ETSI TS 102 034 V1.4.1 (2009-08)

Technical Specification

Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks

Reference
RTS/JTC-DVB-241

Keywords
broadcasting, digital, DVB, IP, satellite, TV, video

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

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 comprising 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 is an updated release of TS 102 034 "Transport of MPEG-2 TS Based DVB Services over IP Based Networks"; it is referred to as DVB-IPTV phase 1.4 and provides extensions to the first set of standardized specifications published by DVB for deployments of DVB services over bi-directional IP networks.

Specifically, it adds support for the following new features:

- Support for Remote Management and Firmware Update Services (RMS-FUS) for managed and unmanaged populations of DVB-IPTV Delivery Network Gateways (DNGs) and Home Network End Devices (HNEDs).
- Regionalisation metadata to enable a HNED to acquire the region in which it is located, from and for a service provider. This is used for the signalling of availability of DVB-IPTV services based on regions.
- Support for Content Download Services (CDSs) for user (pull download) or service provider (push download) initiated download of content items to the HNED.
- Optional retransmission mechanism to provide for protection against packet loss of RTP streams.

As in previous releases of TS 102 034, the DVB-IPTV phase 1.4 work is limited to DVB services [1] encapsulated in MPEG-2 TS [52] covering Live Media Broadcast services (i.e. TV or radio styles), Media Broadcast with Trick Modes and Content on Demand services (CoD). In addition CDSs are supported. These specifications define the mechanisms required in order for a consumer to be able to buy a standard DVB Home Network End Device, take it home, plug it into an IP network, choose and consume DVB services available over the IP network. Clause 4 describes the architectural framework defined for this set of specifications and introduces a Home Network reference model. The contents of the remaining clauses are described below.

1.1 Scope of the present document

1.1.1 What is within the scope

The present document provides specifications to be supported on the interface to the HNED defined as IPI-1 in clause 4 and is based on IP version 4.

It provides a set of technical specifications which covers the following areas:

- The delivery of DVB MPEG-2 TS based services over bi-directional IP networks, both for Live Media Broadcast services (i.e. TV or radio styles) and Content on Demand services. Clause 7 on transport covers the encapsulation of MPEG-2 TS services for streaming delivery over IP and the protocols to be used to access such services. Quality of Service is covered, based on Differentiated Services (DiffServ).
- The Service Discovery and Selection (SD&S) mechanism for DVB based A/V services over bi-directional IP networks. Clause 5 on SD&S defines the service discovery information, its data format and the protocols to use for carriage of this information. Both push and pull models of delivery are supported. Binarisation encoding of SD&S information is specified and can optionally be used if required. Support for advanced codecs, logical channel numbering and signalling regional DVB-IPTV services is provided.
- The use of command and control application-level protocol RTSP to control CoD services and optionally to join multicast services. This is covered in clause 6.
- Clause 8 deals with the assignment of an IP Address to a Home Network End Device (HNED) to get onto the network. The specification is based on DHCP and is restricted to the scenarios where an HNED has a single interface onto the home network and there is a single DNG per home network segment.
- The identification agent for the HNED specified in clause 9 of previous versions of the present document is deprecated and has been deleted.

- Clause 9 now specifies the File Upload System Stub (FUSS) which is mandatory and allows the system software of an HNED to be updated on a power-cycle or reboot. The sending of the system software will be handled by the mechanisms in the optional Remote Management and Firmware Update System for DVB-IPTV Services (RMS-FUS) specification [79].

- Network provisioning specified in clause 10 of previous versions of the present document is deprecated and has been deleted. This functionality is now provided by the Remote Management and Firmware Update Services (RMS-FUS) specification [79].

- Clause 10 now specifies CDSs. CDSs provide the download of content items to a local storage of the HNED via a broadband IP connection. CDSs can be used to provide IPTV services in areas where a broadband connection suitable for streaming services is not available or prone to errors, for simultaneous delivery of multiple content items to HNEDs or for reduced cost offers as the bandwidth consumption may be limited compared to streaming services. Two types of services are supported: push download services where the distribution decision is taken by the service provider (without explicit request from the user) and pull download services where the download is requested by the user. Annex G provides CDS related information, that is expected to be part of other specifications in the future or that is optional for the present document.

- Discovery of Broadband Content Guides (inc. third party). The Broadband Content Guide itself is provided as a separate specification [62].

- Annex E defines an optional protocol for Application Layer FEC (AL-FEC) protection of streaming media for DVB-IPTV services carried over RTP transport. This AL-FEC protocol is a layered protocol with a base layer and zero, one or more optional enhancement layer(s). The base layer is a simple packet-based interleaved parity code based on a subset of [67]. The base layer shall be used wherever AL-FEC is used. The enhancement layer is a Raptor code, as defined in [65] and [66] and may optionally be used to provide further packet loss protection.

- Annex F defines an optional retransmission mechanism (RET) to provide for protection against packet loss of DVB-IPTV services carried over RTP transport. It specifies the mechanism to provide immediate FeedBack (FB) towards the network using RTCP and how to retransmit the missing packets.

NOTE: Packet loss repair can be achieved using the optional AL-FEC solution, the optional retransmission solution or a combination of both solutions.

1.1.2 What is out of the scope

Amongst others, the following subjects are *not* covered in the present document:

- Support for non MPEG-2 TS based services.

- Specific support for Conditional Access or Content Protection.

- Network security and authentication.

- Trick modes (i.e. Pause, Fast Forward, etc.) for Live Media Broadcast services over multicast, e.g. network PVR services.

- IP version 6.

- Configuration of current retail routers and DNGs.

1.1.3 Additional Specifications for Home Network

The present document does not cover home networking. DVB is currently developing a separate specification for home networking. The Home Network Reference Model developed for this purpose is described in [63].

1.1.4 DTDs and XML Schemas

The normative DTDs and XML schemas referenced by the present document are attached as separate files contained in archive ts_102034v010401p0.zip which accompanies the present document. The DTDs and XML schemas included in the present document are informative.

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.

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

[2] ETSI TS 101 162: "Digital Video Broadcasting (DVB); Allocation of Service Information (SI) codes for DVB systems".

[3] ETSI TS 101 812 (V1.3.2): "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.0.3".

[4] IEEE 802-2001: "IEEE Standards for local and metropolitan area networks: overview and architecture".

[5] IEEE 802.1Q-2005: "IEEE standards for local and metropolitan area networks: virtual bridged local area networks".

[6] IEEE 802.2-1989: "Information processing systems - Local area networks - Part 2: logical link control".

[7] IEEE 802.3-2005/Cor 2-2007: "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Corrigendum 2: IEEE Std 802.3an-2006 10GBASE-T Correction".

[8] IEEE P802.11-REVma/D6.0, 2006: Unapproved Draft Standard for Information Technology-Telecommunications and information exchange between systems- Local and metropolitan area network- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

NOTE: This document reflects the combining of the 2003 Edition of 802.11 plus the 802.11g, 802.11h, 802.11i and 802.11j Amendments) (Revision of IEEE Std 802.11-1999).

[9] IEEE 802.1D (2004): "IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges".

[10] IETF RFC 768: "User Datagram Protocol".

[11] IETF RFC 791: "Internet Protocol; DARPA internet protocol; Protocol specification".

[12] IETF RFC 826: "An Ethernet Address Resolution Protocol - or - Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware".

[13] IETF RFC 1034: "Domain names - concepts and facilities".

[14] IETF RFC 1035: "Domain names - Implementation and specification".

[15] IETF RFC 1042: "A Standard for the Transmission of IP Datagrams over IEEE 802 Networks".

[16] IETF RFC 1101: "DNS Encoding of Network Names and Other Types".

[17] IETF RFC 1122: "Requirements for Internet Hosts - Communication Layers".

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

[19] IETF RFC 1738: "Uniform Resource Locators (URL)".

[20] IETF RFC 1630: "Universal Resource Identifiers in WWW".

[21] IETF RFC 3550: "RTP: A Transport Protocol for Real-Time Applications".

[22] IETF RFC 1890: "RTP Profile for Audio and Video Conferences with Minimal Control".

[23] IETF RFC 2030: "Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI".

[24] IETF RFC 2131: "Dynamic Host Configuration Protocol".

[25] IETF RFC 2132: "DHCP Options and BOOTP Vendor Extensions".

[26] IETF RFC 2181: "Clarifications to the DNS Specification".

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

[28] IETF RFC 2241: "DHCP Options for Novell Directory Services".

[29] IETF RFC 2250: "RTP Payload Format for MPEG1/MPEG2 Video".

[30] IETF RFC 2326: "Real Time Streaming Protocol (RTSP)".

[31] IETF RFC 2396: "Uniform Resource Identifiers (URI): Generic Syntax".

[32] IETF RFC 2474: "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers".

[33] IETF RFC 2475: "An Architecture for Differentiated Services".

[34] IETF RFC 2485: "DHCP Option for The Open Group's User Authentication Protocol".

[35] IETF RFC 2486: "The Network Access Identifier".

[36] IETF RFC 2508: "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links".

[37] IETF RFC 2563: "DHCP Option to Disable Stateless Auto-Configuration in IPv4 Clients".

[38] IETF RFC 2610: "DHCP Options for Service Location Protocol".

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

[40] IETF RFC 2782: "A DNS RR for specifying the location of services (DNS SRV)".

[41] IETF RFC 2937: "The Name Service Search Option for DHCP".

[42] IETF RFC 2998: "A Framework for Integrated Services Operation over Diffserv Networks".

[43] IETF RFC 3004: "The User Class Option for DHCP".

[44] IETF RFC 3011: "The IPv4 Subnet Selection Option for DHCP".

[45] IETF RFC 3023: "XML Media Types".

[46] IETF RFC 3046: "DHCP Relay Agent Information Option".

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

[48] IETF RFC 5052: "Forward Error Correction (FEC) Building Block".

[49] IETF RFC 3927: "Dynamic Configuration of IPv4 Link-Local Addresses".

[50] ISO 3166 (all parts): "Codes for the representation of names of countries and their subdivisions".

[51] ISO 639-2: "Codes for the representation of names of languages - Part 2: Alpha-3 code".

[52] ISO/IEC 13818-1 (1996): "Information technology - Generic coding of moving pictures and associated audio information: Systems".

[53] ISO/IEC 13818-9 (1996): "Information technology - Generic coding of moving pictures and associated audio information - Part 9: Extension for real time interface for systems decoders".

[54] "Extensible Markup Language (XML) 1.0 (Fourth Edition)": First published 4 February 2004, revised 16 August 2006, Jean Paoli, Tim Bray, François Yergeau, C. M. Sperberg-McQueen, Eve Maler.

[55] "XML Schema Part 0: Primer Second Edition": First published 2 May 2001, revised 28 October 2004, Priscilla Walmsley, David C. Fallside.

[56] "XML Schema Part 1: Structures Second Edition": First published 2 May 2001, revised 28 October 2004, David Beech, Henry S. Thompson, Murray Maloney, Noah Mendelsohn.

[57] "XML Schema Part 2: Datatypes Second Edition": First published 2 May 2001, revised 28 October 2004, Ashok Malhotra, Paul V. Biron.

[58] ETSI TS 101 154 (V1.8.1): "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream".

[59] ETSI TS 102 323: "Digital Video Broadcasting (DVB); Carriage and signalling of TV-Anytime information in DVB transport streams".

[60] ETSI TS 102 822-3-1 (V1.3.1): "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 3: Metadata; Sub-part 1: Phase 1 - Metadata schemas".

[61] ISO/IEC 23001-1 (MPEG-B): "Information Technology - MPEG Systems Technologies - Binary MPEG format for XML".

[62] ETSI TS 102 539: "Digital Video Broadcasting (DVB); Carriage of Broadband Content Guide (BCG) information over Internet Protocol (IP)".

[63] DVB BlueBooks A109: "DVB-HN (Home Network) Reference Model Phase 1".

[64] UPnP Device Architecture 1.0.

[65] ETSI TS 126 346: "Universal Mobile Telecommunications System (UMTS); Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs; (3GPP TS 26.346 Release 6)".

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

[67] SMPTE specification 2022-1 (2007): "Forward Error Correction for Real-time Video/Audio Transport Over IP Networks".

[68] SMPTE specification 2022-2 (2007): "Unidirectional Transport of Constant Bit Rate MPEG-2 Transport Streams on IP Networks".

[69] ITU-T Recommendation H.610 (07/2003): "Full service VDSL - System architecture and customer premises equipment".

[70] ETSI TS 102 822-3-2 (V1.3.1): "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 3: Metadata; Sub-part 2: System aspects in a uni-directional environment".

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

[72] IETF RFC 3451: "Layered Coding Transport (LCT) Building Block".

[73] IETF RFC 3450: "Asynchronous Layered Coding (ALC) Protocol Instantiation".

[74] IETF RFC 3695: "Compact Forward Error Correction (FEC) Schemes".

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

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

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

[78] IETF RFC 5053: "Raptor Forward Error Correction Scheme for Object Delivery".

[79] ETSI TS 102 824: "Digital Video Broadcasting (DVB); Remote Management and Firmware Update System For DVB IP Services".

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

[81] IETF RFC 3890: "A Transport Independent Bandwidth Modifier for the Session Descrip-tion Protocol (SDP)".

[82] IETF RFC 3388: "Grouping of Media Lines in SDP".

[83] IETF RFC 3555: "MIME Type Registration of RTP Payload Formats" .

[84] IETF RFC 3556 (July 2003): "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth".

[85] IETF RFC 4585 (July 2006): "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)".

[86] IETF RFC 4588 (July 2006): "RTP Retransmission Payload Format".

[87] IETF RFC 3489 (March 2003): "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)".

[88] IETF RFC 3361 (August 2002): "Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers".

[89] IETF RFC 3397 (November 2002): "Dynamic Host Configuration Protocol (DHCP) Domain Search Option".

[90] IETF RFC 3118 (June 2001): "Authentication for DHCP Messages".

[91] IETF RFC 3442 (December 2002): "The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) version"2.

[92] IETF RFC 3495 (March 2003): "Dynamic Host Configuration Protocol (DHCP) Option for CableLabs Client Configuration".

[93] IETF RFC 3825 (July 2004): "Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information".

[94] IETF RFC 3925 (October 2004): "Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4)".

[95] IETF RFC 4280 (November 2005): "Dynamic Host Configuration Protocol (DHCP) Option for Broadcast and Multicast Control Servers".

[96] IETF RFC 4388 (February 2006): "Dynamic Host Configuration Protocol (DHCP) Leasequery".

[97] IETF RFC 4578 (November 2006): "Dynamic Host Configuration Protocol Options for the Intel Preboot eXecution Environment (PXE)".

[98] IETF RFC 4676 (October 2006): "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information".

[99] IETF RFC 4833 (April 2007): "Timezone Options for DHCP".

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

[101] IETF RFC 3679 (January 2004): "Unused Dynamic Host Configuration Protocol (DHCP) Option Codes".

[102] IETF RFC 4039 (March 2005): " Rapid Commit Option for the Dynamic Host Configuration Protocol version 4 (DHCPv4)".

[103] IETF RFC 4702 (October 2006): "The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option".

[104] ETSI TS 102 825-4: "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 4: CPCM System Specification".

[105] IETF RFC 4607 (August 2006): "Source-Specific Multicast for IP".

[106] ETSI TS 102 006 (V1.3.2): "Digital Video Broadcasting (DVB); Specification for System Software Update in DVB Systems".

[107] IETF RFC 3980: "T11 Network Address Authority (NAA) Naming Format for iSCSI Node Names".

[108] IETF RFC 2246: "The TLS Protocol Version 1.0".

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

[110] "SMI Network Management Private Enterprise Codes".

[111] ETSI TS 102 833: "Digital Video Broadcasting (DVB); File Format Specification for the Storage and Playback of DVB Services".

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.

[i.1] ETSI TR 101 211: "Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)".

[i.2] TM3422: "DVB-HN Commercial Requirements Phase 1".

[i.3] IETF draft-ietf-rmt-bb-lct-revised-07 (July 2008): "Layered Coding Transport (LCT) Building Block".

[i.4] draft-ietf-avt-rtcpssm-17 (7 January 2008): "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback".

[i.5] draft-ietf-avt-rtcp-non-compound-06 (July 7, 2008): "Support for Reduced-Size RTCP, Opportunities and Consequences"; I. Johansson, M. Westerlund, Ericsson AB; Internet Draft.

[i.6] draft-ietf-avt-rtp-and-rtcp-mux-07 (August 1, 2007): "Multiplexing RTP Data and Control Packets on a Single Port"; Internet draft.

[i.7] IEEE 1394: "IEEE Standard for High Performance Serial Bus".

[i.8] IETF RFC 3171: "IANA Guidelines for IPv4 Multicast Address Assignments".

[i.9] IEC 62481-1: "Digital Living Network Alliance (DLNA) Home Networked Device Interoperability Guidelines - Part 1: Architecture and Protocols".

[i.10] TS 102 542 (all parts) V1.3.1: "Digital Video Broadcasting (DVB); Guidelines for the implementation of DVB-IPTV Phase 1 specifications".

3 Definitions, abbreviations and notations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

bridge component: OSI layer 2 connecting component, that connects two or more link layer components, not necessarily using different technologies

NOTE: A bridge is usually called either a hub or a (layer 2) switch, where a hub typically forwards all the data coming in on one of the ports to all the other ports and a switch provides some additional functionality such as forwarding packets only to a specific port.

CDS HNED storage: storage on the HNED dedicated to CDSs of a single service provider

component: specific set of functionalities

NOTE: It can offer this functionality to other components in the same device.

connecting component: component which is used to connect link layer components with each other

Content Download Service (CDS): service that provides download delivery of content items to the local storage of the HNED

NOTE: The consumption is independent of the delivery.

content item: an editorially coherent grouping of one or more audiovisual or generic data files which are intended to be consumed in conjunction with each other

content provider: entity that owns or is licensed to sell content or content assets

Content on Demand (CoD): program provided at the request of the end user for direct consumption (real-time streaming)

Content Service Provider (CSP): entity which acquires/licenses content from Content Providers and packages this into a service

Delivery Network (DN): network connecting the Delivery Network Gateway (DNG) and service providers

Delivery Network Gateway (DNG): device that is connected to one or multiple delivery networks and one or multiple home network segments

Destination Transport address: combination of the IP destination address and destination UDP port

Download Session Description: collection of parameters that describe how a content item can be downloaded within a download session using the DVB-IPTV CDS

DVB-IPTV service: one or more programmes under the control of a service provider delivered over IP. The programmes can be made available either as part of a schedule or on demand and either for direct consumption (Live Media Broadcast or Content on Demand Services) or for local storage (CDSs)

DVB CoD RET server: interacts with the RET clients by responding to RET requests with RET packets

DVB LMB RET server: interacts with RET clients, mainly by responding to RET requests with RET packets

DVB RET client: part of the HNED that makes use of the RET protocol to request and receive RET packets from a DVB (CoD/LMB) RET server when it detects packet loss

event: grouping of elementary broadcast data streams with a defined start and end time belonging to a common service

EXAMPLE: First half of a football match, News Flash, first part of an entertainment show.

gateway component: connecting component that connects two or more link layer components of typically different technologies together (it can function at OSI layers 4 through 7)

Home Network End Device (HNED): device that is connected to a home network and which typically terminates the IP based information flow (sender or receiver side)

Home Network Segment (HNS): consists of a single link layer technology and provides a layer 2 connection between home network end devices and/or connecting components

Internet Service Provider (ISP): party offering an Internet access service to the end-user

link layer component: OSI layer 2 component consisting of link layer technology and which is used to provide connectivity between devices

EXAMPLES: Ethernet, DVB-RC, IEEE 802.11 [8] [8].

MPEG-2: Refers to ISO/IEC 13818-1 [52]

NOTE: Systems coding is defined in ISO/IEC 13818- 1 [52]. The real time interface specification is defined in ISO/IEC 13818- 9 [53].

Original RTP session: RTP session where the RTP packets are original packets (not retransmitted)

package: collection of DVB services marketed as a single entity

program: collection of program elements

NOTE: Program elements may be elementary streams. Program elements need not have any defined time base; those that do, have a common time base and are intended for synchronized presentation. Taken from ISO/IEC 13818-1 [52].

pull download service: Content Download initiated by the user

push download service: Content Download initiated by the service provider without explicit request by the user

RET-enabled HNED: HNED that has a DVB RET client

RET-enabled CoD/LMB: CoD/LMB service where RET-enabled HNED can make use of DVB RET protocols for packet loss repair

router component: OSI layer 3 connecting component which connects two or more link layer components to each other, not necessarily of the same type

NOTE: A router is able to select among multiple paths to route packets through the network based on a destination address available in the packet. The only OSI layer 3 type considered is IP.

RTP session: as defined in clause 3 of RFC 3550 [21]

Service Provider (SP): entity providing a service to the end-user

NOTE: See clause 4 on architecture. In the context of the present document, SP will mean a Service Provider providing DVB-IPTV services.

session multiplexing: scheme by which the original stream and the retransmission stream are sent in different RTP sessions

source transport address: combination of the IP source address and source UDP port

SP offering: set of streams or services a Service Provider proposes to the end-user

SSRC multiplexing: scheme by which the original stream and the retransmission stream are sent in the same RTP session with different SSRC values

transport stream: data structure defined in ISO/IEC 13818-1 **[**52**]**

TS Full SI: transport stream with embedded service information as defined by DVB in EN 300 468 [1] with the exception of the network information table NIT

NOTE: This table may be omitted as it has no meaning in the context of IP services.

TS - Optional SI: transport stream with MPEG PSI (PAT and PMT tables) as defined in ISO/IEC 13818-1 [52], all other MPEG-2 and DVB tables are optional

3.2 Abbreviations

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

A/V	Audio/Video
ABNF	Augmented Backus-Naur Form
AL-FEC	Application Layer Forward Error Correction
ALG	Application Level Gateway
ASM	Any Source Multicast
BCG	Broadband Content Guide
BiM	Binary MPEG format for XML
BLP	Bitmask Lost Packet
BNF	Backus-Naur Form
CDS	Content Download Service
CMD	Carousel Multicast Download
CoD	Content on Demand
CoS	Class of Service
CPCM	Content Protection and Copy Management
CPU	Central Processing Unit
CRID	Content Reference IDentifier
CSP	Content Service Provider
CSRC	Contributing SouRCe
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DNG	Delivery Network Gateway
DNS	Domain Name System
DSCP	Differentiated Services CodePoint
DSM-CC	Digital Storage Media - Command and Control
DTD	Document Type Declaration
DTH	Direct To Home
DVB	Digital Video Broadcasting
DVB-RC	Digital Video Broadcasting - Return Channel
DVB-S	Digital Video Broadcasting - Satellite
DVBSTP	DVB SD&S Transport Protocol
ESI	Encoding Symbol ID
FB	FeedBack

NOTE: Typically including negative acknowledgements, i.e., NACK.

FDT	File Delivery Table
FLUTE	File Delivery over Unidirectional Transport
FMT	Feedback Message Type
FUS	Firmware Update Service
FUSS	File Upload System Stub
HN	Home Network
HNED	Home Network End Device

HNN	Home Network Node
HNS	Home Network Segment
HTC	Head-end Time Clock
HTTP	Hyper Text Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ID	IDentifier
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPDC	Internet Protocol DataCasting
IPI	Internet Protocol Infrastructure
IPv4	Internet Protocol version 4
ISN	Initial Sequence Number
ISO	International Organization for Standardization
ISP	Internet Service Provider
LCT	Layered Coding Transport
LMB	Live Media Broadcast
MAC	Media Access Control
MBwTM	Media Broadcast with Trick Modes
MC	MultiCast
MHP	Multimedia Home Platform
MIME	Multipurpose Internet Mail Extension
MPEG	Moving Pictures Expert Group
MPTS	Multiple Program Transport Stream
MTS	MPEG-2 Transport Stream
MTU	Maximum Transmission Unit
NACK	Negative ACKnowledgement
NTP	Network Time Protocol
OSI	Open Systems Interconnection
OSN	Original Sequence Number
PAT	Program Association Table
PCR	(MPEG-2) Program Clock Reference
PLL	Phased Locked Loop
PMT	Program Map Table
PT	Payload Type
QoS	Quality of Service
QRC	Query-Response Channel
RET	RETransmission
RFC	Request For Comments
RMS	Remote Management and Firmware
RR	Receiver Report
RSI	Receiver Summary Information
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SAP	Session Announcement Protocol
SBL	Source Block Length
SBN	Source Block Number
SDES	Source DEScription
SDP	Session Description Protocol
SD&S	Service Discovery and Selection
SI	Service Information
SMD	Scheduled Multicast Download
SNTP	Simple Network Time Protocol
SOAP	Simple Object Access Protocol
SOHO	Small Office/Home Office
SP	Service Provider
SR	Sender Report
SSL	Secure Socket Layer
SSM	Single Source Multicast

SSM	Source Specific Multicast
SSRC	Synchronization Source
STC	(MPEG-2) System Time Clock
TCP	Transmission Control Protocol
TLS	Transaction Layer Security
TOI	Transport Object Identifier
ToS	Type of Service
TS	Transport Stream
TSI	Transport Session Identifier
T-STD	(MPEG-2) Transport Stream System Target Decoder
TV	TeleVision
TVA	TV Anytime
UD	Unicast Download
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
UTF	Unicode Transformation Format
VCR	Video Cassette Recorder
VOD	Video On Demand
WWW	World Wide Web
XML	eXtensible Markup Language
XOR	eXclusive OR

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 [27], for syntax specification.

3.3.1.1 General rules

The following general rules are defined:

```
host            = domainName / ipAddress
domainName      = *(domainNameLabel '.') topLabel ['.']           ; E.g www.example.org_
domainNameLabel = label / aceLabel
label           = ALPHANUM *('-' / ALPHANUM) ALPHANUM             ; E.g. legal-label6
topLabel        = ALPHA *('-' / ALPHANUM) ALPHANUM                ; E.g. com
name            = ALPHA *(('-' ALPHANUM) / ALPHANUM)              ; E.g. legal-name6
aceLabel        = acePrefix punnyCode                             ; Internationalized Domain Name
acePrefix       = 'x' 'n' '-' '-'                                 ; E.g. 'xn--' or 'XN--'
punnyCode       = *('-' / ALPHANUM)
ipAddress       = dottedDecimal / 1*10(DIGIT)                     ; E.g. 80.78.123.11 or 1347320587
dottedDecimal   = 1*3(DIGIT) '.' 1*3(DIGIT)'.' 1*3(DIGIT)'.' 1*3(DIGIT)
version         = 1*3(DIGIT) '.' 1*3(ALPHANUM)                    ; E.g. 1.2A
version         =/ 1*3(DIGIT) '.' 1*3(ALPHANUM) '.' 1*3(ALPHANUM) ; E.g. 1.11C.32
```

3.3.1.2 Core rules

The following set of ABNF core rules derived from [27] are defined:

```
ALPHA   =  %x41-5A / %x61-7A    ; A-Z / a-z
BIT     =  "0" / "1"
CHAR    =  %x01-7F              ; any 7-bit US-ASCII character, excl. NUL
CR      =  %x0D                 ; carriage return
CRLF    =  CR LF                ; Internet standard newline
CTL     =  %x00-1F / %x7F       ; control characters
DIGIT   =  %x30-39              ; 0-9
ALPHANUM=  ALPHA / DIGIT        ; A-Z / a-z / 0-9
DQUOTE  =  %x22                 ; " (Double Quote)
HEXDIG  =  DIGIT / %x41-46 / %x61-66 ;
HTAB    =  %x09                 ; horizontal tab
LF      =  %x0A                 ; linefeed
```

```
LWSP    =  *(WSP / CRLF WSP)     ; linear white space (past newline)
OCTET   =  %x00-FF               ; 8 bits of data
SP      =  %x20                  ; space
VCHAR   =  %x21-7E               ; visible (printing) characters
WSP     =  SP / HTAB             ; white space
```

NOTE 1: The rules for constructing `domainName` is aligned with RFC 1035 [14], RFC 1101 [16] (First mention of labels starting with digits), RFC 1738 [19] (URL), RFC 2181 [26] (Clarifications), RFC 2396 [31] (Including the optional trailing dot), RFC 2486 [35] (URI) and ICANN agreements with domain registrars (`www.icann.org/tlds/agreements/pro/registry-agmt-appc-26aug03.htm` and `www.icann.org/tlds/agreements/name/registry-agmt-appc-13-03jul01.htm`).

NOTE 2: ABNF is used on several places throughout the present document.

4 Architecture

4.1 System structure

In order to describe the complex system that is necessary for the delivery of DVB-services over IP-based networks, the two following clauses describe the inherent functionality. By using these descriptions all elements and interfaces are explained including their interaction in the system.

The Layer Model shows a general overview over the number of interfaces between the domains. The Home Network Reference Model (see figure 2) shows details of the interfaces between the access network, the home network segment and the home network end devices. Clause 4.1.3 shows the relations of the protocols specified and used by the present document to the general TCP/IP- protocol suite.

A dedicated functional architecture for CDSs is defined in clause 10.2. This functional architecture defines the functional blocks and specific interface components at the IPI-1 interface for support of CDSs.

The prime target for standardization by DVB is the interface to the home network end devices, to enable high-volume low-cost equipment. The suite of standards should be complete from layer 1 up to and including the application layer.

4.1.1 Layer model

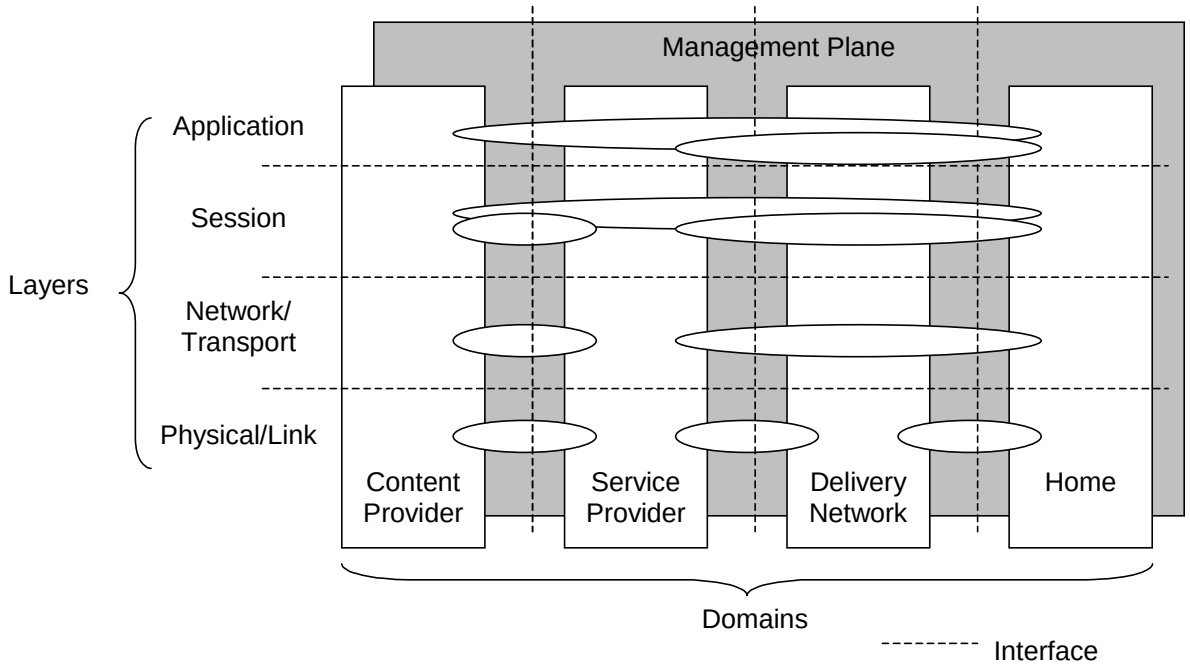


Figure 1: Layer model

The four communicating domains are briefly described as follows:

- Content Provider: the entity that owns or is licensed to sell content or content assets. Although the Service Provider is the primary source for the client at Home, a direct logical information flow may be set up between Content Provider and Home client e.g. for rights management and protection. This flow is shown in the layered model.

- Service Provider: the entity providing a service to the end-user. Different types of service provider may be relevant for DVB services on IP, e.g. simple Internet Service Providers (ISPs) and Content Service Providers (CSPs). In the context of DVB services on IP, the CSP acquires/licenses content from Content Providers and packages this into a service. In this sense the service provider is not necessarily transparent to the application and content information flow.

- Delivery Network: the entity connecting clients and service providers. The delivery system usually is composed of access networks and core or backbone networks, using a variety of network technologies. The delivery network is transparent to the IP traffic, although there may be timing and packet loss issues relevant for A/V content streamed on IP.

- Home: the domain where the A/V services are consumed. In the home a single terminal may be used for service consumption, but also a network of terminals and related devices may be present for this purpose.

As mentioned above the Service Provider entity covers various kinds of Service Provider types, especially broadband ISPs and CSPs. It should be noted that although we treat these two business roles separately, a single company could very well act in both roles. In such a case the end user could be offered a single subscription covering both the ISP and the CSP service offerings (see below).

It is noted that today's Internet business models often involve so called virtual SPs, which means that the SP relies on some other party, typically a wholesale IP network operator, to implement and run all (or parts) of the service production platform. However, in the present document we do not distinguish any virtual SP roles - whether the SP owns the service production platform or "out-sources" the platform is irrelevant for this model since we simply look at the services and functions of each domain. It is also noted that in some countries, the access provider and the ISP may be different parties. In this context, however, those are not treated separately, but the ISP is the only party covered. The "access provider" could for example provide the end device with the IP address. However, in order to simplify the description we cover such potential access provider services/functions under the ISP role.

4.1.2 Home Network Reference Model

The architecture of the DVB home network shall support the following (non-exhaustive) list of possible scenarios taken from TM3422 [i.2]:

1) A home network can be simultaneously connected to multiple and heterogeneous delivery networks.

   As an example, in a typical scenario ADSL and DVB-S are both available at the home. Load balancing may be possible between the different delivery networks in order to optimize the utilization and throughput of the networks and to minimize the delay.

2) End users can choose the service provider.

   As an example, the ISPs and the CSPs may be independent from each other.

3) Different end users in the same home network can select different service providers.

4) End users can access a DVB content from several devices in the home.

5) End users can remotely access the home network for the scheduling of recording sessions.

Based on these scenarios a reference model for DVB home network can be constructed. This reference model is depicted in the Home Network Reference Model document [63] In the present document only the delivery of DVB-IPTV services over broadband delivery networks to DVB-IPTV HNEDs is defined as shown in figure 2. The advanced home network functionality depicted in the Home Network Reference model [63] will be defined in a separate specification.

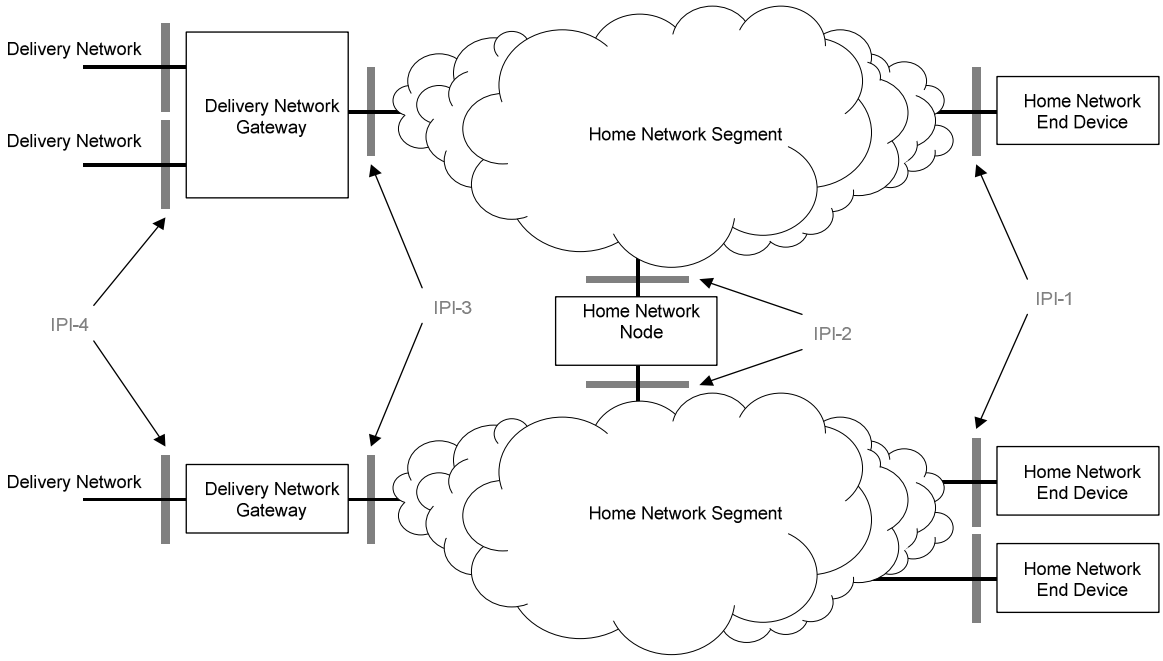


Figure 2: Home Network Reference Model

The Home Network Reference Model considered in the present document, as depicted in figure 2, consists of the Home domain of the Layer Model. Furthermore, it shows the interconnection with the Delivery Network domain. This Home Network Reference Model shows the elements that can be present in the home and their mutual relation. Based on the fact that this is just a reference model, elements can be removed or added as long as the connection between a home network end device and the delivery network is still possible. The collection of all these home network elements forms the Home Network (HN).

The elements present in the Home Reference Model are described as follows:

- Delivery Network Gateway (DNG): the device that is connected to one or multiple delivery networks and one or multiple home network segments. It contains one or more connecting components so that it can interconnect the delivery network(s) with the home network segment(s) on any of the OSI layers. This means, that it can be a so-called "null" device, a wire interconnecting the networks on OSI layer 1; that it can function as a bridge or router interconnecting different link layer technologies; or that it can act as a gateway also providing functionality on the OSI layer 4 and above.

- Home Network Segment (HNS): this element consists of a single link layer technology and provides a layer 2 connection between home network end devices and/or connecting components. The connecting components are not part of a home network segment. So, each home network segment is connected to another home network segment via a connecting component. All HNS form the Home Network, a Home Network is a single "IP-subnet". A home network segment can be wired or wireless.

Due to the fact, that various network technologies can be used by home network segments, the network technology specific name is used to distinguish between them.

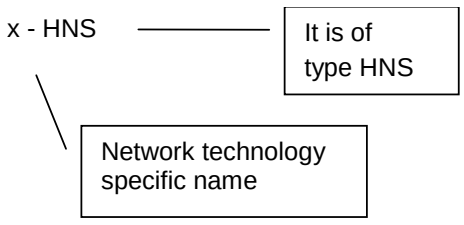


Figure 2a: Home network technology name

Some examples: Ethernet - HNS, Wireless LAN - HNS.

An Ethernet HNS is based on 100BASE-T as specified in IEEE 802.3 [7].

A Wireless LAN HNS is based on IEEE 802.11 as specified in [8].

NOTE: Alternative legacy Ethernet Frame formats (e.g. DIX) are not supported by the present document due to the need to support IEEE 802 framing for QoS.

- Home Network Node (HNN): this device, which contains one or more connecting components, connects two or more home network segments with each other and functions as a bridge. The specification for a HNN that interconnects IEEE 802 [4] LANs (below the MAC service boundary) in a bridged format is defined in IEEE 802.1Q [5]. The present document shall apply to connection between 2 or more 100BASE-T Ethernet HNSs (e.g. an Ethernet switch or hub) and shall apply also to bridging between a 100BASE-T Ethernet HNS and another HNS based on the IEEE 802 MAC layer e.g. a wireless HNS. The HNN shall provide support for QoS via IEEE 802.1D [9] (see clause 11.2).

- Home Network End Device (HNED): the device that is connected to a home network and which typically terminates the IP based information flow (sender or receiver side). This does not imply that this home network end device needs to be the end point of the non-IP based information flow. So, it can still serve as an application level gateway to other non-IP based network technologies. For instance, a DVB stream over IP can be converted to a DVB stream directly over IEEE 1394 [i.8]. This conversion is out of scope of the present document.

- In case the DNG is a "null" device, there is no actual home network. So, in that case the home network end device is directly connected to the delivery network.

The mutual relations presented in the Home Network Reference Model can be described by means of interfaces, which are provided in the figure by using the following naming principle.

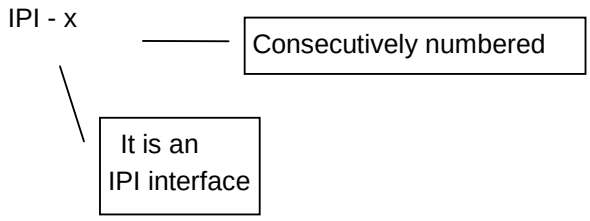


Figure 2b: Interface name

Currently, four interfaces have been defined. Of these interfaces, the IPI-1 interface, as depicted in figure 2, is the primary target for standardization in the present document. The interface description will be independent from the physical layer and link layer technologies used in the home network. The other three interfaces are not specified in the present document but they are taken into account into the Home Network Reference Model document [63].

All IP-based traffic shall be carried transparently over a HNS. Therefore, the interfaces IPI-1, IPI-2 and IPI-3 on a HNS shall comply to the IETF specification RFC 1042 [15]. The Address Resolution Protocol as defined in RFC 826 [12] shall be used.

For the configuration (e.g. IP address) of Home Network End Devices (HNED) DHCP client functionality shall be supported as defined in clause 8. HNED shall also implement Auto IP as defined by the UPnP Device Architecture v1.0 [64] specification so that if a DHCP server is not present on the home network, a link-local network address may be automatically acquired. Each HNED should be uniquely identified by its MAC address (48 bit Ethernet address).

4.1.3 Diagram of the DVB-IPTV Protocol Stack

Figure 3 is a logical diagram of the high-level protocols on the IPI-1 interface, specified in the present document for enabling DVB services over IP-based networks and the associated delivery and network support services. The organization of this protocol stack is based on the hierarchical structure frequently applied in equipment design, i.e. service offering and applications, middleware and functions, IP protocols and transport, and phy/MAC/link layers. This follows the ISO/OSI layering convention in general terms.

The top layer of this stack signifies the service offering intended by the Service Provider. This consists of programs, information about programs, multicast- and/or unicast IP addresses; in short, the essential items needed to enable a DVB service over an IP network.

The middleware and functions layer includes those functions described in the present document and other DVB supporting documents, the text colours are unique to the functions or groups of functions and those colours map down to the IP protocol and transport layer of the diagram.

The colour coding used is:

- QoS = red.
- Multicast service connection and management = black.
- Reliability of delivery = green.
- DVB AV and data services and metadata = blue.
- Remote management and firmware update = yellow.
- Content Download Service (CDS) = brown.
- HNED provisioning and boot procedures = purple.

The IP protocol and transport layer attempts to identify which protocols and transports are required and map the usage of those protocols and transports to the functions of the layers above using the colour coding. Where the protocol is shown in black it indicates that it is required by multiple functions, e.g. DVBSTP, HTTP, etc.

In principle the protocols required for transport of elements of the service offering via IP networking is independent of the physical layers below the IP networking layer and the present document is generally agnostic to the physical layer technology.

The software stack diagram is shown for information only and is not normative.

The HNED is an IP compliant device; on its IPI-1 interface it supports the requirements laid down in RFC 1122 [17]. HTTP, TCP, UDP and IP are available to the HNED as networking and transport protocols.

The following clauses mention the protocols and protocol-related markings, usage of which is specified in the clauses of the present document.

Information for service discovery and selection services is assembled according to the SD&S protocol, specified in clause 5. The SD&S protocol for multicast (push) services is transported in IP packets according to the DVBSTP transport protocol, also specified in clause 5. For unicast (pull) services the SD&S information is transported via HTTP. An SD&S entry point can be implemented using a DNS mechanism, specified in clause 5.

The Real-Time Streaming Protocol (RTSP) is used for delivery and control of content on-demand services and optionally can also be used for control of the delivery of Live Media Broadcast services, e.g. TV and audio (radio) programs. The specification of this usage can be found in clause 6.

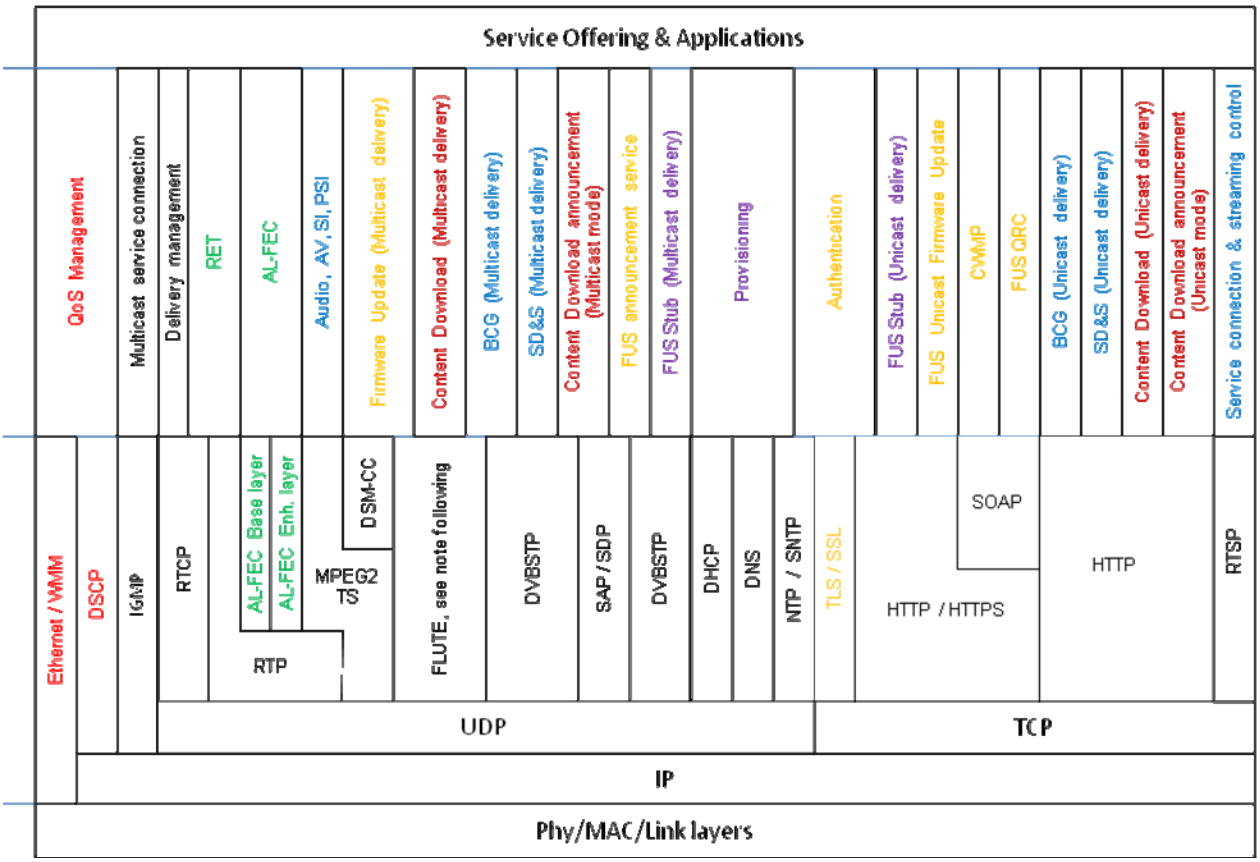
The Audio and Video streams and the Service Information are multiplexed into a valid MPEG-2 Transport Stream, according to [52]. The resulting MPEG-2 packets are encapsulated directly in UDP or in RTP/UDP for streaming delivery, with DSCP packet markings for quality of service. Streaming delivery of MPEG-2 TS on IP is specified in clause 7. The use of RTCP, e.g. to send information to receivers about transmission statistics, and of IGMP to join and leave multicast streams, is also specified in clause 7.

DSCP packet markings and Ethernet priority setting for quality of service are specified in clause 11.

The DHCP protocol is used to configure the HNED with an IP address. The detailed mechanisms and the options for this and related other functions are specified in clause 8. Real time clock services or accurate network time services are implemented using respectively SNTP or NTP protocol.

In this updated version of the present document the initial boot procedure uses a stub file (FUSStub) downloaded over HTTP for unicast or acquired from a multicast DVBSTP service, this stub file mechanism replaces the identification agent method used in previous versions of this present document. For CDSs the HTTP protocol is used for unicast delivery and the FLUTE protocol for multicast delivery.

Content download sessions are described in XML or SDP syntax and delivery is via HTTP (unicast) or SAP (multicast for SDP data) or DVBSTP (multicast for XML data). The mechanisms and protocols are specified in clause 10.



NOTE 1: The profile of DSM-CC used for firmware delivery for RMS-FUS is as specified in TS 102 006 v1.3.2 [106].
NOTE 2: The information exchanged in RTSP may be conveyed in an XML or SDP format.
NOTE 3: TLS/SSL indicates that either TLS or SSL can be used.
NOTE 4: HTTP/HTTPS indicates that either HTTP or HTTPS can be used.

Figure 3: Diagram of the protocol stack for DVB-IPTV services

4.2 Phase 1 scenarios

The present document does not attempt to cover every possible scenario for the HNED. The aim is to cover the main possibilities in use today and likely in the near future, whilst making it simple to implement. The next clauses cover scenarios allowed by the specification.

All scenarios are using DHCP mechanisms to assign IP addresses and other parameters to a HNED. IP traffic is routed on OSI Layer 3 via the DNG to the HNED. HNEDs with static IP addresses are not covered and will be supported by future versions of DVB-IPTV specifications.

4.2.1 Single DNG scenario

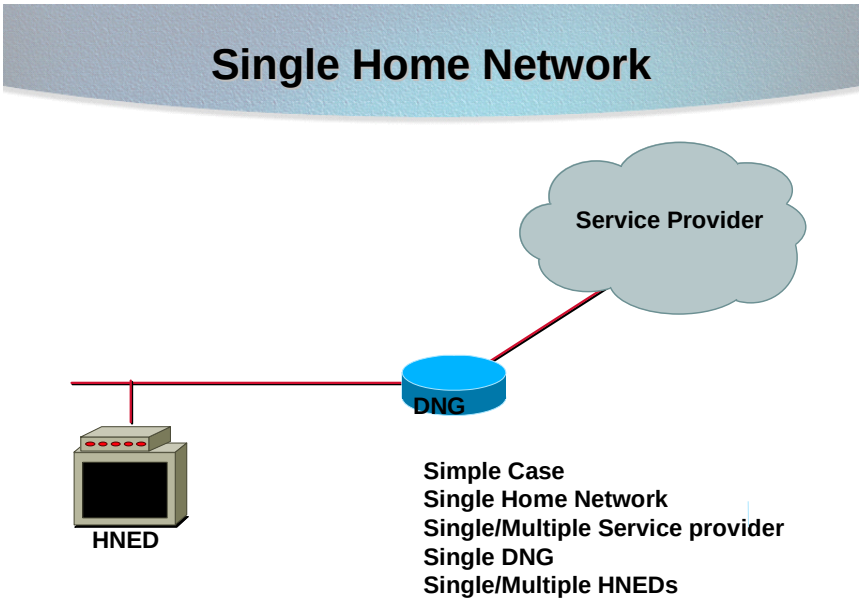


Figure 4: Single home network

In the "Single Delivery Network Gateway" scenario, the home has a single DNG and a single home network. There can be multiple devices on the home network all communicating with each other and via the DNG to the outside world. The use of multiple service providers is allowed, however, the Service Provider that is connected to the DNG routes the packet in the appropriate manner.

4.2.2 Multiple DNGs

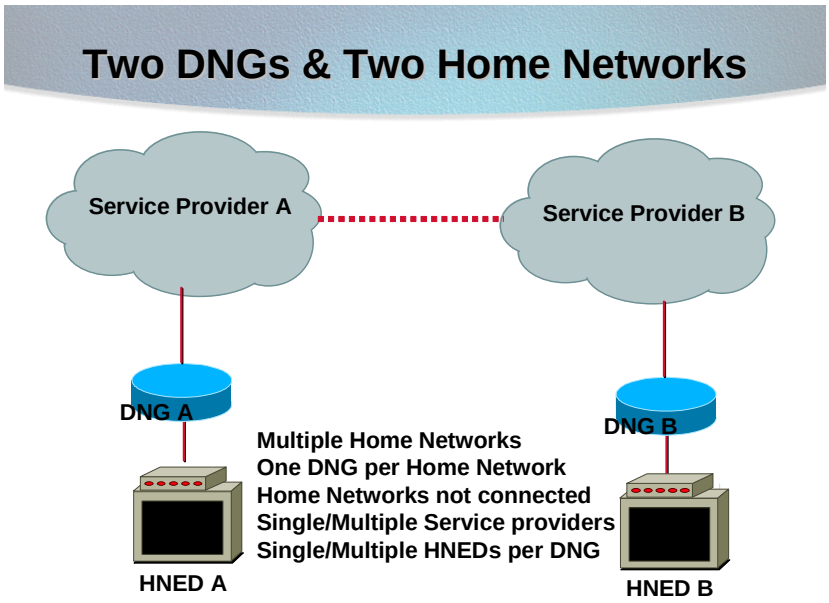


Figure 5: Multiple DNGs

In the "Multiple Delivery Network Gateways" scenario the home has multiple DNGs but each DNG has its own private and separate home network. The separation of the home networks does not mean that the two HNEDs in the diagram cannot communicate; it means that any communication shall go via the Service Provider networks (shown by the red dotted line). The use of Service Provider A and B also does not limit the user to 2 service providers because as in the "Single Delivery Network Gateway" scenario, multiple service providers can be used via the DNG owned Service Provider's network.

4.2.3 DNG and HNED in One Box

If the DNG and HNED are combined into a single box, with the HNED directly connected to the Service Provider's network, then the IP addressing rules described in the present document shall not apply as it is treated as a DNG.

If the combination box has an Ethernet port to allow other devices to be attached to it, then the addressing rules in the present document do apply.

5 Service discovery

5.1 Overview

The present document covers the mechanisms used for service discovery, service selection and the delivery of service discovery information.

Service discovery is the mechanism enabling the discovery of DVB-IPTV services available over bi-directional IP network. The service discovery results in the presentation of a list of services with sufficient information for the user to make a choice and access the chosen service. Selection takes place after the user has made a choice about which service to view.

Live Media Broadcast, CoD and CDSs are covered by the present document. Two types of Live Media broadcast services have been identified: broadcast services with DVB SI [1] embedded in the stream (referenced as "TS Full SI") and broadcast services without in-band SI except for MPEG PSI (referenced as "TS optional SI").

"TS Full SI" is intended for the case where the Service Provider selects traditional DVB broadcast digital TV streams (from different sources) and provides them as they are over IP to the end-user, in the same way that DTV operators aggregate satellite-received streams over cable. In such a case, the minimum amount of information that the Service Provider has to generate specifically for IP delivery is the information needed at the receiver end to be able to locate the different transport streams (similar to the information needed for the scanning phase in cable, satellite or terrestrial networks). Information on individual services is afterwards acquired from the transport stream itself through classical use of service information as defined in DVB-SI [1].

"TS - Optional SI" is intended for the more advanced situation where the Service Provider wants to present its offering but where it cannot afford or does not want to use bandwidth for usual DVB service description information. In that case, the service discovery information has to give the location of the service as well as relevant service information about each service.

The Broadband Content Guide [62] provides CoD and CDS information and program schedule information for Live Media Broadcast services.

CDSs use the BCG for service announcement. Service announcement for CDS is introduced in clause 10.3

Two transport mechanisms are defined to support both push and pull models of delivery for the service discovery information. Both unicast and multicast modes are supported and the same information can be carried over both modes.

The service discovery information shall be represented with and carried as XML records [54] and the XML schemas [55] to [57] describing their syntax and grammar are specified in clause C.1.

5.2 Service Discovery

5.2.1 Service Identification

This clause defines the mechanisms used to identify service providers and services in the context of service discovery.

5.2.1.1 Service Provider (SP)

A SP shall be identified uniquely by the name of the DNS Domain it has registered and controls. The organizations administrating the Internet DNS domain names shall be used as a globally unique registration mechanism that allows these textual SP identifiers to be globally unique names.

5.2.1.2 Service name or service ID

Each service shall be assigned one textual identifier that takes the form of an Internet DNS host name under the DNS domain that the SP controls. Thus a service can be uniquely identified by a concatenation of a service name (managed uniquely by the SP) and the SP's domain name.

The syntax of a textual service identifier is as defined in MHP (clause 14.9 [3]):

<service_name>"."<service_provider_domain_name>

where <**service_name**> is a unique name for the service within the SP's domain and <**service_provider_domain_name**> is an Internet DNS domain name that the SP has rights to control. The <**service_name**> field shall follow the rules defined for Internet DNS names so that the whole textual service identifier is a valid host name to be used in the Internet DNS as defined in RFC 1035 [14].

There are two basic mechanisms for uniquely identifying a service:

- the triplet of numeric identifiers: original_network_id, transport_stream_id and service_id as defined in DVB SI [1];
- a textual service identifier, as defined above.

Either form can be used for identifying a service globally and uniquely.

It should be noted that the DVB triplet (original_network_id, transport_stream_id and service_id) distinguishes between the same service carried by different networks. For example the triplet would consider the channel BBC1 carried by BskyB and by Freeview as two separate services.

For example, the SP CANAL+ is identified by the domain name "canal-plus.com" and a service can be assigned the name "canalplussport.canal-plus.com".

5.2.2 Fragmentation of SD&S Records

5.2.2.1 SD&S Information data types

The following different information types are specified; additional information types may be added in the future:

- SD&S information relating to a SP.
- four types of SD&S information relating to the service offering of a SP.
- Broadband Content Guide Discovery record.
- Regionalisation Discovery record to provide for local services.
- Firmware Announcement Information to allow for upgrade or changes to HNED firmware.

This is to cover the different types of service offering a SP may have. A SP offering can be made up of Live Media Broadcast services ("TS Full SI" or "TS Optional SI" records) or CoD (via the BCG Discovery record). The SP can also reference services provided by another SP or define a package if it chooses to group several services and present them as a single entity.

These different types of SD&S information shall be identified by an 8-bit value called payload ID.

Table 1 lists the different types of SD&S information a SP may use and give the associated value the payload ID takes. The table includes in addition payload IDs for information outside of SD&S (e.g. CDS XML download session descriptions, BCG data) which use the same transport mechanisms for XML data (e.g. DVBSTP).

Table 1: Payload ID values

Payload ID value	SD&S record carried
0x00	Reserved
0x01	SD&S Service Provider Discovery Information
0x02	SD&S Broadcast Discovery Information
0x03	SD&S COD Discovery Information
0x04	SD&S Services from other SPs
0x05	SD&S Package Discovery Information
0x06	SD&S BCG Discovery Information
0x07	SD&S Regionalisation Discovery Information
0x08	FUS Stub file and SD&S RMS-FUS record
0x09 to 0xA0	Reserved
0xA1 to 0xAF	BCG Payload ID values defined in [62]
0xB0	Reserved
0xB1	CDS XML download session description
0xB2	RMS-FUS Firmware Update Announcements [79]
0xB3 to 0xBF	Reserved
0xC1	Application Discovery Information
0xC2 to 0xEF	Reserved
0xF0 to 0xFF	User Private

5.2.2.2 Fragmentation of SD&S records

The SD&S XML records may be of a substantial size, but only part of them are needed by an HNED at any one time. Also, changes to the SD&S records may be localized to part of the records. For these reasons segments shall be supported to allow an SD&S record to be managed as a collection of smaller units. Segments are defined in the context of a single type of SD&S information, i.e. segments are defined for a declared payload ID.

Each segment shall be assigned a segment Id to identify a segment of data for the declared SD&S data type (payload ID). The segment Id shall be a 16-bit value. A segment shall be a well formed and valid XML record.

An 8-bit value shall be used to define the current version of a segment, this version shall be keyed on payload ID together with segment Id. Thus when the data within a segment changes, its version number called segment version shall be incremented. The segment versions of the unchanged segments do not need to change. The segment version is modulo 256, and wraps round.

Records containing SP discovery information (i.e. PayloadID 0x01) shall not be segmented when using the "pull mode". In all other cases, the XML records shall be segmented. Note that a record may be divided into a single segment.

Guidelines on how XML records should be divided into segments are provided with the XML definitions of the records in clause C.1.6.

Figure 6 illustrates the relationship between segments, payload ID and records.

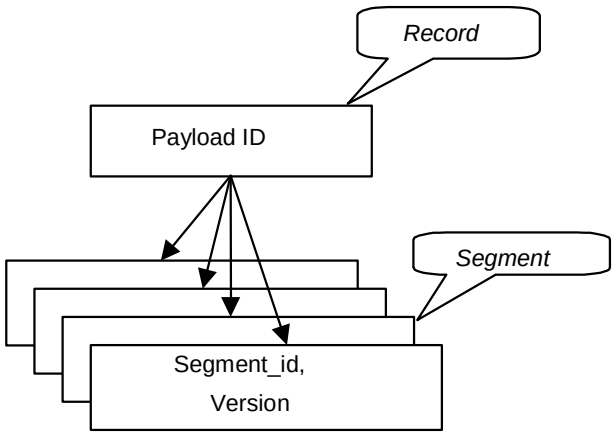


Figure 6: Relationship between records, payload IDs and segments

5.2.2.3 Maximum cycle time

The length of time required to transmit all the segments making up the full set of SD&S Information for a SP is called the Cycle Time. The Maximum Cycle Time shall be set to 30 s.

5.2.3 Steps in service discovery

The service discovery process begins with the discovery of SPs offering DVB-IPTV services over the IP network and continues with the discovery of available services from each SP. The SD&S data model can be found in annex B.

The service discovery process shall bootstrap itself by determining the entry point(s) of the discovery information. This will be specified in clause 5.2.4.

The discovery of SPs offering DVB-IPTV services is done via the acquisition of the SP Discovery Information specified in clause 5.2.5. SPs will publish their offering via the service discovery information as specified in clause 5.2.6.

Figure 7 summarizes the steps of the Service Discovery process. Each step is further described in separate clauses below.

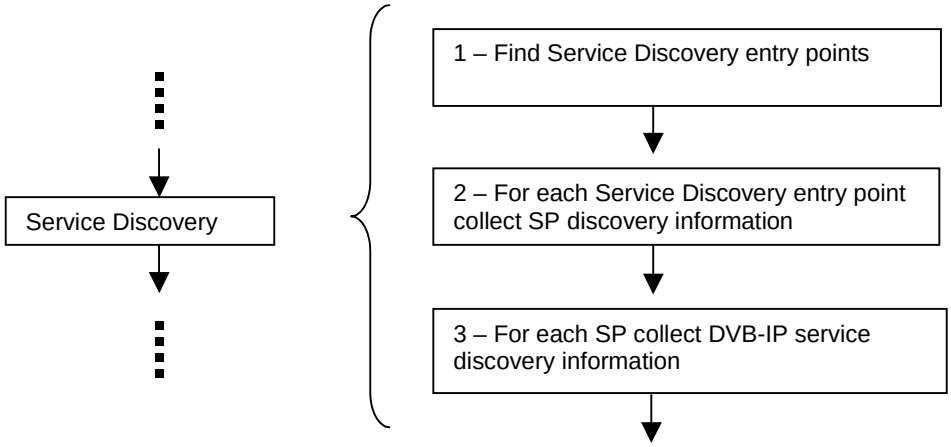


Figure 7: Steps in service discovery

5.2.4 Service discovery entry points

The service discovery process shall bootstrap itself by determining the entry point(s) of the discovery information. The SD&S entry points can be one of the following:

- A well known multicast address registered with IANA that is 224.0.23.14 (DvbServDisc).
- A list of SD&S entry points addresses may be acquired via DNS according to the service location RFC 2782 [40]. The service name is _dvbservdsc, the protocol may be tcp or udp, while the rest of the name is the domain name maintained by DVB for service discovery; this domain name is set to services.dvb.org. So the lookup shall be either _dvbservdsc._tcp.services.dvb.org or _dvbservdsc._udp.services.dvb.org. This requires that the HNED support an SRV cognizant DNS client and according to the specification in RFC 2782 [40]. The DVB organization will maintain the services.dvb.org domain name for service discovery and new SPs should register with DVB to add them to the DNS SRV list. HTTP servers will be found via the tcp protocol method whilst the multicast addresses will be found via the udp protocol method.
- When the HNED connects to the network to request its own address (e.g. during DHCP) it may be provided with domain names via DHCP option 15. A list of SD&S entry points addresses is then acquired via DNS according to the service location RFC 2782 [40] as described above. The service name is _dvbservdsc, the protocol may be tcp or udp, while the rest of the name is the domain name provided via DHCP Option 15. For example the lookup could be _dvbservdsc._tcp.example.com. This requires that the HNED support an SRV cognizant DNS client according to the specification in RFC 2782 [40].

NOTE: The DNS mechanism as described in RFC 2782 [40] may be used in a recursive fashion, i.e. the domain names returned can include ones starting with _dvbservdsc in which case further DNS SRV methods are required to locate the final domain names.

If no portnumber is specified, the default portnumber shall be 3937 (dvbservdsc) as assigned by IANA.

The HNED shall look for SD&S entry points in the priority order defined below. When one of the steps below provides at least one entry point then the HNED shall stop searching for new entry points:

1) The domain names returned by DHCP option 15 shall be used in conjunction with the DNS mechanism defined above. If the method does not resolve to one or more valid domain names or returns an error, then the HNED shall go to the next step.

2) The HNED joins the IANA registered multicast address; if no valid DVBSTP packets are received within a minimum period of 2 cycles of SD&S Information delivery (maximum cycle period specified in clause 5.2.2.3) then the HNED shall go to the next step.

3) The DVB constructed DNS method defined above shall be used, if it does not resolve to one or more valid domain names or returns an error, then the HNED shall go to the next step.

4) If no entry point has been found through the steps above there shall be the option for the user to enter the URL [19] or an IP address and optional portnumber of an entry point manually.

5.2.5 SP discovery information

The first stage in the service discovery is the SP discovery phase. This enables the discovery of SPs offering DVB-IPTV services on the network and the acquisition of the location information of the various SPs' offering(s).

This SP Discovery Record shall be carried in a record containing the information listed in table 2. The SP Discovery Information may be multicast (push model) or retrieved on request (pull model). One or both models shall be supported by the server. Both models shall be supported by the client.

A SP Discovery Information record may aggregate discovery information on several SPs. This is intended to be useful when minimizing the number of records acquired, such as when the act of acquiring a record has an overhead associated with it. For example, a single HTTP request could retrieve the complete list of SPs providing DVB-IPTV services on the network.

Table 2: SP(s) discovery record

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional/ Conditional
ServiceDiscovery type:	/ServiceDiscovery	
ServiceDiscovery@Version	Version of this record. A change in this value indicates a change in one of the ServiceProviderDiscovery Records.	O
ServiceProvider type (one entry per service provider):	/ServiceDiscovery/ServiceProviderDiscovery/ServiceProvider	
ServiceProvider@DomainName	An internet DNS domain name registered by the SP that uniquely identifies the SP	M
ServiceProvider@Version	Version of the SP(s) Discovery record; the version number shall be incremented every time a change in any of the records that comprise the service discovery information for this SP occurs.	M
ServiceProvider@LogoURI	Pointer to a SP logo for potential display. The pointer shall be a URI [20].	O
Name	Name of the SP for display in one or more languages; one SP name is allowed per language code, and at least one language shall be provided (though not necessarily more than one).	M
Description	Description of the SP for potential display in one or more languages; one description is allowed per language code.	O
OfferingListType type (one entry per offering):	/ServiceDiscovery/ServiceProviderDiscovery/ServiceProvider/Offering	
Push@Source Push@Address Push@Port	Port number and IP address of the multicast location of the DVB-IPTV Offering Records which describe the offerings that the SP makes available. This element is optional.	O M (see note 1) M (see note 1)
Pull@Location	This URI [20] encodes the location of the DVB-IPTV Offering(s) Records which describe the offerings that the SP makes available.	O
PayloadList type:	/ServiceDiscovery/ServiceProviderDiscovery/ServiceProvider/Offering/Pull/PayloadId /ServiceDiscovery/ServiceProviderDiscovery/ServiceProvider/Offering/Push/PayloadId	
PayloadId@Id	Indicates the type of service discovery information available at the DVB-IPTV offering location. For example, this can be of type broadcast discovery or CoD discovery. The different values of this field are set out in table 1 in clause 5.2.2.1.	O
Segment@ID	Indicates which segment carries service discovery information of type PayloadId@Id for this SP.	C (see note 2)
Segment@Version	Version number of the segment identified by Segment@ID.	O
NOTE 1: The Mandatory here means that if the Optional parent element is transmitted, then this field shall be present. NOTE 2: The list of segment Ids and version number is provided to inform the HNED of the segments making up the DVB-IPTV offering discovery record. This list is mandatory when SD&S information is provided on request (i.e. "pull mode") as this is the only way for the HNED to know what segments to request. This list is optional when multicasting the SD&S information ("push" mode).		

This record implements both the SP Discovery Information and the SP components of the Data Model presented in annex B.

The location of the DVB-IPTV offering is optional to enable a SP to signal its presence even when it is not transmitting any service.

5.2.6 DVB-IPTV service discovery information

5.2.6.1 DVB-IPTV Offering Record

The DVB-IPTV Offering record shall contain at least the fields defined in table 3 followed by fields relating to the actual SP offering. A SP offering is made up of Live Media Broadcast services ("TS Full SI" or "TS Optional SI" records) or CoD (via the BCG Discovery record). The SP can also reference services provided by another SP. The discovery information relating to these referenced services such as the location of the service will need to be acquired directly from the SP providing the service. A SP can also define a package if it chooses to group several services and present them as a single entity.

This DVB-IPTV Offering (OfferingBase XML type) record will not be used, except where it is inherited by one of the subsequent records.

Table 3: DVB-IPTV Offering Record

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional/ Conditional
@DomainName	An internet DNS domain name registered by the SP that uniquely identifies the SP.	M
@Version	Version of the DVB-IPTV Offering record, the version number shall be incremented every time a change in the DVB-IPTV Offering record is made.	C (see note)
NOTE: The version number of the DVB-IPTV offering record is mandatory when the record is provided on request (i.e. "pull mode") and is optional when the record is multicasted (i.e. "push mode").		

This record implements the DVB-IPTV Offering component of the Data Model.

5.2.6.2 Broadcast discovery record

5.2.6.2.1 Broadcast discovery record - TS Full SI

The "TS Full SI" Broadcast Discovery Information Record (BroadcastOffering XML type) is derived from the DVB-IPTV Offering Record. It provides all the necessary information to find available live media broadcast services which have embedded SI. Information on individual services is afterwards acquired from the transport stream itself through classical use of service information as defined in DVB-SI. This record implements the Broadcast Discovery Information [TS Full SI] and the linked Service(s) Location and Service(s) Description Location, and by inheritance the DVB-IPTV Offering components of the Data Model in annex B. This record shall include all attributes in table 3, and in addition shall contain the following fields.

Table 4: "TS Full SI" Discovery Information

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
BroadcastOffering type:	/BroadcastDiscovery	
IPServiceList type (one per service list):	/BroadcastDiscovery/ServiceList	
ServicesDescriptionLocation	If present, this shall contain the identifier(s) of the BCG Record(s) for the BCG Discovery element that carries the information on this offering.	O
ServicesDescriptionLocation @preferred	If present and set to true, specifies that this location contains the preferred BCG. The default value for this attribute is false. There shall be only one preferred BCG.	O
IPService type (one entry per service):	/BroadcastDiscovery/ServiceList/SingleService	
TextualIdentifier@DomainName	An internet DNS domain name registered by the SP that uniquely identifies the SP. If this is not present, then the DNS domain name from the DVB-IPTV Offering record is used.	O
TextualIdentifier@ServiceName	A unique host name for the service within the SP's domain.	M
DVBTriplet@OrigNetId	Identifies the network Id of the originating delivery system.	M
DVBTriplet@TSId	Identifies the Transport Stream.	M

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
DVBTriplet@ServiceId	Identifies a service from any other service within theTS. The service Id is the same as the program number in the corresponding program map table.	M
MaxBitrate	Specifies the maximum bitrate (in kbits/s) of the overall stream carrying the service excluding any FEC or other layers and calculated according to TIAS value in RFC 3980. NOTE: Other layers may be carried on the same multicast address, and appropriate calculations should me made as necessary.	O
ServiceLocation type (one entry per service location):	/BroadcastDiscovery/ServiceList/SingleService/Service Location At least one of IPMulticastAddress or RTSPURL shall be present.	
IPMulticastAddress	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation /IPMulticastAddress Signals the use of IGMP to access the service and provides the transport address and other parameters at which the service may be accessed. At least one of IPMulticastAddress or RTSPURL shall be present.	O
IPMulticastAddress@Source	Optionally the IP unicast address of the source of the TS may be provided.	O
IPMulticastAddress@Address	Provides the multicast group address at which the service may be accessed.	M (see note)
IPMulticastAddress@Port	Provides the port at which the service may be accessed.	M (see note)
IPMulticastAddress@Streaming	Optionally indicates RTP or direct UDP streaming. In case the parameter is not provided, RTP streaming is assumed.	O
CNAME	Optionally provides the canonical name when RTP streaming is used.	O
ssrc	Optionally provides the ssrc identifier value when RTP streaming is used.	O
FECBaseLayer	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation /IPMulticastAddress/FECBaseLayer Contains the multicast address and port of the AL-FEC stream. This element shall be present if the FECEnhancementLayer element is present.	M if FEC used
FECBaseLayer @Address	IP Multicast Address for FEC Base Layer (SMPTE-2022-1 [67]). If the IP multicast address is omitted, then the FEC flow is assumed to be on the same multicast address as the original data.	O if FEC used
FECBaseLayer @Source	IP Multicast Source Address for FEC Base Layer (SMPTE-2022-1 [67]). If the IP multicast source address is omitted, then the FEC flow is assumed to be on the same multicast source address as the original data.	O if FEC used
FECBaseLayer @Port	UDP port for FEC Base Layer.	M if FEC used
FECBaseLayer @MaxBitrate	Specifies the maximum bitrate (in kbits/s) of this Layer of the FEC flow, and calculated according to TIAS value in RFC 3980.	O if FEC used
FECBaseLayer @PayloadTypeNumber	RTP payload type number for FEC Base layer. It shall be 96 (the 1st dynamic payload number) for SMPTE 2022 compatibility. If not present, the value 96 shall be inferred.	O if FEC used
FECBaseLayer @RTSPControlURL	The RTSP URL to be used for RTSP control messages (SETUP) for this FEC Layer.	M if FEC Base Layer used in conjunction with RTSP
FECEnhancementLayer	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation /IPMulticastAddress/FECEnhancementLayer Contains the multicast address and port of the AL-FEC enhancement stream(s). This element shall only be present if the FECBaseLayer element is present. This element may be repeated for multiple layers.	M if FEC Enhancement Layer used
FECEnhancementLayer@Address	IP Multicast Address for FEC Enhancement Layer (Raptor). If the IP multicast address is omitted, then the FEC flow is assumed to be on the same multicast address as the original data.	O if FEC Enhancement Layer used

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
FECEnhancementLayer@Source	IP Multicast Source Address for FEC Enhancement Layer (Raptor). If the IP multicast source address is omitted, then the FEC flow is assumed to be on the same multicast source address as the original data.	O if FEC Enhancement Layer used
FECEnhancementLayer@Port	UDP port for FEC Enhancement Layer.	M if FEC Enhancement Layer used
FECEnhancementLayer@MaxBitrate	Specifies the maximum bitrate (in kbits/s) of this Layer of the FEC flow, calculated according to TIAS value in RFC 3980 [107].	O if FEC Enhancement Layer used
FECEnhancementLayer@PayloadTypeNumber	RTP payload type number for FEC enhancement layer.	O if FEC Enhancement Layer used
FECEnhancementLayer@TransportProtocol	Transport Protocol of enhancement layer. In the current version the Identifier is restricted to UDP/FEC and RTP/AVP. If this element is omitted, the UDP/FEC shall be assumed for the protocol identifier.	O if FEC Enhancement Layer used
FECEnhancementLayer@RTSPControlURL	The RTSP URL to be used for RTSP control messages (SETUP) for this FEC Layer.	M if FEC Enhancement Layer used in conjunction with RTSP
IPMulticastAddress@FECMaxBlockSizePackets	This indicates the maximum number of stream source packets that will occur between the first packet of a source block (which is included) and the last packet for that source block (source or repair).	O if FEC used
IPMulticastAddress@FECMaxBlockSizeTime	The maximum transmission duration of any FEC Block in milliseconds (source and repair packets)	O if FEC used
IPMulticastAddress@FECObjectTransmissionInformation	The FEC Object Transmission Information for the Raptor code. If an FECEnhancementLayer element is included then this element SHALL be included.	M if FEC Enhancement used
RTPRetransmission	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/ IPMulticastAddress/RTPRetransmission Signals the use of RTP Retransmission (RET) and the parameters associated with retransmission. NOTE: Parameters and attributes whose name starts with "dvb" are defined in the present document, in annex F. Other parameters and attributes are also defined within the relevant references.	M if RET used
RTCPReporting	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/ IPMulticastAddress/RTPRetransmission/RTCPReporting This element signals the transport addresses and parameters associated with the RTCP reporting in the original MC RTP session.	M if RET used
RTCPReporting@ DestinationAddress	List of IP addresses (minimum 1 IP address) OR single DNS SRV RR. The IP address selected from this list by the HNED or resolved by the HNED, is used as the IP Destination Address of RTCP packets issued by HNED, i.e. IP address of RTCP target. At the same time it is the IP Source address of the unicast RTP RET packets. This address value may be overruled by an URL which is achieved either in-band (through RTCP RSI messages) or an URL that is obtained via DHCP.	O if RET used
RTCPReporting@DestinationPort	UDP Destination Port of RTCP packets issued by HNED.	M if RET used
RTCP Reporting@dvb-t-ret	Minimum time a receiver should wait for a requested RET packet (per retransmission request/attempt) before issuing another retransmission request for the same packet(s). This time period has as starting point the sending of the RTCP FB message, and is expressed in milliseconds. If not present, it is up to the HNED to choose an appropriate delay time with which failed retransmissions are re-attempted.	O if RET used

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
RTCP Reporting@rtcp-bandwidth	Amount of bandwidth an RTP receiver may use for its RTCP reporting (kb/s). Default is 5 % of RTP stream bandwidth when this attribute is not present.	O if RET used
RTCP Reporting@rtcp-rsize	Indicates that RTCP FB messages can be transmitted in reduces size format, i.e. as stand-alone RTCP reports (also knows as "non-compound"). Default behaviour is that RTCP FB messages are transmitted as compound RTCP reports.	O if RET used
RTCP Reporting@ trr-int	Minimum period for compound RTCP reporting, in ms. Default value is zero when this attribute is not present.	O if RET used
RTCP Reporting@ dvb-disable-rtcp-rr	Is present when HNED shall disable RTCP RR reporting. Default RTCP RR is enabled when this attribute is not present, i.e. that the default value of this field is "false".	O if RET used
RTCP Reporting@ dvb-enable-bye	When present, HNED shall issue BYE following rules as described in annex F. Default BYE usage is disabled when this attribute is not present.	O if RET used
RTCP Reporting@dvb-t-wait-min	Upon packet loss detection, the HNED shall issue an RTCP FB message in an interval randomly chosen between dvb-t-wait-min and dvb-t-wait-max (both expressed in ms). Default value for dvb-t-wait-min is 0 ms.	O if RET used
RTCP Reporting@dvb-t-wait-max	Upon packet loss detection, the HNED shall issue an RTCP FB message in an interval randomly chosen between dvb-t-wait-min and dvb-t-wait-max (both expressed in ms). Default value for dvb-t-wait-max is 0 ms.	O if RET used
RTCP Reporting@dvb-ssrc-bitmask	Contains a 32-bit wide bitmask. Those HNEDs for which their SSRC match the SSRC inside the original MC streams on the bit positions that are set to 1 in the bitmask, shall set both dvb-t-wait-min and dvb-t-wait-max to zero, overruling the signalled values dvb-t-wait-min and dvb-t-wait-max. Default all bit positions in the bitmask are 1, meaning that the dvb-t-wait-min and dvb-t-wait-max are not overruled.	O if RET used
RTCP Reporting@dvb-rsi-mc-ret	Signals that the RSI packets of the original MC RTP session are distributed in the MC RET session.	O if RET used
RTCP Reporting@dvb-ssrc-upstream-client	SSRC of upstream client for which LMB server translates RTCP FB message into RTCP FF message.	O if RET used
UnicastRET	BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/ IPMulticastAddress/RTPRetransmission/UnicastRET This element signals the transport addresses and parameters associated with the unicast RET session.	M if RET used
UnicastRET@SourcePort	UDP Source Port of unicast RTP RET packets. If not present, the port number in these packets SHALL match "RTPRetransmission/RTCP Reporting@DestinationPort".	O if RET used
UnicastRET@DestinationPort	UDP Destination Port of unicast RTP RET packets. If not present, this port number matches the source port of the RTCP packets issued by the HNED for RTCP reporting in the original session.	O if RET used
UnicastRET@rtx-time	Amount of time (in milliseconds) an RTP packet payload is available for retransmissions.	M if RET used
UnicastRET@rtcp-mux	If present, signals that RTCP and RTP RET packets issued by LMB RET server are multiplexed on the same destination port. If not present, then it follows standard definition, i.e. "RTPRetransmission/UnicastRET@Destination Port" plus one.	O if RET used
UnicastRET@DestinationPortForRTCP Reporting	UDP destination port of RTCP packets issued by HNED in RET session. If this attribute is not present, then RTCP RR on the RET stream shall be disabled by the HNED	O if RET used
UnicastRET@trr-int	Minimum period for compound RET RTCP reporting, in ms. Default value is zero when attribute is not present.	O if RET used

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
UnicastRET@dvb-original-copy-ret	When this is signaled, the identical RTP packet copy format will be used as format for the RET packets. Default the RFC 4588 [86] RET format is used when attribute is not present.	O if RET used
UnicastRET@RTP PayloadTypeNumber	Dynamic RTP payload type number of RET RTP packets. Attribute only present if RFC 4588 [86] RET format is used.	O if RET used
UnicastRET@ssrc	SSRC identifier value in unicast RET RTP packets. Attribute only present if RFC 4588 [86] RET format is used.	O if RET used
UnicastRET@RTSPControlURL	The RTSP URL to be used for controlling the unicast RET stream.	M if RET used in conjunction with RTSP
MulticastRET	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation / IPMulticastAddress/RTPRetransmission/MulticastRET Signals the use of Multicast RTP Retransmission (RET) repair. These parameters are only used if RTP retransmission is enabled and there is a multicast error repair service.	M if MC RET used
MulticastRET@GroupAddress	Single IP address OR single DNS SRV RR representing the IP Group Address of MC RET.	M if MC RET used
MulticastRET@SourceAddress	Single IP address OR single DNS SRV RR. This is the IP Source Address of the MC RTP RET packets. If not present, the IP Source Address of the MC RTP RET packets takes the same value as the IP Source Address of the unicast RTP RET packets.	O if MC RET used
MulticastRET@DestinationPort	UDP Destination port of MC RET RTP Packets.	M if MC RET used
MulticastRET@rtx-time	Amount of time (in milliseconds) an RTP packet payload is available for retransmissions.	M if MC RET used
MulticastRET@rtcp-mux	If present, signals that RTCP and RTP RET packets are multiplexed on the same destination port in the MC RET session. If not present, then it follows standard definition, i.e. "RTPRetransmission/MulticastRET@DestinationPort" plus one.	O if MC RET used
MulticastRET@dvb-original-copy-ret	When this is signaled, the identical RTP packet copy format will be used as format for the MC RET packets. Default the RFC 4588 [86] RET format is used when attribute is not present.	O if MC RET used
MulticastRET@RTPPayloadTypeNumber	Dynamic RTP payload type number of MC RET RTP packets. Attribute only present if RFC 4588 [86] RET format is used.	O if MC RET used
MulticastRET@ssrc	SSRC of MC RET packets. Attribute only present if RFC 4588 [86] RET format is used.	O if MC RET used
RTSPURL	Signals the use of RTSP to access the service and provides the URL at which the service description may be accessed. This URL is also the aggregate URL when control URLs are present for FEC streams. At least one of IPMulticastAddress or RTSPURL shall be present.	O

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
RTSPURL@RTSPControlURL	A URL which can be used to issue RTSP Control commands for the main audio-video stream.	M if RTSP used in conjunction with FEC or RET
AudioAttributes	Signals details of the audio coding algorithms and purpose that the service may use. This shall take the form of the AudioAttributtes element defined in clause 6.3.5 of TS 102 822-3-1 [60] and used in TS 102 323 [59]. The classification scheme used for the Coding element shall either be defined by TS 102 323 [59], or provided by the present document. If this element is omitted, then the default value of MPEG-1 or MPEG-2 layer 2 backwards compatible, mono or stereo shall be used ; specifically this shall be the legacy value from TS 101 154 [58].	O
VideoAttributes	Signals details of the video coding that may be used by the service. This shall take the form of the VideoAttributes element defined in clause 6.3.5 of TS 102 822-3-1 [60] and used in TS 102 323 [59]. The classification scheme used for the Coding element shall either be defined by TS 102 323 [59], or provided by the present document. If this element is omitted, then the default value of MPEG-2 coded video, operating at MP@ML at a frame rate of 25Hz shall be used; specifically this shall be the legacy value from TS 101 154 [58].	O
ServiceAvailability	/BroadcastDiscovery/ServiceList/SingleService/ServiceAvailability This element provides support for regionalisation. It allows each service to have a list of 'cells' (regions) with which the service is associated. By default, all the single services are available everywhere. There shall be at most one ServiceAvailability element for each CountryCode.	O
CountryCode	/BroadcastDiscovery/ServiceList/SingleService/ServiceAvailability/CountryCode This element indicates the country for which the availability is being defined. This element shall be of the 2-letter format specified in ISO-3166 [50].	M (see note)
CountryCode@Availability	This flag indicates whether the service is available in the country specified by CountryCode. The default is TRUE. When TRUE, the service is available in the specified country with the exception of those regions identified by Cells. When FALSE, the service is not available in the specified country with the exception of those regions identified by Cells.	O
Cells	A list of string identifiers representing geographical regions in the country identified by CountryCode. The Cells listed represent the exception to the value supplied by the flag, i.e. the negation of the Availability flag applies to any listed cells.	O
NOTE: The Mandatory here means that if the Optional parent element is transmitted, then this field shall be present.		

5.2.6.2.2 Broadcast discovery record - TS Optional SI

The "TS - Optional SI" Broadcast Discovery Information Record is derived from the DVB-IPTV Offering Record. It provides all the necessary information to create a list of available services with sufficient information for the user to make a choice and gives the necessary information on how to access the service. The "TS Optional SI" Broadcast Discovery Information implements the Broadcast Discovery Information [TS Optional SI] and the linked Service(s) Location and Service Description Location, and by inheritance the DVB-IPTV Offering, components of the Data Model in annex B. The record shall include all attributes in table 3, and in addition shall contain the following fields.

NOTE: The "TS - Optional SI" Broadcast Discovery Information Record is the same as the "TS Full SI" Broadcast Discovery Information Record except for the addition of the SI element.

Table 5: "TS - Optional SI" discovery information

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
BroadcastOffering type:	/BroadcastDiscovery.	
IPServiceList type (one per service list):	/BroadcastDiscovery/ServiceList.	
ServicesDescriptionLocation	If present, this shall contain the identifier(s) of the BCG Record(s) for the BCG Discovery element that carries the information on this offering.	O
ServicesDescriptionLocation @preferred	If present and set to true, specifies that this location contains the preferred BCG. The default value for this attribute is false. There shall be only one preferred BCG.	O
IPService type (one entry per service):	/BroadcastDiscovery/ServiceList/SingleService.	
TextualIdentifier@DomainName	An internet DNS domain name registered by the SP that uniquely identifies the SP. If this is not present, then the DNS domain name from the DVB-IPTV Offering record is used.	O
TextualIdentifier@ServiceName	A unique host name for the service within the SP's domain.	M
DVBTriplet@Original Network Id	Identifies the network Id of the originating delivery system.	M
DVBTriplet@TS Id	Identifies the Transport Stream.	M
DVBTriplet@Service Id	Identifies a service from any other service within the TS. The service Id is the same as the program number in the corresponding program map table.	M
MaxBitrate	Specifies the maximum bitrate (in kbits/s) of the overall stream carrying the service excluding any FEC or other layers and calculated according to TIAS value in RFC 3980 [107]. NOTE: Other layers may be carried on the same multicast address, and appropriate calculations should me made as necessary.	O
ServiceLocation type (one entry per service location):	/BroadcastDiscovery/ServiceList/SingleService/Service Location. At least one of IPMulticastAddress or RTSPURL shall be present.	
IPMulticastAddress	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/IPMulticastAddress. Signals the use of IGMP to access the service and provides the transport address and other parameters at which the service may be accessed. At least one of IPMulticastAddress or RTSPURL shall be present.	O
IPMulticastAddress@Source	Optionally the IP unicast address of the source of the TS may be provided.	O
IPMulticastAddress@Address	Provides the multicast group address at which the service may be accessed.	M (see note)
IPMulticastAddress@Port	Provides the port at which the service may be accessed.	M (see note)
IPMulticastAddress@Streaming	Optionally indicates RTP or direct UDP streaming. In case the parameter is not provided, RTP streaming is assumed.	O
CNAME	Optionally provides the canonical name when RTP streaming is used.	O
ssrc	Optionally provides the ssrc identifier value when RTP streaming is used.	O
FECBaseLayer	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/IPMulticastAddress/FECBaseLayer Contains the multicast address and port of the AL-FEC stream. This element shall be present if the FECEnhancementLayer element is present.	M if FEC used
FECBaseLayer@ Address	IP Multicast Address for FEC Base Layer (SMPTE-2022-1 [67]). If the IP multicast address is omitted, then the FEC flow is assumed to be on the same multicast address as the original data.	O if FEC used
FECBaseLayer@ Source	IP Multicast Source Address for FEC Base Layer (SMPTE-2022-1 [67]). If the IP multicast source address is omitted, then the FEC flow is assumed to be on the same multicast address as the original data.	O if FEC used

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
FECBaseLayer@Port	UDP port for FEC Base Layer.	M if FEC used
FECBaseLayer@MaxBitrate	Specifies the maximum bitrate (in kbits/s) of this Layer of the FEC flow, and calculated according to TIAS value in RFC 3980.	O if FEC used
FECBaseLayer@PayloadTypeNumber	RTP payload type number for FEC Base layer. It shall be 96 (the 1st dynamic payload number) for SMPTE 2022 compatibility. If not present, the value 96 shall be inferred.	O if FEC used
FECBaseLayer@RTSPControlURL	The RTSP URL to be used for RTSP control messages (SETUP) for this FEC Layer.	M if FEC Base Layer used in conjunction with RTSP
FECEnhancementLayer	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/IPMulticastAddress/FECEnhancementLayer Contains the multicast address and port of the AL-FEC enhancement stream(s). This element shall only be present if the FECBaseLayer element is present. This element may be repeated for multiple layers.	M if FEC Enhancement Layer used
FECEnhancementLayer@Address	IP Multicast Address for FEC Enhancement Layer (Raptor). If the IP multicast address is omitted, then the FEC flow is assumed to be on the same multicast address as the original data.	O if FEC Enhancement Layer used
FECEnhancementLayer@Source	IP Multicast Source Address for FEC Enhancement Layer (Raptor). If the IP multicast source address is omitted, then the FEC flow is assumed to be on the same multicast address as the original data.	O if FEC Enhancement Layer used
FECEnhancementLayer@Port	UDP port for FEC Enhancement Layer.	M if FEC Enhancement Layer used
FECEnhancementLayer@MaxBitrate	Specifies the maximum bitrate (in kbits/s) of this Layer of the FEC flow, calculated according to TIAS value in RFC 3980 [107].	O if FEC Enhancement Layer used
FECEnhancementLayer@PayloadTypeNumber	RTP payload type number for FEC enhancement layer.	O if FEC Enhancement Layer used
FECEnhancementLayer@TransportProtocol	Transport Protocol of enhancement layer. In the current version the Identifier is restricted to UDP/FEC and RTP/AVP. If this element is omitted, the UDP/FEC shall be assumed for the protocol identifier.	O if FEC Enhancement Layer used
FECEnhancementLayer@RTSPControlURL	The RTSP URL to be used for RTSP control messages (SETUP) for this FEC Layer.	M if FEC Enhancement Layer used in conjunction with RTSP
IPMulticastAddress@FECMaxBlockSizePackets	This indicates the maximum number of stream source packets that will occur between the first packet of a source block (which is included) and the last packet for that source block (source or repair).	O if FEC used
IPMulticastAddress@FECMaxBlockSizeTime	The maximum transmission duration of any FEC Block in milliseconds (source and repair packets).	O if FEC used
IPMulticastAddress@FECObjectTransmissionInformation	The FEC Object Transmission Information for the Raptor code If an FECEnhancementLayer element is included then this element SHALL be included.	M if FEC Enhancement used
RTPRetransmission	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/IPMulticastAddress/RTPRetransmission Signals the use of RTP Retransmission (RET) and the parameters associated with retransmission. NOTE: Parameters and attributes whose name starts with "dvb" are defined in the present document, in annex F. Other parameters and attributes are also defined within the relevant references.	M if RET used

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
RTCPReporting	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/IPMulticastAddress/RTPRetransmission/RTCPReporting This element signals the transport addresses and parameters associated with the RTCP reporting in the original MC RTP session.	M if RET used
RTCPReporting @ Destination Address	List of IP addresses (minimum 1 IP address) OR single DNS SRV RR. The IP address selected from this list by the HNED or resolved by the HNED, is used as the IP Destination Address of RTCP packets issued by HNED, i.e. IP address of RTCP target. At the same time it is the IP Source address of the unicast RTP RET packets. This address value may be overruled by an URL which is achieved either in-band (through RTCP RSI messages) or an URL that is obtained via DHCP.	O if RET used
RTCPReporting @ DestinationPort	UDP Destination Port of RTCP packets issued by HNED	M if RET used
RTCP Reporting@dvb-t-ret	Minimum time a receiver should wait for a requested RET packet (per retransmission request/attempt) before issuing another retransmission request for the same packet(s). This time period has as starting point the sending of the RTCP FB message, and is expressed in milliseconds. If not present, it is up to the HNED to choose an appropriate delay time with which failed retransmissions are re-attempted.	O if RET used
RTCP Reporting@rtcp-bandwidth	Amount of bandwidth an RTP receiver may use for its RTCP reporting (kb/s). Default is 5 % of RTP stream bandwidth when this attribute is not present.	O if RET used
RTCP Reporting@rtcp-rsize	Indicates that RTCP FB messages can be transmitted in reduces size format, i.e. as stand-alone RTCP reports (also knows as "non-compound"). Default behaviour is that RTCP FB messages are transmitted as compound RTCP reports.	O if RET used
RTCP Reporting@ trr-int	Minimum period for compound RTCP reporting, in ms. Default value is zero when this attribute is not present.	O if RET used
RTCP Reporting@ dvb-disable-rtcp-rr	Is present when HNED shall disable RTCP RR reporting. Default RTCP RR is enabled when this attribute is not present, i.e. that the default value of this field is "false".	O if RET used
RTCP Reporting@ dvb-enable-bye	When present, HNED shall issue BYE following rules as described in annex F. Default BYE usage is disabled when this attribute is not present.	O if RET used
RTCP Reporting@dvb-t-wait-min	Upon packet loss detection, the HNED shall issue an RTCP FB message in an interval randomly chosen between dvb-t-wait-min and dvb-t-wait-max (both expressed in ms). Default value for dvb-t-wait-min is 0 ms.	O if RET used
RTCP Reporting@dvb-t-wait-max	Upon packet loss detection, the HNED shall issue an RTCP FB message in an interval randomly chosen between dvb-t-wait-min and dvb-t-wait-max (both expressed in ms). Default value for dvb-t-wait-max is 0 ms.	O if RET used
RTCP Reporting@dvb-ssrc-bitmask	Contains a 32-bit wide bitmask. Those HNEDs for which their SSRC match the SSRC inside the original MC streams on the bit positions that are set to 1 in the bitmask, shall set both dvb-t-wait-min and dvb-t-wait-max to zero, overruling the signalled values dvb-t-wait-min and dvb-t-wait-max. Default all bit positions in the bitmask are 1, meaning that the dvb-t-wait-min and dvb-t-wait-max are not overruled.	O if RET used
RTCP Reporting@dvb-rsi-mc-ret	Signals that the RSI packets of the original MC RTP session are distributed in the MC RET session.	O if RET used
RTCP Reporting@dvb-ssrc-upstream-client	SSRC of upstream client for which LMB server translates RTCP FB message into RTCP FF message.	O if RET used

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
UnicastRET	BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/IPMulticastAddress/RTPRetransmission/UnicastRET This element signals the transport addresses and parameters associated with the unicast RET session.	M if RET used
UnicastRET@SourcePort	UDP Source Port of unicast RTP RET packets. If not present, the port number in these packets SHALL match "RTPRetransmission/RTCP Reporting@DestinationPort".	O if RET used
UnicastRET@DestinationPort	UDP Destination Port of unicast RTP RET packets. If not present, this port number matches the source port of the RTCP packets issued by the HNED for RTCP reporting in the original session.	O if RET used
UnicastRET@rtx-time	Amount of time (in milliseconds) an RTP packet payload is available for retransmissions.	M if RET used
UnicastRET@rtcp-mux	If present, signals that RTCP and RTP RET packets issued by LMB RET server are multiplexed on the same destination port. If not present, then it follows standard definition, i.e. "RTPRetransmission/UnicastRET@Destination Port" plus one.	O if RET used
UnicastRET@DestinationPortForRTCP Reporting	UDP destination port of RTCP packets issued by HNED in RET session. If this attribute is not present, then RTCP RR on the RET stream shall be disabled by the HNED.	O if RET used
UnicastRET@trr-int	Minimum period for compound RET RTCP reporting, in ms. Default value is zero when attribute is not present.	O if RET used
UnicastRET@dvb-original-copy-ret	When this is signaled, the identical RTP packet copy format will be used as format for the RET packets. Default the RFC 4588 [86] RET format is used when attribute is not present.	O if RET used
UnicastRET@RTP PayloadTypeNumber	Dynamic RTP payload type number of RET RTP packets. Attribute only present if RFC 4588 [86] RET format is used.	O if RET used
UnicastRET@ssrc	SSRC identifier value in unicast RET RTP packets. Attribute only present if RFC 4588 [86] RET format is used.	O if RET used
UnicastRET@RTSPControlURL	The RTSP URL to be used for controlling the unicast RET stream.	M if RET used in conjunction with RTSP
MulticastRET	/BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/IPMulticastAddress/RTPRetransmission/MulticastRET Signals the use of Multicast RTP Retransmission (RET) repair. These parameters are only used if RTP retransmission is enabled and there is a multicast error repair service.	M if MC RET used
MulticastRET@GroupAddress	Single IP address OR single DNS SRV RR representing the IP Group Address of MC RET.	M if MC RET used
MulticastRET@SourceAddress	Single IP address OR single DNS SRV RR. This is the IP Source Address of the MC RTP RET packets. If not present, the IP Source Address of the MC RTP RET packets takes the same value as the IP Source Address of the unicast RTP RET packets.	O if MC RET used
MulticastRET@DestinationPort	UDP Destination port of MC RET RTP Packets.	M if MC RET used
MulticastRET@rtx-time	Amount of time (in milliseconds) an RTP packet payload is available for retransmissions.	M if MC RET used
MulticastRET@rtcp-mux	If present, signals that RTCP and RTP RET packets are multiplexed on the same destination port in the MC RET session. If not present, then it follows standard definition, i.e. "RTPRetransmission/MulticastRET@DestinationPort" plus one.	O if MC RET used
MulticastRET@dvb-original-copy-ret	When this is signaled, the identical RTP packet copy format will be used as format for the MC RET packets. Default the RFC 4588 [86] RET format is used when attribute is not present.	O if MC RET used

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
MulticastRET@RTPPayloadTypeNumber	Dynamic RTP payload type number of MC RET RTP packets. Attribute only present if RFC 4588 [86] RET format is used.	O if MC RET used
MulticastRET@ssrc	SSRC of MC RET packets. Attribute only present if RFC 4588 [86] RET format is used.	O if MC RET used
RTSPURL	Signals the use of RTSP to access the service and provides the URL at which the service description may be accessed. This URL is also the aggregate URL when control URLs are present for FEC streams. At least one of IPMulticastAddress or RTSPURL shall be present.	O
RTSPURL@RTSPControlURL	A URL which can be used to issue RTSP Control commands for the main audio-video stream.	M if RTSP used in conjunction with FEC or RET
SI type:	/BroadcastDiscovery/ServiceList/SingleService/SI	
SI@ServiceType	Specifies the type of service; it shall be coded as per DVB SI standard [1]. Examples are digital television service, digital radio sound service, mosaic service, data broadcast service, DVB MHP service, etc.	M (see note)
SI@PrimarySISource	Indicates which source of service information to give priority (XML record or DVB SI) in case DVB SI tables are present.	O
Name	Name of the service for display in one or more languages; one Service name is allowed per language code, and at least one language shall be provided (though not necessarily more than one).	M
Description	Description of the service for potential display in one or more languages; one description per language code maximum.	O
ServiceDescriptionLocation	If present, this shall contain the identifier(s) of the BCG Record(s) for the BCG Discovery element that carries the information on this service. If this element is present, it shall be used in preference to the ServicesDescriptionLocation.	O
ServiceDescriptionLocation@preferred	If present and set to true, specifies that this location contains the preferred BCG. The default value for this attribute is false. There shall be only one preferred BCG.	O
ContentGenre	Indicates one or more genre of the service (not individual programmes). For example movie/drama channel or news/current affairs channel. This shall use the first level coding defined by DVB 1 as content_nibble_level_1.	O
CountryAvailability	Gives a list of countries and/or groups of countries where the service is intended to be available, and/or a list of countries and/or groups where it is not. This field is deprecated and Service Availability should be used instead.	Deprecated
AnnouncementSupport	The announcement support element identifies the type of spoken announcements that are supported by the service (for example emergency flash, road traffic flash, etc.). Furthermore, it informs about the transport method of the announcement and gives the necessary linkage information so that the announcement stream can be monitored.	O
Replacement Service	Identifies a service replacement service which may be selected automatically by the HNED when the service being decoded fails.	O
MosaicDescription	The mosaic description element identifies the elementary cells of a mosaic service, groups different elementary cells to form logical cells, and establishes a link between the content of all or part of the logical cell and the corresponding service or package information.	O

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
AudioAttributes	Signals details of the audio coding algorithms and purpose that the service may use. This shall take the form of the AudioAttributtes element defined in clause 6.3.5 of TS 102 822-3-1 [60] and used in TS 102 323 [59]. The classification scheme used for the Coding element shall either be defined by TS 102 323 [59], or provided by the present document. If this element is omitted, then the default value of MPEG-1 or MPEG-2 layer 2 backwards compatible, mono or stereo shall be used; specifically this shall be the legacy value from TS 101 154 [58].	O
VideoAttributes	Signals details of the video coding that may be used by the service. This shall take the form of the VideoAttributes element defined in clause 6.3.5 of TS 102 822-3-1 [60] and used in TS 102 323 [59]. The classification scheme used for the Coding element shall either be defined by TS 102 323 [59], or provided by the present document. If this element is omitted, then the default value of MPEG-2 coded video, operating at MP@ML at a frame rate of 25 Hz shall be used; specifically this shall be the legacy value from TS 101 154 [58].	O
ServiceAvailability	/BroadcastDiscovery/ServiceList/SingleService/ ServiceAvailability This element provides support for regionalisation. It allows each service to have a list of 'cells' (regions) with which the service is associated. By default, all the single services are available everywhere. There shall be at most one ServiceAvailability element for each CountryCode.	O
CountryCode	/BroadcastDiscovery/ServiceList/SingleService/ ServiceAvailability/CountryCode This element indicates the country for which the availability is being defined. This element shall be of the 2-letter format specified in ISO 3166 [50].	M (see note)
CountryCode@Availability	This flag indicates whether the service is available in the country specified by CountryCode. The default is TRUE. When TRUE, the service is available in the specified country with the exception of those regions identified by Cells. When FALSE, the service is not available in the specified country with the exception of those regions identified by Cells.	O
Cells	A list of string identifiers representing geographical regions in the country identified by CountryCode. The Cells listed represent the exception to the value supplied by the flag, i.e. the negation of the Availability flag applies to any listed cells.	O
NOTE: The Mandatory here means that if the Optional parent information is transmitted, then this field shall be present.		

By default, the IP Service Discovery Information shall take precedence over the DVB SI tables when present in the transport stream.

5.2.6.3 Content on Demand (CoD) discovery record

Note that the use of this Record is deprecated and should not be used. The Broadband Content Guide Record (clause 5.2.6.6) should be used instead. This record is retained solely for legacy reasons.

The Content on Demand Discovery Record provides all the necessary information to discover the CoD servers available on the network and the location of their catalogue of contents. It does not provide any information on individual contents. The Content on Demand Discovery Record implements the CoD Discovery Information and Content Description Location, and by inheritance the DVB-IPTV Offering, components of the Data Model in annex B. The component Content Location is deliberately not implemented; it is intended that this information is retrieved from the provider, possibly after negotiation. The record shall include all attributes in table 3, and in addition shall contain the following fields.

Table 6: Content on demand discovery record

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
CoDOffering type:	/CoDDiscovery.	
Catalogue@Id	Identifies a CoD Provider/Server; This Id is allocated by the SP.	M
Name	Name of the CoD offering catalogue for display in one or more languages; one name is allowed per language code, and at least one language shall be provided (though not necessarily more than one).	M
Description	Description of the CoD general offering catalogue for potential display in one or more languages; one description per language code.	O
Locator	One or more URI [20] where the aggregated content descriptions can be found (catalogue/metadata).	M

An HTTP request on the " Locator" URI [20] shall return a record compliant to a schema that will be specified in a later revision of the present document.

5.2.6.4 "Service From other Services Providers" record

A SP can reference individual services or a complete offering provided by another SP. Supplying its textual service identifier references a service. Supplying the SP's DNS domain name without a service list references an entire SP's offering. Discovery information relating to a service, or SP, such as the location of the service will need to be acquired directly from the SP providing the service, and is not "pointed to" from this record.

The "Services From other SPs" Record implements the Services From other SPs and linked Service Id, and by inheritance the DVB-IPTV Offering, components of the Data Model in annex B. The record shall include all attributes in table 3, and in addition shall contain the following fields.

Table 7: Services from other SPs record

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
ReferencedServices type:	/ServicesFromOtherSP.	
@Domain	An internet DNS domain name registered by the referenced SP that uniquely identifies the SP being referenced.	M
Service@Name	A unique host name for the service within the referenced SP's domain for each service from the referenced provider. Not required if the entire set of offerings from the SP is referenced.	O

5.2.6.5 Package discovery record

The Package Discovery Record provides a means for a collection of services to be marketed as, or grouped into, a single entity.

The Package Discovery Record implements the Package Discovery Information, linked Service Id and Description Location, and by inheritance the DVB-IPTV Offering, components of the Data Model in annex B. The record shall include all attributes in table 3, and in addition shall contain the following fields.

Table 8: Package discovery information

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
Packaged Services type:	/PackageDiscovery.	
Package@Id	Identifies a package; this ID is allocated by the SP.	M
Package@Visible	A Boolean which indicates in combination with the PackageAvailability element, whether this package shall be presented to the user. The default value is true.	O
PackageName	Name of the package for display in one or more languages; one name per language code maximum.	M
PackageDescription	If present, this shall contain the identifier(s) of the BCG Record(s) for the BCG Discovery element that carries the information on this package.	O
PackageDescription@preferred	If present and set to true, specifies that this location contains the preferred BCG. The default value for this attribute is false. There shall be only one preferred BCG.	O
CountryAvailability	Gives a list of countries and/or groups of countries where the package is intended to be available, and/or a list of countries and/or groups where it is not. This field is deprecated and Package Availability should be used instead.	Deprecated
PackageReference	This shall be the Id(s) of package(s) that are included in the current package.	O
Service	List of services forming the package, comprising:	M
TextualID@ DomainName	An internet DNS domain name registered by the SP that uniquely identifies the SP. If this is omitted the SP Domain Name from the inherited DVB-IPTV Offering is used.	O
TextualID@ServiceName	A unique host name for the service within the SP's domain.	M
DVBTriplet	The DVB triplet by which the service may be known.	O
DescriptionLocation	The URI [20] of additional service description provided in the context of a package; this is not required to acquire a service.	O
DescriptionLocation @preferred	A Boolean flag which indicates if the description location indicated is the preferred location for this information. Only one location for any service may have this flag set to true.	O
LogicalChannelNumber	The logical channel number of the service.	O
PackageAvailability	This element provides support for regionalisation. It allows each package to have a list of 'cells' (regions) with which the package is associated. By default, the package is available everywhere. There shall be at most one PackageAvailability element for each CountryCode.	O
CountryCode	This element indicates the country for which the availability is being defined. This element shall be of the 2-letter format specified in ISO 3166 [50].	M
CountryCode@Availability	This flag indicates whether the package is available in the country specified by CountryCode. The default is TRUE. When TRUE, the package is available in the specified country with the exception of those regions identified by Cells. When FALSE, the package is not available in the specified country with the exception of those regions identified by Cells.	O
Cells	A list of string identifiers representing geographical regions in the country identified by CountryCode. The Cells listed represent the exception to the value supplied by the flag, i.e. the negation of the Availability flag applies to any listed cells.	O

A service may belong to more than one package. A service does not have to be part of any package.

The package discovery information does not enable the discovery of new services. Discovery information relating to a service, or SP, such as the location of the service will need to be acquired directly from the SP providing the service, and is not "pointed to" from this record.

Additional information on services can optionally be provided in the context of a package.

Where the PackageAvailability element is used, there may be multiple packages transmitted, each one corresponding to a specific set of regions. However, for any given HNED there shall only be a single package that both has the Visible attribute set to true and that has the PackageAvailability element that match the values held by the HNED.

NOTE: This means that once an HNED has found a visible package that matches the CountryCode, and if present Cell, values, the HNED has found the package it should use.

A package may include another package using the PackageReference element, in which case the Visible attribute of the referenced package shall be ignored.

5.2.6.6 Broadband Content Guide record

The Broadband Content Guide Record provides a means to discover the locations of guides listing the content that is available, either live (e.g. through a Broadcast Offering) or via CoD or via CDSs. A provider discovered through this shall offer a service as described in TS 102 539 [62].

For CDSs the location of download session descriptions distributed via multicast can be provided. This allows a HNED to cache the download session descriptions distributed via multicast (see clause 10.4.2).

Table 9: Broadband Content Guide Discovery record

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
BCGOffering type:	/BCGDiscovery.	
BCG	BCG record.	M
BCG@Id	Identifies a Broadband Content Guide Provider/Server; this Id is allocated by the SP.	M
BCG@Version	Version of this record. A change in this value indicates a change in one of the BCG Records.	O
Name	Name of the Broadband Content Guide offering for display in one or more languages; one name is allowed per language code, and at least one language shall be provided (though not necessarily more than one).	M
Description	Description of the Broadband Content Guide for potential display in one or more languages; one description per language code.	O
Logo	A pointer to an optional logo for the content guide.	O
Type	This indicates if the content guide relates to live programs, CoD, both, or some other form of content. The extensible classification scheme provided in the present document shall be used.	O
TargetProvider	The domain name of the provider whose content is described by this BCG (for example Canal+). The domain name shall be the same as a domain name present in the ServiceList.	O
TransportMode	The location where the broadband content guide may be found.	M
DVBSTP	Specifies the location at which the content guide is available using the DVBSTP protocol, and details the relevant segments that are being transmitted.	O
HTTP@Location	Specifies the location at which the guide may be found.	M (if the HTTP element is used)
HTTP@SOAP	Indicates if the guide may be queried using the SOAP protocol rather the mechanism outline in clause 5.4.2. The default value of this attribute is "false".	O
PayloadList type (one entry per payload ID)	/BCGDiscovery/BCG/TransportMode/DVBSTP/PayloadId /BCGDiscovery/BCG/TransportMode/HTTP/PayloadId	
PayloadId@Id	Indicates the type of service discovery information available at the DVB-IPTV offering location. The different values of this field are set out in clause 4.1.2.1 of TS 102 539 [62].	C (see note 1)
Segment@ID	Indicates which segment carries BCG data delivery unit of type PayloadId@Id for this BCG provider.	C (see note 2)
Segment@Version	Version number of the segment identified by Segment@ID.	O
BCGProviderName	The name of the BCG provider (for example "Telerama"). This field shall be identical to the textual string of the Publisher attribute of the TVAMain element in the BCG metadata	O
CDSDownloadSessionDescriptionLocation (one entry per	Specifies locations at which CDS download session descriptions are available via multicast distribution.	O

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
multicast channel)		
DVBSTP	Specifies the location (multicast address, port and source address) at which CDS XML download session descriptions are available using the DVBSTP protocol and details the relevant segments.	O
SAP	Specifies the location (multicast address, port and source address) at which CDS SDP download session descriptions are available using the SAP protocol.	O
NOTE 1: The PayloadList is mandatory when the transport mode is HTTP and it is a container-based delivery (i.e. it is not a SOAP server; the SOAP attribute is not present or is set to "false"). The PayloadList is optional when the transport mode is DVBSTP. The PayloadList shall not be used when the transport mode is HTTP and a SOAP attribute is set to "true". NOTE 2: The Segment element is mandatory when the transport mode is HTTP and the SOAP attribute is not present or set to "false".		

There are two means for locating the BCG of a given Service (typically the current broadcast channel being watched by the user):

- by finding the BCG ID in the ServiceList or SingleService element where the current service is located (using the ServicesDescriptionLocation or ServiceDescriptionLocation field);
- by finding the ServiceProvider domainName of the current service and parsing all BCGs in BCGDiscovery to find a matching TargetProvider.

In the case where there are several references to BCGs in the ServiceList or SingleService, the preferred BCG is optionally signalled using a boolean attribute "preferred".

5.2.6.7 HNED Cell ID Discovery – Regionalisation Discovery Record

An HNED is located geographically in a region which is defined by the SP and identified using a string identifier called a Cell ID which is unique within a country (i.e. the location of the HNED can be defined using the country code and Cell ID together). This identifier is used in the PackageAvailability element in table 8 and the ServiceAvailability element in tables 4 and 5 to indicate which package and services can be received by the HNED.

The HNED obtains its location from the DHCP server via the DHCP option 99 RFC 4676 [98] as described in clause 8.1.1.11. The HNED can then retrieve its Cell ID by:

- either requesting it by sending its location information to a server - URI retrieved from the Regionalisation Offering (Pull mode see clause 5.2.6.7.1);
- or matching the location information against the table pushed to it on an address available from the Regionalisation Information record (Push mode see clause 5.2.6.7.2).

The resulting identifier is the Cell ID which, in combination with the country code, can be matched against information in the PackageAvailability and ServiceAvailability elements to show which package and services can be received by the HNED in its region.

Unlike the rest of Service Discovery and Selection information, the format of the Regionalisation information received by the HNED is different depending on the use of Push or Pull modes, (i.e. the data transmitted in the Push or Pull mode is not interchangeable). However, the way the Push and Pull locations are advertised in the Regionalisation Offering is identical to other SD&S Offerings. The Regionalisation Offering within the SP Discovery Record shall provide the IP address of either the Push or the Pull location, or both. If both a Push and Pull locations are provided then the HNED shall use the Pull location first and only on failure try the Push location. It is recommended to use pull mode.

If no Regionalisation Offering is defined within the SP Discovery information, then the HNED shall use the domain name of the SP as the Pull location and follow the process defined for the Pull mode to try and retrieve its Cell ID, unless the relevant regionalisation information is already available to the HNED through some other proprietary means.

5.2.6.7.1 Obtaining the Cell ID via HTTP (Pull mode)

The pull mode uses the HTTP protocol with the HTTP POST method. The HNED sends a POST message to the URI retrieved from either the Regionalisation Offering or to the domain name in the SP Discovery Record if there is no Regionalisation Offering present. The body of the Post method shall include the Country Code and all Civic Address information retrieved via DHCP option 99 (see clause 8.1.1.11). The server replies to the HTTP POST with the Cell ID defined for the location.

The POST message shall be in the following format and shall use the HTTP congestion avoidance mechanism specified in clause 9.1.2. The POST Request may contain other headers conforming to RFC 2616 [39].

```
'POST /dvb/sdns/CellID HTTP/1.1' CRLF
'Host: ' host CRLF
'Content-type: text/xml' CRLF
<?xml version="1.0" encoding="UTF-8"?>
<CountryCode>"CountryCode"</CountryCode>
<CA type="CAtype" value="CAtype Value">
    <CA type="CAtype" value="CAtype Value">
        ...
            <CA type="CAtype" value="CAtype Value"/>
        ...
    </CA>
</CA>
```

Where:

host is the domain name used to address the POST

CountryCode is the value retrieved from the DHCP option 99, for example "FR" for France

A list of CA XML elements where:

- **CAtype** is the CAtype number used to identify the CAtype value obtained from the DHCP option 99, for example "24" for postcode/zipcode.
- **CAtype Value** is the value of the CAtype, for example for a CAtype of postcode/zipcode, the value can be "75011" in Paris, France or "95134" in California, USA.

The CA elements can be listed in any order, but it is recommended to follow the same order as in the DHCP message.

When the DHCP GEOCONF_CIVIC data have multiple languages set of parameters (CAtype="0"), the HNED shall provide all of them to retrieve the Cell ID.

The response from the HTTP server shall be:

```
'HTTP 200 OK' CRLF
'Content-type: text/xml' CRLF
'Date: ' "date time" CRLF
<?xml version="1.0" encoding="UTF-8"?>
<Cell ID="Cell ID"/>
```

Where:

date time is the date and time, for example 2002-09-26 18:35:39 UTC

Cell ID is the resulting Cell ID that the SP has supplied for the location provided by the CAType values.

NOTE: This may be a string such as "Paris" or a number such as "42".

Following is an example where the DHCP GEOCONF_CIVIC option has supplied the location of the HNED as:

- Country code = "FR"
- CAtype = "0", CAvalue = "fr"
- CAtype = "128", CAvalue = "Latn"
- CAtype = "1", CAvalue = "IDF"
- CAtype = "3", CAvalue = "Paris"

- CAtype = "24", CAvalue = "75011"
- CAtype = "132", CAvalue = "private CA parameter"

The resulting request from the HNED is:

```
POST /dvb/sdns/CellID HTTP/1.1
Host: cellid.tv5.fr
Content-type: text/xml
<?xml version="1.0" encoding="UTF-8"?>
<CountryCode>FR</CountryCode>
<CA type="0" value="fr">
    <CA type="128" value="Latn">
        <CA type="1" value="IDF">
            <CA type="3" value="Paris">
                <CA type="24" value="75011">
                    <CA type="132" value="private CA parameter"/>
                </CA>
            </CA>
        </CA>
    </CA>
</CA>
```

And the response is:

```
HTTP 200 OK
Content-type: text/xml
Date: "2009-01-26 18:35:39 UTC"
<?xml version="1.0" encoding="UTF-8"?>
<Cell ID="Paris East"/>
```

5.2.6.7.2 Obtaining the Cell ID via the Regionalisation Discovery Record (Push mode)

The Regionalisation Discovery Record allows CellIDs to be sent to all HNEDs to allow for a pure push model. The HNED then checks the DHCP GEOCONF_CIVIC option 99 content and matches it against the table to find the Cell ID for its location.

The Regionalisation Discovery Record implements the Regionalisation Discovery Information and by inheritance the DVB-IPTV Offering components of the Data Model in annex B. The record shall include all attributes in table 3, and in addition shall contain the fields in table 10.

Table 10: Regionalisation Discovery Information

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
RegionalisationOffering type:	/RegionalisationDiscovery.	
Cell	/RegionalisationDiscovery/Cell.	
Cell@ID	CellID value corresponding to the Standard Location parameters as provided in the following CA elements.	M
CountryCode	/RegionalisationDiscovery/Cell/CountryCode Country Code for this cell. This element shall be of the 2-letter format specified in ISO 3166 [50].	M
CA	/RegionalisationDiscovery/Cell/CA Standard Location parameter element. Several values are allowed for the same CA type attribute. When another CA type attribute is included, it is done as a sub XML element.	M
CA@type	Type of the Civic Address parameter, as specified in RFC 4676 [98].	M
CA@value	Value for the Civic Address type.	M

The Regionalisation Discovery Record can be large so it is recommended to use compression (see table 10). It is not necessary to carry all the CAtypes within the Regionalisation Discovery record. The HNED shall minimize processing time by stopping when the CAtype does not match the value supplied by the DHCP server.

All Civic Address parameters may not be used, and they may be used in any order within the Regionalisation Discovery Record, compared to the DHCP message. Thus the HNED shall parse the XML structure and can stop parsing when a non-matching type/value pair is found. When all type/value pairs are matched, and the end of the CA elements tree has been reached, it means that a match has been found and that the Cell ID has been retrieved.

When the DHCP GEOCONF_CIVIC data have multiple languages set of parameters (CAtype="0"), it is sufficient to match one of these to retrieve the Cell ID.

Following is an example where the DHCP GEOCONF_CIVIC option has supplied the location of the HNED as:

- Country code = "FR".
- CAtype = "0", CAvalue = "fr".
- CAtype = "128", CAvalue = "Latn".
- CAtype = "1", CAvalue = "IDF".
- CAtype = "3", CAvalue = "Paris".
- CAtype = "24", CAvalue = "75011".
- CAtype = "132", CAvalue = "private CA parameter".

Following is the relevant part of the long Regionalisation Discovery Record that was pushed to the HNED which resulted in the Cell ID being "Paris East", "Ile de France", "Paris and Suburb".

```
<cell ID="Paris East">
    <CountryCode>FR</CountryCode>
    <CA type="0" value="fr">
        <CA type="128" value="Latn">
            <CA type="1" value="IDF">
                <CA type="3" value="Paris">
                    <CA type="24" value="75003" />
                    <CA type="24" value="75004" />
                    <CA type="24" value="75005" />
                    <CA type="24" value="75010" />
                    <CA type="24" value="75011" />
                    <CA type="24" value="75012" />
                    <CA type="24" value="75019" />
                    <CA type="24" value="75020" />
                </CA>
            </CA>
        </CA>
    </CA>
</cell>
```

5.2.6.8 Provision of RMS-FUS Information

The RMS-FUS may optionally use SD&S to provide information about the FUS and RMS services which are available to the HNED.

The aim is to describe the locations (entry points) for the FUS and the RMS to allow the HNED to use the services offered on the network. For the FUS the necessary locations are the push (multicast) and pull (unicast) connections to the announcement services, and the Query-Response Channel (QRC) used by the HNED to query the FUS for firmware updates. For the RMS only the management channel, which is a unicast service, is required.

The record includes enough information for the HNED to identify the FUS or RMS to be sure that only the appropriate connections are made.

Sufficient information should be available in the SD&S record to allow the "ownership" status of an HNED to be changed, for example to make an unmanaged HNED into a managed one, to change management agencies or to change from managed to unmanaged. These operations will depend on all the support mechanisms and appropriate firmware updates being available on the RMS and FUS servers, the mechanisms are not specified in this present document.

The record described below in table 11 may include descriptions of the locations of several FUS Providers, each of which may use multiple FUS announcement locations. The firmware update descriptions for which the connection locations are provided may be delivered using either DVBSTP/UDP or SDP/SAP/UDP and an attribute is included to

identify which protocol will be needed. The record may also include an address to connect to the RMS, which is assumed to be a single unicast address.

The RMS-FUS Information record may be multicast (push model) or retrieved on request (pull model). One or both models shall be supported by the server while both models shall be supported by the client.

Table 11: FUS Firmware Announcement Update record

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional/ Conditional
RMSFUSDiscoverytype:	Entry addresses for RMS and FUS Firmware Announcement.	
@Version	Version of this record. A change in this value indicates a change in one of the RMS or FUS Records.	M
FUSProviderType (one entry per FUS provider):	/RMSFUSDiscovery/FUSProvider	
FUSName	Name of the FUS Provider for display in one or more languages; one FUS Provider name is allowed per language code, and at least one language shall be provided (though not necessarily more than one).	M if FUS used
FUSID	ID of FUS Provider, is identical to ID used in RMS-FUS XML documents	M if FUS used
Description	Description of the FUS Provider for potential display in one or more languages; one description is allowed per language code.	O
@LogoURI	Pointer to a FUS Provider logo for potential display. The pointer shall be a URI.	O
FUS@Announcement	/RMSFUSDiscovery/FUSProvider/FUSAnnouncement	
FUSAnnouncement Description	Description of the firmware update for potential display in one or more languages; one description is allowed per language code.	O
MulticastAnnouncementAddress@Source MulticastAnnouncementAddress@Address MulticastAnnouncementAddress@Port MulticastAnnouncementAddress@Protocol	Multicast address of the entry point to access the Update Announcement Messages for the FUS. The "MulticastAnnouncementAddress@Source" and "MulticastAnnouncementAddress@Protocol" attributes are optional. Announcement messages available at the entry address provided may be provided over DVBSTP/UDP or SDP/SDP/UDP, if absent default is DVBSTP.	M if FUS used
FUSUnicastAnnouncement	URI of the Update Announcement Message available from the FUS.	O
QRCLocation	URI of the QRC service, there should be only one per FUSProvider. This is not normally used for managed systems.	O
RMSProviderType (one entry per RMS provider):	/RMSFUSDiscovery/RMSProvider.	
RMSName	Name of the RMS Provider for display in one or more languages; one FUS Provider name is allowed per language code, and at least one language shall be provided (though not necessarily more than one).	M if RMS used
RMSID	ID of RMS Provider, is identical to ID used in RMS-FUS XML documents.	M if RMS used
Description	Description of the RMS Provider capability for potential display in one or more languages; one description is allowed per language code.	O
RMSProvider@LogoURI	Pointer to a RMS Provider logo for potential display. The pointer shall be a URI.	O
RMSProvider@RMSLocation	URI of the management channel, there should be only one per FUSProvider.	M if RMS used

5.3 Service Selection

A streaming based service may be accessed by an individual HNED in the following ways:

- using RTSP;
- using IGMP.

Live Media Broadcast services are delivered over IP multicast; they are streamed continuously and do not need to be initiated by each HNED. End devices can join and leave multicast services simply by issuing the appropriate IGMP messages. The element "Service Location" in the service discovery records gives all the information required to issue the appropriate IGMP message. No control of the stream, for example pause or fast-forward, is allowed.

Optionally for Live Media Broadcast services, SPs may choose to require the HNED to engage in explicit set up and tear down phases (possible reasons include accounting, conditional access, etc.). In such systems, the higher-layer session protocol, RTSP [30], shall be used. The element "Service Location" in the service discovery record signals the use of RTSP and gives all the information necessary to issue the appropriate RTSP method. Parameters required for the IGMP message will be acquired via the SETUP method from RTSP. See clause 6 on RTSP for the specification of the DVB-IPTV RTSP profile.

Media Broadcast with Trick Mode services are similar to Live Media Broadcast but delivered over IP unicast to enable control of the stream.

Content on Demand Services and Media Broadcast with Trick Mode Services are delivered using IP unicast and are intended for a specific user and need to be initiated explicitly by the end device. RTSP shall be used to access such services. Clause 6 on RTSP [30] specifies which methods to use.

Service selection for CDSs is covered in clause 10 of the present document.

5.4 Transport mechanisms

This clause specifies the protocols that are used to transport the SP Discovery Information and the Service Discovery Information. Two mechanisms are defined, one for multicast and one for unicast.

DVB defined a new protocol for the delivery of XML records over multicast. This protocol is called DVB SD&S Transport Protocol (DVBSTP) and is specified in clause 5.4.1. It shall be used to transport the SD&S information over multicast.

The protocol HTTP [39] shall be used to transport the SD&S information over unicast.

The two transport mechanisms shall be interchangeable in all steps and carry the same content encoded in the same way.

5.4.1 Protocol for multicast delivery of SD&S information

When the service discovery information is transmitted using multicast UDP packet, the protocol DVBSTP defined in this clause shall be used. All values defined below shall be transmitted in normal IP network byte order (most significant byte first).

The DVBSTP protocol is also used for the multicast delivery of Broadband Content Guide data [62], for the multicast delivery of firmware announcement messages [79] and for the multicast delivery of CDS XML download session descriptions as defined in clause 10.4.2.

A URI scheme for DVBSTP is introduced in clause G.3.

5.4.1.1 Syntax

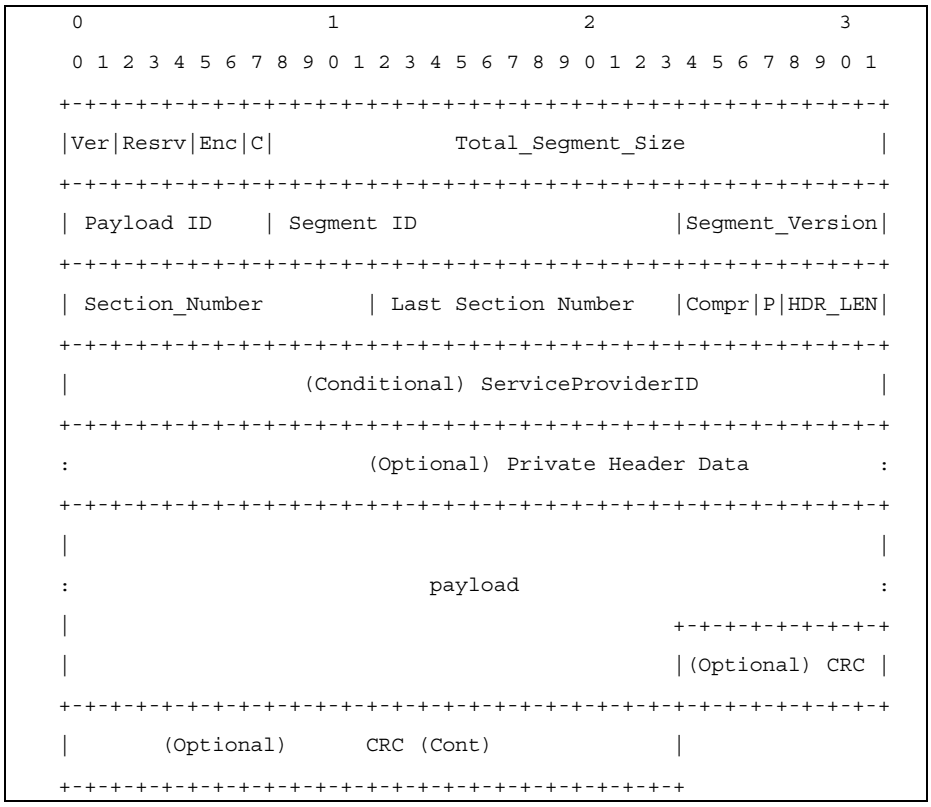


Figure 8: Syntax SD&S multicast delivery protocol

5.4.1.2 Semantics

Protocol Version (Ver): The protocol version. This 2 bit field shall have the value "00".

Reserved (Resrv): These 3 bits are reserved and shall take the value "000".

Encryption (Enc): This 2 bit field shall be used to signal the presence of encryption. It shall take the value "00" to indicate that the payload is not encrypted. The syntax, semantics, behaviour and meaning of other values are not defined.

CRC flag (C): If the value is "1", this indicates the presence of a 32-bit CRC at the end of the packet. 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: A 24 bit field that specifies a size in bytes. For uncompressed data (i.e. Compression is "000"), this is the cumulative size of all the payloads of all the sections comprising the segment (i.e. ignoring headers and CRC, if present).

For compressed data that is usable in the compressed form (e.g. BiM), this is the cumulative size of all the payloads of all the sections (see also clause 5.4.1.3.1) comprising the segment (i.e. ignoring headers and CRC, if present) - this is referred to as the "transmitted size". For compressed data that shall be decompressed before use (e.g. zlib), this is the size of the segment once decompressed by the specified algorithm (note that this may not be the same size as that of the original XML) - this is referred to as the "decompressed size". The definition of the compression field value shall also define which of these two interpretation of total segment size shall apply.

Payload ID: A 8 bit value used to identify the type of data being carried within the payload. The values this may take are set out in table 1.

Segment ID: A 16 bit value used to identify a segment of data for the declared PT (Payload ID) (see note).

NOTE: For example, you may have multiple Broadcast Discovery Information records, and each one will be assigned a unique Id.

Segment version: An 8 bit value used to define the current version of the segment being carried. I.e. version is keyed on Payload ID together with Segment ID. Thus when the data within a segment changes, the segment version fields of all packets that comprise that segment ID and payload ID change. No other payload version fields are necessarily changed. The segment version is modulo 256, and wraps round.

The segment version should only change at the start of a segment. However, to handle packet loss, a receiver should cope with the segment version changing at any point in the segment.

Section number: A 12 bit field identifying the number of this section. The first section in a segment shall be 0.

Last Section number: A 12 bit field which specifies the last section number (the one with the highest section number) in a segment.

Compression (Compr): A 3 bit field used to indicate the compression scheme, if any, used on the payload. All segments of a given payload ID shall share the same compression value. The meanings of these values are given in table 12. GZIP is only available with payload ID 0x08 for use with RMS/FUS or for payload ID 0x07 with the Regionalisation Information Record.

Table 12: Compression values

Compression value	Meaning	Total Segment Size Meaning
000	No Compression	Transmitted Size
001	BiM (as defined in the present document)	Transmitted Size
010	GZIP	Transmitted Size
011 to 110	Reserved	
111	User Private	User Defined

ProviderID Flag (P): Flag signalling if the ServiceProviderID field is present. The value "1" defines the presence of the ServiceProviderID field in the header.

Private Header Length (HDR_LEN): A 4 bit field counting the number of 32 bit words in the header immediately following the header length field, or the Provider ID field if present. This is used to signal the presence of private header data. If no additional header data is sent, then this shall have the value "0000". The Provider ID field is not considered part of the private header, and so is not counted by the Private Header Length field.

ServiceProvider ID: A 32-bit number that is used to identify the SP. This number shall be an IPv4 address, as detailed in clause 5.4.1.3. It is the responsibility of the SP to ensure that this address is appropriately maintained with the appropriate authorities and maintains a unique value within the scope it is used. Note that the ServiceProvider ID is only for use by HNED and not for any network filtering.

A SP ID field is mandatory unless the provider knows that no other SPs can use the same multicast address.

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: The payload of the packet, which is an integral number of bytes. The size of the payload can be calculated from the size of the received packet minus the size of the header (including the optional ProviderID field, if present and any optional private header data present) and the CRC (if present). Note that the payload may be zero bytes in length.

CRC: An optional 32-bit CRC. The standard CRC from ISO/IEC 13818-1 [52], annex A, shall be used. It shall be applied to the payload data of all sections comprising a segment. This field is not necessarily aligned with a 32 bit boundary.

5.4.1.3 Usage

5.4.1.3.1 Use of sections

The size of segments may be substantially larger than that supported by the underlying network. To allow efficient delivery of data, it is necessary to be able to divide the segments into smaller units for delivery. The section mechanism provides this functionality.

Each section shall be sent in exactly one UDP datagram, and each UDP datagram shall carry exactly one section.

To assemble the entire segment, an HNED collects the payload from all the sections and orders them based on their section numbers. Only after an entire segment has been assembled can the CRC, if present, be checked.

Figure 9 illustrates the relationship between sections, segments and records.

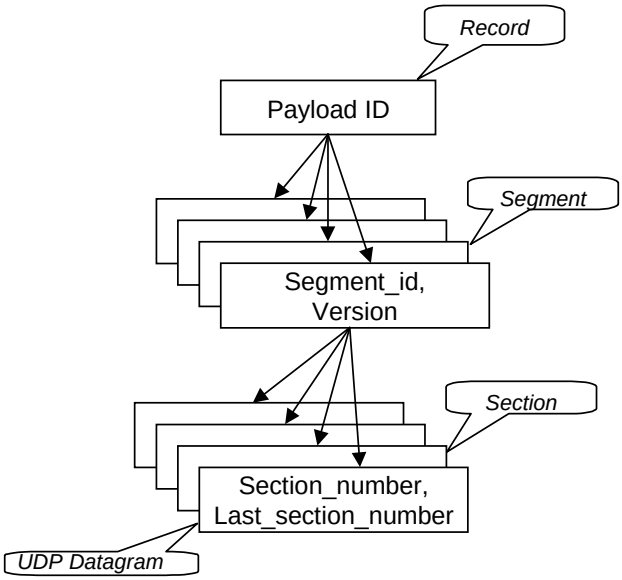


Figure 9: Relationship between records, segments and sections

5.4.1.3.2 Maximum section size

The amount of data that can be encapsulated in each UDP packet, and therefore the potential size of a section, is limited by the maximum size of the IP datagram (65 535 octets for IPv4), minus the UDP and multicast protocol header sizes. To avoid network fragmentation, it is recommended to set the maximum size such that the underlying Maximum Transmission Unit (MTU) of the network is not exceeded.

Exceeding the network's MTU will lead to IP fragmentation, which significantly increases the effective loss rate of a network. This is because if one fragment is lost, the remaining fragments will need to be discarded by the receiver. It also puts additional load on the routers performing fragmentation and on the end-systems re-assembling the fragments.

For an IEEE Ethernet-based network, with an MTU of 1 492 bytes, the maximum section size should be limited to a maximum of 1 452 bytes. Where additional IP, UDP or multicast protocol options are used, then this value should be reduced by the appropriate amount.

If the section size is set such that fragmentation occurs, any end devices that do not support fragmentation will be unable to receive the SD&S payload. It is therefore recommended that SPs set the DF (Do not Fragment) bit in the IP header. With this bit set, routers will return an ICMP "fragmentation needed and DF set" message if the packet length exceeds the destination network's MTU. The SP can adjust the payload size, if such messages are received. IP (RFC 791 [11]) requires that all hosts shall be prepared to accept datagrams of at least 576 octets.

5.4.1.3.3 Use of ProviderID field

Filtering packets on the basis of the source IP address of a packet limits the transmission of packets to sources whose IP addresses is constant and known to the HNED. The ProviderID field overcomes this limitation. It allows an HNED to filter the packets without inspecting or decompressing them. It is expected that the ProviderID field will only be used with SP Discovery records, i.e. when PayloadID is 0x01, since the discovery process will thereafter ensure that only multicast addresses of interest will be received.

If a provider does not have, and is not able to get, a suitable IPv4 address that is unique within the needed scope (that of the network carrying the UDP packets), then the "original_network_id" defined in TS 101 162 [2] may be used. This is mapped into the IPv4 address range using the bottom section of the special 0.0.0.0/8 address range (the "this" network), i.e. 0.0.0.0/16. As an example, an original_network_id of 0x1234 would be represented as 0.0.18.52.

5.4.1.3.4 Repetition rates

The population of receiving devices (HNEDs) will be dynamically changing. It is not assumed that any HNED stores the SD&S data permanently, so the data shall be continually retransmitted. This also provides a degree of reliability, as any corrupted or lost data can be received on the next repetition. To provide flexibility, different segments within a record (payload id) may be repeated more frequently if desired (e.g. to support faster access to some parts of the record). Similarly, different records may be repeated at different rates.

The full cycle to transmit all the segments of the SD&S records for a SP shall not exceed the Maximum Cycle Time defined in clause 5.2.2.3. A segment may be transmitted several times as required during the cycle and different segments may be transmitted at different rates.

This means that an HNED can assume that the complete SD&S information set of a SP has been transmitted after the Maximum Cycle Time.

5.4.2 Protocol for unicast delivery of SD&S Information

In the pull model of delivery of SD&S information, HTTP [39] Protocol shall be used for all communication between the HNED and the SD&S server(s).

When the HNED requests SD&S information, it shall use the following format:

```
'GET /dvb/sdns' request ' HTTP/1.1' CRLF
    'Host: ' host CRLF
```

where request = sp_discovery_request/service_discovery_request.

<request> is used to identify the specific type of request. Two requests have been defined:

- **sp_discovery_request** for a request for discovery information relating to SPs; or
- **service_discovery_request** for a request for discovery information relating to the service offering of a SP.

For the **sp_discovery_request <host>** is the IP address of the SD&S server obtained as specified in clause 5.2.4. For the service_discovery_request <host> is the address specified in the field "Location of the SP Discovery Record" as defined in clause 5.2.5.

The request may contain other header fields conforming to the RFC 2616 [39].

The response to the HTTP requests above shall return the appropriate XML records defined in clause 5.2.6 unencrypted. The HNED should evaluate the message returned from the SD&S server simply to ensure that it contains a 200 series success status. If a 200 series success status is not returned then a retry should occur according to the congestion avoidance mechanism defined in clause 9.1.2.

The maximum size of data returned through unicast delivery shall be bounded by the maximum size of the multicast delivery segment, as defined in clause 5.1.4.2.

After receiving a 200 series success status, the TCP connection is closed.

The HTTP client and server should negotiate a suitable compression using the Accept-Encoding header in the following way: both the client and server SHALL support the Accept-Encoding header (as defined in HTTP/clause 1.1 [39]).

In addition to this, clients and servers that choose to transfer SD&S data in a BiM encoded form SHALL signal BiM encoded content with a proper Content-Encoding header upon transmission, and SHALL NOT change the Content-Type corresponding to their content.

The content coding token corresponding to the BiM encoding shall be x-bim.

In case the transferred data is encoded in the BiM format, the client SHALL have acquired the DVB-TVA-init prior to acquiring the SD&S segments.

5.4.2.1 SP Discovery request

The sp_discovery_request shall return the SP discovery record as defined in clause 5.2.5 for one or all SPs operating on the network. The request has one parameter which can take the value ALL to request discovery information relating to all SPs or the domain name of a specific SP to request discovery information relating to the specified SP. When using the "pull mode", records containing SP discovery information (i.e. Payload ID 0x01) shall not be segmented. This SP discovery record shall exist in two forms, as a single XML record with the list of discovery information for the complete set of SPs operating on the network and as a collection of XML records, one per SP.

The sp_discovery_request shall comply with the following format:

```
sp_discovery_request  = sp_discovery?id='ALL'/SPId
```

where

```
SPId = domainName as defined in clause 3.3
```

This leads to the following two possible requests:

```
'GET /dvb/sdns/sp_discovery?id=ALL HTTP/1.1' CRLF
    'Host: ' host CRLF
```

and

```
'GET /dvb/sdns/sp_discovery?id=MyDomainName HTTP/1.1' CRLF
    'Host: ' host CRLF
```

The host contains the IP address of the SD&S entry point(s) acquired as described in clause 5.2.4. The sp_discovery_request shall not be issued more than once per Maximum Cycle Time.

5.4.2.2 Service Discovery request

The service_discovery_request shall return the service discovery record as defined in clause 5.2.6 describing the service offering of a specific SP. The request has three mandatory parameters which take the domain name of the SP, the type of service offering (i.e. payload ID) and the segment ID. Optionally a segment version may be specified in the request, this will indicate to the server the current version of the segment that the HNED has.

When the segment version is specified, the response to the request shall return the service discovery record for the specified segment only if a new version is available. The version number of the returned segment can be found in the XML record. If the segment has not changed then the server shall return status code "204" as per the RFC 2616 [39] to indicate that the request has been processed successfully but that there is no entity-body to return.

When the segment version is not specified, the response to the request shall return the service discovery record for the specified segment.

When a record is not found, the server shall return status code "404" as per the RFC 2616 [39]; the HNED will then need to issue the appropriate sp_discovery_request to check whether the segment Id is still valid.

The HNED should only issue a service_discovery_request for the valid segment Ids as listed in the SP discovery record.

The service_discovery_request shall comply with the following format:

```
service_discovery_request= 'service_discovery?id='SPId
'&Payload='PayloadId'&Segment='SegmentItem
```

SPId is a domain name as defined above in clause 5.4.2.1.

```
PayloadId       = 2 HEXDIG; any hex number from 00 to ff
SegmentId       = 4 HEXDIG;any hex number from 0000 to ffff
SegmentItem     = SegmentId 0*1('&Version='VersionNumber)
```

SegmentItem is a SegmentId with an optional field for the version number.

```
VersionNumber = 2 HEXDIG; any hex number from 00 to ff
```

For example the following request can be constructed to request the service discovery information relating to the broadcast offering of a SP with DomainName as identifier:

```
'GET /dvb/sdns/service_discovery?id=MyDomainName&Payload=02&Segment=0001 HTTP/1.1' CRLF
    'Host: ' host CRLF
```

The host contains the IP address of the service discovery server of the SP; this address is obtained by resolving the URL contained in the field "PullURL" as documented in clause C.1.2.16 of the present document. The service_discovery_request should be used for the first acquisition of the SD&S information and then only when a change is detected in one of the segments.

5.4.3 Signalling of changes

Changes in the SP offering or the SP discovery information shall be signalled by incrementing the version number of the SP discovery information.

The Service Discovery Information describing the offering of a SP is divided up into segments per type of service discovery information. A change in the offering will translate to a change in the associated segment. Any change in the data carried in a segment shall be signalled by incrementing the segment version of a segment.

The HNED shall monitor the SP discovery record(s) on a regular basis to detect any change in version numbers. Upon detection of a new version of the SP discovery record, the HNED shall check if the SP description needs updating and then shall check if there is any change in the service offering. The HNED will determine which part of the service offering has changed by checking the segment version number of each segment the HNED wants to monitor. The HNED shall then only acquire the segments which have changed.

When using the pull mode, the SP discovery record shall not be checked more than once per Maximum Cycle Time.

In the case where the list of segments is provided in the SP discovery record (mandatory in the "pull" mode, optional in the "push" mode), the addition or removal of segments shall be detected by looking at the list of valid segment Ids for a SP.

When using the "push" mode, in the case where the list of segments is not provided in the SP discovery record and the SP discovery information changes without a change in the offering, it is accepted that the HNED will also check the version number of all the segment Ids it wants to monitor by joining the appropriate multicast address even though there has not been a change in the offering.

In the push mode, in the case where the list of segments is not provided in the SP discovery record, a segment shall be considered as deleted if no packet has been received for this segment for a minimum period of twice the Maximum Cycle Time.

As the DVB-IPTV offering record does not contain any information on the segment it forms (i.e. Segment Id), it is recommended that the HNED should keep a record of the Segment Id together with the relevant DVB-IPTV offering record.

5.5 Encoding

5.5.1 Introduction

SD&S segments may be encoded with BiM [61]. However, the network provider shall also make accessible non-encoded SD&S segments either in the PULL or the PUSH mode, or both, so that HNEDs without a BiM implementation can still obtain non-encoded SD&S segments. In the case where one encoded and one non-encoded multicast stream are delivered, the HNED may discriminate between the streams according to the "compression" flag of the DVBSTP header.

NOTE: If the SP delivers a BCG, then the HNED is expected to support BiM encoding. In this case, it is recommended to use compression of SD&S.

5.5.2 Usage of BiM

5.5.2.1 Introduction

The format is compatible with the BiM format used in TS 102 323 [59] for the transport of TV-Anytime information.

5.5.2.2 DVB-TVA-Init and InitialDescription

In DVB, the DVB-TVA-init (see table 42 in [59]) is used to configure parameters required for the decoding of the binary Access Units and to transmit the initial state of the decoder (DecoderInit message).

The EncodingVersion parameter in the DVB-TVA-Init SHALL be set to "0xF0".

In the DecoderInit field, at least one schema URN shall be transmitted. Consequently, the field NumberOfSchemas of the DecoderInit shall be greater or equal to 1 and the field SchemaURI[0] of the DecoderInit shall be set to urn:dvb:metadata:iptv:sdns:2008-1. DVBContextPath of additional schemas are specified by the ContextPathCode in IEC 23001-1 [61].

As each SD&S segment is a valid stand-alone XML document tree, no initial description is required. Therefore, the InitialDescription() field of the DecoderInit message shall be empty.

5.5.2.3 BiM Access Unit

Each SD&S segment is transported in a DVBBiMAccessUnit as defined in TS 102 323 [59] (clause 9.4.2.3) with the following constraints:

1) As each segment is transported independently, NumberOfFUU should be equal to 1.

2) The table 55 in TS 102 323 [59] is updated with the following values.

Value	Description	EquivalentStartType
0x0030	serviceDiscovery	sdns:ServiceDiscovery type

where: sdns = urn:dvb:metadata:iptv:sdns:2008-1

5.5.2.4 Codec

The BiM decoder used to decode SD&S segments SHALL use by default the Zlib codec, as defined in TV-Anytime (see clause 4.2.4.4 in TS 102 822-3-1 [60]), for decoding string data. This will be signalled in the DecoderInit using the ClassificationScheme "urn:tva:metadata:cs:CodecTypeCS:2004" defined in TS 102 822-3-2 [70].

6 RTSP Client

6.1 Usage of RTSP in DVB

In this clause the use of the *Real Time Streaming Protocol* (RTSP) [30] for a playback capable HNED is specified.

NOTE: A recording capable HNED is not specified in the present document.

RTSP is an application-level protocol for control over the delivery of data with real-time properties. Here the use of RTSP for a classical broadcast like type of delivery of video (TV) and audio (radio) and as well as for on-demand delivery of video and audio is specified.

6.1.1 Service selection

The Service Discovery and Selection process as described in clause 5 shall provide the HNED with the necessary RTSP information for accessing the RTSP based service in question. Depending on the number of streams composing the service, there can be multiple RTSP URLs in the SD&S record:

- If control URLs are present for FEC streams, then the /BroadcastDiscovery/ServiceList/SingleService/ServiceLocation/RTSPURL is the "aggregate" URL for the entire service except for the retransmission stream. When the service is composed of a single stream, this URL is used for all RTSP messages (SETUP, PLAY, TEARDOWN, etc.). In any case, this URL shall be used for the DESCRIBE message when description retrieval is needed.

- The "control" URLs are used when several streams compose the service. They are used to SETUP each stream separately. They can be used to control (PLAY, PAUSE) each stream separately when needed. They are used to TEARDOWN each stream separately. The control URLs are:

    - RTSPURL@RTSPControlURL: this URL is used to control the main audio-video stream;

    - FECBaseLayer@RTSPControlURL: this URL is used to control the FEC Base layer stream;

    - FECEnhancementLayer@RTSPControlURL: this URL is used to control the FEC Enhancement layer stream;

    - UnicastRET@RTSPControlURL: this URL is used for the RTSP control messages (SETUP) for unicast RET stream.

As an example the HNED listens to a multicast address and port number to get the SD&S description, which is presented to the user and from which subsequently the user can make a selection. When the service is selected, the HNED can use the associated RTSP URL(s) to access the service. The URL(s) indicate whether the session control is based on RTSP. When this is the case, the HNED shall use RTSP to access the service in question.

When the service uses retransmission or AL-FEC, it may be necessary to retrieve additional session description information to setup the session. See clause. 6.3.1.

6.1.2 Session transport

DVB compliant HNEDs should use a persistent TCP connection for exchanging RTSP messages with the RTSP server. It is recommended to use a persistent TCP connection; otherwise there is no reliable way for the RTSP server to reach an HNED that is behind a firewall. Persistent TCP connections [39] are used in general to avoid using a separate transport connection for each request/response transaction; this is useful, for example, if the server intends to send asynchronous RTSP ANNOUNCE messages (see table 13) to the HNED.

Multimedia streams, encapsulated as described in clause 7 and controlled by an RTSP server can be transmitted in either unicast or multicast mode. However, in multicast mode trick mode operation like *paus*e, *fast forward* and similar can not be done.

6.1.3 Service information

The HNED uses service information to inform the user about the kind - and availability of services, to locate and to access them. This information needs to be kept up-to-date.

Where possible, the RTSP server can send asynchronously service information to the HNED by using the ANNOUNCE method (see table 13). Alternatively, the HNED can poll the server with the aid of a DESCRIBE method (see table 13) to detect whether the service information is updated. This can be used e.g. in the case a transient connection is used between the HNED and the RTSP server.

When AL-FEC and/or RET is used according to annexes E and F, the session description parameters for LMB services shall be included in the SD&S IPMulticastAddress ServiceLocation type element or/and in the information available via the RTSP URL present in the SD&S RTSPURL ServiceLocation type element; both elements are in the Broadcast Discovery Record.

For LMB services, an RTSP URL may also be available through CRID resolution as described in BCG [62] or alternatively may also be available directly in the ProgramURL element of the tva:ScheduleEvent to confirm XML structure. If present, the CRID resolution is the recommended mechanism.

For CoD and MBwTM services, an RTSP URL shall be used to obtain the session description [62]. This RTSP URL may be available through CRID resolution as described in TS 102 539 [62] or alternatively may also be available directly in the ProgramURL element of the tva:onDemandProgram XML structure. If present, the CRID resolution is the recommended mechanism.

Whenever an RTSP URL is used by the HNED to retrieve the session description, either for LMB or COD/MBwTM services, the HNED shall issue an RTSP DESCRIBE message to obtain the session description.

The ANNOUNCE and DESCRIBE methods are used for conveying the service information to the HNED.

6.1.4 Security considerations

As this DVB specification is based on RTSP and HTTP, the same security considerations apply as with these protocols (see related RFCs).

NOTE: It was decided not to specify security and authentication for DVB-IPTV Phase 1.

6.2 Profiles

6.2.1 Profile definitions

This DVB specification defines the following three RTSP profiles:

- Live Media Broadcast (LMB).
- Media Broadcast with Trick Modes (MBwTM).
- Content on Demand (CoD).

Each RTSP profile contains a subset of the methods and headers defined in the RTSP protocol. The relationship between the RTSP profiles is such that the "Live Media Broadcast" profile is a subset of the "Media Broadcast with Trick Modes", which is in turn a subset of the "Content on Demand" one.

NOTE: The RTSP profile used depends on the application and on whether the service in question is delivered in unicast or multicast mode. Only the LMB is delivered in multicast mode.

6.2.2 Live media broadcast

The Live Media Broadcast RTSP Profile is characterized as the equivalent of the traditional broadcast like TV and radio. The actual media streams are delivered in multicast mode only. This means that the presentation is linear and that there is no support for trick mode operation like pause, fast forward and similar. The presentation is available as part of a continuous flow of events and not on demand.

6.2.3 Media broadcast with trick modes

The Media Broadcast with Trick Modes RTSP Profile is characterized as the equivalent of the Live Media Broadcast one with the addition of support for trick mode operation like pause, fast forward and similar. Therefore the actual media streams are delivered in unicast mode only. The presentation is available as part of a continuous flow of events. The difference with CoD Profile is that the user cannot initiate it.

6.2.4 Content on Demand (CoD)

The CoD RTSP Profile adds to the Media Broadcast with Trick Modes the ability to initiate the start (and stop) of a presentation as an isolated event. This means that this profile supports pause, fast forward and similar as well as the possibility to access media on a time of the user"s choosing. Therefore the actual media streams are delivered in unicast mode only.

6.3 RTSP methods

table 13 specifies the RTSP methods to be supported by the IPI-1 interface for unicast mode of delivery. This applies to MBwTM and CoD profiles.

Table 13: RTSP methods for unicast mode

RTSP Method	Direction: H = HNED; S = Server;	IETF	DVB Requirement
ANNOUNCE	H→S	MAY	MAY
ANNOUNCE	S→H	MAY	**SHOULD**
DESCRIBE	H→S	SHOULD	SHOULD
GET_PARAMETER	H→S	MAY	**SHOULD**
GET_PARAMETER	S→H	MAY	MAY
OPTIONS	H→S	SHALL	SHALL
OPTIONS	S→H	MAY	MAY
PAUSE	H→S	SHOULD	**SHALL**
PLAY	H→S	SHALL	SHALL
REDIRECT	S→H	MAY	**SHALL**
SETUP	H→S	SHALL	SHALL
TEARDOWN	H→S	SHALL	SHALL

6.3.1 DVB specific usage of RTSP methods

6.3.1.1 ANNOUNCE

The ANNOUNCE method can be used to update asynchronously the service information at the HNED. This can be used for example in a LMB to update the service name.

The DVB RTSP client is required to support the reception of descriptions in XML format. For the broadcast profiles (LMB and MBwTM) the ANNOUNCE method shall contain the BroadcastOffering XML complex structure as described in clause 5.2.6.2 (see also clause C.1.4.2). For other, on-demand content, the ANNOUNCE method shall contain the XML complex structure described in table 14 (see also clause C1.5)

The MIME Type in the Content-Type header (see table 16) for such message shall be `text/xml` and the content of the Content-Encoding header and XML description shall be UTF-8. See RFC 3023 [45] on XML Media Types.

Table 14: RTSP ANNOUNCE and DESCRIBE Information

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
CoDAnnounceDescribe		
CoDAnnounceDescribe@Streaming	This attribute shall indicate the streaming format, as per the attribute of the same name in table 4.	O
CoDAnnounceDescribe@RTSPControlURL	This element shall be identical to that described in table 4 with the addition that unicast sessions using RET aggregate URL is also allowed when session multiplexing is used.	O
ContentDescription	This element shall contain the information according to the TV-Anytime [60] type BasicContentDescriptionType, as used in the Broadband Content Guide [64].	M
FECInfo		
FECBaseLayer	This element shall be identical to that described in table 4 with the exception of the port attribute which may be optional (see clause C.1.3.13).	M if FEC used
FECEnhancementLayer	This element shall be identical to that described in table 4 with the exception of the port attribute which may be optional (see clause C.1.3.13).	M if FEC Enhancement Layer used
FECInfo@FECMaxBlockSizePackets	This indicates the maximum number of stream source packets that will occur between the first packet of a source block (which is included) and the last packet for that source block (source or repair).	O if FEC used

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
FECInfo@FECMaxBlockSizeTime	The maximum transmission duration of any FEC Block in milliseconds (source and repair packets).	O if FEC used
FECInfo@FECObjectTransmissionInformation	The FEC Object Transmission Information for the Raptor code. If an FECEnhancementLayer element is included then this element SHALL be included.	M if FEC Enhancement Layer used
RETInfo		
RTPRetransmission	This element shall be identical to that described in table 4, except that the MulticastRET element, and the attributes dvb-enable-bye, dvb-t-wait-min, dvb-t-wait-max, dvb-ssrc-bitmask, dvb-rsi-mc-ret and dvb-ssrc-upstream-client shall not be present. NOTE: The RTCPReporting@ Destination Address element is mandatory for RET-enabled CoD services.	M if RET used

Additionally, DVB RTSP client supporting retransmission according to annex F, should understand session descriptions in SDP format [76]. The MIME Type of SDP descriptions is `application/sdp` and the SDP description itself also uses UTF-8. The HNED may include `application/sdp` in the Accept Header to explicitly indicate support for SDP.

6.3.1.2 DESCRIBE

The DVB RTSP client is required to support the reception of descriptions in XML format as supported for the ANNOUNCE method and described in clause 6.3.1.1.

The MIME Type for XML descriptions shall be `text/xml` and theXML descriptions shall be UTF-8. See RFC 3023 [45] on XML Media Types. The HNED shall always include `text/xml` when the Accept header is used.

Additionally, DVB RTSP client supporting retransmission according to annex F, should understand session descriptions in SDP format [76]. The MIME Type of SDP descriptions is `application/sdp` and the SDP description itself also uses UTF-8. The HNED may include `application/sdp` in the Accept Header to explicitly indicate support for SDP.

6.3.1.3 GET_PARAMETER

The MIME Type in the Content-Type header of a GET_PARAMETER request or response shall be `text/parameters` and the content of the Content-Encoding header shall be UTF-8.

In the request, each parameter name is followed by a colon (":") and is separated by white space, and may be on separate lines or all on the same line. Parameters in the response are expected to be returned one per line in the form:

```
parameter = name ":" *(VCHAR) CR
```

See also clause 3.3 for correct notation.

Table 15 defines the minimal set of GET_PARAMETER parameters that shall be supported by the IPI-1 interface, in the case the GET_PARAMETER method is supported.

Table 15: GET_PARAMETER parameters

GET_PARAMETER parameter	Result	Description

Stream-state	<current stream state>	This parameter retrieves the current stream state.
		Possible returned values are:
		`playing`
		`paused`
		`stopped`

position	NPT	This parameter retrieves the current time position in a CoD multimedia session. The position is the number of seconds from the beginning of the multimedia session in NPT format. This can be used for indication by the HNED to the user how far the presentation of the current session has advanced in time. E.g. the result of a GET_PARAMETER request with the parameter "position" can be: `position: npt=12:05:35.3-` This parameters is undefined for LMB and MBwTM multimedia sessions.

6.3.1.4 SETUP

The HNED should not issue a SETUP request more than once for the same stream or multimedia session before issuing a TEARDOWN request.

6.3.2 Headers

6.3.2.1 RTSP request header fields

Table 16 presents the RTSP header fields that are generated by the HNED and are either mandatory or recommended for the IPI-1 interface.

Table 16: RTSP headers generated by the HNED

RTSP Request Header	IETF	DVB requirement	Remarks on usage for DVB
Accept	MAY	SHOULD	At least the media type: text/xml shall be supported. Other presentation description content types are optional.
Accept-Language	MAY	SHOULD	
Bandwidth	MAY	SHOULD	
Content-Encoding	SHALL	SHALL	
Content-Length	SHALL	SHALL	
Content-Type	SHALL	SHALL	The content types: text/xml and text/parameters shall be supported.
Cseq	SHALL	SHALL	The sequence number shall fit within an unsigned 32-bit number.
Timestamp	MAY	**N.A.** *for LMB* **SHOULD** *for CoD*	
If-Modified-Since	MAY	**SHOULD**	
Proxy-Required	SHALL	SHALL	
Range	MAY	**SHOULD**	
Require	SHALL	SHALL	
Scale	MAY	**N.A.** *for LMB* **SHOULD** *for CoD.*	At least the following scale factors should be supported: -4: fast rewind -2: rewind 0: pause 1: normal play 2: forward 4: fast forward
Session	SHALL	SHALL	

RTSP Request Header	IETF	DVB requirement	Remarks on usage for DVB
Transport	SHALL	SHALL	The HNED may supply multiple transport options from which the RTSP server may choose. The HNED shall support `RTP/AVP/UDP` transport for RTP streaming. It shall support `MP2T/H2221/UDP` and `RAW/RAW/UDP` for direct UDP streaming. The following transport configuration parameters should be provided by the HNED to help configuring intermediaries: `unicast`, `multicast` and `client_port`.
User-Agent	MAY	**SHOULD**	The following format for the User-Agent header is recommended: `User-Agent = "User-Agent" ":" deviceID " HNED V1.0"` See also clause 3.3. E.g.: `User-Agent: PHILIPS-CE/HN3200/A6743ABCD201 HNED V1.0`
NOTE 1: The column IETF presents the request headers required to be supported according to the IETF RTSP specification: RFC 2326 [30]. The DVB requirement columns present the request headers required to be supported for DVB. NOTE 2: The key words in bold indicate where the DVB specification differs from the IETF. NOTE 3: The HNED may generate RTSP request headers that are not listed in table 16.			

Table 17 presents the RTSP header fields that are supported by the HNED (either mandatory or recommended) on the IPI-1 interface.

Table 17: RTSP headers parsed and understood by the HNED

RTSP Response Header	IETF	DVB requirement	Remarks on usage for DVB
Allow	MAY	**SHOULD**	
Connection	SHALL	SHALL	
Content-Encoding	SHALL	SHALL	
Content-Language	SHALL	SHALL	
Content-Length	SHALL	SHALL	
Content-Type	SHALL	SHALL	
Cseq	SHALL	SHALL	It is expected that the server generates sequence numbers that fit within an unsigned 32-bit number.
Expires	MAY	**SHOULD**	
Last-Modified	MAY	**SHOULD**	
Location	SHALL	SHALL	
Public	MAY	**SHOULD**	
Range	MAY	MAY	
RTP-Info	SHALL	SHALL for RTP streaming N.A. for UDP streaming	
Scale	MAY	**N.A.** *for LMB* **SHOULD** *for MBwTM and CoD.*	At least the following scale factors should be supported: -4: fast rewind -2: rewind 0: pause 1: normal play 2: forward 4: fast forward
Retry-After	MAY	**SHOULD**	
Server	MAY	**SHOULD**	The content of this header is left to the implementation of the RTSP server.
Session	SHALL	SHALL	It is expected that the RTSP server uses the timeout parameter with this header.

RTSP Response Header	IETF	DVB requirement	Remarks on usage for DVB
Transport	SHALL	SHALL	`RTP/AVP/UDP` transport shall be supported for RTP streaming. `MP2T/H2221/UDP` and `RAW/RAW/UDP` shall be supported for direct UDP streaming. Furthermore, the HNED should support (and the server is expected to provide) at least the following transport configuration parameters: `unicast`, `multicast`, `destination`, `port`, `client_port`, `source` and `server_port`. These parameters can help intermediaries in forwarding the multimedia stream in question.
Timestamp	MAY	**SHOULD**	
Unsupported	SHALL	SHALL	
NOTE 1: The column IETF presents the response headers required to be supported according to the IETF RTSP specification: RFC 2326 [30]. The DVB requirement columns present the response headers required to be supported for DVB.			
NOTE 2: The key words in bold indicate where the DVB specification differs from the IETF.			
NOTE 3: The HNED may ignore RTSP response headers that are not listed in table 17.			

6.3.2.2 Transport Header parameters required for direct UDP encapsulation

The following additional "transport-protocol/profile/lower-transport" value sets for the RTSP "Transport:" header are defined for UDP streaming:

- MP2T/H2221/UDP; and
- RAW/RAW/UDP.

This indicates that an MPEG2 transport stream is used and transported directly over UDP (without RTP).

6.4 Status codes in response to requests

Table 18 lists the RTSP and HTTP status codes that the RTSP enable HNED shall be able to interpret.

Table 18: RTSP response codes

Status Code	Description
200	*"OK"*
275	*"OK - Request forwarded"*
300	*"Multiple Choices"*
301	*"Moved Permanently"*
302	*"Moved Temporarily"*
304	*"Not Modified"*
400	*"Bad Request"*
401	*"Unauthorized"*
403	*"Forbidden"*
404	*"Not Found"*
405	*"Method Not Allowed"*
406	*"Not Acceptable"*
408	*"Request Time-out"*
410	*"Gone"*
411	*"Length Required"*
412	*"Precondition Failed"*
413	*"Request Entity Too Large"*
414	*"Request-URI Too Large"*

Status Code	Description
415	*"Unsupported Media Type"*
451	*"Parameter Not Understood"*
453	*"Not Enough Bandwidth"*
454	*"Session Not Found"*
455	*"Method Not Valid in This State"*
456	*"Header Field Not Valid for Resource"*
457	*"Invalid Range"*
459	*"Aggregate operation not allowed"*
460	*"Only aggregate operation allowed"*
461	*"Unsupported transport"*
462	*"Destination unreachable"*
463	*"Destination required"*
500	*"Internal Server Error"*
501	*"Not Implemented"*
503	*"Service Unavailable"*
505	*"RTSP Version not supported"*
551	*"Option not supported"*
NOTE 1: Particular response codes will be raised with a particular profile only. NOTE 2: The HNED shall use the most significant digit of the status code to identify its severity, in the case that the given status code is unknown to the HNED.	

6.5 The use of RTSP with multicast

Optionally, it is possible to use RTSP for joining multicasts of Live Media Broadcasts.

NOTE 1: In principle a multicast does not support trick mode operation, therefore it cannot be used with the MBwTM and CoD RTSP profiles.

Using RTSP for joining multicast gives intermediaries the opportunity to inspect the nature of the multimedia session. Specifically, firewalls will be able to ascertain the incoming port being used i.e. this will allow them to open the ports and do any necessary port forwarding. Furthermore, it can be useful if the RTSP server wishes to count the number of receivers "tuned-in".

IGMP shall be used (next to RTSP) to signal to IP network to forward the multicast in question, when the media streams are delivered in multicast mode. During the set up of the multimedia session, an IGMP JOIN message shall be issued by the HNED for joining the given multicast. Furthermore, the HNED shall issue an IGMP LEAVE message, when it leaves the multicast.

NOTE 2: It is mandatory that IGMP version 3 [47] is used for all such messages on the IPI-1 interface.

The transport configuration parameters: `destination` and `source` (see table 17) shall be used by IGMP. The former shall signal the multicast address, the latter can be used by IGMP version 3 to signal the source address of the multicast for *Source-Specific Multicast* (SSM) (see RFC 3376 [47]).

NOTE 3: RFC 2326 [30] specifies that by default a multimedia stream is delivered in multicast mode, when no indication is given by RTSP whether the mode of delivery is unicast or multicast. See also the transport configuration parameters: `unicast` and `multicast` in table 17.

For multicast mode of delivery, table 19 presents the RTSP methods to be supported by the IPI-1 interface.

Table 19: RTSP methods for multicast mode

RTSP Method	Direction: H = HNED; S = Server;	DVB Requirement	Remark
ANNOUNCE	H→S	MAY	
ANNOUNCE	S→H	**SHOULD**	The multicast server can use this method to update asynchronously the service information.
DESCRIBE	H→S	SHOULD	
GET_PARAMETER	H→S	**SHOULD**	
GET_PARAMETER	S→H	MAY	
OPTIONS	H→S	SHALL	The HNED can use this method to request from the RTSP server which methods it supports.
PAUSE	H→S	**N.A.**	
PLAY	H→S	SHALL	This method can be used to signal to the intermediaries that the delivery of the multicast is about to start. The Range and Scale request headers should not be used (see tables 16 and 17).
REDIRECT	S→H	**SHALL**	The multicast server can use this method for load balancing.
SETUP	H→S	SHALL	This method can be used by the intermediaries to allocate resources, open ports, etc. The SETUP method will have no effect on the multicast RTSP server's state. The server can use this method to count the number of HNEDs "listening".
TEARDOWN	H→S	SHALL	This method can be used by the intermediaries to reverse the effect of the SETUP method i.e. close ports, de-allocate resources, etc. The TEARDOWN method will have no effect on the multicast RTSP server's state. The server can use this method to count the number of HNEDs "listening".
NOTE 1: The keywords in bold indicate where the DVB specification differs from the IETF. NOTE 2: The RTSP methods RECORD and SET_PARAMETER are not supported.			

7 Transport of MPEG-2 TS for real-time services

The present document covers the delivery of DVB services over IP networks, as described in clause 4. The initial registration and configuration of the end-device (including IP address assignment), and the means of discovering and choosing a DVB service are covered in other clauses of the present document. This clause concentrates on the format of the service as it appears on the IP network and the requirements on that network for correct and timely delivery of real-time services (Live Media Broadcast and CoD). In accordance with clause 4, clause 7 pertains to the interface IPI-1 of the home network end device.

The present document has been designed to meet the requirements of direct-to-home (DTH) content delivery via IP, as specified in clause 4.

The transport of MPEG-2 TS for non real-time services (CDSs) is covered in clause 10.

7.1 Transport stream encapsulation

The present document can be used to encapsulate any TS 101 154 compliant MPEG-2 Transport Stream (MTS) [56], whether containing single or multiple programs. Those transport streams that contain multiple Program Clock References (PCRs) shall, by definition, be constant bitrate streams. Transport streams containing a single clock reference may be constant or variable bitrate.

NOTE: However, in the case of variable bitrate, the bitrate between PCRs is constant as defined by MPEG-2.

The Content Service Provider (CSP) may receive transport streams (e.g. from a satellite feed) that contain multiple programs. The CSP may choose to decompose these transport streams and generate separate single program transport streams (SPTSs) for each program, or to transmit the Multiple Program Transport Stream (MPTS) in its entirety. This is an operational decision.

All transport streams shall be TS 101 154 [58] compliant, and shall be encapsulated either in RTP (Real-time Transport Protocol) according to RFC 3550 [21] in conjunction with RFC 2250 [29] or directly in UDP (User Datagram Protocol) according to ITU-T Recommendation H.610 [69].

7.1.1 Real-time Transport Protocol (RTP) encapsulation

RFC 3550 [21] indicates that RTP should use an even UDP port number, with the corresponding RTCP stream using the next higher (odd) port number.

Each IP packet [11] is made up of the standard IP header, a UDP header, an RTP header and an integer number of 188-byte MPEG-2 transport stream packets. See figure 10. There is no requirement for every RTP packet in a stream to contain the same number of transport stream packets. The receiver should use the length field in the UDP header to determine the number of transport stream packets contained in each RTP packet.

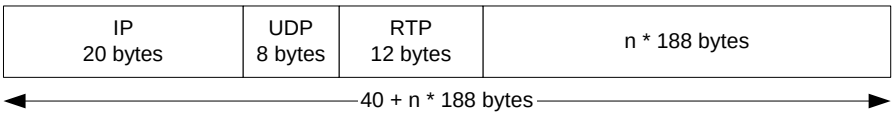


Figure 10: Minimal packet format (IPv4) for RTP encapsulation

The number of transport stream packets that can be encapsulated in each IP packet is limited by the maximum size of the IP datagram (65 535 octets for IPv4). Care should be taken not to exceed the underlying maximum transmission unit of the network. Exceeding the network's MTU will lead to IP fragmentation, which significantly increases the effective loss rate of a network. This is because if one fragment is lost, the remaining fragments will need to be discarded by the receiver. It also puts additional load on the routers performing fragmentation and on the end-systems re-assembling the fragments.

For any Ethernet-based network, with an MTU of 1 492 bytes (IEEE 802.3 [7] frame with LLC) or 1 500 bytes (IEEE 802.3 [7] frame without LLC, see IEEE 802.3 [7] and IEEE 802.2 [6]), the number of MPEG packets per IP packet should be limited to seven (giving a maximum packet length of 1 356 bytes). Where IP or RTP header options are used the number of MTS packets per IP packet may need to be less than seven to stay within the MTU.

If the RTP payload is set such that fragmentation occurs, any end devices that do not support fragmentation will be unable to receive the stream. It is therefore recommended that CSPs set the DF (Do not Fragment) bit in the IP header. With this bit set, routers will return an ICMP "fragmentation needed and DF set" message if the packet length exceeds the destination network's MTU. The CSP can adjust the payload size if such messages are received. IP (RFC 791 [11]) requires that all hosts shall be prepared to accept datagrams of at least 576 octets.

The CSP may choose not to calculate the UDP checksum and set this value to zero (as per RFC 768 [10]).

The RTP header is shown in figure 11.

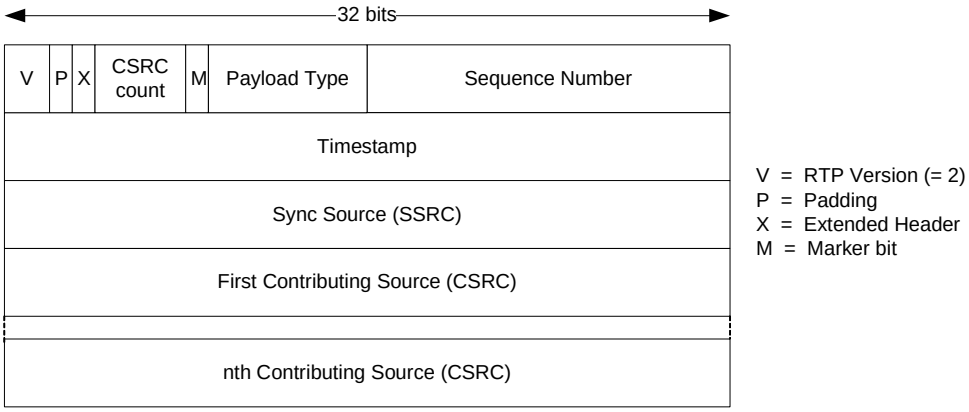


Figure 11: RTP header format

The PT shall be set to MP2T (33), as specified in RFC 1890 [22].

The 16-bit sequence count in the RTP header should be used by the receiver to reorder out-of-order packets, delete duplicates, and detect packet loss.

The 32-bit timestamp in the RTP header is derived from a 90 kHz clock source that may be, but is not required to be, locked to the clock reference of one of the programs in the transport stream. This clock shall conform to the accuracy and slew constraints for MPEG-2 system clocks as defined in ISO/IEC 13818-1 [52].

Other fields are completed as per RFC 3550 [21] and RFC 2250 [29]. Optional CSRC fields should be ignored by the end device.

For most streams, the RTP/UDP/IP overhead of 40 bytes per RTP packet will be low (for example 3 % with a 1 316 byte payload). Although header compression could be beneficial in certain low bit rate applications, the additional complexity at the receiver is not justified. As such, header compression (such as RFC 2508 [36]) shall not be used.

7.1.1.1 Real-time Transport Control Protocol (RTCP)

The RTP specification defines a second protocol - the Real-time Transport Control Protocol (RTCP). It is intended to provide feedback on the network reception quality from each participant and is also used to enable participants to determine the other participants in a session.

RFC 3550 [21] defines two separate RTCP message sets. RTCP Compound Sender Reports are sent by the sender to each receiver and are used to inform receivers about transmission statistics (number of packets and bytes sent). RTCP Compound Receiver Reports are sent periodically from each receiver back to the sender to inform the sender about reception statistics (e.g. delay and jitter).

The IPI-1 interface shall not generate RTCP (Compound) Receiver Reports, unless the HNED is RET-enabled (see annex F). This decision is based on scalability as for large scale deployments Receiver Reports can generate a large volume of traffic at the sender.

The IPI-1 interface shall accept Sender Reports. CSPs are recommended to send Sender Reports to enable HNEDs to synchronize independent transport streams accurately (for picture in picture or other applications). If CSPs choose to send Sender Reports the time between repeat transmissions shall not exceed 10 s.

For two-way applications the RTCP specification allows senders to include Receiver Report fields within Sender Reports. These fields shall not be included in Sender Reports generated by CSPs.

An HNED may have the capability to receive and decode multiple transport streams simultaneously (picture in picture for example). The problem here is how to synchronize the two streams given that they are independently timed from independent clocks that have arbitrary values. For this application, sender reports should be used to convey the relationship between the RTP timestamp values and real time. Each sender report contains two timestamps taken at the same instance, one of the RTP clock source and the other of the wall clock time as determined by the Network Time Protocol (NTP) [18].

The sender reports allow the end device to calculate at what offset the two streams need to run to keep them in synchronization. The end device does not need to support NTP to synchronize multiple streams. The CSPs should use NTP in order to generate their sender reports. To enable correct synchronization at the receiver, CSPs should synchronize their NTP clocks to within 20 ms of each other (either by deriving them from a common clock or by some other means).

7.1.2 Direct User Datagram Protocol (UDP) encapsulation

In case of managed IP networks that can provide guarantees concerning packet loss, jitter and packet routing (e.g. no packet re-ordering), the transport stream may be directly encapsulated in UDP as defined in ITU-T Recommendation H.610 [69].

Each IP packet [11] is made up of the standard IP header, a UDP header, and an integer number of 188-byte MPEG-2 transport stream packets. See figure 12. There is no requirement for every UDP packet in a stream to contain the same number of transport stream packets. The receiver should use the length field in the UDP header to determine the number of transport stream packets contained in each UDP packet.

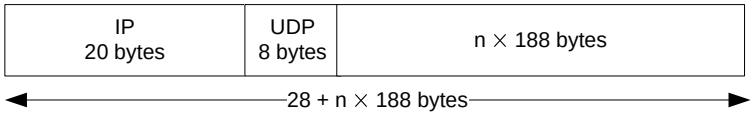


Figure 12: Minimal packet format (IPv4) for UDP encapsulation

The number of transport stream packets that can be encapsulated in each IP packet is limited by the maximum size of the IP datagram (65 535 octets for IPv4). Care should be taken not to exceed the underlying maximum transmission unit of the network. Exceeding the network's MTU will lead to IP fragmentation, which significantly increases the effective loss rate of a network. This is because if one fragment is lost, the remaining fragments will need to be discarded by the receiver. It also puts additional load on the routers performing fragmentation and on the end-systems re-assembling the fragments.

For any Ethernet-based network, with an MTU of 1 492 bytes (IEEE 802.3 [7] frame with LLC) or 1 500 bytes (IEEE 802.3 [7] frame without LLC, see IEEE 802.3 [7] and IEEE 802.2 [6]), the number of MPEG packets per IP packet should be limited to seven (giving a maximum packet length of 1 356 bytes). Where IP header options are used the number of MTS packets per IP packet may need to be less than seven to stay within the MTU.

If the UDP payload is set such that fragmentation occurs, any end devices that do not support fragmentation will be unable to receive the stream. It is therefore recommended that CSPs set the DF (Do not Fragment) bit in the IP header. With this bit set, routers will return an ICMP "fragmentation needed and DF set" message if the packet length exceeds the destination network's MTU. The CSP can adjust the payload size if such messages are received. IP (RFC 791 [11]) requires that all hosts shall be prepared to accept datagrams of at least 576 octets.

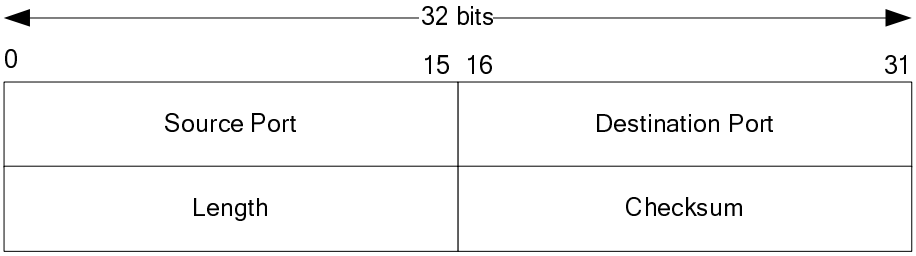


Figure 13: UDP header format

Setting of the source port is optional. If not used the CSP shall set it to zero. The CSP may choose not to calculate the UDP checksum and set this value to zero (as per RFC 768 [10]).

7.1.3 Detection and Usage of RTP and direct UDP encapsulation (Informative)

The use of RTP or direct UDP encapsulation is signaled by SD&S (see clause 5.2.6.2) for multicast and RTSP (see clause 6.3.2) for unicast streaming. In addition it is possible for a device to detect the use of RTP or direct UDP encapsulation. This shall be done by looking for the value 0x47 in the first byte after the UDP header. In case of direct UDP encapsulation this is the first byte of a 188 byte MPEG2-TS packet which always has the value 0x47 (synchronization byte of transport stream header). In case of RTP encapsulation this is the first byte of the RTP header. Its value is always different from 0x47. So in case the byte has the value 0x47 then direct UDP encapsulation is used, whilst if it has any other value then RTP encapsulation is used.

7.1.4 Embedded Service Information (SI)

For transport streams with optional SI (TS - optional SI), all MPEG-2 [52] and DVB [1] tables other than those required by TS 101 154 [58] are optional.

TS - optional SI transport streams are intended for the more advanced situation where the SP wants to present its offering but where it cannot afford or does not want to use bandwidth for usual DVB service description information.

Where transport streams with SI (TS - Full SI) are transported over IP, they shall be compliant with EN 300 468 [1] and TR 101 211 [i.1] and contain all necessary DVB SI with the exception of the network information table NIT. This table may be omitted as it has no meaning in the context of IP services.

7.2 Network requirements

The IP network shall comply with the mandatory network requirements to guarantee successful delivery and decoding by compliant HNEDs.

7.2.1 Mandatory constraints

7.2.1.1 Packet Jitter

MAXIMUM 40 ms peak-to-peak.

Packet jitter is defined as the variation in delay between the source of the stream and the end device. The peak-to-peak jitter, J, implies that the deviation in network delay, d, is bounded by $-J/2 \leq d \leq +J/2$. To be more precise, the HNED shall comply with the MPEG-2 Real Time Interface Specification (ISO/IEC 13818-9 [53]) with t_jitter = 20 ms.

7.2.1.2 Direct User Datagram Protocol (UDP) Packet Reordering

If the HNED is using direct User Datagram Protocol (UDP) then the network shall not allow packet reordering.

7.2.2 Recommended constraints

The recommended constraints are given for information only. They are provided as typical values that users might consider acceptable. Failure to meet these recommendations will not prevent the system operating successfully, but may significantly degrade the user's experience.

7.2.2.1 Packet loss

MAXIMUM one noticeable artefact per hour.

The IP packet error rate that results in this quality level depends on the transport stream bit rate. For a 4 Mb/s transport stream with seven transport stream packets per IP packet, one error per hour is equivalent to an IP packet error rate of less than 1×10^{-6}.

When AL-FEC and/or RET is used according to annexes E and F then the acceptable IP packet loss rate may be higher.

7.2.2.2 Multicast timing

LEAVE time: MAXIMUM 500 ms
JOIN time: MAXIMUM 500 ms

These constraints are intended to bound the time taken to join and leave multicast groups. The use of IGMPv3 joins and leaves is defined in clause 7.3.1. The "LEAVE time" is the maximum time that should elapse between an end device emitting an IGMP multicast LEAVE and it receiving any further packets of the associated flow. The "JOIN time" is the maximum time that should elapse between an end device emitting an IGMP multicast JOIN and the first packet of that flow arriving at the end device.

7.3 Service initiation and control

The present document supports the delivery of DVB services either to a single user (using IP unicast), or to many users simultaneously (using IP multicast). These two delivery mechanisms are intended to support different types of service - multicast will be used to deliver "traditional" broadcast DVB services, whereas unicast can be used for personalized DVB services such as video on demand.

7.3.1 Multicast services

Multicast-capable networks will typically restrict the distribution of multicast streams until such time that an end device signals that it is interested in receiving the stream. This signalling is achieved using the Internet Group Management Protocol (IGMP). The IPI-1 interface shall support IGMP version 3 as defined in RFC 3376 [47].

IGMP version 3 adds support for "source filtering"; that is, the ability for a system to report interest in receiving packets only from specific source addresses (or from all but specific source addresses) sent to a particular multicast address. This facility eases the allocation of IPv4 multicast addresses.

To receive a service, the HNED shall perform a group JOIN according to IGMPv3. The JOIN shall include the list of valid source addresses returned by the Service Discovery mechanism if provided.

To terminate reception of a service, the HNED shall perform a group LEAVE according to IGMPv3.

Services delivered over IP multicast are streamed continuously and do not need to be initiated by each end device. HNEDs can join and leave multicast services simply by issuing the appropriate IGMP messages. However, SPs may choose to require the end device to engage in explicit set up and tear down phases (possible reasons include accounting, conditional access, etc.). In such systems, a higher-layer session protocol, such as RTSP, would be used. When a session protocol is used, the IGMP JOIN and LEAVE messages shall be issued when appropriate (for example when the set up and tear down phases are completed).

7.3.2 Unicast services

Services delivered using IP unicast are intended for a specific user and need to be initiated explicitly by the end device. Once the flow is established, many applications will require stream control from the end device (typically VCR-like controls for a VOD service).

Unicast services will be initiated and controlled using the DVB profile of the Real Time Streaming Protocol (RTSP) as defined in clause 6.

7.4 Quality of Service

In order to provide the required Quality of Service (QoS) MPEG2 TS real-time streams shall be assigned to the "real-time video bearer" traffic types as defined in clause 11.

8 IP Address allocation and network time services

8.1 IP Addressing and routing

8.1.1 IP Address assignment

The HNED requires one IP address per interface, which will be obtained from a DHCP server. The DHCP server can provide other information as detailed in clause 8.1.1.4.

8.1.1.1 Dynamic Addressing only

The IP address, subnet mask, DNS Server address(es), default gateway, gateway and, if necessary, WINS/NetBIOS servers shall only be allocated dynamically via DHCP.

Static addressing using whatever method is not recommended.

8.1.1.2 Dynamic Host Configuration Protocol (DHCP)

DHCP is defined in a number of RFCs of which the main ones are RFC 2131 [24] and RFC 2132 [25]. The protocol consists of a number of messages that have the same fixed format as shown in figure 14.

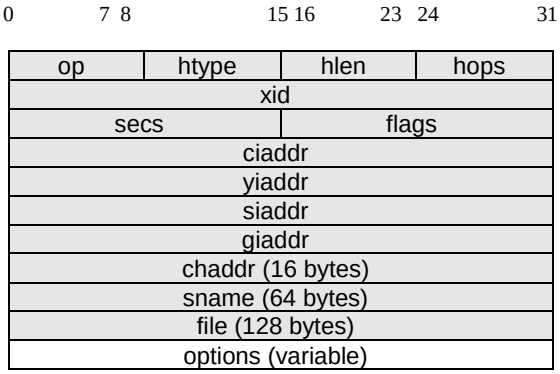


Figure 14: DHCP Format

The messages contain a variable size options part that allows the message to carry additional information other than the IP address. The present document divides the specification of the DHCP client in the HNED into the messages and options.

8.1.1.3 DHCP messages

The DHCP client shall support all the messages of RFC 2131 [24] and RFC 2132 [25].

DHCP requires a client identifier which is the MAC address in Ethernet or Ethernet like products (RFC 2131 [24] and RFC 2132 [25]). This identifier shall be unique.

8.1.1.4 DHCP options

The DHCP option number space (1 to 254) is split into two parts. The site-specific option codes (128 to 254) are defined as "Private Use", and are implementation dependent.

The public option codes (0 to 127, 255) are defined by a range of RFCs in addition to RFC 2132 [25] and are detailed in table 20.

Table 20: DHCP options table

Option description	Reference (RFC 2132 [25] unless otherwise stated)	Option number	Support on IPI-1
Pad Option	3.1	0	Mandatory
Subnet Mask	3.3	1	Mandatory
Time Offset	3.4	2	Optional
Router Option	3.5	3	Mandatory
Time Server Option	3.6	4	Mandatory
Name Server Option	3.7	5	Optional
Domain Name Server Option	3.8	6	Mandatory
Log Server Option	3.9	7	Optional
Cookie Server Option	3.10	8	Optional
LPR Server Option	3.11	9	Optional
Impress Server Option	3.12	10	Optional
Resource Location Server Option	3.13	11	Optional
Host Name Option	3.14	12	Optional
Boot File Size Option	3.15	13	Optional
Merit Dump File	3.16	14	Optional
Domain Name	3.17	15	Mandatory
Swap Server	3.18	16	Optional
Root Path	3.19	17	Optional
Extensions Path	3.20	18	Optional
IP Forwarding Enable/Dizble Option	4.1	19	Optional
Non-Local Source Routing Option	4.2	20	Optional
Policy Filter Option	4.3	21	Optional
Max. Datagram Reassembly Size	4.4	22	Optional
Default IP TTL	4.5	23	Optional
Path MTU Aging Timeout	4.6	24	Optional
Path MTU Plateau Option	4.7	25	Optional
Interface MTU Option	5.1	26	Optional
All Subnets are Local Option	5.2	27	Optional
Broadcast Address Option	5.3	28	Optional
Perform Mask Discovery Option	5.4	29	Optional
Mask Supplier Option	5.5	30	Optional
Perform Router Discovery Option	5.6	31	Optional
Router Solicitation Address Option	5.7	32	Optional
Static Route Option	5.8	33	Optional
Trailer Enacapsulation Option	6.1	34	Optional
ARP Cache Timeout	6.2	35	Optional
Ethernet Encapsulation Option	6.3	36	Optional
TCP Default TTL Option	7.1	37	Optional
TCP Keepalive Interval Option	7.2	38	Optional
TCP Keepalive Garbage Option	7.3	39	Optional
Network Information Service Domain Option	8.1	40	Optional
Network Information Servers Option	8.2	41	Optional
Network Time Protocol Servers Options	8.3	42	Mandatory if NTP used
Vendor Specific Info	8.4	43	May be used with DSL Forum TR-069 [100] as the RMS.
NetBIOS over TCP/IP Name Server Option.	8.5	44	Optional
NetBIOS over TCP/IP Datagram distribution server option	8.6	45	Optional
NetBIOS over TCP/IP Node Type Option	8.7	46	Optional (see clause 8.1.1.4.2)
NetBIOS over TCP/IP Scope Option	8.8	47	Optional (see clause 8.1.1.4.2)
X Window System Font Server Option	8.9	48	Optional
X Window System Display Manager Option	8.10	49	Optional
Requested IP Address	9.1	50	Mandatory
IP Address Lease Time	9.2	51	Mandatory
Option Overload	9.3	52	Mandatory

Option description	Reference (RFC 2132 [25] unless otherwise stated)	Option number	Support on IPI-1
DHCP Message Type	9.6	53	Mandatory
Server Identifier	9.7	54	Mandatory
Parameter Request List	9.8	55	Mandatory
Message	9.9	56	Mandatory
Max DHCP Message Size	9.10	57	Mandatory if DHCP message size exceeds 378 bytes, otherwise Optional
Renewal (T1) Time Value	9.11	58	Mandatory
Rebinding (T2) Time Value	9.12	59	Mandatory
Vendor class identifier	9.13	60	May be used with DSL Forum TR-069 [100] as the RMS.
Client-identifier	9.14	61	Mandatory
Network Information Service+ Domain Option	8.11	64	Optional
Network Information Service+ Servers Option	8.12	65	Optional
TFTP Server Name	9.4	66	Optional
Bootfile Name	9.5	67	Mandatory see clause 9
Mobile IP Home Agent Option	8.13	68	Optional
SMTP Server Option	8.14	69	Optional
POP3 Server Option	8.15	70	Optional
NNTP (News) Server Option	8.16	71	Optional
Default WWW Server Option	8.17	72	Optional
Default Finger Server Option	8.18	73	Optional
Default IRC Server Option	8.19	74	Optional
StreetTalk Server Option	8,20	75	Optional
StreetTalk Directory Assistance Server Option	8.21	76	Optional
User Class	RFC 3004 [43]	77	Mandatory
SLP (Service Location Protocol) Directory Agent	RFC 2610 [38]	78	Optional
SLP Service Scope Option	RFC 2610 [38]	79	Optional
Rapid Commit	RFC 4039 [102]	80	Optional
Client FQDN (Fully Qualified Domain Name)	RFC 4702 [103]	81	Optional
Relay Agent Information	RFC 3046 [46]	82	Optional
iSNS (Internet Storage Name Service)	RFC 4039 [102]	83	Optional
NDS Servers	RFC 2241 [28]	85	Optional
NDS Tree Name	RFC 2241 [28]	86	Optional
NDS Context	RFC 2241 [28]	87	Optional
BCMCS Controller Domain Name list	RFC 4280 [95]	88	Optional
BCMCS Controller IPv4 address option	RFC 4280 [95]	89	Optional
Authentication	RFC 3118 [90]	90	Optional
client-last-transaction-time option	RFC 4388 [96]	91	Optional
associated-ip option	RFC 4388 [96]	92	Optional
Client System Architecture	RFC 4578 [97]	93	Optional
Client Network Device Interface	RFC 4578 [97]	94	Optional
LDAP (Lightweight Directory Access Protocol)	RFC 3679 [101]	95	Optional
UUID/GUID-based Client Identifier	RFC 4578 [97]	97	Optional
User Authentication Protocol List	RFC 2485 [34]	98	Optional
GEOCONF_CIVIC (used for CellID Locatization)	RFC 4676 [98]	99	Mandatory
PCode (IEEE 1003.1 TZ String)	RFC 4833 [99]	100	Optional
TCode (Reference to the TZ Database)	RFC 4833 [99]	101	Optional
NetInfo Parent Server Address	RFC 3679 [101]	112	Optional
NetInfo Parent Server Tag	RFC 3679 [101]	113	Optional
URL	RFC 3679 [101]	114	Optional
Autoconfigure	RFC 2563/2.0 [37]	116	Mandatory that this option is not implemented

Option description	Reference (RFC 2132 [25] unless otherwise stated)	Option number	Support on IPI-1
Name Service Search (Search order)	RFC 2937 [41]	117	Optional
Subnet Selection	RFC 3011 [44]	118	Mandatory
DNS domain search list	RFC 3397 [89]	119	Optional
SIP Servers DHCP Option	RFC 3361 [88]	120	Optional
Classless Static Route Option	RFC 3442 [91]	121	Optional
CableLabs Client Configuration	RFC 3495 [92]	122	Optional
GeoConf Option	RFC 3825 [93]	123	Optional
Vendor-Identifying Vendor Class	RFC 3925 [94]	124	Optional
Vendor-Identifying Vendor-Specific	RFC 3925 [94]	125	Optional
PXE Options	RFC 4578 [97]	128,129.130, 131,132,133, 134 and 135	Optional
End Option	3.2	255	Mandatory

8.1.1.4.1 Max DHCP message size

The maximum DHCP message size option is mandatory when the DHCP message size exceeds 378 bytes, however under 378 bytes it is not required.

8.1.1.4.2 NetBIOS over TCP/IP options

The NetBIOS over TCP/IP options shall be implemented if the HNED requires connectivity to servers that use NetBIOS over TCP/IP. If there is no requirement to connect to a NetBIOS/WINS server then these options shall not be implemented.

8.1.1.4.3 DHCP user class option (RFC 3004)

This shall be implemented in the DHCP client and provision shall be made for multiple user classes. It is not possible for the user to change these class names, however the Remote Management System may add additional class names. Following are the class IDs currently defined.

The class designator should be:

Table 21: Class Designators

Class Name	Description
dvb-ip-stb-video	HNED that is using the IP address for decoding standard DVB video streams
dvb-ip-stb-voice	HNED that is using the IP address for voice over IP
dvb-ip-stb-data	HNED that is using the IP address for non-specific data such as web pages
Vendor defined class names	Subject to registration with DVB

8.1.1.4.4 DHCP relay agent information

There should be no need to implement the DHCP Relay Agent Option (RFC 3046 [46]) in the HNED.

8.1.1.5 DHCP server unavailable

If the remote DHCP server is unavailable for some reason, then products on the home network should still be able to communicate. The method shall use RFC 3927 [49].

8.1.1.6 Multiple DHCP servers

The scenarios currently do not allow multiple DHCP servers on the same home network whether internal or external to the DNG.

8.1.1.7 DNS Server allocation and default gateway

DNS server allocation shall happen via DHCP. A default gateway shall be specified by DHCP.

8.1.1.8 Universal plug and play

Currently there is no need to implement any aspect of Universal Plug and Play in the HNED but it can be added as an option.

8.1.1.9 Server Implementation

If a DHCP server is implemented in an HNED then it shall be possible to enable and disable the server to allow only a single active DHCP server on the network.

8.1.1.10 RTP Retransmission Server Address and future DVB DHCP Extensions

The RTP retransmission server address can be delivered by a DHCP option, however, the available options have been used up, so all future DVB options will be structured as "Vendor-Identifying Vendor Specific Information Options" according to RFC 3925 [94] The Enterprise number assignment for DVB is 2696 [Reference [110]: http://www.iana.org/assignments/enterprise-numbers]. The HNED should send the Vendor Identifying Class Option (124) first with the DVB enterprise number assignment (2696) and a vendor-class-data N of 14 resulting in a data-Len of 1.

The HNED, if using the RTP retransmission option and receiving the server address via DHCP, shall receive the Vendor-Identifying Vendor Specific Information Option (125) containing the DVB enterprise number assignment (2696) with a suboption code of 10 and a suboption containing the a comma-delimited list of the IP addresses or URLs of the RTP retransmission servers. The servers shall be in the order of priority from first to last server to connect to. The method for connecting to the server and assuring its operation is vendor specific.

8.1.1.11 Location Parameter for CellID

The location parameter for CellID is obtained using DHCP option 99 as defined in the RFC 4676 [98] GEOCONF_CIVIC. This option allows the DHCP server to supply country and postal address information of the HNED based on the client address (chaddr) of the HNED sent to the DHCP server. The HNED must use this option and the DHCP server must supply the appropriate civic address elements for CellID to work.

Figure 15 shows the format of the GEOCONF_CIVIC option. The "what" field must have a value of "2" and the country code must represent the country of location of the HNED.

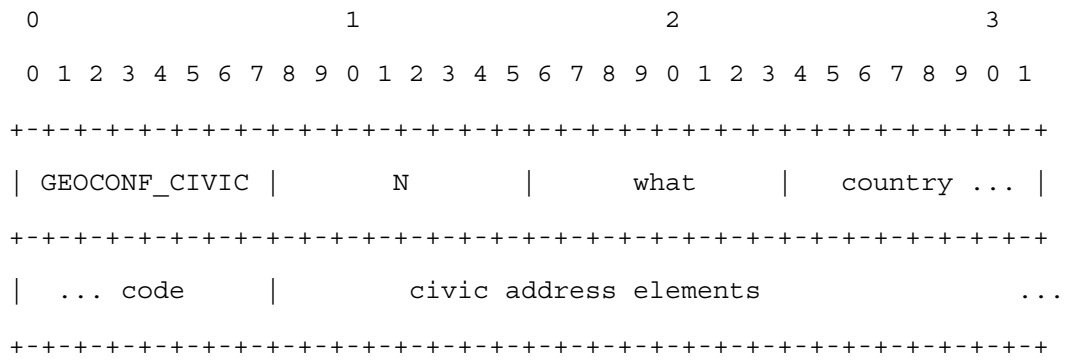


Figure 15: GEOCONF_CIVIC Format

A postal address consists of several parts which vary by country hence the option divides them up into different fields with an index known as the CAType (see table 22). The HNED shall be able to accept any of the CATypes including the private information fields. It is recommended that where a postal/zip code provides sufficient location information that this should be used. All fields must be encoded in UTF-8. The CAtype list should in numerical order.

If a Network SP would like to use some other civic address element to indicate location, for example the name of the DSLAM, then it must use a CAType outside of the range 0 to 128 for proprietary purpose with the appropriate value.

The contents of the civic address elements, once received by the HNED from the DHCP server, are then used to obtain the CellID by either sending them to the SD&S server (see clause 5.2.6.7.1) or by matching against the SD&S provided table (see clause 5.2.6.7.2).

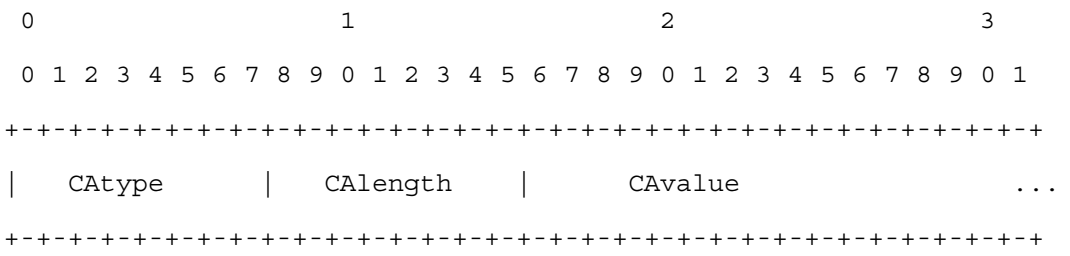


Figure 16: Civic Address Elements

Table 22: Example of Some Civic Address Types (CAtype)

CAtype	NENA	PIDF	Description	Examples
0			Language	i-default
16	PRD	PRD	leading street direction	N
17	POD	POD	trailing street suffix	SW
18	STS	STS	street suffix or type	Ave, Platz
19	HNO	HNO	house number	123
20	HNS	HNS	house number suffix	A, 1/2
21	LM	LMK	landmark or vanity address	Columbia University
22	LOC	LOC	additional location information	South Wing
23	NAM	NAM	name (residence and office occupant)	Joe's Barbershop
24	ZIP	PC	postal/zip code	10027-1234
25			building (structure)	Low Library

8.2 Network time services

The HNED will require network time services for a real-time clock, logging and optionally for the transport stream. These services divide into two:

1) Network time services for applications such as a real-time clock with accuracy of 100 ms.

2) Network time services for the transport stream with accuracy better than 50 ms.

It should be noted that both services can co-exist simultaneously.

8.2.1 Real-Time Clock or other applications with an accuracy of 100 ms

The real time clock in the HNED should be implemented using RFC 2030 [23], Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI. The addresses of the SNTP servers should come from the Time Server DHCP option (4).

8.2.2 Accurate time services for the transport stream

As an option, Network Time Protocol (Version 3) as detailed in RFC 1305 [18] should be implemented when time services with an accuracy of 1 ms to 50 ms are needed. The IP addresses of the time servers should come from the Network Time Server DHCP option (42). The Network Time Protocol should be tried first and only on failure shall Simple Network Time Protocol be used. A null Network Time Server DHCP option (42) means no server is available and Simple Network Time Protocol should be used. It may be desirable to implement a secure real-time clock mechanism for security reasons, if required this should use Network Time Protocol version 3 with the authentication option outlined in Appendix C of RFC 1305 [18].

9 File Upload System Stub (FUSS) to Enable Optional Updates of the System Software of an HNED

This chapter replaces the chapter "Identification Agent for the transport of DVB Services over IP based networks" in the previous versions of the present document. It is intended to work with the updated and separate "Remote Management and Firmware Update System for DVB-IPTV Services" [79] to allow the system software of an HNED to be updated on a power-cycle or reboot. The updating of the system software after power-cycle or reboot will be handled by the mechanisms in the optional "Remote Management and Firmware Update System for DVB-IPTV Services" specification [79].

The FUSS shall be supported in every HNED and its use is mandatory, however, downloading and replacement of the system software pointed to by the stub, whilst strongly recommended, should only be done once vendor specific security measures have been passed.

The procedure to upgrade the firmware of the HNED consists of 4 steps:

1) Obtaining the Stub File either via unicast or multicast. The filename in the unicast case is always "dvb-ipi-fus-stub.dvb".

2) Examination of the Stub File to find possible upgrade candidates.

3) (optional) Downloading the upgrade.

4) (optional) Execution of vendor supplied security measures and replacement of current firmware.

9.1 Obtaining the Stub File

On startup of the device, the device shall find out a URL or IP address for the stub file in the following priority with the following methods:

1) Check the DHCP next server "*siaddr"* field. If "*siaddr*" contains a valid unicast IP address then the device shall obtain the stub file using HTTP(S) with the URL: http(s)://siaddr/dvb-ipi-fus-stub.dvb. If "*siaddr*" contains a valid multicast address then the device shall obtain the file using DVBSTP as described below.

2) If the "*siaddr*" field is set to 0 or is an invalid IP address then the device shall check the bootfile DHCP option (67). The bootfile option shall contain the fully qualified URI for the file which should use HTTP(S) for unicast as in method 1 above, or can contain a single multicast IP address for downloading using DVBSTP as described below. If there are filenames or URIs without the dvb extension then they must be skipped. If there are multiple URIs with the extension dvb then they must all be tried in no particular order.

3) If there is no bootfile name or IP address in the bootfile option then the device shall listen to a globally reachable and public IGMPv3/SSM address of 232.255.255.254 as defined in RFC 3171 [i.9]. The HNED shall listen for DVBSTP for a maximum of 10 s on this address.

4) The device manufacturer has the option of hard coding a URL or IP address into the box for use with HTTP or DVBSTP.

9.1.1 Using DVBSTP to Obtain the Stub File via Multicast

Once the multicast address has been obtained, the HNED shall listen on the multicast address on the port number 3937 (dvbservdsc) as assigned by IANA. The HNED listens for payload ID 0x08 and Segment ID 0x00 to find the payload containing the Stub File. It uses the ServiceProviderID, if present, to select whether the Stub File is meant for this HNED.

Clause 5.4.1 describes the use of DVBSTP for obtaining in SD&S data. The use of the semantics in clause 5.4.1.2 shall be followed with the exceptions below:

Compression (Compr): The FUS Stub file should be fairly small so it should have no need to be compressed, thus this value must be 000.

ProviderID Flag (P): This flag signals if the ServiceProviderID field is present in SD&S but in FUSS indicates whether multiple FUS Stub providers are being used. The value "1" defines the presence of the ServiceProviderID field in the header and that the SP is multicasting multiple FUS Stub Files to the HNEDs. The setting of the ProviderID Flag and use of the SP ID is optional.

ServiceProvider ID: A 32-bit number that is used to identify the FUS Stub provider without examining the payload. The 32-bit number shall be formed from the 24-bit ManufacturerOUI with the remaining 8-bits set to 0 to be reserved for later usage. The HNED must check the ServiceProviderID if the ProviderID flag is set to 1, and must then compare the lower 24-bits of the content of the ServiceProviderID to its ManufacturerOUI. If the ServiceProviderID is the same as its ManufacturerOUI then the DVBSTP payload should be taken, otherwise the whole DVBSTP message should be ignored as it is for a different type of HNED and the HNED should return to examining the multicast traffic.

CRC: The optional 32-bit CRC should be used if there is no Manifest header within the payload. The standard CRC from 13818-1 [52], annex A, shall be used. It shall be applied to the payload data of all sections comprising a segment. This field is not necessarily aligned with a 32 bit boundary.

9.1.2 Using HTTP(S) to Obtain the Stub File via Unicast

The unicast address for the FUS Stub file may be provided in the "siaddr" field of the DHCP message.

- If the siaddr carries a valid unicast IP address and the HNED carries a certificate to support the SSL/TLS operation, the FUS Stub File may be obtained using the URL: https://siaddr/dvb-ipi-fus-stub.dvb based on the "siaddr" supplied. TLS is specified in RFC 2246 [108] and its association with HTTP in RFC 2818 [109].

- If the siaddr carries a valid unicast IP address and no certificate is present or the HTTPS is unsuccessful the operation should be repeated using the URL:http://siaddr/dvb-ipi-fus-stub.dvb.

Alternatively the fully qualified HTTP or HTTPS URI of the FUS Stub file may be carried in DHCP option 67 in the "bootfile name", e.g. HTTPS://10.1.5.51/stub_repository/ dvb-ipi-fus-stub.dvb".

9.1.2.1 HTTP Congestion avoidance mechanism

A congestion avoidance mechanism is required in case of a power cut or other failure that causes a large number of HNEDs to send data at startup so overloading the FUS servers.

Each time the HNED attempts to contact the HTTP(S) server, a Backoff timer shall be initialized to a value of 2 seconds. Immediately before each attempt to establish a connection, a random delay of between Backoff and 2×Backoff seconds shall be imposed. Upon failure to establish this connection, the Backoff timer shall be doubled and the connection will be retried. When doubling of the Backoff timer results in an arithmetic overflow (just before the 16th attempt when Backoff is a 16 bit unsigned integer), retry attempts should be abandoned.

9.2 Stub File Format

The Stub File format is a simple text like format that is simple to parse and compact. The contents are a subset of the metadata defined in annex B of "RMS Remote Management and Firmware Update System for DVB-IPTV Services" [79]. It may either be sent in compact or long form. The compact form uses the "Coding" representation while the long form uses the full names enclosed in "[ ]"for easier human reading. All files have a header "[_DVB-STUB-HEADER-v1.0].

The compact form represents the elements by a coding number shown in table 23 which have an "=" appended and then the value. The elements shall be separated by a ";" character, and if any ";" characters occur in the strings they must be expressed as escape values.

Example of long form:

[_DVB-STUB-HEADER – v1.0]

[DeviceClassInfo]
ManufacturerOUI = 4567
ProductClass = "Fred"
HardwareVersion = "1.01"
SoftwareVersion = "2.003"
SignedPackage = 0

[SoftwarePackageInfo]
Packagename = "Fred"
Packagesize = 12345
FootprintSizeVolatile = 5000000
FootprintSizeNonVolatile = 25000000
SignedPackaged = 0

[ResourceAccessInfo]
URL=http://download.cisco.com/STB-Software/fred1001.bin

Example of same long form information in the compact form:

[_DVB-STUB-HEADER – v1.0]

1a=4567;1b="Fred";1c="1.01";1d="2.003";2a="Fred";2b=12345;2c=5000000;2d=25000000;
2e=0;3a=http://download.cisco.com/STB-Software/fred1001.bin.

The URI can be used two ways:

1) **Unicast only:** This may point directly to a file image for downloading from the FUS directly

2) **Multicast and Unicast:** This can point to a pointer message in the multicast announcement service or to the description announcement message souced from the FUS which identifies the download.

If the final image file is to be made up of several component files, the URL must point to the description announcement message sourced from the FUS, either directly or through a pointer.

Table 23: Stub File Format Elements

Element description		Coding	Type	Mandated/ Optional/ Conditional	Description
DeviceClassInfo	ManufacturerOUI	"1a="	24 bit number	M	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.
	ProductClass	"1b="	String	O	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.
	HardwareVersion	"1c="	String	O	A string identifying the particular CPE hardware model and version.

Element description		Coding	Type	Mandated/ Optional/ Conditional	Description
	SoftwareVersion	"1d="	String	O	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.
SoftwarePackageInfo	PackageName	"2a="	String	O	Opaque string with no specific requirements for its format. The value is to be interpreted based on the HNED's vendor-specific package naming conventions.
	PackageSize	"2b="	Long integer (bytes)	O	The size of the package in bytes.
	FootprintSizeVolatile	"2c="	Long integer (bytes)	O	Required available size of installed image in memory e.g. RAM that is erased at power-off or reboot.
	FootprintSizeNonVolatile	"2d="	Long integer (bytes)	O	Required available size of installed image in memory e.g. Flash that is kept after power-off or reboot.
	SignedPackaged	"2e="	Boolean (0 or 1)	O	Switch indicating that a manifest is used - 0 = false, 1 = true, for the file reached by the URL below.
ResourceAccessInfo	URL	"3a="	IPv4 URI	M	This URI may identify: - The location of a unicast download - The "dvb-mcast" URI (defined in clause G.3) for the multicast pointer or announcement message, - Multicast address for the multicast pointer or announcement message; in this case the "Protocol" field below shall be used.
	Protocol	"3b="	Integer	M for multicast except when "dvb-mcast" URI used	The multicast protocol used for the IP address given by the URL. See table 24. It is not required if the "ResourceAccessInfo" field above provides "dvb-mcast" URI defined in clause G.3 but it shall be present if the "ResourceAccessInfo" field above provides the multicast address only.

Where a multicast service is identified, the use of the "dvb-mcast" URI form of the URL is recommended over the use of the multicast address/protocol fields. The "dvb-mcast" URI is defined in clause G.3.

Table 24: ResourceAccess Info Protocol

Description	Value
SAP	1
DVBSTP	2
Flute	3
DSMCC	4

10 Content Download Service (CDS)

10.1 Overview

CDSs allow for the download of content items to a local storage of the HNED via a broadband IP connection. A CDS can be used to provide IPTV services in areas where a broadband connection suitable for streaming services is not available or prone to errors, for simultaneous delivery of multiple content items to HNEDs or for reduced cost offers as the bandwidth consumption may be limited compared to streaming services.

DVB-IPTV CDSs shall support two different service modes:

The **push download service** mode that is defined as a distribution of content items where the distribution decision is taken by the SP, without explicit request from the user.

The **pull download service** mode provides for download of content items at the explicit request of a user.

In support of these two service modes, the CDS delivery system supports two "download modes": multicast download and unicast download. The protocol used for the multicast download mode is the File Delivery over Unicast Transport (FLUTE) protocol [71] and may be combined with a file repair mechanisms. The unicast download mode is based on the HTTP 1.1 protocol [39]. Download of a file from a single server and download of the file in chunks from multiple servers are supported. A reception reporting procedure allows the HNED to report the successful download of content.

NOTE: While the push download service mode might most often be realized using the multicast download mode and the pull service mode might most often be realized by the unicast download mode, other combinations are possible according to SP requirements. For example, a push download service to a small population of HNEDs can make use of unicast download and a pull download service for popular content items that is expected to be downloaded by a large number of users can make use of carousel multicast download.

The CDS functions enable to download content items. Content items consist of one or more files (e.g. A/V file and related metadata). The available content items, the related files for download and the download mechanisms are announced to the HNED using the BCG and dedicated *download session descriptions*. The HNED either automatically initiates the download (push download service mode) or acts on a user request (pull download service mode).

While the content download mechanisms as such are agnostic to the file formats that are transferred, the present document exclusively specifies the download of content encapsulated into an MPEG2 transport stream and related BCG metadata. The usage of the specification for other content formats is not in the scope of the present document.

CDSs are transparent to any content protection systems and therefore do not prevent the implementation of content protection systems that build on the DVB CPCM specifications [104]. If authentication is required to set up unicast connections between the CDS HNED and the CDS Network function for either session descriptions or file download, it is recommended to use the methods described in RMS/FUS specification [79], clause 5.4.

10.2 Functional Architecture

To support CDSs, the CDS functional architecture according to figure 17 may serve as a reference architecture. The architecture includes logical interfaces between HNED and other CDS network functional components. The present specification aims at specifying these functional components and interfaces. All CDS interfaces are part of the IPI-1 interface.

NOTE: All functions identified in the figure are logical rather than physical. No physical device is implied. The arrow direction indicates the main message flow.

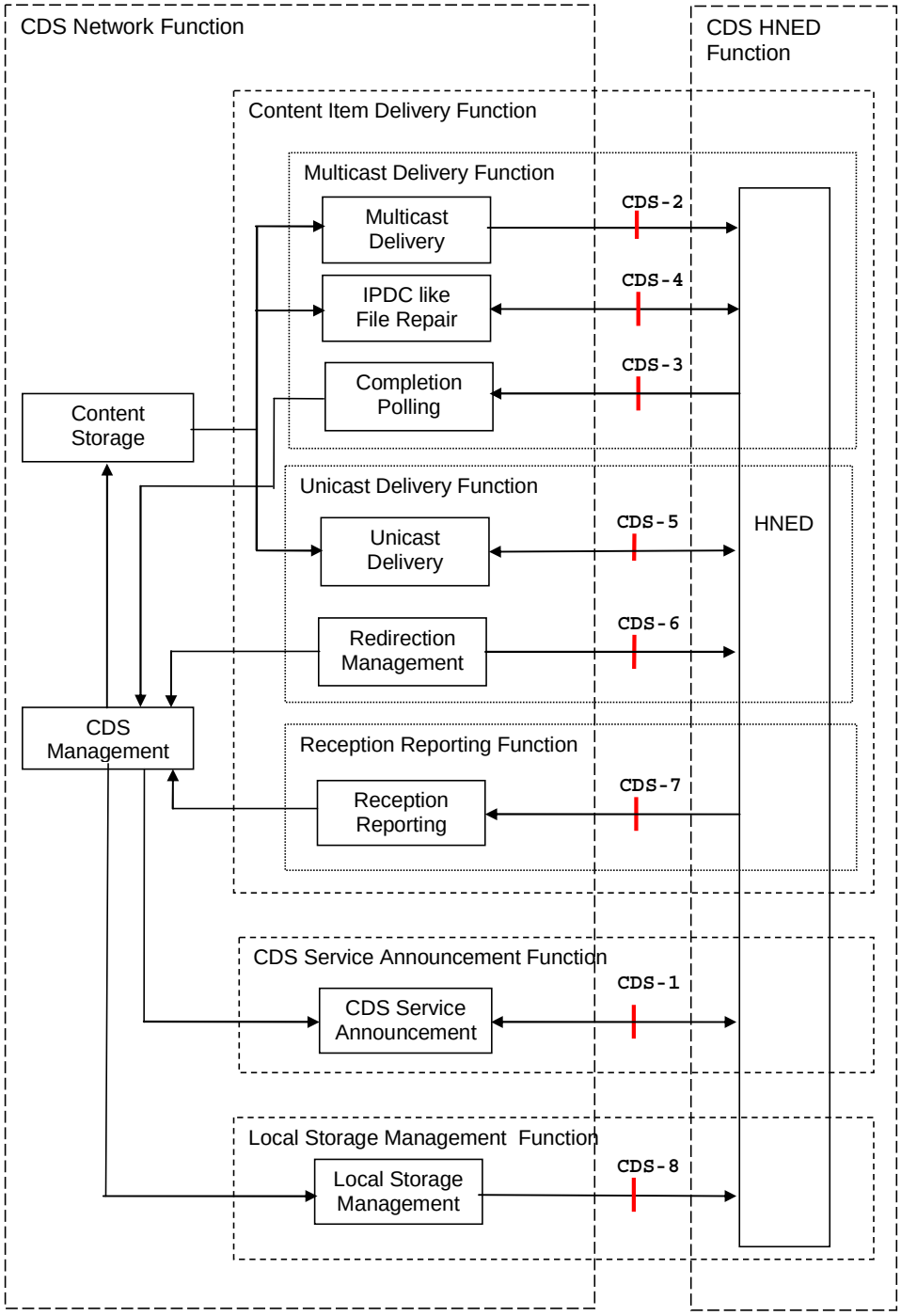


Figure 17: Content Download System Functional Architecture

Interfaces without any numbering are not in the scope of the present document.

10.2.1 CDS Functional Components

An overview of the functional components is provided:

CDS HNED: The user device aims at providing an easy, fast and secure access to the IPTV services. HNEDs that support CDS services shall implement the CDS HNED functions of the present document and shall provide storage dedicated to CDSs. A prescriptive description of the overall behavior of the HNEDs is out of scope of the present document.

CDS Announcement: The CDS Announcement function advertises the availability of content items for pull service or push service download mode as well as the corresponding download session parameters. The details of the service announcement within CDS are introduced in clause 10.3.

Multicast download functions: The multicast download mode reliably distributes content items to a group of receivers simultaneously. The details of the multicast download functions are introduced in clause 10.6.2. The multicast download functions include:

- **Multicast download:** The multicast download component downloads content items to HNEDs. The multicast download component is based on the FLUTE protocol. The details of the FLUTE protocol are introduced in clause 10.6.2.2.
- **Completion polling:** This component is used by the CDS network function to determine when all HNEDs of the multicast group have completed the reception of the contents to be able to stop a multicast download. The details of the completion polling are introduced in clause 10.6.2.5.
- **File repair:** The file repair enables the repair of incomplete files after the multicast download session has been completed. Two types of file repair are defined. For CDS specific file repair the HNED uses the unicast download component of the unicast download function (see below) to download the missing parts of the file over the CDS-5 interface. For IPDC like file repair dedicated repair data is requested by the HNED and provided by the CDS network function. The details of the file repair are introduced in clause 10.6.2.6.

Unicast download function: The unicast download mode aims at reliably distributing content items to individual receivers. The details of the unicast download functions are introduced in clause 10.6.3.

Unicast download: The unicast download component aims at distributing the content items to individual HNED's upon their request. This download mode is based on the HTTP protocol [39]. The details of the unicast download component are introduced in clause 10.6.3.

Redirection management: This component aims at redirecting the unicast download requests to alternative download sources such as a single alternative server, a multicast session on which the requested content is available or a list of multiple servers each of which providing a different portion of the requested content. Moreover this component indicates to the HNED when to carry out the redirection requests e.g. immediately or at some later time. Details on redirection management are provided in clause 10.6.3.4.

Reception Reporting: After a successful download of file chunks, files or content items the HNED may inform the reception reporting function on the successful download. This function offers the possibility for the CDS network to collect statistics about the content download activity and can be used for monitoring or for adapting the download strategy dynamically. The reception reporting function is introduced in clause 10.6.5.

Local storage management: This function allows the CDS network to manage CDS storage and content on the HNED. The details of the storage management are introduced in clause 10.7.

CDS management: This component controls all other CDS functions. This function is not within the scope of the present document.

CDS network content storage: The CDS network content storage function prepares and stores the content items and associated metadata before they get delivered to the HNEDs. This function is not within the scope of the present document. Only the content item and file formats are addressed and are described in clause 10.4.

10.2.2 CDS Interfaces

CDS defines eight interfaces between the CDS network functions and the CDS HNED function. All interfaces are part of the IPI-1 interface. Table 25 provides the interfaces between the CDS HNED functions and the CDS Network functions. The reference to the clause of the present document specifying each one of the interfaces is given in table 25.

Table 25: CDS interfaces on IPI-1

Interface	Description	Protocols	Clause
CDS-1	Carries CDS service announcement information to the HNEDs	BCG XML/DVBSTP XML/HTTP XML/SOAP SDP/SAP SDP/HTTP	10.310.5
CDS-2	Multicast download of content items from the network to the HNEDs	FLUTE	10.6.2.2
CDS-3	Multicast completion polling interface notifies the multicast content download status to the CDS Network	LCT ext. UDP	10.6.2.5
CDS-4	IPDC like file repair	HTTP	10.6.2.6
CDS-5	Unicast download of the content items from the network to the HNEDs	HTTP	10.6.3
CDS-6	Carries the redirection information of a unicast download request to the HNEDs	XML/HTTP SDP/HTTP	10.6.3.4
CDS-7	Notifies the successful completion of the content download to the CDS Network	XML/HTTP	10.6.5
CDS-8	Carries information to manage the HNED local storage	BCG	10.7

10.2.3 CDS Protocol Stack

Figure 18 shows the protocols used over the IPI-1 interface for the support of CDSs. The top layer of the stack signifies the application (content item and service announcement). At the bottom the IP layer serves as the common network transport layer and physical and data link layers.

NOTE: The protocol stack is provided for information only as it can not express all functions in CDS in sufficient detail.

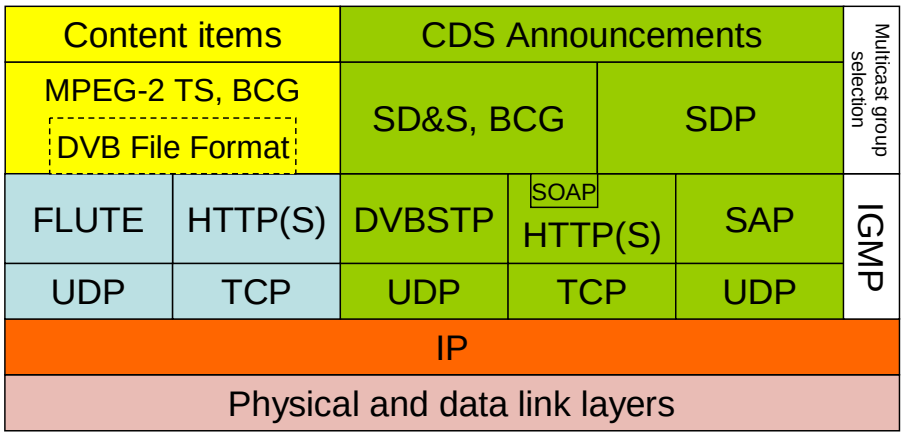


Figure 18: CDS protocol stack

10.3 CDS Announcement through BCG

The CDS Announcement functions provide the CDS HNED functions with information about content items that are offered by the CDS network function for push and pull download service modes to the CDS HNED. This includes metadata for the content items, the availability for download and download session description information. The CDS Announcement information is exclusively delivered over the CDS-1 interface.

10.3.1 Usage of SD&S, BCG and TVA for CDS

The TV Anytime (TVA) based Broadband Content Guide (BCG) as defined in TS 102 539 [62] shall be used for CDS announcement and the announcement of individual content items. Specifically, the metadata fragments as defined in TS 102 539 [62], clause 6 with extensions shall be used for the announcement of content items for download. An extended version of the OnDemandProgramType is defined in clause G.1.1 and is introduced for the announcement of the content items available in the pull download service mode. A new PushDownloadType is defined in clause G.1.2 and is introduced for the announcement of the content items available in the push download service mode. The PushDownloadType is introduced as part of the ProgramLocationType. For this purpose the ProgramLocationType is extended as defined in clause G.1.3. CRID resolution SHALL be performed as defined in TS 102 539 [62], clause 5. The locator for CDS can be a URI locator or an extended on-demand decomposed binary locator as defined in clause G.1.4. The URIs specifically used for CDS in the locators and ProgramURL are defined in clause 10.3.2.

NOTE: CDS Announcement requires extension of the BCG as well as extension of TVA. The BCG OnDemandProgram Type and the on-demand decomposed binary locator are extended in order to differentiate between streaming and download modes and with content download specific information (see clause G.1.1). A new BCG type *PushDownloadType* is introduced. Relevant specifications are expected to be updated in their next releases. To provide a consistent CDS specification in the present document, these extensions are collected in clause G.1.

Content items that are available for pull download are announced via the BCG in the same way as it is done for streaming CoD content items. Information about the content item itself is provided by the Content Description Metadata (see TS 102 822-3-1 [60], clause 6.3) and information about the actual download session is provided by the extended OnDemandProgram Type Instance Description Metadata and/or by the URI locator or extended on-demand decomposed binary locator provided by the CRID resolution.

Content items that are available within a pull download service can be selected by the user for download. The CDS HNED shall initiate the download accordingly.

Content items that are available for push download service are announced via the BCG PushDownloadType Instance Description Metadata (see clause G1.2). CDS HNEDs that have subscribed to the push download service shall autonomously download these content items. PushDownloadType content item instances shall not be announced to the user. After the successful download the content item can be announced via OnDemandProgramType Instance Description Metadata and/or by the URI locator or extended on-demand decomposed binary locator provided by the content resolution as available for consumption to the user with the URI pointing to the content item on the CDS HNED storage (see clause 10.3.2). The OnDemandProgramType metadata can be provided via the normal BCG mechanisms or as part of the content download.

The description of a content download session requires several parameters. These parameters are provided by a dedicated *download session description* mechanism outside of the BCG. The download session description contains all relevant information to reliable download a content item. Download session descriptions may be described in XML or SDP syntax. CDS HNEDs shall support download session descriptions in XML format and may support download session descriptions in SDP format. The BCG instance description metadata and the locators provide the link to the download session description. This link is referred to as *CDS URI* and is specified in clause 10.3.2. The transport of download session descriptions may be unicast or multicast. The transport methods for CDS download session description are specified in clause

10.5.5.

The SD&S BCG discovery record (see clause 5.2.6.6) may provide information about download session description being delivered via multicast. This allows the HNED to listen to the announced multicast delivery and to cache the download session descriptions. In case a specific download session description is requested from a multicast delivery the HNED can access it from the cache and does not have to wait until this download session description is sent out on the multicast delivery.

10.3.2 URIs for Download Session Description

The link to the CDS download session description may be provided by:

- the ProgramURL of the PushDownloadType or the OnDemandProgramType; as well as
- the URI of the URI locator or the Extended-On-demand decomposed binary locator.

The syntax of the URIs used for the different CDS download session description protocols and transport methods are specified in this clause. Four different URIs schemas are specified taking into account different download session description methods (XML and SDP) and transport mechanisms (unicast and multicast). CDS HNEDs shall support CDS URIs that locate XML-based download session descriptions, i.e. the locators specified in clauses 10.3.2.1 and 10.3.2.2 and may support CDS URIs that locate SDP-based download session descriptions, i.e. the locators specified in clauses 10.3.2.3 and 10.3.2.4.

10.3.2.1 CDS XML Multicast Locator

CDS content may be located by a reference to an XML-based download session description delivered over multicast. The actual multicast delivery of XML-based session descriptions is defined in clause 10.5.5.1 based on DVBSTP. In this case, the XML segments with download session descriptions are constantly sent on a multicast group. The multicast group information (multicast address, port and optional source address), the SegmentID and the optional ServiceProviderID have to be provided in order to access the specific XML segment containing the referenced download session description.

As the segment may contain several download session description records the Download-Session-ID has to be provided in addition in the download session description URI.

The CDS HNED function shall extract the specific download session description referenced by the Download-Session-ID from the delivered segment. The Download-Session-ID is part of each session description record as defined in clause 10.5.3.

The DVB-MCAST URI for DVBSTP as defined in clause G.3.2 is used for referencing the multicast delivery of an XML session description. The payload is always provided and shall be set to ‘dvbstp'. The dvbstpPayloadID is always provided and shall be set to the value "b1".

For CDSs and content items located by a CDS XML Multicast Locator, the following format shall be used:

```
'dvb-mcast://' [ src-host '@' ] mcast-addr [':' port] '?payload=dvbstp' ['&service-provider='
ServiceProviderID] '&dvbstp-payload=' b1 ['&segment=' SegmentID] ['#? dvb-cds-session-id=' Download-
Session-ID]
```

For instance, the following sample shows an URI referencing an XML-based download session description to be delivered over dvbstp:

```
dvb-mcast://132.45.1.1@230.100.1.1:1000?payload=dvbstp&dvbstp-payload=b1&segment=23#?dvb-cds-
session-id=20
```

10.3.2.2 CDS XML Unicast Locator

CDS content may be located by a reference to a XML-based download session description delivered over unicast. The actual unicast delivery of XML-based session descriptions is defined in clause 10.5.5.2 using http. The download session description is provided in an XML segment from the host application. The session description URI needs to provide the host, optional port, application (/dvb/cds/session_description), and the Segment ID have to be provided in order to access the specific XML segment.

A HTTP URI is used to reference the XML-based download session description. In case the segment contains several session description records the SessionID has to be provided in addition. The CDS HNED function SHALL extract the specific download session description defined by the Download-Session-ID from the delivered segment. The Download-Session-ID is part of each session description record as defined in clause 10.5.3.

NOTE 1: A DVBSTP service provider ID is not provided. It is assumed that the host application supports a single SP.

NOTE 2: A DVBSTP payload ID is not provided. The application /dvb/cds/sesion_description already indicates that a session description type of payload is requested.

NOTE 3: The segment version is not provided in the URI as always the latest version of the segment shall be used. It might be included automatically in the request by the HNED in case the segment is already in the local cache.

For CDSs and content items located by a CDS XML Unicast Locator, the following format shall be used:

```
'http://' host [':' port ] '/dvb/cds/session_description?Segment=' SegmentID ['#?dvb-cds-session-id=' Download-Session-ID]

SegmentId           = 4*4 HEXDIG; any hex number from 0x0000 to 0xffff
Download-Session-ID = String
```

For instance, the following sample shows an URI referencing an XML-based download session description to be delivered over http:

http://announcements.provider1.org:80/dvb/cds/session_description?Segment=a0ff#?dvb-cds-session-id=102

10.3.2.3 CDS SDP Multicast Locator

CDS content may be located by a reference to a SDP-based download session description delivered over multicast. The actual multicast delivery of SDP-based download session descriptions is defined in clause 10.5.5.3. The session descriptions are constantly send on a multicast group. The multicast group information (multicast address, port and optional source address) and the Download-Session-ID have to be provided in the download session description URI in order to access the specific SDP session description. The Download-Session-ID is part of each session description record.

For CDSs and content items located by a CDS SDP Multicast Locator, the DVB-MCAST URI for SAP as defined in clause G.3.3 shall be used. The payload is always provided and shall be set to 'sap'.

```
'dvb-mcast://' [ src-host '@' ] mcast-addr [':' port] '?payload=sap' '#?sdp-session-id=' Download-Session-ID
```

For instance, the following sample shows an URI referencing an SDP-based download session description to be delivered over multicast:

```
dvb-mcast://132.45.1.1@230.100.1.1:1000?payload=sap#?sdp-session-id=12
```

10.3.2.4 CDS SDP Unicast Locator

CDS content may be located by a reference to a SDP-based download session description delivered over unicast. The actual unicast delivery of the SDP-based download session descriptions is defined in clause 10.5.5.4. The session description is provided as file. The host and the path to the file and the filename have to be defined in order to access the file. The default file extension is ".sdp", however other extensions can be used.

A HTTP URI is used to reference the SDP file. An SDP file contains a single session description. The Download-Session-ID therefore does not need to be defined in the reference. However, if it is defined the HNED shall only accept the download session description in the delivered file if the Download-Session-ID in the reference and of the session description record match. In case they do not match, the download session description shall be ignored.

For CDSs and content items located by a CDS SDP Unicast Locator, the following format shall be used:

```
'http://' host [':' port ] '/' path '/' filename  ['#?sdp-session-id=' Download-Session-ID]

Download-Session-ID = String
```

For instance, the following sample shows an URI referencing an SDP-based download session description to be delivered over unicast:

http://announcements.provider1.org:80/session_announcements/session14.sdp

10.3.3 URI for files on the CDS HNED storage

After the successful download of a content item, the HNED can access the files of the content item from the CDS HNED storage. In order to reference a file located on the CDS HNED storage from the BCG metadata (e.g. ProgramURL of OnDemandProgramType) and locators (e.g. URI locator, URI of Extended On-demand decomposed binary locator) the DVB CDS Local URI scheme shall be used.

```
'dvb-cds-local://'File-Reference

File-Reference = absolute path

<absolute-path> as defined in clause 10.5.2
```

The *File-Reference* which identifies a specific file on the CDS HNED storage shall be the same as provided in the download session description or FLUTE FDT for that specific file. The CDS network, i.e. the SP, shall ensure that the *File-Reference* is unambiguous.

The content type of the file is available from the download procedure, either from the download session description (*File-Content-Type*), the FLUTE FDT or the HTTP session.

In case of a push download the metadata which announces the downloaded content item to the user after the download shall use this DVB CDS Local URI.

In case of a pull download the metadata will indicate that the content item is available for pull download. After the download however the CDS HNED knows that the content item is available for local play out and it shall replace that pull download information with a link to the file on the CDS HNED storage for immediate play out.

10.4 CDS Content Item and File Formats

10.4.1 General

The content item download procedures defined in clause 10.6 are in general transparent to the formats of the files that are delivered. CDS however is focused on the download of content items that consist of one or more audio/video files and optional related metadata. Specifically, the present document primarily specifies the download of content items based on the MPEG-2 TS file format [56].

CDS differentiates between content item formats and file formats. Content item formats provide a high-level description of the content item, i.e. the collection of files in one session description, and also provide hints to the CDS HNED function, how to handle the consumption and play-out of the content item. The content item format is part of the download session description as *Content-Item-Format* parameter. Supported content item formats are defined in clause 10.4.3.

Content items generally consist of one or several files whereby each file has a specific file format or content type. The content type may signaled in the download session description in the *File-Content-Type* attribute, in entity header field *Content-Type* of the http request or reply or in the *Content-Type* field of the FDT. Content types should be registered MIME media types. Supported file formats and the associated MIME media types are defined in clause 10.4.2.

10.4.2 File Formats and Media types

10.4.2.1 MPEG-2 Transport Stream file format

The CDS HNED shall support the reception and consumption of content files in MPEG-2 transport stream format. An MPEG-2 transport stream file consists of the concatenated 188 bytes transport stream packets stored in the order they are delivered. The transport stream SHALL be compliant to TS 101 154 [58].

MIME media type:

- video/mp2t (for video and combined video and audio content), see RFC 3555 [83].
- audio/mp2t (for audio only content).

NOTE 1: Despite RFC 3555 [83] defines the MIME media type video/mp2t only for transfer over RTP, it is re-used here for the purpose of CDS.

NOTE 2: Audio/mp2t is not a registered MIME media type.

10.4.2.2 BCG Metadata file format

The CDS HNED shall support BCG metadata files as defined in TS 102 539 [62]. The XML file shall contain at least an element of the type tva:ProgramInformationType describing the associated content. The XML file may be delivered as uncompressed or BiM compressed textual schema-valid XML file. In case BiM compression is used the *Content-Encoding* parameter in the FLUTE FDT and HTTP header shall be set to "x-bim".

After the successful download of a content item the CDS HNED shall interpret the downloaded BCG metadata files as part of it BCG processing as defined in TS 102 539 [62].

MIME media type:

- application/xml.

10.4.2.3 DVB File Format

The CDS may support files in the DVB File Format as specified in TS 102 833 [111].

MIME media type:

- video/vnd.dvb.file (for video and combined video and audio content).
- audio/vnd.dvb.file (for audio content).

NOTE 1: Files complying to the DVB File Format specification shall contain descriptive metadata as specified in TS 102 833 [111].

NOTE 2: The DVB File Format can support content formats other than the MPEG-2 Transport Stream. At present it is expected that the DVB file format would only contain the MPEG-2 Transport Stream representations of content, but in the future other representations may be supported.

10.4.3 Content Item Formats

A single content item may include a single file or may consist of several files. All files of a content item SHALL be announced in a single download session description and are downloaded in a single download session. The HNED shall keep track of this association between content item and files. The download session description may announce the content item format in the *Content-Item-Format* attribute.

The following content item formats SHALL be supported:

- *Content-Item-Format=0:*
  In this case, the download session description describes the download of any collection of files. The CDS HNED SHOULD interpret the Content-Type of the first file in the download session description for the consumption of the content item.

- *Content-Item-Format=1:*
  In this case, the download session description describes the download of a single MPEG-2 Transport Stream with the content type according to clause 10.4.2.1.

- *Content-Item-Format=2:*
  In this case, the download session description describes the download of an MPEG-2 Transport Stream with the format according to clause 10.4.2.1 and associated BCG metadata according to clause 10.4.2.2. The CDS HNED shall always interpret the BCG metadata before accessing the MPEG-2 Transport Stream.

The following content item formats MAY be supported:

- *Content-Item-Format=3:*
  In this case, the download session description describes the download of a file in DVB File format to clause 10.4.2.3 where the file encapsulates an MPEG-2 TS.

If the content item format is not present, then *Content-Item-Format=0* shall be assumed.

NOTE 1: By the use for *Content-Item-Format*=0, it is not prevented that other formats are carried within CDS services, especially if the appropriate MIME media type is defined.

NOTE 2: Details of which codec(s) are used by the file may be signaled within the AVAttributes field within the descriptive metadata associated with the file. An HNED is able to inspect this field and to decide if it wishes to download the file.

10.5 CDS Download Session Description

10.5.1 Overview

Each content item is acquired in a download session that is described by a download session description. A download session description is composed of several download session parameters and provides information to initiate or join download sessions and to reliably download content items. The download of content items requires the download of one or several individual files.

The acquisition of the individual files of a content item requires the construction of a unique reference link for each file. The referencing methods of file locations in CDS download session descriptions are introduced in clause 10.5.2. The download session parameters and their semantics are introduced in clause 10.5.3. Different types of download sessions are described in clause 10.5.4 along with the assigned download session parameters. The following download session types are distinguished:

- Scheduled Multicast Download (SMD) Session.
- Carousel Multicast Download (CMD) Session.
- Unicast Download (UD) Session.

The CDS HNED function SHALL support all mandatory features of all three types of download sessions.

Download session descriptions can be provided in XML or in SDP syntax. The CDS HNED SHALL support XML syntax. The CDS HNED MAY support SDP syntax. Both download session descriptions support the same parameter set. The XML syntax for the download session parameters is defined in clause C.2.3. The SDP syntax for the download session parameters is defined in clause G.2.

The transport of download session descriptions is defined in clause 10.5.5.

10.5.2 Referencing file locations for download

The downloading of content items includes the download of one or more files within a download session. The download session description provides the information about all files that have to be downloaded for a specific content item. In addition, each file within a content item may be downloaded from different locations. Therefore, the download session descriptions for unicast download may define alternative sources in the initial session description or in the description of a redirection of the actual file download (see clause 10.6.3). In case of multicast download the files have to be identified within the File Delivery Table (FDT) and the file repair procedures have to know the location of the repair data (see clause 10.6.2).

In CDS file locations are uniquely referenced using the URI scheme defined in RFC 3986 [80] with some restrictions. The following definitions as defined in RFC 3986 [80] are re-used in the present specification:

- *Generic URI* <URI> as defined in RFC 3986 [80], clause 3.
- *Scheme syntax component* <scheme> as defined in RFC 3986 [80], clause 3.1.
- *Authority syntax component* <authority> as defined in RFC 3986 [80], clause 3.2.
- *Query syntax component* <query> as defined in RFC 3986 [80], clause 3.4.
- *Fragment syntax component* <fragment> as defined in RFC 3986 [80], clause 3.5.
- *Absolute URI* <absolute-URI> as defined in RFC 3986 [80], clause 4.3.

- *Relative reference* <relative-ref> as defined in RFC 3986 [80], clause 4.2.
- *Absolute path* <path-absolute> as defined in RFC 3986 [80], clause 3.3.

NOTE: The *absolute path* syntax <path-absolute> is a special case of the *relative reference* syntax <relative-ref>.

The following definition is used in the context of the present document:

- *HTTP-Server Base URI* <http-server-base-URI> is an <absolute-URI> with a fixed scheme of "http://" and an <authority> component (host, port) only. The URI shall not contain any absolute path, query or fragment syntax component.

The following principles apply for referencing files:

- A file location is referenced by a *target URI* that has the syntax of an absolute URI <absolute-URI>. By the definition of the absolute URI syntax, a fragment syntax component shall not be used. Furthermore, the target URI referencing the file location SHALL not use any *query* syntax component.
- A CDS HNED may construct a target URI by *reference resolution* as defined in RFC 3986 [80], section 5. To create a target URI by the application of reference resolution, the CDS HNED requires
  - a *base URI* of syntax <http-server-base-URI>;
  - a *relative reference* of syntax <path-absolute>.
- In the CDS context the base URI identifies the server part and the files that have to be downloaded are referenced by an absolute path syntax component <path-absolute>.

10.5.3 Download Session Description Parameters

Download session descriptions contain several parameters that describe the content item format as well as the methods on how to acquire the content item. The semantics of the parameters is described.

10.5.3.1 General Parameters

The following download session parameters are applicable to any download session.

Service-Provider-Domain: The SP domain is an Internet DNS domain name registered by the SP that uniquely identifies the SP. There SHALL be exactly one occurrence of a *Service-Provider-Domain* parameter in a download session description.

Download-Session-ID: The download session ID is a numeric string and identifies a specific session description. There SHALL be exactly one occurrence of a *Download-Session-ID* parameter in a download session description.

Download-Session-Version: The download session version identifies the version of a specific session description. It is an integer value between 0 and 255 which is increased by 1 for each new version of a session description. It overflows from 255 to 0. There SHALL be exactly one occurrence of a *Download-Session-Version* parameter in a download session description.

Content-Item-Format: The content item format describes the format of the content item. If the content item format is not present, then *Content-Item-Format=0* shall be assumed. For details on content-item formats refer to clause 10.4.3.

Download-Session-Mode: The download session type specifies one of the three download session modes: "SMD", "CMD" or "UD" according to clause 10.5.4. There SHALL be exactly one occurrence of a *Download-Session-Mode* parameter in a download session description.

NOTE 1: The differentiation between single server and multiple servers unicast download is based on the unicast download parameters and not the *Download-Session-Mode*.

Download-Session-Time-Information: The *Download-Session-Time-Information* provides the information when a CDS download session is active and the HNED can join it to perform the download. In case of a Unicast and Carousel Multicast Download the start and end time of the active time window shall be defined. In case of a Scheduled Multicast Download only the start time of the session shall be defined. The information is provided at the session level.

NOTE 2: This information is identical to the availability for download information in BCG the extended OnDemandProgramType and Extended On-demand decomposed binary locator defined above.

Successful reception of the advertised content items shall be reported to the CDS network function. To enable this, the HNED requires reception reporting parameters:

For each reception reporting server:

- Reception-Reporting-Server-URI: HTTP URI of the reception reporting server.
- In case more than one server is provided the CDS HNED function shall randomly select one of the provided servers. In the absence of the parameter, reception reporting is not supported for this download session.

If at least one reception reporting server is provided then the following parameters MAY be provided:

- Reception-Reporting-Mode: Defines the level of details provided by the reception reporting:
  - Reception-Reporting-Mode=0: Content item reporting
  - Reception-Reporting-Mode=1: Content item and file reporting
  - Reception-Reporting-Mode=2: Content item, file and chunk reporting

In the absence of the parameter, the CDS HNED function SHALL assume *Reception-Reporting-Mode=0*. In case of multicast download, *Reception-Reporting-Mode*=2 shall not be used.

Reception-Reporting-Offset-Time: This parameter defines the offset time for the reception reporting back-off time (see clause 10.6.5). It is an integer value expressed in seconds. In the absence of the parameter, the CDS HNED function SHALL assume *Reception-Reporting-Offset-Time* =0.

Reception-Reporting-Random-Time-Period: This parameter defines the random time period for the reception reporting back-off time (see clause 10.6.5). It is an integer value expressed in seconds. In the absence of the parameter, the CDS HNED function SHALL assume *Reception-Reporting-Random-Time-Period*=0.

10.5.3.2 Unicast Download Related Parameters

The following download session parameters are applicable to unicast download session.

For each file that has to be downloaded:

File-Reference: This is a reference to the file to be downloaded. The syntax of this reference SHALL conform to the *absolute path* syntax <path-absolute>.

File-Content-Type: Content-Type of file as defined in clause 10.4.2 referring to a registered MIME-type.

File-Length: Length of file (as defined in RFC 2616 [39] for the Content-Length entity header field).

File-Digest: Base64 of 128 bit MD5 digest of the file as defined in RFC 2616 [39] for the Content-MD5 entity header field.

In case download of a file in individual chunks is provided:

Chunk-Length: Length of a chunk of the file. The file is divided into chunks of constant length as defined by this parameter (except for the last chunk which could be shorter depending on the file length) that can be downloaded separately.

For each chunk (chunks are numbered from 1 to n in the order they make up the file):

Chunk-Digest: Base64 of 128 bit MD5 digest of chunk as defined in RFC 2616 [39] for the Content-MD5 entity header field.

For each server where the file is available for download:

Server-Base-URI: The base URI of the file location represents the server from which the file can be downloaded. The *Server-Base-URI* syntax of this reference SHALL conform to the <http-server-base-URI> syntax (see clause 10.5.2).

In case download of a file in individual chunks is provided:

Available-Chunk-List: List of all chunks of a file that are available on that server (*chunks are numbered from 1 to n in the order they make up the file)*. If the parameter is not provided the whole file (all chunks) is available on the server.

"Available-Chunks-List" grammar using conventions of [39], clause 2:

```
chunks-list         =   1#( single-chunk-num | chunk-range-spec )
single-chunk-num    =   1*DIGIT
chunk-range-spec    =   first-chunk-num "-" last-chunk-num
first-chunk-num     =   1*DIGIT
last-chunk-num      =   1*DIGIT
```

Examples of valid "Available-Chunks-List":

- 6.
- 8-11.
- 3, 14, 29.
- 5-12, 22, 36.

In case multiple servers and chunk information (at least *File-Length* and *Chunk-Length*) is provided, the download of the file can be distributed over these servers (see clause 10.6.3.3), otherwise the file is downloaded from a single server (see clause 10.6.3.2).

10.5.3.3 Multicast Download Related Parameters

The following download session parameters are applicable to multicast download sessions.

For each file that has to be downloaded:

File-Reference: This is a reference to the file to be downloaded. The syntax of this reference SHALL conform to the *absolute path* syntax <path-absolute>. In the absence of the parameter, the CDS HNED function SHALL download all files transported by the FLUTE session. In case *Content-Item-Format=0* is used and the content item contains more than one file, this field is mandatory.

To join a FLUTE multicast download session the following parameters based on the definitions in TS 102 472 [66], clause 6.1.13 are used.

IP-Source-Address: Source IP address of the multicast group of the FLUTE session. There shall be exactly one IP source address per multicast file download session.

Transport-Session-Identifier: Transport Session Identifier (TSI) of the session. The TSI together with the *IP-Source-Address* uniquely identifies a FLUTE session for a given IP source address during the time that the session is active, and also for a large time before and after the session is active. There SHALL be exactly one occurrence of this parameter in a FLUTE session description. The TSI shall be an integer value.

FEC-Encoding-ID: Describes the FEC scheme. Two schemes are supported:

- *FEC-Encoding-ID*=0: Compact No-Code FEC scheme.
- *FEC-Encoding-ID*=1: Raptor FEC scheme.

If the *FEC-Encoding-ID* is not provided, the CDS HNED function shall assume *FEC-Encoding-ID*=0.

Note that FEC Object Transmission Information (OTI) SHALL be delivered using the ALC/LCT extension header EXT_FTI or the FDT (see clause 10.6.2.2).

Number-Of-Channels: Number of FLUTE/LCT channels of the FLUTE session. The multiple channel attribute parameter indicates to the receiver that the sender is using multiple channels in the FLUTE session to transmit data. The attribute also indicates the number of channels used by the sender. In absence of this parameter, a CDS HNED function shall understand that exactly one FLUTE channel is used for the multicast download session.

For each channel:

IP-Multicast-Address: IP multicast address for each FLUTE channel. There SHALL be exactly one occurrence of this parameter for each channel in a FLUTE session description.

IP-Multicast-Port-Number: Port number for each FLUTE channel. There SHALL be exactly one occurrence of this parameter for each channel in a FLUTE session description.

Max-Bandwidth: Maximum bandwidth to be used by each FLUTE channel. The TIAS bandwidth modifier as defined in RFC 3890 [81] shall be used. In the absence of the parameter, no maximum bandwidth limit shall be assumed.

Furthermore, the order of FLUTE channels in the download session description provides the order in which the HNED shall join and leave the channels.

To participate in completion polling in a scheduled multicast download session, the CDS HNED function needs to know the following parameters associated with the completion polling. In the absence of the parameters, completion polling is not applied for this download session.

Completion-Poll-Response-Server-Address: IP address to which CDS HNED function will send completion poll responses.

Completion-Poll-Response-Server-Port-Number: Port number for completion poll responses.

If the file repair mechanism for multicast download is supported (see clause 10.6.2.6), the CDS HNED function needs to know the following parameters associated with file repair mechanism.

For each recovery server:

Recovery-Server-Base-URI: The base URI of a unicast recovery server. The *Recovery-Server-Base-URI* syntax of this reference SHALL conform to the <http-server-base-URI> syntax (see clause 10.5.2).

If a *Recovery-Server-Base-URI* is provided then the following parameters MAY be provided:

Recovery-Mode: Describes which file repair procedure to be applied, an IPDC-like file repair procedure or a specific CDS file repair procedure.

- *Recovery-Mode=0:* CDS file repair procedure.
- *Recovery-Mode=1:* IPDC-like file repair procedure.

In the absence of the parameter, the CDS HNED function SHALL assume *Recovery-Mode=0*.

Recovery-Offset-Time: This parameter defines the offset time for the file repair back-off time (see clause 10.6.2.6). It is an integer value expressed in seconds. In the absence of the parameter, the CDS HNED function SHALL assume *Recovery-Offset-Time*=0.

Recovery-Random-Time-Period: This parameter defines the random time period for the file repair back-off time (see clause 10.6.2.6). It is an integer value expressed in seconds. In the absence of the parameter, the CDS HNED function SHALL assume *Recovery-Random-Time-Period* =0.

10.5.4 Download session Modes

CDS download sessions may be operated in one of four modes:

- Scheduled Multicast Download (SMD) Session.
- Carousel Multicast Download (CMD) Session.
- Unicast Download (UD) session with Single Server (SS) downloads.
- Unicast Download (UD) session with Multiple Server (MS) downloads.

The download session descriptions of the each of a download sessions requires and permits certain parameters from the list of download session parameters in clause 10.5.3. Table 26 provides the mapping of download session parameters to download session types: "M" refers to a mandatory parameter and SHALL be included by the CDS network function in a download session description. "O" refers to an optional parameter and MAY be included by the CDS network function in an announcement of the respective download session. "N" refers to the case that this parameter SHALL not be included for this download session mode. The table also provides the type for each parameter.

Table 26: Mapping of Download session Parameters to Download sessions Modes

Parameter	Type	SMD	CMD	UD	
				SS	MS
General parameters					
Service-Provider-Domain	domain name	M		M	
Download-Session-ID	unsigned integer	M		M	
Download-Session-Version	unsigned integer	M		M	
Content-Item-FormatType	unsigned integer(2)	O		O	
Download-Session-Mode	syntax specific	M		M	
Download-Session-Time-Information	syntax-specific	M		M	
Reception-Reporting-Server-URI (one per sever)	<http-server-base-URI>	O		O	
Reception-Reporting -Offset-Mode (one per sever)	unsigned integer (2)	O		O	
Reception-Reporting -Offset-Time (one per server)	unsigned integer(64)	O		N	
Reception-Reporting-Random-Time-Period (one per server)	unsigned integer(64)	O		N	
Unicast Download Related Parameters					
File-Reference (one per file)	<path-absolute>	N	N	M	M
File-Content-Type (one per file)	MIME type	N	N	O	O
File-Length (one per file)	unsigned integer	N	N	O	M
File-Digest (one per file)	base64	N	N	O	O
Chunk-Length (one per file)	unsigned integer	N	N	N	M
Chunk-Digest (one per file and chunk)	base64	N	N	N	O
Server-Base-URI (one per file and server)	<http-server-base-URI>	N	N	M	M
Available-Chunk-List (one per file and server)	list of unsigned integer	N	N	N	O
Multicast Download Related Parameters					
File-Reference (1…n)	<path-absolute>	O	O	N	
IP-Source-Address	IP address or fully qualified domain name	M	M	N	
Transport-Session-Identifier	unsigned integer (48)	M	M	N	
FEC-Encoding-ID	unsigned integer (8)	O	O	N	
Number-Of-Channels	unsigned integer (4)	O	O	N	
IP-Multicast-Address (one per channel)	IP address	M	M	N	
IP-Multicast-Port-Number (one per channel)	unsigned integer (16)	M	M	N	
Max-Bandwidth (one per channel)	unsigned integer	O	O	N	
Completion-Poll-Response-Server-Address	IP address or fully qualified domain name	O	N	N	
Completion-Poll-Response-Server-Port-Number	unsigned integer (16)	M	N	N	
Recovery-Server-Base-URI (one per server)	<http-server-base-URI>	O	O	N	
Recovery-Mode	unsigned integer(1)	O	O	N	
Recovery-OffsetTime	unsigned integer(64)	O	O	N	
Recovery- Random-Time-Period	unsigned integer(64)	O	O	N	

NOTE: M=Mandatory.
O=Optional.
N=Not applicable.

10.5.5 Transport of download session descriptions

The XML-based and SDP-based download session descriptions are provided to the HNED by use of unicast or multicast transport via the CDS-1 interface.

10.5.5.1 Multicast transport of XML-based download session descriptions

For the multicast transport of XML session descriptions, the DVBSTP protocol as defined in clause 5.4.1 is used with the following CDS specific profile:

- The Payload ID field shall be set to 0xB1 as specified in table 1.
- XML-based session descriptions can be delivered un-compressed or BiM compressed as defined in clause 5.5.2.

The CDS XML fragments are constantly sent out on the multicast group in a carousel manner. In order for the HNED to access a specific segment it may have to wait until the segment is sent out on the multicast group. In case the multicast download session is announced to the HNED beforehand via the SD&S BCG discovery record (see clause 5.2.6.6) the HNED can constantly listen to the multicast group and cache the latest versions of the XML segments. The HNED can in this case access the relevant segment immediately from the cache without the need to wait until the segment is send out on the multicast group.

In case the segment contains more than one session description record the optional *Download-Session-ID* fragment identifier in the referencing URI (see clause 10.3.2.1) identifies the specific session description record.

Where multicast is used to distribute the download session description information, XML records may be segmented, that is divided up into smaller units, to enable easier processing in the HNED, or variable access times. Note that a record may be divided into a single segment. Each segment shall contain an integral number of download session elements as defined above (specifically, a segment shall not contain part of a download session element). Each segment shall be valid and well formed. Segment IDs need not be contiguous.

The multicast group may be shared between several SPs. The optional ServiceProvider ID field of the DVBSTP header is in this case used to identify the SP.

10.5.5.2 Unicast transport of XML-based download session descriptions

The unicast transport of XML session descriptions is aligned with the SD&S unicast transport as defined in clause 5.4.2. The HTTP Protocol shall be used for the communication between the HNED and the CDS session description server(s). A session description request is defined for the delivery of a specific session description record. The request shall return a XML segment with one or more download session description records.

NOTE: In case the returned segment contains more than one session description record the optional Download-Session-ID fragment identifier in the locator identifies the specific session description record. The specific session description record is always extracted by the HNED. The request always delivers the whole segment to the HNED.

The request has one mandatory parameter that takes the SegmentID. Optionally a segment version may be specified in the request, this will indicate to the server the current version of the segment at the HNED.

The HNED may cache segments. In case a download session description from the same segment is requested later the HNED can provide the version of the cached segment, which can be found in the XML record, in the request. The response to the request shall return the service discovery record for the specified segment only if a newer version is available. If the segment has not changed then the server shall return status code "204" as per the RFC 2616 [39] to indicate that the request has been processed successfully but that there is no entity-body to return.

When the segment version is not specified, the response to the request shall return the actual version of the specified segment. When a record is not found, the server shall return status code "404" as per the RFC 2616 [39].

The download session description request shall comply with the following format:

```
'GET /dvb/cds/session_description' '?Segment=' SegmentItem 'HTTP/1.1' CRLF
'Host: ' host [':' port ] CRLF
```

The SegmentItem parameter is a SegmentId with an optional field for the version number.

```
SegmentItem     = SegmentId 0*1('&Version='VersionNumber)
SegmentId       = 4*4 HEXDIG; any hex number from 0x0000 to 0xffff
VersionNumber   = OCTET; any hex number from 0x00 to 0xff
```

Note that a payload ID as defined for the service discovery request in clause 5.4.2.2 is not provided as the request type of "/dvb/cds/session_description" already indicates that session description information is requested.

For example the following request can be constructed to request the session description records of segment 1 from the server sdes.dvb.org:

```
'GET /dvb/cds/session_description?Segment=0001 HTTP/1.1' CRLF
'Host: xyz.company.com:1022' CRLF
```

10.5.5.3 Multicast transport of SDP-based download session descriptions

The multicast transport of SDP session descriptions uses the SAP Protocol as defined in RFC 2974 [77] with the following CDS specific profile:

- The PT is "application/sdp".
- The payload can be compressed using zlib.
- Authentication is not supported.

SDP session descriptions are constantly sent out on the multicast channel in a carousel manner. In order for the HNED to access a specific session description it may have to wait until it is send out on the multicast channel. In case the multicast delivery channel is announced to the HNED in advance in the BCG discovery record (see clause 5.2.6.6), the HNED may constantly listen to the multicast channel and cache the latest versions of the session descriptions. The HNED can in this case access the relevant session description immediately from the cache without the need to wait until the segment is sent out on the multicast channel.

10.5.5.4 Unicast transport of SDP-based download session descriptions

The unicast transport of SDP session descriptions uses the HTTP Protocol for all communication between the HNED and the CDS session description server(s). The HNED requests a SDP file from the CDS session description server. The file shall contain a single SDP session description as defined in clause G.2. The Content-Type for the SDP file shall be "application/sdp". The file is delivered uncompressed.

10.6 CDS Content Item Download

10.6.1 Overview

CDS content item download is concerned with the reliable distribution of content items to a single HNED or a population of HNEDs in non real-time manner.

A content item may consist of one or more files. These can be video, audio, combined video and audio and related metadata files as defined in clause 10.4. CDS supports the download of all files associated with a content item as part of a single download session. Within a CDS session the files are delivered either via unicast or multicast download. A CDS session is announced either being unicast or multicast, but not a mixture of both. However by the use of unicast file repair for a multicast session or multicast redirection of a unicast download session, the download mode may change during a session.

CDS is only concerned with the download of the content item, but not with the play out and presentation of the content item. Appropriate consumption and presentation of a content item that consist of several files has to be ensured by the use of the *Content-Item-Format.*

A CDS HNED function SHALL support:

- all mandatory features of multicast content download as specified in clause 10.6.2;
- all mandatory features of unicast content download as specified in clause 10.6.3; and
- all mandatory features of the reception reporting procedures as specified in clause 10.6.5.

Before initiating the download of a content item, it is assumed that the CDS HNED function has access to a download session description that describes the procedures on how to download the referenced content item (see clause 10.5).

The CDS HNED function has completed the download of the content item only if all files of the accessed content item have been completely downloaded.

If the *Download-Session-Mode* is "SMD" or "CMD", the CDS network and CDS HNED functions SHALL apply the multicast content download procedures specified in clause 10.6.2.

If the *Download-Session-Mode* is "UD", the CDS network and CDS HNED functions SHALL apply the unicast content download procedures specified in clause 10.6.3.

Clause 10.6.4 provides guidelines for the parallel download of content items from one or multiple servers.

If the download session description contains at least one Reception-Reporting-Server-URI, the CDS network and CDS HNED functions SHALL apply the reception reporting procedures as specified in clause 10.6.5.

10.6.2 Multicast Content Download

10.6.2.1 Overview

Multicast download modes provide download of content items to multiple HNEDs using IP multicast. It is therefore suitable for efficiently downloading the same content items to many receivers. The availability of content items via Multicast download mode is advertised in the download session description.

Multicast Content Download is organized in download sessions. A download session is characterized as an instance of the CDSs with a start time and optionally an end time as well as addresses of the IP flows used for the download of the files between the start and end time. The start and end times are signaled in the *Download-Session-Time-Information* parameter.

A download session description refers to the download of exactly one content item. However, a multicast session may include the files for more than one content item. In this case the download session description of specific content item identifies which files belong to that content item and only these files will be downloaded. Otherwise all files shall be downloaded.

The *Download-Session-Mode* parameter indicates if Scheduled Multicast Download (SMD) or Carousel Multicast Download (CMD) is applied.

In SMD mode, a multicast session is explicitly scheduled by the network to begin at a start time. HNEDs MAY choose to join the session at the appointed time. The CDS network function SHOULD use completion polling in SMD mode.

In the CMD mode, a session is scheduled to be available over a long period of time and CDS HNEDs MAY join and leave at any time. The CDS network function SHALL NOT use completion polling in CMD mode.

For the actual multicast file distribution, at the appointed session start time, the CDS Network Multicast Server Function begins distribution of the files using the FLUTE protocol [71]. Details on the use of FLUTE in CDS are specified in clause 10.6.2.2. The CDS HNEDs join the session by joining one or more of the IP Multicast groups. For multicast channel selection the IGMPv3 protocol RFC 3376 [47] is used. Procedures on which and how many IP Multicast groups are joined is specified in clause 10.6.2.3.

For SMD mode sessions the SP SHOULD continue the session until all receivers still joined to the group have received the content items being downloaded. The length of the session can be determined using the completion polling function as specified in clause 10.6.2.5.

Furthermore, for SMD sessions, the CDS HNED function SHOULD join the multicast session latest at the appointed start time. In case a CDS HNED joins the scheduled session after the appointed start time and CDS system applies the completion mechanism, the HNED SHALL not respond to the completion poll request messages of the CDS network functions.

In case the CDS HNED cannot complete the download of the content item during the time the multicast session is active, CDS provides file repair mechanisms. These file repair mechanisms are described in clause 10.6.2.6.

10.6.2.2 FLUTE Transport Protocol in CDS

The File deLivery over Unidirectional Transport (FLUTE) protocol [71] SHALL be used for CDS multicast download. The usage of the FLUTE protocol is closely aligned with the Delivery Protocol for File Delivery Services in TS 102 472 [66]. In addition to the basic protocol as specified in RFC 3926 [71], the CDS multicast download is comprised of parts that further specify how FLUTE is used. The purpose of file delivery is to deliver content items in files. A file may contain any type of data (e.g. Audio/Video file, Binary data, Still images, Text, BCG metadata). In the present document the term "file" is used for all objects carried by FLUTE (with the exception of the FDT Instances).

FLUTE is built on top of the Asynchronous Layered Coding (ALC) protocol instantiation [73]. ALC combines the Layered Coding Transport (LCT) building block [71] a congestion control building block and the Forward Error Correction (FEC) building block [48] to provide congestion controlled reliable asynchronous delivery of content to an unlimited number of concurrent receivers from a single sender. See figure 19 for an illustration of FLUTE building block structure.

FLUTE is carried over UDP/IP, and is independent of the IP version and the underlying link layers used.

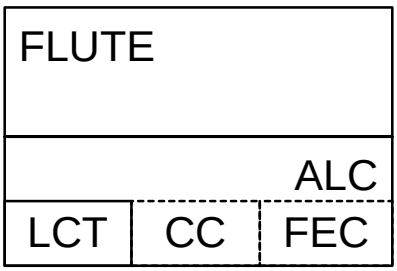


Figure 19: Building block structure of FLUTE

ALC uses the LCT building block to provide in-band session management functionality. The LCT building block has several specified and under-specified fields that are inherited and further specified by ALC. ALC uses the FEC building block to provide reliability. The FEC building block allows the choice of an appropriate FEC code to be used within ALC, including using the no-code FEC code that simply sends the original data using no FEC coding.

ALC is under-specified and generally transports binary objects of finite or indeterminate length. FLUTE is a fully-specified protocol to transport files (any kind of discrete binary object), and uses special purpose objects - the File Delivery Table (FDT) Instances - to provide a running index of files and their essential reception parameters in-band of a FLUTE session.

The CDS HNED and the CDS network SHALL implement all the mandatory parts of the FLUTE specification RFC 3926 [71], as well as ALC RFC 3450 [73] and LCT RFC 3451 [72] features that FLUTE inherits.

In addition, several optional and extended aspects of FLUTE, as described in the following clauses, SHALL be supported by the CDS HNED and network functions.

10.6.2.2.1 Segmentation of files

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.

10.6.2.2.2 Symbol Encoding Algorithm

The applied Symbol Encoding Algorithm is signaled in the download session parameter *FEC-Encoding-ID*.

The "Compact No-Code FEC scheme" [76] (*FEC-Encoding-ID=0*, also known as "Null-FEC") SHALL be supported.

The "Raptor FEC Scheme" (*FEC-Encoding-ID=1*) as defined in [66], clause 8 and RFC 5053 [78] consists of two distinct components:

- Source block and source packet construction and reception.
- Repair packet construction and reception and Raptor FEC encoding and decoding.

The CDS HNED function SHALL support the source block and source packet construction and reception for the "Raptor FEC Scheme". Support of the Source Block and Source Packet construction component requires support of the FEC Payload ID and FEC Object Transmission Information defined in [66], clauses 8.1.2 and 8.1.3 as well as the source packets constructed according to [66], clauses C.3.1 and C.3.2.1.

The CDS HNED function that supports repair packet construction and reception and Raptor FEC encoding and decoding requires support of annex C of [66].

10.6.2.2.3 Use of multiple FLUTE channels

A file (or some encoding symbols of a file) MAY be sent simultaneously or at different times over multiple channels.

The number of FLUTE channels is signaled in the delivery parameter *Number-of-Channels*.

The use of multiple FLUTE channels for a FLUTE session SHALL be supported by CDS HNEDs and MAY be supported by CDS network functions.

The HNED SHALL support the reception of at least 16 FLUTE channels within one FLUTE session.

Multiple channels may be used to provide multicast rate adaptation according to clause 10.6.2.3.

10.6.2.2.4 Blocking Algorithm

In the case of the Compact no-Code FEC Scheme (*FEC-Encoding-ID=0*) [76], the "Algorithm for Computing Source Block Structure" described within the FLUTE specification SHALL be used.

In the case of the Raptor FEC Scheme (*FEC-Encoding-ID=1*), the source block construction algorithm described in TS 102 472 [66], clause C.3.1, SHALL be used.

10.6.2.2.5 Congestion Control

No mechanisms for multicast congestion control are provided in the present specification. Multicast rate adaptation as introduced in clause 10.6.2.3 may be used for the purpose of congestion control.

10.6.2.2.6 Content encoding of files for transport

Files may be content encoded for transport in the file delivery method using the GZip algorithm RFC 1952 [75]. Terminals 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".

10.6.2.2.7 Further Considerations

For informative ALC packet size considerations, refer to TS 102 472 [66], clause 6.1.8.

For normative procedures on signaling the end of file delivery and end of file download session, refer to TS 102 472 [66], clause 6.1.9.

Spanning files over several download sessions SHALL not be used.

File grouping as specified in TS 102 472 [66], clause 6.1.11 SHALL not be used. The grouping of files for a content item is signaled by the download session description.

File versioning as specified in TS 102 472 [66], clause 6.1.12 SHALL not be used. Instead content item versioning as defined in clause 10.6.6 is used.

10.6.2.2.8 Signaling of Parameters with FLUTE

10.6.2.2.8.1 Signaling of Parameters with basic ALC/FLUTE Headers

FLUTE and ALC mandatory header fields SHALL be as specified in the FLUTE specification RFC 3926 [71] and the ALC specification RFC 3450 [73], with the following additional specializations:

- In FLUTE the following applies: The length of the CCI (Congestion Control Identifier) field SHALL be 32 bits and it is assigned a value of zero (C=0).
- The Transmission Session Identifier (TSI) field SHALL be of length 16 bits (S=0, H=1, 16 bits) or 32 bits (S=1, H=0) when TOI is an identifier of 32 bits.
- The Transport Object Identifier (TOI) field SHOULD be of length 16 bits (O=0, H=1) or 32 bits (O=1, H=0).
- Only Transport Object Identifier (TOI) 0 (zero) SHALL be used for FDT Instances.

- The following features SHALL be used for signaling the end of session; the following features SHOULD be used for signaling an end of object transmission to the receiver prior to the FDT expiry date:
    - The Close Session flag (A) for indicating the end of a session as described in clause to TS 102 472 [66], clause 6.1.9.
    - The Close Object flag (B) for indicating the end of an object as described in clause to TS 102 472 [66], clause 6.1.9.

In FLUTE the following applies:

- The LCT header length (HDR_LEN) SHALL be set to the total length of the LCT header in units of 32-bit words.
- For "Compact No-Code FEC scheme" (*FEC-Encoding-ID*=0), the payload ID SHALL be set according to RFC 3695 [74] such that a 16 bit Source Block Number (SBN) and then the 16 bit ESI are given.

10.6.2.2.8.2 Signaling of Parameters with FLUTE Extension Headers

FLUTE extension header fields EXT_FDT, EXT_FTI, EXT_CENC according to RFC 3926 [71] SHALL be used as follows:

- EXT_FTI SHALL be included in every FLUTE packet carrying symbols belonging to any FDT Instance.
- FDT Instances SHALL not be content encoded and therefore EXT_CENC SHALL not be used.

In FLUTE the following applies:

- EXT_FDT is in every FLUTE packet carrying symbols belonging to any FDT Instance.
- FLUTE packets carrying symbols of files (not FDT instances) SHALL not include the EXT_FDT.

The optional use of EXT_FTI for packets carrying symbols of files (not FDT instances) SHALL comply to FLUTE for the signaling of FEC Object Transmission Information associated to FEC Encoding 0.

When Raptor forward error correction code (*FEC-Encoding-ID*=1) is used, the EXT_FTI format as defined in TS 102 472 [66], clause 8.1.3, SHALL be used.

10.6.2.2.8.3 Signaling of parameters with FDT instances

The FLUTE FDT Instance schema defined in clause 10.6.2.2.8 SHALL be used. Some of the data elements can be included at the FDT-Instance or at the file level. In this case, the data element values in the file element override the same in the FDT Instance element. In addition, the following applies to both the FDT-Instance level information and all files of a FLUTE session.

The inclusion of these FDT Instance data elements is mandatory according to the FLUTE specification:

- Content-Location (URI of a file); this shall be an *absolute path* syntax <path-absolute> as defined in clause 10.5.2. No server information SHALL be included.
- TOI (Transport Object Identifier of a file instance);
- Expires (expiry data for the FDT Instance).

Additionally, the inclusion of the following FDT Instance data elements is mandatory:

- Content-Length (source file length in bytes);
- Content-Type (content Mime type). This attribute SHALL be either in the FDT-Instance or File element or in both.

The inclusion of the following FDT Instance data elements is optional and depends on the FEC Scheme:

- FEC-OTI-Maximum-Source-Block-Length;

- FEC-OTI-Encoding-Symbol-Length;
- FEC-OTI-Max-Number-of-Encoding-Symbols;
- FEC-OTI-Scheme-Specific-Info.

These optional FDT Instance data elements may or may not be included for FLUTE in CDS:

- Complete (the signaling that an FDT Instance provides a complete, and subsequently not modifiable, set of file parameters for a FLUTE session may or may not be performed according to this method);
- FEC-OTI-FEC-Encoding-ID (the default value is FEC Encoding ID 0);
- FEC-OTI-FEC-Instance-ID;
- Content_Encoding;
- Transfer_length;
- Content-MD5 (Checksum of the file as defined in RFC 3926 [71]).

10.6.2.2.9 FDT Structure

Table 27 provides an overview of the FDT structure. The corresponding XML schema is given in TS 102 472 [66], clause 6.1.15. For details on syntax and semantics refer to RFC 3926 [71].

Table 27: Overview - FLUTE File Delivery Table (FDT) structure (for details, refer to RFC 3926 [71])

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
FDT-Instance-Attributes	**Common Attributes for all the files described by the FDT instance**	
Expires	expiry time of the FDT Instance.	M
Complete	when present and TRUE, signals that no new data will be provided in future FDT Instances within this session.	O
Content-Type	content type.	O
Content-Encoding	Content encoding.	O
FDT-Instance-Delivery-Attributes	Attributes related to the delivery of all files described by the FDT instance	
FEC-OTI-FEC-Encoding-ID	Identification of FEC algorithm.	O
FEC-OTI-FEC-Instance-ID	FEC instance depending on the FEC algorithm identification.	O
FEC-OTI-Maximum-Source-Block-Length	The maximum number of source symbols per source block.	O
FEC-OTI-Encoding-Symbol-Length	Length of encoding symbols in bytes.	O
FEC-OTI-MaxNumber-Of-Encoding-Symbols	Maximum Number of Encoding Symbols that can be generated for a source block.	O
FEC-OTI-Scheme-Specific-Info		O
File Attributes (one per file)		
Content-Type	MIME media type of content.	O
Content-Encoding	Compression.	O
Content-Location	<path-absolute>.	M
Content-Length	Size of the content.	M
Content-Digest	Hash of the content (MD5).	O

Element/Attribute Name		Element/Attribute Description	Mandated/ Optional
	Content-Delivery-Attributes	**Attributes related to the delivery of the file**	
	TOI	Transport Object Identifier.	M
	Transfer-Length	Size of the transport object carrying the content.	O
	Bandwidth-Requirement	Aggregate rate of sending packets to all channels.	O
	FEC-OTI-FEC-Encoding-ID	Identification of FEC algorithm.	O
	FEC-OTI-FEC-Instance-ID	FEC instance depending on the FEC algorithm identification.	O
	FEC-OTI-Maximum-Source-Block-Length	The maximum number of source symbols per source block.	O
	FEC-OTI-Encoding-Symbol-Length	Length of encoding symbols in bytes.	O
	FEC-OTI-MaxNumber-Of-Encoding-Symbols	Maximum Number of Encoding Symbols that can be generated for a source block.	O
	FEC-OTI-Scheme-Specific-Info		O
NOTE 1: Mandatory (M) here means that if the Optional parent element is transmitted, then this field SHALL be present. NOTE 2: Mandatory means that the CDS network function shall signal the corresponding element.			

10.6.2.3 Multicast Rate Adaptation

Multicast rate adaptation is supported by the use of multiple FLUTE channels for a single FLUTE session. All channels are transported via different multicast groups.

10.6.2.3.1 CDS network procedures

To support multicast rate adaptation, the CDS network multicast download function SHOULD use multiple FLUTE channels in combination with "Raptor FEC Scheme". The number of FLUTE channels is advertised in the download session description in the parameter *Number-Of-Channels*.

Each FLUTE channel is transported via a dedicated multicast group identified by a unique *IP-Multicast-Address.* In addition to support multicast rate adaptation the CDS network function SHOULD also signal the *Max-Bandwidth parameter*, for each channel. By the order of the channels listed in the download session description, the CDS network function defines in which order the HNED shall join the channels.

The CDS network function SHALL not exceed the maximum bandwidth advertised by the *Max-Bandwidth* parameter, for each multicast group.

The distribution of source packets as well as FEC packets, if applicable, to different multicast groups may be done in various fashions by the CDS network function and is beyond the scope of the present document.

In case Raptor FEC is supported by all CDS HNED functions, the CDS network function SHOULD evenly distribute FLUTE/UDP packets across multicast groups according to the data rate for each group.

The allocation of bandwidth to multicast groups may be done in various fashions by the CDS network function and is beyond the scope of the present document.

10.6.2.3.2 CDS HNED procedures

By the reception of the download session description the CDS HNED has access to the number of multicast groups in this session (*Number-of-Channels)* as well as for each multicast group access to:

- multicast group identifier, IP-Source-Address, IP-Multicast-Address and IP-Multicast-Port-Number;
- the maximum bandwidth, Max-Bandwidth;
- the multicast group order, defined by the order in which the channels are listed in the download session description.

CDS HNEDs may join the session by joining one or more of the multicast groups. HNEDs SHOULD join a number of multicast groups such that the total bandwidth approximates the HNEDs available bandwidth. The HNED can detect the current available bandwidth by measuring the incoming data rate from the subscribed multicast groups. If this is less than the sum of the advertised rates of the subscribed multicast groups, then this measured rate equals the available bandwidth (at that time).

During reception of a CDS multicast session or sessions, the HNED SHALL calculate the following two figures:

Subscribed CDS bandwidth This is the total bandwidth of the subscribed multicast groups being the sum of the rates of the subscribed multicast groups as advertised in the download parameter *Max-Bandwidth*.

Observed CDS bandwidth: This is the total observed data rate of incoming CDS data at any given time. The timescale on which observed bandwidth is measured and the exact bandwidth measurement algorithm are implementation specific. The timescale should be less than the frequency with which multicast group membership is adjusted and should be greater than one tenth of this time.

The HNED SHOULD adjust multicast group membership on a continuous basis such that the *Subscribed CDS bandwidth* is the least possible value which is not less than the *Observed CDS bandwidth*. The possible values of Subscribed CDS bandwidth are constrained by the advertised multicast groups and their data rates. The HNED SHALL NOT adjust its CDS multicast group membership more than once every 30 s.

The CDS HNEDs SHALL join and leave multicast groups in the order as they are listed in the download session description. Multicast groups that are listed first shall be joined first and left last.

10.6.2.4 File download from the FLUTE session

The files that have to be downloaded by the HNED from the FLUTE session are provided in the download session description (by the *File-Reference* parameters). If the *File-Reference* parameter is not present all files of the FLUTE session SHALL be downloaded.

In case *File-Reference* parameters are provided the HNED compares them against the *Content-Location* URIs in the Flute FDT in order to identify the files in the FLUTE session.

In case that from a previous download session for the same content item (identified by the same CRID) a file with the same <path-absolute> already exists on the local storage and a MD5 digest and content length is provided for the file in the FLUTE FDT the HNED compares that against the content length and MD5 digest of the file on the local storage. If they are the same the file SHOULD not be downloaded. Otherwise the file on the CDS HNED storage SHALL be deleted and the new version of the file SHALL be downloaded.

If requested by the download session description, the HNED SHALL perform a reception reporting as defined in clause 10.6.5 after the successful download of all files of the content item.

10.6.2.5 CDS Network-based Session Completeness

10.6.2.5.1 Basic Principle

If the download session is a scheduled multicast session, i.e., *Download-Session-Mode* ="SMD", the CDS network function SHOULD use a completion polling mechanism to determine when to stop the multicast download session at the Multicast File Server.

CDS HNEDs, which have completely received the file or files being distributed, SHALL leave the subscribed multicast groups and terminate their participation in this download session.

As a result, the multicast tree will shrink as time passes and receivers complete the reception. However, it is likely that not all receivers will complete at the same time due to:

- different receivers have different incoming available bandwidth,
- different receivers experience different packet loss levels.

For this purpose the CDS Multicast Download Function periodically sends a "Completion Poll" message within the FLUTE session. The Completion Poll contains a single, 32-bit field, "POLL_MASK" which HNEDs use to determine whether or not to send a reply to the server. For this purpose, each CDS HNED function is configured with a fixed "pseudo-random" 32-bit number, referred to as POLL_MASK_X. If not specified otherwise the HNED SHALL use the 4 least significant bytes of its MAC address.

To prevent the possibility of message implosion at the server when there are many HNEDs, the HNEDs use the POLL_MASK field from the Completion Poll message to determine whether to reply. The poll request is received only by those HNEDs that have not completed the reception of the file. With the reception of the poll request the CDS

HNED calculates the logical AND of its internal POLL_MASK_X and the POLL_MASK value provided by the server. If the result is zero, the HNED replies to the Completion Poll message by sending a completion poll response message using the completion poll response server connection information.

The Completion Poll response message uses a simple UDP-based protocol. The message is specified in clause 10.6.2.5.2 and contains:

- The sequence number of the request message to which the message refers to.
- The source address and TSI of the FLUTE session.
- The HNED's local poll mask.
- The CDS HNED's estimation of the remaining time it requires to receive the file or files.

NOTE: No special reliability mechanisms are required for the Completion Poll message or the reply. Since the Completion Poll with POLL_MASK=0 will be repeated several times before ending the session the chance that the session ends too early (whilst receivers are still listening) can be made exceedingly low.

The completion poll message formats, the CDS network procedures, and the HNED procedures are described.

10.6.2.5.2 Message formats

10.6.2.5.2.1 Completion Poll Request

The Completion Poll Request is a LCT Header Extension as defined in [i.4]. The Congestion Poll LCT Header Extension SHALL be included in a normal FLUTE packet associated with a transport object and the normal rules for LCT header settings apply.

The format of the Completion Poll Request LCT Header Extension is shown in figure 20.

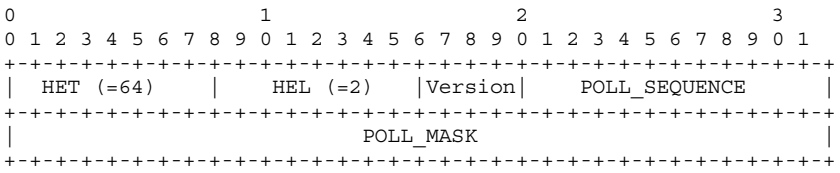


Figure 20: Completion Poll Request LCT Header Extension

Header Extension Type (8 bits) (HET): This field identifies the header extension as a Completion Poll Request. The value SHALL be set to 64.

NOTE: The value "64" is not yet registered with IANA as the LCT draft is draft-ietf-rmt-bb-lct-revised-07.txt [i.4] *is not yet published. Unless unknown surprise happen to DVB will get value 64. The procedure for DVB is to go ahead with value 64 in the present document and submit a registration to IANA as soon as the LCT RFC is published.*

Header Extension Length (8 bits) (HEL): This field gives the length of the header extension in units of 32-bit words.

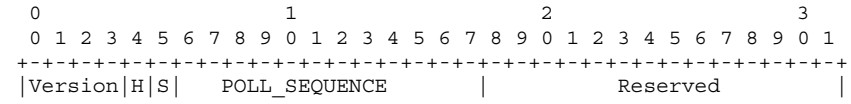
Version (4 bits): The protocol version. In this version of the protocol this field SHALL be set to zero.

POLL_SEQUENCE (12 bits): A sequence number for the Completion Poll Request.

POLL_MASK (32 bits): A value chosen by the CDS network function to support filtering of poll responses.

10.6.2.5.2.2 Completion Poll Response

The completion poll response message shown in figure 21 is sent over UDP transport to the address specified in the *Completion-Poll-Response-Server-Address* and the port *Completion-Poll-Response-Server-Port-Number*.



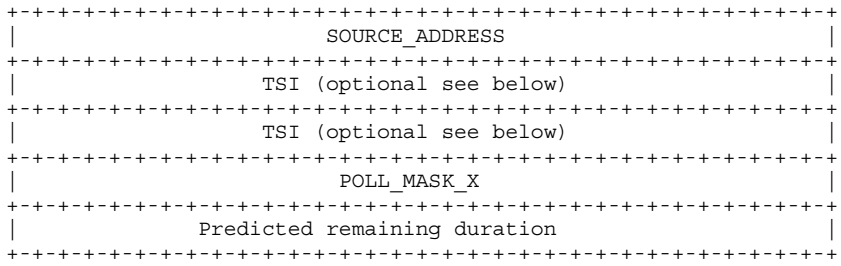


Figure 21: Completion Poll Response

Version (4 bits): This SHALL be set to zero in this version of the protocol.

Transport Session Identifier flag (S, 1 bit): This is the number of full 32-bit words in the TSI field. The TSI field is 32*S + 16*H bits in length, i.e. the length is either 0 bits, 16 bits, 32 bits, or 48 bits. It shall be identical to the S-flag of the FLUTE session to which this completion poll response corresponds.

Half-word flag (H, 1 bit): The TSI field is a multiples of 32-bits plus 16*H bits in length. Shall be identical to the H-flag of the FLUTE session to which this completion poll response corresponds.

Reserved (10 bits): This SHALL be set to zero in this version of the protocol. The receiver SHALL ignore this field.

POLL_SEQUENCE (12 bits): This SHALL be set to the POLL_SEQUENCE value of the Completion Poll Request message which triggered this response.

SOURCE_ADDRESS (32 bits): This SHALL be set to the IPv4 Source Address of the FLUTE session as provided in the download session description.

TSI (0, 32, 64 bits): Depending on the S and H flag, the length of the field is:

- 0 bit for S=H=0;
- 32 bit if S=1 and H=0 or S=0 and H=0;
- 64 bit for S=H=1.

The first 32*S+16*H bit of this field SHALL be identical to the TSI-value of the FLUTE session to which this completion poll response corresponds. The last 16*H bit shall be set to "0".

POLL_MASK_X (32 bits): This SHALL be set to the POLL_MASK_X value used by the client when processing the Completion Poll Request.

Predicted Remaining Duration (32 bits): This SHALL be set to an estimate of the remaining time in seconds that will be required for this receiver to complete reception of the file, or zero if no estimate can be made. The estimation of the remaining time is implementation specific.

10.6.2.5.3 CDS network procedures

In the scheduled download session mode (*Download-Session-Mode*="SMD") the CDS network function SHOULD use completion poll mechanism.

The CDS network function SHALL indicate the use of completion polling by the provisioning of completion poll response server information (*Completion-Poll-Response-Server-Address* and *Completion-Poll-Response-Server-Port-Number*) in the download session description.

The Completion Poll procedure is initiated by the CDS network function by including a Completion Poll Request message in one or more packets of the multicast transmission of the base FLUTE channel. If there are multiple multicast groups, the Completion Poll Request SHALL only be included in the base FLUTE channel.

The CDS network function sets the POLL_SEQUENCE value according to the number of executions of the Completion Poll procedure for the session so far: for the first Completion Poll Request procedure, the POLL_SEQUENCE value SHALL be zero and it SHALL be incremented by one for each new Completion Poll Request procedure.

The CDS network function should set the POLL_MASK value preferably based on an estimate of the number of active receivers and a target number of responses. Note that before execution of the Completion Poll procedure, there is no way to estimate this number and so the server should use a default, maximal value. The target number of response is implementation-specific.

Initially, the CDS network function should choose a POLL_MASK value with a large number of non-zero bits. The probability that a HNED sends a reply to the Completion Poll is then very low (in fact it is 2^{-b}, where b is the number of non-zero bits in POLL_MASK.) Even if there are many receivers still listening to the session, the response to the Completion Poll will be small.

If there are no replies to a Completion Poll message after a short time, the CDS network function SHOULD repeat the message with fewer non-zero bits. This process is repeated until either a reply is received, or a number of Completion Poll message have been sent with POLL_MASK = 0 (in which case all HNEDs should reply) and there are no replies. In this last case the session ends.

The Completion Poll Request SHOULD be included in *Nrepeat* packets in each multicast group, in which case the POLL_SEQUENCE and POLL_MASK values SHALL be the same in every message. The recommended value of *Nrepeat* is 20.

On receipt of a Completion Poll Response message, the server SHALL check the POLL_SEQUENCE field of the received response against the POLL_SEQUENCE value included in the last sent Completion Poll Request message. If they are different, the received message SHALL be discarded.

Otherwise, the CDS network function SHALL calculate the logical AND of the received POLL_MASK_X value and the POLL_MASK of the last sent Completion Poll Request message. If the result is non-zero the server SHALL discard the received message.

Otherwise, the CDS network function MAY use the Predicted Remaining Duration field to determine the remaining duration of the session. The exact use of the Predicted Remaining Duration field is outside the scope of the present document.

The CDS network function SHOULD count the number of received responses within a timeout of *Tresponse*. The recommended value of *Tresponse* is 10 seconds. This number, multiplied by 2^w, where *w* is the poll weight used to calculate the POLL_MASK field of the Completion Poll Request may be used as the estimated number of active receivers for the next Completion Poll Request.

Finally, if no responses are received to the Completion Poll Request within *Tresponse*, and if the poll weight, *w*, used was non-zero, the CDS network function SHOULD repeat the Completion Poll procedure using poll weight *w*-1.

10.6.2.5.4 CDS HNED procedures

The CDS HNED SHALL determine the end of file delivery according to the procedures provided in clause 10.6.2.2.

If a CDS HNED has completely received all the files being defined by the download session description, the download session is completed for this HNED and it SHALL leave the subscribed multicast groups terminating its participation in the download session. It will therefore also no longer receive completion poll request messages.

If completion poll response server information (*Completion-Poll-Response-Server-Address* and *Completion-Poll-Response-Server-Port-Number*) is provided in the scheduled download session mode (*Download-Session-Mode*="SMD") and the HNED is participating in this download session the CDS HNED SHALL expect the reception of Completion Poll Request messages on the first joined multicast group.

On receipt of a Completion Poll Request message, the HNED first checks the POLL_SEQUENCE value. If the HNED has previously processed a Completion Poll Request with a POLL_SEQUENCE value greater than or equal to this one, then the message SHALL be discarded.

Otherwise, the HNED checks the received POLL_MASK value. Each HNED is provided, by implementation-specific means, with a fixed pseudo-random 32-bit number POLL_MASK_X. The HNED calculates the logical AND of the received POLL_MASK and the provided POLL_MASK_X. If this calculated value is non-zero, the Completion Poll Request SHALL be discarded.

Otherwise, the HNED SHALL construct and send a Completion Poll Response message according to the procedures of clause 10.6.2.5.2 and send it to the address indicated in the *Completion-Poll-Response-Server-Address* parameter of the

download session description and the UDP destination port indicated in the *Completion-Poll-Response-Server-Port-Number* parameter of the download session description.

A single message is sent from HNED to the server over UDP containing identification of the session the message refers to, the POLL_SEQUENCE, as well as the CDS HNEDs estimate of the remaining time it requires to receive all the files of this download session.

An HNED compliant to this version of the protocol SHALL ignore the Version field of the Completion Poll Request and MUST ignore any additional data after the POLL_MASK field.

An HNED compliant to this version of the protocol SHALL set the Version field of the Completion Poll Response to zero.

10.6.2.6 File Repair Procedure

10.6.2.6.1 General Procedure

The purpose of the File Repair Procedure is to repair and complete files for which the multicast download session was not completed successfully.

Incomplete reception of a multicast download session may occur for different reasons. Examples are that the HNED is forcibly disconnected from a session, or that an HNED joins a scheduled session after the appointed start time and the download cannot be completed, etc.

The CDS network function indicates the availability of a file repair procedure for this download session by the providing the base URI (see clause 10.5.2) of one or more recovery servers in the download session description (*Recovery-Server-Base-URI* parameter).

Moreover a *Recovery-mode* flag in the download session description indicates to the CDS HNED which kind of file repair procedures to apply: an IPDC-like file repair procedure or a specific CDS file repair procedure.

The CDS HNED shall only initiate the file repair procedure in case the content version of the content item has not changed. For the detailed behavior see clause 10.6.6.

In case of an IPDC-like file repair procedure the CDS HNED follows exactly the File Repair Strategy described in TS 102 472 [66], clause 7.3. The general procedure is then:

- The HNED identifies the missing data from a file delivery according to TS 102 472 [66], clause 7.3.3.
- It waits for the Back-off Time as defined in clause 10.6.2.6.3 of the present document.
- It sends requests for the missing parts of the file according to TS 102 472 [66], clause 7.3.6.
- The CDS network function responds to the message with repair data according to TS 102 472 [66], clause 7.3.7.
- Optionally a redirection of the file repair to another repair server or a multicast session that delivers the same files as defined in TS 102 472 [66], clauses 7.3.7 and 7.3.8 can be used.
    - In case of a redirection to a unicast repair server the returned URI SHALL contain the server base URI in <http-server-base-URI> syntax.
    - In case of a redirection to a multicast repair service the returned URI shall point to the download session description of the multicast download session as defined in clause 10.5, i.e. the returned URI shall be any URI as defined in clause 10.3.2. Both unicast and multicast transport of the session description are supported. The CDS HNED shall then follow the multicast content download procedures defined in this clause to join the session and download the missing data.

In case of a specific CDS file repair the procedure basically follows the IPDC-like file repair procedures except that the unicast download procedures as specified in clause 10.6.3 are used for the file repair procedure. In this case the general procedure is the following:

- The HNED identifies the missing data from a file delivery according to clause 10.6.2.6.2.

- By applying *reference resolution*, it generates a request-URI in *absolute URI* syntax <absolute-URI> constructed from the *Recovery-Server-Base-URI* in <http-server-base-URI> syntax of a randomly selected recovery server and the *File-Reference* in <path-absolute> syntax of the file with the missing data.
- It waits for the Back-off Time as defined in clause 10.6.2.6.3.
- It sends a HTTP requests for the missing parts of the file using the request-URI. The HTTP range header of the request identifies the missing and requested data.
- The CDS network function responds to the request and returns the data or initiates a redirection as defined in clause 10.6.3.4.
- The response could be a redirection as defined in clause 10.6.3.4.

Note that in this case HNEDs which entirely miss a file in the multicast session SHOULD simply use the Unicast download mode in straightforward fashion.

10.6.2.6.2 Identification of file repair needs

At the end of a file delivery (see clause 10.6.2.2) the HNED identifies its file repair needs associated to the delivered content item. The FLUTE stack provides the receiver with sufficient information to determine the source block and encoding symbol structure of each file and corresponding FDT instance.

In case of IPDC-like file repair (*Recovery-mode*=1), the procedures in TS 102 472 [66], clause 7.3.3 shall be applied. In case of CDS-based file repair (*Recovery-mode*=0) the CDS HNED function should invert the Source Blocking as specified in clause 10.6.2.2 to map the received encoding symbols to a partially received file. From this information, the receiver is able to determine the ranges of missing bytes sufficient to complete the reception and recovery of the file and request those bytes range over the recovery procedure.

In the case that the Raptor FEC scheme is used, the receiver SHOULD take into account any Raptor parity symbols that have already been received when determining the ranges of missing bytes sufficient to complete the reception of the specific file. Specifically, the acquired data through the repair procedure should be mapped to encoding symbols by the use of the Source Blocking as specified in clause 10.6.2.2 and if appropriate, FEC decoding should be applied to recover source blocks and the entire file.

Every time the end of a multicast download session is detected, every CDS HNED SHALL check whether there is missing content in the session in comparing the files received in the FLUTE session against the files announced in the service advertisement (e.g. download session description or FLUTE FDT). If some parts of the delivered content are missing the CDS HNED SHALL request sufficient data to recover the entire content item.

10.6.2.6.3 Distribution of Recovery requests over time

To resolve the problem of feedback implosion at each end of file delivery and at the end of the download session every request messages to a unicast download mode server is delayed in adopting the same strategy than in TS 102 472 [66], clause 7.3.4. The offset time and random time period parameters are provided by the session announcement (*Recovery-Offset-Time* and *Recovery-Random-Time*).

10.6.3 Unicast Content Download

10.6.3.1 General

The Unicast content download mode provides download of content items to single HNEDs using IP unicast based on HTTP as defined in RFC 2616 [39]. The availability of content items via Unicast download mode is advertised in the download session descriptions according to clause 10.5.

Unicast Content Download is also organized in download sessions. A download session is characterized as an instance of the CDS with a start time and optionally an end time as well as download URIs for the files corresponding to the content item between the start and end time. The start and end times are signaled in the *Download-Session-Time-Information* parameter.

The download session parameter *Download-Session-Mode*="UD" SHALL be used by the CDS network function to indicate unicast download.

Individual files of different content items may be either downloaded from a single server (single server unicast download) as defined in clause 10.6.3.2 or from multiple servers (single server unicast download) as defined in clause 10.6.3.3.

In case of a single server unicast download, redirection to an alternative single server download, to a multiple server download or to a multicast download session as defined in clause 10.6.3.4 may be performed.

In case that from a previous download session for the same content item (identified by the same CRID) a file with the same *File-Reference* already exists on the local storage and a MD5 digest and content length is provided for the file in the download session description the HNED compares that against the content length and MD5 digest of the file on the local storage. If they are the same the file SHOULD not be downloaded. Otherwise the file on the CDS HNED storage SHALL be deleted and the new version of the file SHALL be downloaded.

10.6.3.2 Single server unicast download

Single server file download SHALL be performed if no file chunk information (missing *File-Length* and *Chunk-Length* parameters) or only a single server location (*Server-Base-URI* parameter) is provided for the file.

The HNED SHALL generate the request-URI by *reference resolution* with the base URI *Server-Base-URI* of a randomly selected server out of the list of announced servers for the file and the relative reference *File-Reference* and initiates a HTTP transfer of the file using the request URI. In case the File-Content-Type is present in the download session description an Accept header SHALL be included in the request with the specified Content-Type.

The CDS network function (HTTP server) may respond with:

- the requested file;
- a redirection request as defined in clause 10.6.3.4;
- a "503" (Service Unavailable) status code and a "Retry-After" response header which indicates to the HNED to retry the initial file request after the delta time or after the date and time provided by the "Retry-After" header;
- a "410" (Gone) status code which indicates that the download session is terminated (see clause 10.6.6).

If the server does not respond as above, e.g. a "500" (Internal Server Error) status code or a "503" (Service Unavailable) without a "Retry-After" response, the CDS HNED shall select another server out of the list of servers announced for that file and start a new file download. The CDS HNED shall continue this procedure until the request was successful or all announced servers have been tried.

In case reception reporting servers are defined in the download session description, the HNED shall perform reception reporting as defined in clause 10.6.5 after the successful download of a file and/or all files of the content item.

10.6.3.3 Multiple server unicast download

Multiple server file download SHALL be performed if file chunk information (at least *File-Length* and *Chunk-Length* parameters) and multiple server locations (*Server-Base-URI* parameters) are provided for a certain file. In this case the following procedures SHALL apply.

In this case the HNED shall randomly distribute the download of individual chunks of the file over the server locations provided for the file, taking into account the availability of the chunks on the individual servers (*Available-Chunk-List* parameter). The HNED shall ensure that all chunks of the file are downloaded.

NOTE 1: The method for distributing the requests over the server locations is outside the scope of the present document.

For each selected server the HNED SHALL generate the request-URI by *reference resolution* with the base URI *Server-Base-URI* and the relative reference *File-Reference*.

For each chunk that shall be downloaded from the server the HNED calculates the byte range based on the constant chunk length (*File-Chunk-Length* parameter) and the chunks position. The byte range SHALL be calculated as defined for the HTTP range header in RFC 2616 [39]. It should be noted that the length of the last chunk could be shorter than the constant chunk length.

The HNED initiates a HTTP request for the chunk using the request-URI and including the byte range of the chunk that is requested from this server into the HTTP range header. In case the Content-Type of the file is specified (File-Content-Type parameter) an Accept header shall be included in the request with the specified MIME media type.

The CDS network function (requested HTTP server) shall return the requested chunk with a status code of 206 (Partial content) and indicate the delivered content range in the Content-Range header.

A HNED may combine the requests for several chunks from the same server location (same request-URI) into a single HTTP request by including the byte ranges of all the chunks into the range header. The server shall respond to a request for multiple none consecutive byte ranges (chunks) with a multipart message (multipart media type "multipart/byteranges") as defined in RFC 2616 [39]. Each byte range is delivered in a separate part of the multipart message.

NOTE 2: The use of a single combined request or several individual requests of chunks from the same server location is outside the scope of the present document.

If the MD5 digest is provided for the chunks in the download session description (*Chunk-Digest* parameter) the HNED SHALL verify the correct reception of the chunk by comparing that digest with the digest of the received chunk. If they are different the received chunk SHALL be discarded.

In case the server returns a 410 (Gone) status code the download session is terminated (see clause 10.6.6).

If a chunk cannot be downloaded successfully the HNED shall try to download the chunk from another server if available.

The HNED constructs the file from the received chunks. If the MD5 digest is provided for the file in the download session description (*File-Digest* parameter) the HNED SHALL verify the correct reception of the file by comparing that digest with the digest of the file constructed from the received chunks. If they are different the received file SHALL be discarded.

The HNED may try to download the file again from other servers if available.

In case reception reporting servers are defined the HNED shall perform reception reporting as defined in clause 10.6.5 after the successful download of a chunk and/or file and/or the content item.

NOTE 3: It is worth to note that nothing in the multiple server unicast download procedures makes any assumptions about the location or topology of the various servers. For example, servers may be network-based or may reside on HNEDs to support advanced distributed media delivery. However, server locations and management is not in the scope of the present document.

10.6.3.4 Redirection

A single server unicast download request might be redirected to:

- an alternative single server file download (see clause 10.6.3.4.1);
- a multiple server file download (see clause 10.6.3.4.2);
- a multicast download session (see clause 10.6.3.4.3).

Redirection is indicated by the CDS network function by providing a HTTP response with a redirection status code (3xx) to the initial HTTP request of the HNED.

10.6.3.4.1 Alternative single server redirection

For a redirection to an alternative single server file download the CDS network function SHALL response to the file download request with a status code of "302" (Found). A new *base URI* with syntax of <http-server-base-URI> for the alternative server is provided by the location field of the response. A "Retry-After" response header may be provided which indicates to the HNED to perform the redirection after the delta time or after the date and time provided by the "Retry-After" header.

The HNED shall initiate a single server file download after the delta time or data and time defined by the "Retry-After" header or immediately if this header is not provided. The single server file download procedures as defined in clause 10.6.3.2 SHALL be performed. The new *base URI* provided by the redirection SHALL be used as the new *Server-Base-URI*.

10.6.3.4.2 Multiple server redirection

For a redirection to a multiple server file download the CDS network function SHALL respond to the file download request with a status code of "300" (Multiple Choices). The entity of the response contains the description for the multiple server file download. A "Retry-After" response header may be provided which indicates to the HNED to perform the redirection after the delta time or after the date and time provided by the "Retry-After" header assuming that this time falls into the download session time announced in the download session description (see clause 10.5).

The description for the multiple server file download uses the semantics and syntax defined in clause 10.6.3.3 for a unicast download session. The CDS HNED shall support XML syntax and may support SDP syntax. The syntax is indicated by the appropriate Content-Type in the response. For the interpretation of the redirection information see clause 10.6.3.4.4.

NOTE 1: The same redirection method is also used for the multicast download redirection. The HNED will know from the Download-Session-Mode parameter which kind of redirection is used. A value of "UD" indicates a unicast redirection. A value of "SMD" or "CMD" indicates a multicast download redirection.

The HNED uses the original *File-Reference* of the file to identify the relevant information in the download session description.

NOTE 2: The download session description may contain information for more than one file, for example if the download session description for the whole content item is reused. Each file is uniquely identified by its *File- Reference* parameter.

The HNED shall initiate the multiple server download after the delta time or data and time defined by the "Retry-After" header or immediately if this header is not provided.

The multiple server file download procedure defined in clause 10.6.3.3 SHALL be used.

It should be noted that the download session description may define a single server file download instead of a multiple server file download for the file. In this case the single server file download procedure as defined in clause 10.6.3.2 shall be used.

10.6.3.4.3 Multicast download redirection

For a redirection to a multicast download the CDS network function responses to the file download request with a status code of "300" (Multiple Choices). The entity of the response contains the description for the multicast download. A "Retry-After" response header SHALL not be used. Information about the time of the multicast download session is provided in the session description.

The description for the multicast download uses the semantics and syntax defined in clause 10.5 for multicast download session. The CDS HNED shall support XML syntax and may support SDP syntax. The syntax is indicated by the appropriate MIME type in the response. For the interpretation of the redirection information see clause 10.6.3.4.4.

NOTE 1: The same redirection method is also used for the multiple server download redirection. The HNED will know from the Download-Session-Mode parameter which kind of redirection is used. A value of "SMD" or "CMD" indicates a multicast download redirection. A value of "UD" indicates a unicast redirection.

The HNED shall initiate the multicast download after the delta time or data and time defined by the "Retry-After" header or immediately if this header is not provided.

The multicast download procedure defined in clause 10.6.2 shall be used. The HNED SHALL use the *File-Reference* parameter of the file to identify it in the FDT of the FLUTE session.

NOTE2 : The FLUTE session and download session description may contain more than one file, for example if the redirection points to a FLUTE session for the whole content item. Each file is uniquely identified by its path-absolute relative reference (File-Path-Absolute parameter).

The present document does not define the procedures used by the CDS network function to determine whether redirection to a multicast session should be used. However, by way of example, one possibility for deciding when to establish a multicast session vs. serving the request via unicast download mode servers, and for coordinating the establishment of the session across multiple servers is described here:

- Unicast download mode requests may be received by many different unicast download mode servers. These servers inform the CDS management function whenever they begin serving a content item.
- The CDS management function is responsible for detecting when a substantial number of users request the same content item and, based on this, establishing a Multicast download session for that content item.
- When a Multicast Download session is established, the CDS management function sends a Session Advertisement over a local multicast group (for example using the Session Announcement Protocol (SAP)). This Session Advertisement is received by the various unicast download mode servers. The unicast download mode server will then redirect subsequent requests for the same content item to the multicast download session.
- Unicast download mode file servers which receive a session announcement for a multicast distribution session may choose to terminate unicast download mode sessions that are in progress for the content item. This SHOULD cause HNEDs to re-request the item, receiving the multicast session advertisement in response.

Redirection to both carousel and scheduled multicast download sessions is supported. However carousel multicast download is more likely to be used as it does not require the HNED to join the session at a specific time.

10.6.3.4.4 Interpretation of redirection information

The redirection information provided in the entity body by a "300" (Multiple Choices) response uses the download session description information as defined in clause 10.5 either in XML or SDP format. This information shall be interpreted as follows:

- *Service-Provider-Domain* and *Download-Session-ID* SHALL be the same as in the original request. If this is not the case the redirection information SHALL be ignored.
- *Download-Session-Version* might be different.
- *Content-Item-Format* information SHALL be the same as in the original request in case it is provided.
- *Download-Session-Mode* SHALL be provided and the indicated download mode shall be used for the redirection.
- *Download-Session-Time-Information* SHALL be provided. In case of unicast download ("UD") the HTTP redirection Retry-After information SHALL be taken into account. In case the time calculated based on the Retry-After information is outside the announced download session time window the HNED shall perform the redirection at the earliest time that fits into the announced download session time window. In case of a multicast download ("CMD" or "SMD") the HNED shall perform the redirection at the earliest time that fits to the announced download session time information.
- In case Reception Reporting information (*Reception-Reporting-Server-URI, Reception-Reporting –Mode, Reception-Reporting -Offset-Time, Reception-Reporting-Random-Time-Period*) is provided, it SHALL be the same as in the original request.
- Unicast or multicast specific information has to be provided for the redirected file download. The *File-Reference* parameter for the redirected file has to be the same as in the original download session description. The HNED SHALL use the original *File-Reference* parameter to identify the file specific parameters in the redirection information. In case of a redirection to a multicast download the HNED shall use the original *File-Reference* parameter to identify the file in the FDT of the FLUTE session (compare against the FDT Content-Location).

The CDS HNED shall update initial download session description with the information received in the redirection information.

10.6.4 Parallel downloads

A CDS HNED may perform parallel download of multiple content items in parallel download sessions. This depends on time of availability of the content items, the requested content items by the user in the pull download mode and the announced content items by the SP in the push download mode. The parallel content item download sessions can have different download session modes.

In case of parallel multicast downloads of multiple content items, the CDS HNED should adapt the multicast rate adaptation as introduced in clause 10.6.2.3 and share the observed bandwidth among the multicast downloads.

In case of any unicast download the HNED may perform parallel download of the files of a single content item. The details of parallel file downloads is outside the scope of the present document.

In case of a multiple server file download the HNED may perform parallel download of file chunks. The details of parallel file chunk downloads is outside the scope of the present document.

10.6.5 Reception Reporting

10.6.5.1 General

The CDS network function SHALL indicate the use of reception reporting by provisioning one or more *Reception-Reporting-Server-URIs* in the download session description.

The *Reception-Reporting-Mode* parameter defines the details of the reporting. This can be:

- content item reporting (Reception-Reporting-Mode=0);
- content item and file reporting (Reception-Reporting-Mode=1); and
- content item, file and chunk reporting (Reception-Reporting-Mode=2).

If the *Reception-Reporting-Mode* parameter is not provided, content item reporting (*Reception-Reporting-Mode=0)* SHALL be used. If chunk reporting is requested for the content item, it shall only be used for multiple servers file downloads. For files that are downloaded from a single server, *Reception-Reporting-Mode=1* (file reporting) shall be used.

The CDS HNED SHALL determine if the item(s) for which reporting is requested (e.g. content item, file, chunk) are successfully downloaded as defined in the download specifications above and send the reception reporting reports to a reception reporting server. The reception reporting server SHALL be chosen randomly from the list of reception reporting servers provided in the download session description (*Reception-Reporting-Server-URI* parameter). In case of a multicast download session the request shall be delayed by the back-off time as defined in clause 10.6.5.2. The reception reporting server shall respond as defined in clause 10.6.5.4.

10.6.5.2 Distribution of Reception reporting request over time

To resolve the problem of feedback implosion in case of multicast download, every request messages to a reception reporting server is delayed in adopting the same strategy used in TS 102 472 [66], clause 7.3.4. The offset time and random time period parameter are provided by the download session description (*Reception-Reporting-Offset-Time* and *Reception-Reporting-Random-Time-Period* parameter).

NOTE: The "Offset-Time" and "Random-Time-Period" used for delivery confirmation in multicast download mode may have different values from those used for file repair.

This back-off timing mechanism for the reception reporting is not used in unicast download mode. In this case the CDS HNED function SHALL initiate the reception reporting process immediately after the verification of download completion.

10.6.5.3 Reception reporting message

The HNED SHALL send a Reception Report request using the HTTP 1.1 POST request RFC 2616 [39] carrying XML formatted reception reporting message.

Table 28 describes the parameters of the Reception Reporting message. The corresponding XML schema is provided in clause C.2.4.

For the successful download of a content item a content item reception report message shall be sent which includes information about the content item and all files of the content item as provided by the download session description. For each file it is indicated whether the download was performed or not performed in case the latest version of the file as identified by the file length and digest was already available at the HNED.

For the successful download of a file, a file reception report message shall be sent. In case the file download was not performed as the latest version of the file as identified by the file length and digest was already available at the HNED no reception reporting message shall be send.

For the successful download of a file chunk a chunk reception report message shall be sent.

Table 28: Reception Reporting message

Parameter	Description	Type	Usage
Reporting type	Content item, file or chunk report.	inherent	all messages
Client-ID	Identification of the client.	string	all messages
Push-Action	Indicates that the download was initiated by a PushDownloadType (see clauses 10.3.1 and G.1.2).	boolean	all messages
CRID	Content Reference Identifier as provided by the BCG.	URI	all messages
Content-Version	Content Version number (see clause 10.6.6).	unsigned integer (8)	all messages
Download-Session-Parameters			
Service-Provider-Domain	Service provider domain (see clause 10.5.3).	domain name	all messages
Download-Session-ID	Identification of the download session (see clause 10.5.3).	unsigned integer	all messages
Download-Session-Version	Version of the download session (see clause 10.5.3).	unsigned integer	all messages
Files (one per file)			
File-Reference	Relative reference of the file (see clause 10.5.2).	<path-absolute>	all messages
Download-Action	Indicates for the file whether the download was performed or not performed in case the latest version of the file as identified by the file length and digest was already available at the HNED.	values: download, skipped	content item messages only
Chunks (one per chunk)			
Byte-Range	Byte range of the chunk of the file (as defined for HTTP range header RFC 2616 [39]).	first byte position (unsigned integer); last byte position (unsigned integer)	Chunk message only

The Mime type of XML reception reporting message SHALL be set to application/xml.

The reporting for several items of the same type (e.g. several chunks in a chunk reception report message, several files in a file reception reporting message) can be aggregated into a single message.

Multiple messages of different types can be aggregated into a single HTTP request using Multipart MIME (multipart/mixed).

The "Client-ID" provides a unique identification of the CDS HNED that sends the reception reporting message. The specific value of the client ID and its provisioning at the HNED is outside the scope of the specification. If not specified otherwise the HNED SHALL use its MAC address as Client-ID.

NOTE: The HNED IP address cannot be assumed a unique identification as the HNED may use a private IP address in case it is located in a home network with network address translation.

10.6.5.4 Reception report response message

The reception reporting server shall respond with a HTTP response with status code "200" (OK) to signal successful reception and processing of a reception report. Other status codes may be used in error cases as defined in RFC 2616 [39]. The HNED SHALL in case of a response with an error status code or in case no response is received resend the reception reporting message to an alternative server if provided.

10.6.6 Content Version Numbering

A content item announced for download might have errors that prevent the correct play out of the content item. In order to provide an updated version of such a content item a Content Version number is provided in the BCG instance description metadata (OnDemandProgramType and PushDownloadType Content Version attribute, see clauses G.1.1 and G1.2) and decomposed binary locator (Extended On-demand decomposed binary locator content_version attribute, see clause G.1.4).

In case of a change to any file of a content item, the CDS network function shall setup a new download session for the content item and announce the new download session in the BCG with a new content version number (using the same CRID). A new multicast download session SHALL use a different TSI.

In case the change occurs while the download session for the old content version is active the CDS network function shall stop this download session. For unicast download sessions the servers SHALL response with a HTTP status code of 410 "Gone" to any download request. For multicast download sessions the CDS network function SHALL terminate the FLUTE delivery the session. CDS HNEDs that actively participate in a download session SHALL terminated their participation in this outdated download session (including file repair, redirection and reception reporting actions) and delete the already downloaded data as soon as they receive an updated BCG announcement with a new content version number. For unicast download the HNED SHALL in addition check for an updated content version number in case it receives a HTTP status code of 410 "Gone" to a file download request. For multicast download the HNED SHALL in addition check for an updated content version number before it starts a file repair in case the file download is incomplete.

The HNED shall join the new download session as announced in order to download the updated content item.

In case the HNED has already successfully downloaded the content item and terminated its participation in the download session, the HNED shall download the updated content item if it receives a BCG PushDownloadType announcement with a new content version. For pull download announcements (via the OnDemandProgramType or Extended On-demand decomposed binary locator), the HNED shall check for a new content version before the play out of the content item. In case a new content version is available it SHALL be downloaded before the play out is started.

The files of the outdated content version shall be deleted and replaced by the updated content version. Files that have not changed (indicated by the same <path-absolute>, file length and MD5 digest) SHOULD be kept from the old version and not be downloaded again.

10.6.7 Priority settings

CDS unicast and multicast download sessions SHALL use the "Best effort data" traffic type with the associated IP DSCP and Ethernet priority marking as defined in clause 11. Thus CDS download traffic is unable to cause congestion except for other CDS download traffic itself or any other traffic using this traffic type.

10.7 CDS HNED Storage Management

The current version of the specification provides a limited set of content and storage management functionality.

If the HNED supports CDS, it shall dedicate a sufficient amount of storage to the CDS. This dedicated storage is referred to CDS HNED Storage.

NOTE 1: A 4 MBit/s MPEG-2 TS stream would for example requires roughly 1,8 GByte of storage per hour.

Before acquiring a new content item the HNED SHALL verify that sufficient space on the CDS HNED storage is available. If storage is not sufficient the HNED SHALL not initiate the download session for the content item.

The CDS network function MAY monitor the CDS HNED Storage by tracking reception reporting of content items of individual HNEDs. However, the detailed usage of reception reporting for Storage Management is outside the scope of the present document.

A downloaded content item may have an associated "*ExpiryTime*" provided in the BCG OnDemandProgramType, PushDownloadType or on-demand decomposed binary locator (see clauses G.1.1, G.1.2 and G.1.4). When this "*ExpiryTime*" is due, the CDS HNED function SHALL automatically delete all the files that are associated with the content item from the CDS HNED storage.

Specifically, the download of new content items shall not be prevented by content items on the CDS HNED Storage with expired "*ExpiryTime*" (e.g. due to insufficient storage space).

NOTE 2: Content item deletion only refers to the removal of the content item from the CDS HNED storage. Any content item that is has been moved outside or is moved outside the CDS HNED storage - even if it was acquired through CDS - is not affected by this deletion process. The permission to move the content item from the CDS HNED storage to a private storage on the HNED or even to a different device is outside the scope of the present document. Hence, the content item deletion process is no secure way of preventing access to the content item after the deletion. Content protection mechanisms shall be used if restricted access to the content shall be provided.

11 Quality of Service

For the network to provide the required Quality of Service (QoS) to the end user there shall be a method for determining the type of data contained in each datagram and a mechanism for prioritizing the traffic based on this classification.

The method of classification will follow the Differentiated Services model described in RFC 2475 [33]. IP packets passing over the IPI-1 interface shall be appropriately marked at the originating source, as described in clause 11.1.

NOTE: It is assumed that other guideline documents will be needed to recommend good practice within both the home and the Service Provider(s) domain.

11.1 DSCP packet marking

The Differentiated Services marking uses the 8-bit Type of Service field in the IP header and is described in RFC 2474 [32]. Networks compliant with RFC 2474 [32] use 6 bits of this ToS field to contain the differentiated services codepoint - a numeric value used within the network to manage queuing policies. Networks not compliant with RFC 2474 [32] use a 3-bit field within the ToS to determine precedence.

Within IP networks designed to carry DVB services, the markings detailed in table 29 shall be used. It is recommended that the full DSCP value be used.

Table 29: DSCP markings

Traffic Type	IP DSCP Value	Corresponding IP Precedence
Voice Bearer (see note 1)	0b110000	0b110
Real-time Video Bearer (high priority) (see note 2)	0b100010	0b100
Real-time Video Bearer (lower priority) (see note 3)	0b100100	0b100
Voice and Video Signalling	0b011010	0b011
Best effort data	0b000000	0b000
NOTE 1: The voice bearer is listed here to ensure that there is no interference with DVB-IPTV services. NOTE 2: Normal marking for real-time video. NOTE 3: Use of this marking is application dependent. It is intended to allow a CSP to suggest that some video packets are less important than others.		

11.2 Ethernet Priority

The interfaces IPI-1, IPI-2 and IPI-3 on an Ethernet MAC based HNS shall support IEEE 802.1Q [5], with defined user priority classes. The IEEE 802.1D [9] field shall be supported in an IEEE 802.1Q [5] compliant Ethernet frame. The marking shall be based on the DiffServ CodePoint (DSCP) marking method [42] as described in clause 11.1.

Table 30: DSCP Values and corresponding Ethernet IEEE 802.1D marking

Traffic type	IP DSCP value	Corresponding IEEE 802.1D User Priority value
Voice bearer (see note)	0b110000	0b110
Video bearer (high priority)	0b100010	0b100
Video bearer (lower priority)	0b100100	0b100
Video signalling	0b011010	0b011
Best effort data	0b000000	0b000
NOTE: The voice bearer is listed here to ensure that there is no interference with DVB-IPTV services.		

For a HNS based on Ethernet MAC these DSCP values are used to map a traffic type onto the corresponding IEEE 802.1D [9] priority codes. Packets shall be marked using the Layer 2 Class of Service (CoS) settings in the User Priority bits of the IEEE 802.1D [9] portion of the 802.1Q header. These can be mapped to the IP Precedence/DSCP bits in the Type of Service (ToS) byte of the IPv4 header. Note that the 802.1Q header adds an additional 4 bytes of data into an Ethernet frame header. The IEEE 802.1D [9] priority field is one of the fields in the 802.1Q header, and is a 3 bit field. Any switching device that implements the IEEE 802.1Q [5] specification can use the user-priority field to determine the scheduling class a packet belongs to.

Note that mapping the IP precedence field is easy, as it can be copied to the user-priority field directly, as both the fields are 3 bits long. To map the DSCP field to the user-priority field, the DSCP shall be shifted right by 3 bits, i.e. the user-priority field is the first 3 bits of the DSCP field. To map the user-priority field to the DSCP field, the user-priority field shall be tested for values that match the user-priority value in Column 3. If the user-priority value does not match any of the values shown in column 3, the packet shall be marked with a DSCP value which is the user-priority shifted left by 3 bits.

Annex A (informative): MPEG2 Timing Reconstruction

This annex describes one way in which RTP timestamps can be used to reconstruct an MTS that is encapsulated in RTP packets using RFC 2250 [29] and transported over a jitter-inducing network e.g. IP or Ethernet. This description is for information only and is not a normative part of the present document.

The Transport Stream System Target Decoder (T-STD) is defined fully in ISO/IEC 13818-1 [52]. It is a conceptual decoder model used to define terms precisely and to model the decoding process. The input to the T-STD is a MTS. A MTS may contain multiple MPEG programs with independent time bases. However, the T-STD decodes only one program at a time.

Data from the MTS enters the T-STD at a piecewise constant rate. The *i*th byte enters at time *t(i)*. The time at which this byte enters the T-STD can be recovered from the input stream by decoding the input PCR fields, encoded in the MTS packet adaptation field of the program to be decoded and, by counting the bytes in the complete MTS between the successive PCRs of the program to be decoded. The value encoded in the PCR field indicates the time *t(i)*, where *i* refers to the byte containing the last bit of the PCR.

For all other bytes the input arrival time *t(i)* is computed from *PCR(i")* and the transport rate at which the MTS arrives. The transport rate is determined as the number of bytes in the MTS between the bytes containing the last bit of two successive PCR fields of the same program plus one, divided by the difference between the time values encoded in these same two PCR fields (see also figure A.1):

$$t(i) = \frac{PCR(k-1)}{27\ MHz} + \frac{i - i''}{R(i)} \qquad \text{(A.1)}$$

Where:

i	is the index of any byte in the MTS for *i'' < i < i'*.
i''	is the index of the byte containing the last bit of the most recent PCR field applicable to the program being decoded.
PCR(k-1)	is the time encoded in the PCR field in units of the 27 MHz system clock.
R(i)	is the transport rate which is calculated as follows:

$$R(i) = \frac{(i'-i'') \times 27\ MHz}{PCR(k) - PCR(k-1)} \quad \text{(A.2)}$$

Where:

i'	is the index of the byte containing the last bit of the immediately following PCR applicable to the program being decoded.
and	$i'' < i \leq i'$.

Note that equation A.2 assumes that the transport rate between two successive PCRs is constant, but that the transport rate may change at any PCR. Note furthermore that the transport rate for multi-program transport streams is typically constant, but that the transport rate of a single-program transport stream may vary within the piece-wise constant rate concept defined by equation A.2. (See also ISO/IEC 13818-1 [52]).

A tolerance is specified for the PCR values. The PCR tolerance is defined as the maximum inaccuracy allowed in received PCRs. This inaccuracy may be due to imprecision in the PCR values or to PCR modification during remultiplexing. Note that it does not include errors in packet arrival time due to network jitter or other causes. The PCR tolerance is ±500 ns. In the T-STD model, the inaccuracy will be reflected as an inaccuracy in the calculated transport rate *R(i)* of equation A.2.

A.1 Clock recovery in a RTP receiver

It is assumed that a jitter-smoothing network adapter is inserted between a network's output and an MPEG-2 decoder. The network adapter exploits the RTP timestamps to achieve jitter smoothing. The MPEG-2 decoder is assumed to conform to the real-time MPEG-2 interface specification [53]. This interface requires an MPEG-2 decoder with more jitter tolerance than the idealized decoder of the STD. The network adapter processes the incoming jittered bit stream and outputs a system stream whose actual byte delivery schedule conforms to the real-time specification.

Note that for immediate decoding the network adapter approach may not be necessary or cost effective. Instead a single stage of clock recovery can be used.

According to RFC 2250 [29], each RTP packet contains a timestamp derived from the sender's 90 KHz clock reference. This timestamp is the *target transmission time* of the first byte of the RTP payload i.e. the "ideal" time that the packet should be fed into the IP network. It is assumed that the time between the last byte put in the RTP packet and the time value inserted as the RTP timestamp into the packet is constant. In this way the RTP timestamp is the time of the last byte that entered the RTP packet plus some constant delay. Note that the boundary of the IP network may still be somewhat vague and this may affect the jitter process i.e. the transmitter can also add some (scheduling and processing) jitter to the packet before it appears on the (IP) network. However, the receiver should be able to handle this additional jitter adequately.

In this regard, the difference between the (RTP) *target transmission time* and the (MPEG) *target delivery time* is a time constant plus the (constant) delay imposed on the delivery of the MTS to the RTP receiver. Both can be ignored, because they are constant, and hence for the RTP receiver the target transmission time is functionally equivalent to the target delivery time.

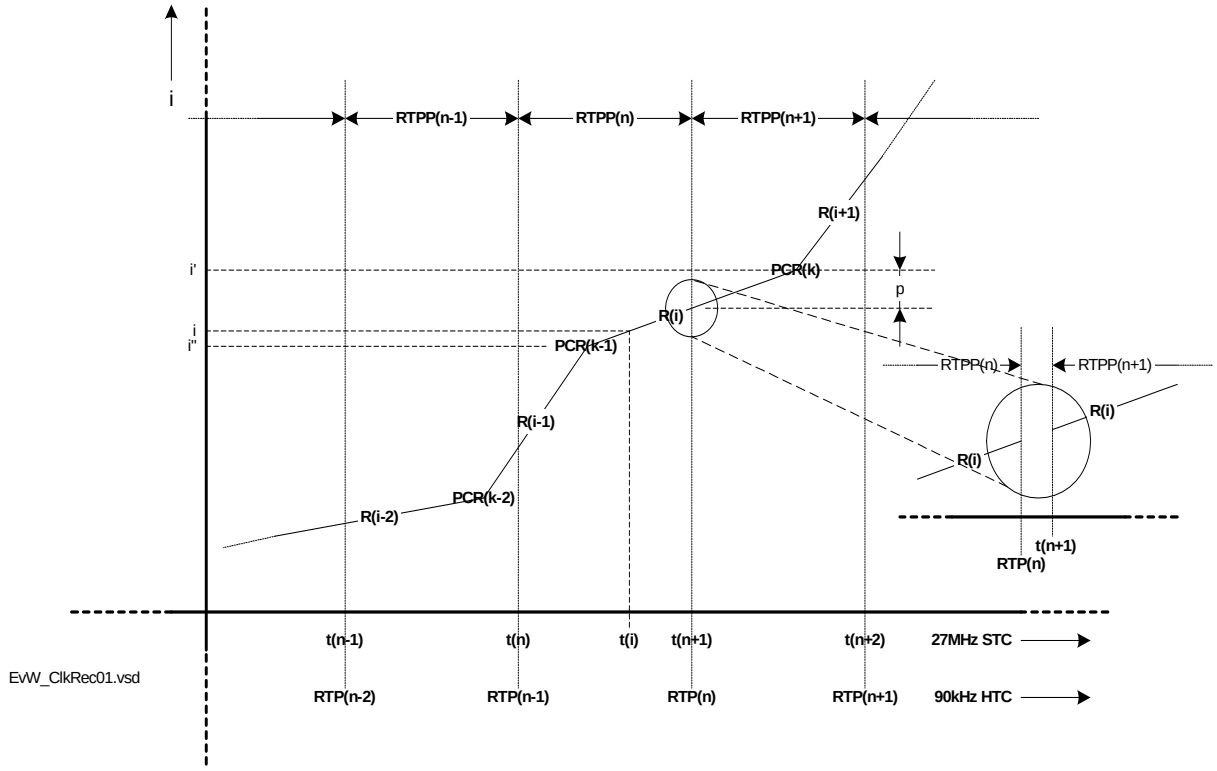


Figure A.1: Timing

In terms of the MPEG-2 system time clock, the first byte of the payload of *RTP Packet* (n+1), referred to as RTPP(n+1) in figure A.1, enters the T-STD at time *t(n+1)*. Time *t(n+1)* can be recovered as follows:

$$t(n+1) = \frac{PCR(k)}{27\ MHz} - \frac{p}{R(i)} \qquad \text{(A.3)}$$

where:

n+1	is the index of the RTP packet i.e. the value n+1 in RTPP(n+1).
k	is the index of the first PCR in RTPP(n+1).
p	is the number of bytes preceding the byte that contains the last bit of PCR(k).
PCR(k)	is the time encoded in the first PCR of the MPEG program that is selected as reference to reconstruct the MTS.
R(i)	is the transport rate of the transport stream between PCR(k-1) and PCR(k) of the MPEG program that is selected as reference to reconstruct the MTS, as calculated by equation A.2.

The target transmission time *RTP(n)* plus a constant delay, expressed in units of the 90 kHz Head-end Time Clock (HTC) of the sender corresponds to time value t(n+1) of the first byte of RTPP(n+1). Time value t(n+1) is expressed in units of the 27 MHz MPEG-2 STC. In many, if not all cases, it is reasonable to assume that the drift between the HTC and the STC can be ignored for the duration of the transport stream contained in one RTP packet and between two consecutive RTP packets.

Therefore, if desired, it is also possible to map the value of any contained PCR to a 90 kHz value of the sender, as follows:

$$PCR(k) \cong RTP(n) + 90\ kHz \times \frac{(p+1)}{R(i)} \qquad \text{(A.4)}$$

The mapping information between the STC and the 90 kHz clock of the sender can be used to reconstruct the MPEG-2 transport stream at the receiver.

Note that there is an uncertainty of about 11 μs (1/90 kHz), due to the 90 kHz resolution of the RTP time stamps. This is perceived by the receiver as delivery jitter and conforms to the MPEG-2 real-time interface specification [53]. A well-constructed 27 MHz STC PLL should be able to remove this jitter.

Note that the RTP timestamps can be derived from an arbitrary 90 kHz HTC, which may be, but is not required to be, locked to the STC of one of the programs in the MTS.

A.2 Recommendation

To use this two-stage MTS reconstruction method based on RTP timestamps, it is recommended that the time between putting the last byte in the RTP packet and inserting the RTP timestamp value into the RTP packet is constant.

Annex B (informative): SD&S data model

Figure B.1 provides a graphic representation of the DVB-IPTV service discovery model.

The boxes in bold are the components required to establish the list of DVB-IPTV services available from different SPs.

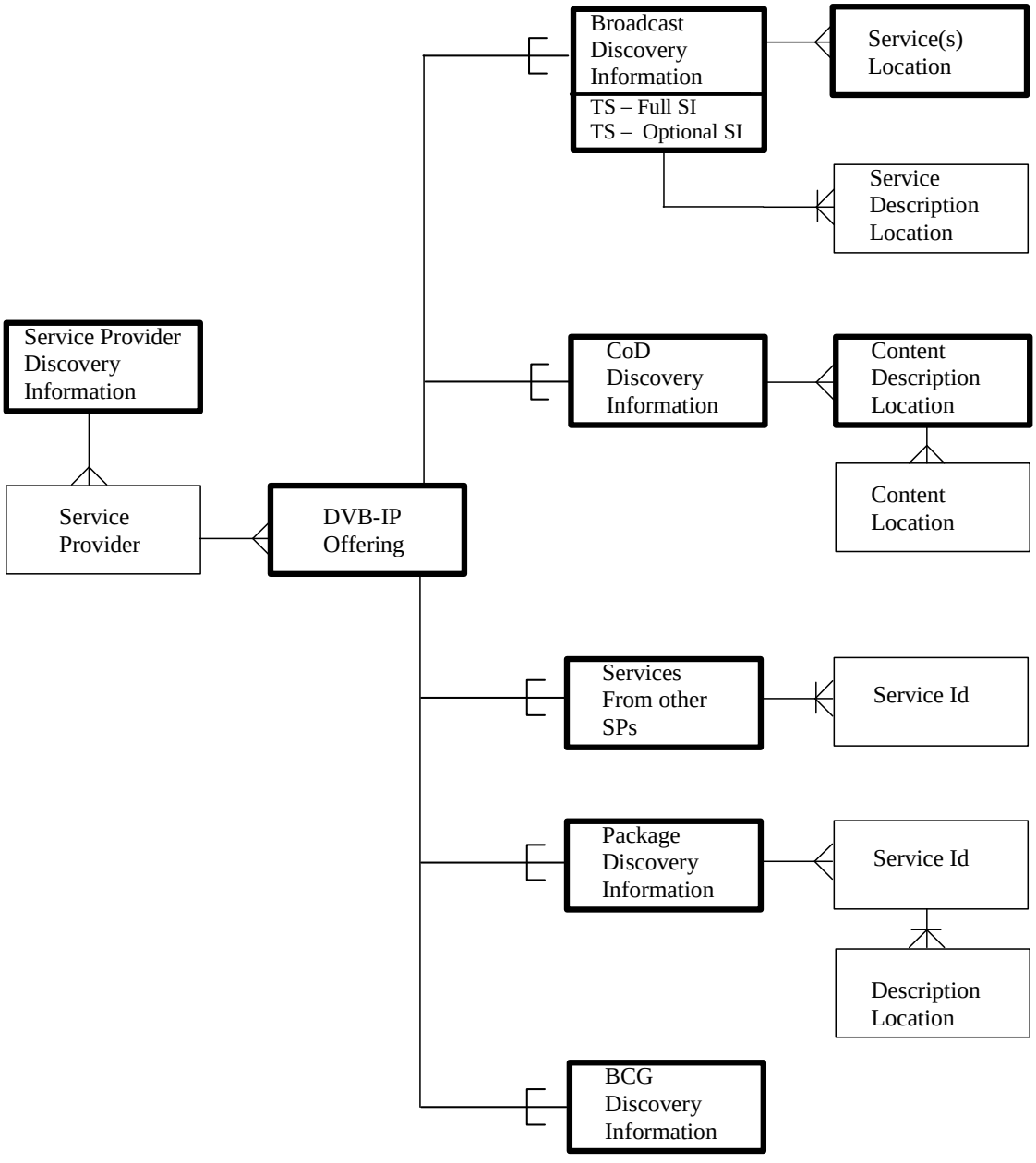


Figure B.1: Proposed data model for DVB-IPTV service discovery information

The SP Discovery Information enables the discovery of SPs offering DVB-IPTV services. SPs publish their offer via the service discovery information. A SP offer is made of services of type broadcast or CoD.

The "TS - Full SI broadcast discovery information" component is used when full DVB SI is available in-band.

The "TS - Optional SI broadcast discovery information" component is used when complete service description is not available in-band.

The "CoD discovery information" is used for SPs that would like to describe their CoD offer.

The model allows SPs to reference individual services or a complete offering from another SP which it has a commercial agreement with.

The "Package discovery information" is used by SPs that would like to group several services and present them as a single entity. The package information does not enable the discovery of new services; the package discovery information references services which have to be discovered via the two other components in the model called Broadcast and CoD Discovery Information. Additional information on services can optionally be provided in the context of a package.

Using the data model above, the HNED first builds the list of DVB-IPTV SPs operating on the network, then in a second stage the list of DVB-IPTV services is established by acquiring the service discovery information for each SP.

The model allows the entry point to the service discovery and selection mechanism to be a specific SP, in this case the information relating to the SP and the list of services for this SP may be acquired from the same location.

This model might be easily extended by adding new types of discovery information if new types of SP offers are identified.

Annex C (normative): Schemas

C.1 SD&S XML schemas

The following clauses define the various types and elements that are used in the SD&S XML schema. The full normative XML schema is available as the file sdns handbook v1.4 r10.xsd in archive ts_102034v010401p0.zip which accompanies the present document.

C.1.1 Namespace

The namespace of the service discovery schema is urn:dvb:metadata:iptv:sdns:2008-1.

C.1.2 Simple types

C.1.2.1 DescriptionLocation

```
<xsd:simpleType name="DescriptionLocation">
    <xsd:restriction base="xsd:anyURI"/>
</xsd:simpleType>
```

A URI that specifies the location of further information.

C.1.2.2 DomainType

```
<xsd:simpleType name="DomainType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="((.|\n|\r)*)?(\.(.|\n|\r)*)+"/>
    </xsd:restriction>
</xsd:simpleType>
```

This type describes a "domain name" type. It is recommended that domains names comply with the "preferred name syntax" of clause 3.5, RFC 1034 [13].

C.1.2.3 Genre

```
<xsd:simpleType name="Genre">
    <xsd:restriction base="xsd:byte">
        <xsd:minInclusive value="0"/>
        <xsd:maxInclusive value="15"/>
    </xsd:restriction>
</xsd:simpleType>
```

This type describes the content genre, which is encoded as a number in the range 0 to 15, as detailed in the content_nibble_level_1 field of the content_descriptor, as in table 26 in EN 300 468 [1].

C.1.2.4 Hexadecimal3bit

```
<xsd:simpleType name="Hexadecimal3bit">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0-7]"/>
    </xsd:restriction>
</xsd:simpleType>
```

A 3 bit number represented as a single hexadecimal digit.

C.1.2.5 Hexadecimal4bit

```
<xsd:simpleType name="Hexadecimal4bit">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0-9a-fA-F]"/>
    </xsd:restriction>
</xsd:simpleType>
```

A 4 bit number represented as a single hexadecimal digit.

C.1.2.6 Hexadecimal8bit

```
<xsd:simpleType name="Hexadecimal8bit">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0-9a-fA-F]{1,2}"/>
    </xsd:restriction>
</xsd:simpleType>
```

An 8 bit number, represented as one or two hexadecimal digits.

C.1.2.7 Hexadecimal16bit

```
<xsd:simpleType name="Hexadecimal16bit">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0-9a-fA-F]{1,4}"/>
    </xsd:restriction>
</xsd:simpleType>
```

A 16 bit number represented as between one and four hexadecimal digits.

C.1.2.8 Hexadecimal32bit

```
<xsd:simpleType name="Hexadecimal32bit">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0-9a-fA-F]{8}"/>
    </xsd:restriction>
</xsd:simpleType>
```

A 32-bit number represented as 8 hexadecimal digits.

C.1.2.9 Integer6bit

```
<xsd:simpleType name="Integer6bit">
    <xsd:restriction base="xsd:unsignedShort">
        <xsd:minInclusive value="0"/>
        <xsd:maxInclusive value="63"/>
    </xsd:restriction>
</xsd:simpleType>
```

A 6 bit decimal number in the range 0 to 63.

C.1.2.10 IPorDomainType

```
<xsd:simpleType name="IPOrDomainType">
    <xsd:union memberTypes="dvb:IPType dvb:DomainType"/>
</xsd:simpleType>
```

Either an IP address (see IPType), or a domain name (see DomainType).

C.1.2.11 IPType

```
<xsd:simpleType name="IPType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern
value="((([1-9]?[0-9])|(1[0-9][0-9])|(2[0-4][0-9])|(25[0-5]))\.){3}(([1-9]?[0-9])|(1[0-9][0-9])|(2[0
-4][0-9])|(25[0-5]))"/>
    </xsd:restriction>
</xsd:simpleType>
```

An IPv4 dotted address of the form a.b.c.d. All four components are mandatory and in decimal.

C.1.2.12 ISO-3166-List

```
<xsd:simpleType name="ISO-3166-List">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="\c\c\c(,\c\c\c)*"/>
    </xsd:restriction>
</xsd:simpleType>
```

A comma separated list of one or more country codes as defined in ISO 3166 [50].

C.1.2.13 ISO 639-2

```
<xsd:simpleType name="ISO639-2">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="\c\c\c"/>
    </xsd:restriction>
</xsd:simpleType>
```

A three letter language code, as defined in ISO 639-2 [51].

C.1.2.14 OrigNetId

```
<xsd:simpleType name="OrigNetId">
    <xsd:restriction base="xsd:unsignedShort"/>
</xsd:simpleType>
```

The original_network_id, as defined in TS 101 162 [2], which also specifies the management of this number space. This value shall be in decimal.

C.1.2.15 PrimarySISource

```
<xsd:simpleType name="PrimarySISource">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Stream"/>
        <xsd:enumeration value="XML"/>
    </xsd:restriction>
</xsd:simpleType>
```

This type is used to indicate if the specified SI is the primary (with the value "XML") or in the stream (with the value "Stream").

C.1.2.16 PullURL

```
<xsd:simpleType name="PullURL">
    <xsd:restriction base="xsd:anyURI">
        <xsd:pattern value=".*/dvb/sdns/.*"/>
    </xsd:restriction>
</xsd:simpleType>
```

This is used to specify the location from which information can be pulled.

C.1.2.17 RTSP

```
<xsd:simpleType name="RTSP">
    <xsd:restriction base="xsd:anyURI">
        <xsd:pattern value="rtsp://.*"/>
    </xsd:restriction>
</xsd:simpleType>
```

This type is describes an RTSP URL.

C.1.2.18 Service

```
<xsd:simpleType name="Service">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="(.|\n|\r)+"/>
    </xsd:restriction>
</xsd:simpleType>
```

This is the name of a service, as specified in TS 101 812 [3], clause 14.9. It is recommended that this follows the rules for an internet DNS name as specified in RFC 1035 [14] and subsequent updates.

C.1.2.19 ServiceID

```
<xsd:simpleType name="ServiceId">
    <xsd:restriction base="xsd:unsignedShort"/>
</xsd:simpleType>
```

The service_id, as defined in EN 300 468 [1]. This value shall be in decimal.

C.1.2.20 ServiceType

```
<xsd:simpleType name="ServiceType">
    <xsd:restriction base="dvb:Hexadecimal8bit"/>
</xsd:simpleType>
```

An eight bit hexadecimal value (see Hexadecimal8bit) encoding the "type" of a service. The values and meanings are defined in EN 300 468 [1], table entitled "service type coding".

C.1.2.21 StreamingType

```
<xsd:simpleType name="StreamingType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="rtp"/>
        <xsd:enumeration value="udp"/>
    </xsd:restriction>
</xsd:simpleType>
```

This type is used to indicate if RTP (with the value "rtp") or direct UDP (with the value "udp") streaming is used.

C.1.2.22 TransportProtocolType

```
<xsd:simpleType name="TransportProtocolType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="RTP-AVP"/>
        <xsd:enumeration value="UDP-FEC"/>
    </xsd:restriction>
</xsd:simpleType>
```

A string that may be used to signal the transport and FEC type used for delivery.

C.1.2.23 TSId

```
<xsd:simpleType name="TSId">
    <xsd:restriction base="xsd:unsignedShort"/>
</xsd:simpleType>
```

The transport_stream_id as defined in EN 300 468 [1]. This value shall be in decimal.

C.1.2.24 Version

```
<xsd:simpleType name="Version">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0-9a-fA-F]{2}"/>
    </xsd:restriction>
</xsd:simpleType>
```

A number conveying the version of a table or record. This value will increase with changes to the table or record, modulo 256. This value shall be in hexadecimal.

C.1.3 Complex types and attribute groups

C.1.3.1 AnnouncementSupport

```
<xsd:complexType name="AnnouncementSupport">
    <xsd:sequence>
        <xsd:element name="Announcement" maxOccurs="unbounded">
            <xsd:complexType>
                <xsd:choice minOccurs="0">
                    <xsd:element name="TextualIdentifier" type="dvb:TextualIdentifier"/>
                    <xsd:element name="DVBTriplet" type="dvb:DVBTriplet"/>
                </xsd:choice>
                <xsd:attribute name="Type" type="dvb:Hexadecimal4bit" use="required"/>
                <xsd:attribute name="ReferenceType" type="dvb:Hexadecimal3bit" use="required"/>
                <xsd:attribute name="ComponentTag" type="dvb:Hexadecimal8bit" use="optional"/>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="SupportIndicator" type="dvb:Hexadecimal16bit" use="required"/>
</xsd:complexType>
```

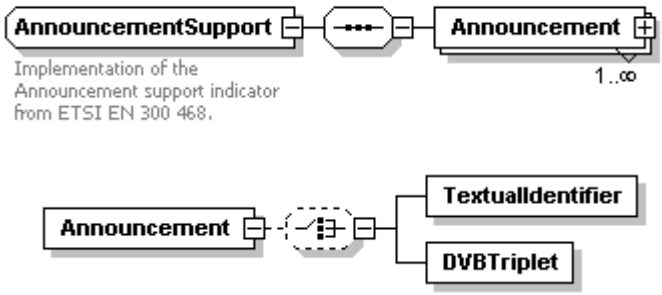


Figure C.1: AnnouncementSupport

This is an XML representation of the Announcement Support Indicator in EN 300 468 [1]. The meanings and values of attributes and elements are defined in EN 300 468 [1].

C.1.3.2 BasicMulticastAddressAttributesType

```
<xsd:attributeGroup name="BasicMulticastAddressAttributesType">
    <xsd:attribute name="Source" type="dvb:IPOrDomainType" use="optional"/>
    <xsd:attribute name="Address" type="dvb:IPOrDomainType" use="required"/>
    <xsd:attribute name="Port" type="xsd:unsignedShort" use="required"/>
</xsd:attributeGroup>
```

This common group of attributes is used to carry the basic multicast address, without any information on FEC or RET channels.

C.1.3.3 CDSDownloadSessionDescriptionLocationType

```
<xsd:complexType name="CDSDownloadSessionDescriptionLocationType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="DVBSTP">
            <xsd:complexType>
                <xsd:attributeGroup ref="dvb:BasicMulticastAddressAttributesType"/>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="SAP">
            <xsd:complexType>
                <xsd:attributeGroup ref="dvb:BasicMulticastAddressAttributesType"/>
            </xsd:complexType>
        </xsd:element>
    </xsd:choice>
</xsd:complexType>
```

Figure C.2: CDSDownloadSessionDescriptionLocationType

This type is used to carry both the multicast address of the CDS Download Session, and indicate if this is carried as SAP (in which case element SAP is used) or SD&S XML (in which case element DVBSTP is used) .

C.1.3.4 Cell

```
<xsd:complexType name="Cell">
    <xsd:sequence>
        <xsd:element name="CountryCode" type="xsd:string "/>
        <xsd:element name="CA" type="dvb:CivicAddress" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string" use="required"/>
</xsd:complexType>
```

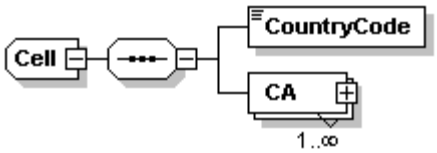


Figure C.3: Cell

C.1.3.5 Civic Address

```
<xsd:complexType name="CivicAddress">
    <xsd:sequence>
        <xsd:element name="CA" type="dvb:CivicAddress" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Type" type="xsd:string" use="required"/>
    <xsd:attribute name="Value" type="xsd:string" use="required"/>
</xsd:complexType>
```

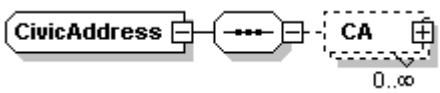


Figure C.4: CivicAddress

C.1.3.6 CommonCastRETType

```
<xsd:attributeGroup name="CommonCastRETType">
    <xsd:attribute name="ssrc" type="xsd:unsignedInt" use="optional"/>
    <xsd:attribute name="RTPPayloadTypeNumber" type="xsd:unsignedInt" use="optional"/>
    <xsd:attribute name="dvb-original-copy-ret" type="xsd:boolean" use="optional"/>
    <xsd:attribute name="rtcp-mux" type="xsd:boolean" use="optional" default="false"/>
    <xsd:attribute name="DestinationPort" type="xsd:unsignedInt" use="optional"/>
    <xsd:attribute name="rtx-time" type="xsd:unsignedInt" use="required"/>
</xsd:attributeGroup>
```

This attribute group is a collection of attributes common across both unicast and multicast RET.

C.1.3.7 CountryAvailabilty

```
<xsd:complexType name="CountryAvailability">
    <xsd:attribute name="Countries" type="dvb:ISO-3166-List" use="required"/>
    <xsd:attribute name="Available" type="xsd:boolean" default="true"/>
</xsd:complexType>
```

This is an XML representation of the Country availability descriptor in EN 300 468 [1]. The meanings and values of attributes and elements are defined in EN 300 468 [1].

C.1.3.8 DescriptionLocationBCG

```
<xsd:complexType name="DescriptionLocationBCG" mixed="true">
    <xsd:simpleContent>
        <xsd:extension base="dvb:DescriptionLocation">
            <xsd:attribute name="preferred" type="xsd:boolean" use="optional"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

A URI that specifies the location of BCG information with an optional boolean attribute for signaling the preferred BCG. There shall be no more than one instance of preferred set to true in each relevant scope.

C.1.3.9 DVBSTPTransportModeType

```
<xsd:complexType name="DVBSTPTransportModeType">
    <xsd:complexContent>
        <xsd:extension base="dvb:PayloadList">
            <xsd:attributeGroup ref="dvb:MulticastAddressAttributes"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

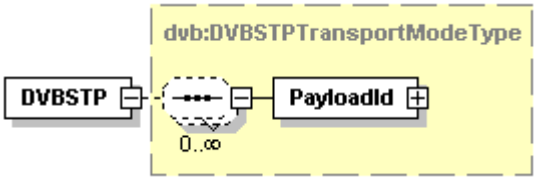


Figure C.5: DVBSTPTransportModeType

C.1.3.10 DVBTriplet

```
<xsd:complexType name="DVBTriplet">
    <xsd:attribute name="OrigNetId" type="dvb:OrigNetId" use="required"/>
    <xsd:attribute name="TSId" type="dvb:TSId" use="required"/>
    <xsd:attribute name="ServiceId" type="dvb:ServiceId" use="required"/>
</xsd:complexType>
```

This is a representation of the identifier for a service in a classic DVB system.

C.1.3.11 FECAttributeGroupType

```
<xsd:attributeGroup name="FECAttributeGroupType">
    <xsd:attribute name="FECMaxBlockSize" type="xsd:unsignedShort" use="optional"/>
    <xsd:attribute name="FECMaxBlockTime" type="xsd:unsignedShort" use="optional"/>
    <xsd:attribute name="FECOTI" type="xsd:base64Binary" use="optional"/>
</xsd:attributeGroup>
```

This attribute group represents the common FEC information used in enhancement layers.

C.1.3.12 FECInfoType

```
<xsd:complexType name="FECInfoType">
    <xsd:sequence>
        <xsd:element name="FECBaseLayer" type="dvb:FECLayerAddressType"/>
        <xsd:element name="FECEnhancementLayer" type="dvb:FECLayerAddressType" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="dvb:FECAttributeGroupType"/>
</xsd:complexType>
```

C.1.3.13 FECLayerAddressType

```
<xsd:complexType name="FECLayerAddressType">
    <xsd:attribute name="Address" type="dvb:IPOrDomainType" use="optional"/>
    <xsd:attribute name="Source" type="dvb:IPOrDomainType" use="optional"/>
    <xsd:attribute name="Port" type="xsd:unsignedShort" use=" optional"/>

    <xsd:attribute name="MaxBitrate" type="xsd:positiveInteger" use="optional"/>
    <xsd:attribute name="RTSPControlURL" type="xsd:anyURI" use="optional"/>
    <xsd:attribute name="PayloadTypeNumber" type="xsd:unsignedInt" use="optional"/>
    <xsd:attribute name="TransportProtocol" type="dvb:TransportProtocolType" use="optional"/>
</xsd:complexType>
```

The Port attribute is optional as this type may be used in multiple places, however it shall be present in some cases. Specifically, the type is only optional when it is used in conjunction with CoDAnnounceDescribe and an RTSP based URL. In this case the relevant information port is obtained via the SETUP response message.

Where this type is used in the context of the BaseLayer, the PayloadTypeNumber shall have a default value of 96 (i.e. if there is no PayloadTypeNumber in a BaseLayer, the value 96 is inferred) and the TransportProtocol shall not be present.

Where this type is used in the context of the EnhancementLayer, the TransportProtocol shall have a default value of UDP/FEC (i.e. if there is no TransportProtocol in an EnhancementLayer, the value UDP/FEC is inferred).

C.1.3.14 FUSAnnouncementType

```
<xsd:complexType name="FUSAnnouncementType">
    <xsd:sequence>
        <xsd:element name="FUSAnnouncementDescription" type="xsd:string" minOccurs="0"/>
        <xsd:element name="MulticastAnnouncementAddress" type="dvb:RMSFUSMulticastAddressType"
minOccurs="0"/>
        <xsd:element name="FUSUnicastAnnouncement" type="xsd:anyURI" minOccurs="0"/>
        <xsd:element name="QRCLocation" type="xsd:anyURI" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

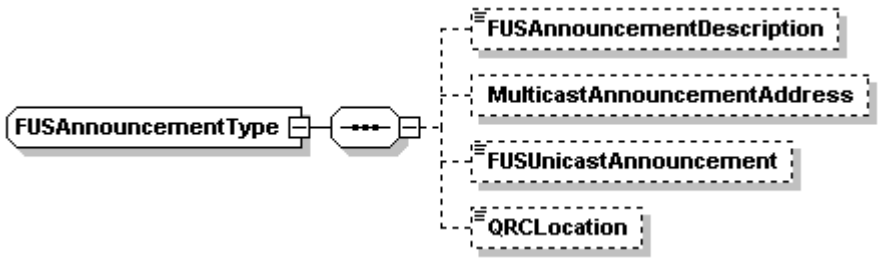


Figure C.6: FUSAnnouncementType

C.1.3.15 FUSType

```
<xsd:complexType name="FUSType">
    <xsd:sequence>
        <xsd:element name="FUSName" type="dvb:MultilingualType" maxOccurs="unbounded"/>
        <xsd:element name="FUSID" type="xsd:decimal"/>
        <xsd:element name="Description" type="dvb:MultilingualType" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="FUSAnnouncement" type="dvb:FUSAnnouncementType" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="LogoURI" type="xsd:anyURI" use="required"/>
</xsd:complexType>
```

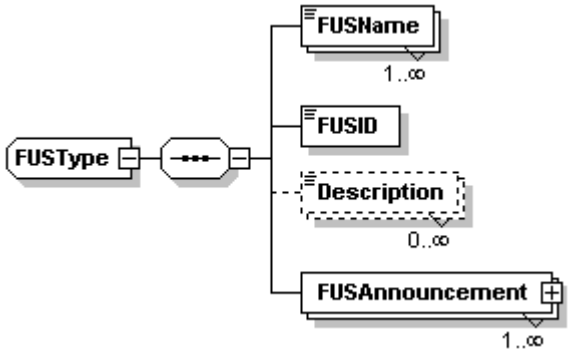


Figure C.7: FUSType

C.1.3.16 HTTPTransportModeType

```
<xsd:complexType name="HTTPTransportModeType">
    <xsd:complexContent>
        <xsd:extension base="dvb:PayloadList">
            <xsd:attribute name="Location" type="dvb:PullURL" use="required"/>
            <xsd:attribute name="SOAP" type="xsd:boolean" default="false"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

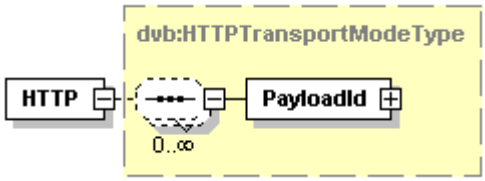


Figure C.8: HTTPTransportModeType

C.1.3.17 IPService

```
<xsd:complexType name="IPService">
    <xsd:sequence>
        <xsd:element name="ServiceLocation" type="dvb:ServiceLocation"/>
        <xsd:element name="TextualIdentifier" type="dvb:TextualIdentifier"/>
        <xsd:element name="DVBTriplet" type="dvb:DVBTriplet"/>
        <xsd:element name="MaxBitrate" type="xsd:positiveInteger" minOccurs="0"/>
        <xsd:element name="SI" type="dvb:SI" minOccurs="0"/>
        <xsd:element name="AudioAttributes" type="tva:AudioAttributesType" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="VideoAttributes" type="tva:VideoAttributesType" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="ServiceAvailability" type="dvb:ServiceAvailabilityType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

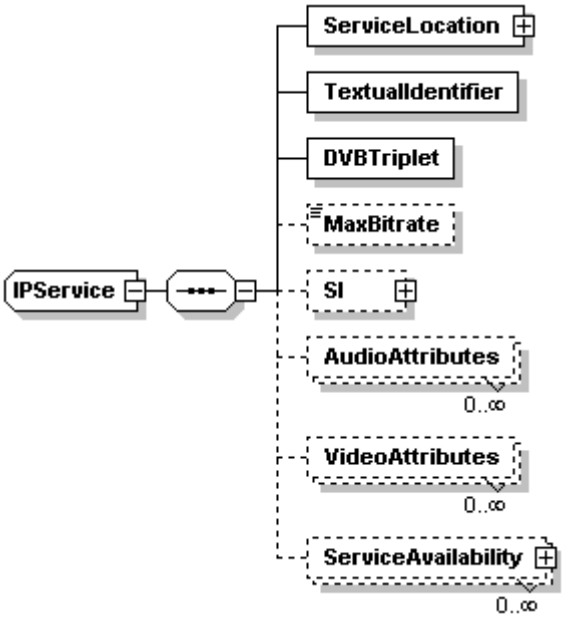


Figure C.9: IPService

This provides information on a single IP service, giving the location(s) at which it may be found, and the identifiers by which it is referred to. Optionally it may also include SI information about the service. The MaxBitrate field describes the peak bitrate at which the service will operate.

Table C.1: IP Service Fields

Name	Definition
`TextualIdentifier`	The Textual identifier by which the service is known. If the domain name is missing, it is taken from the context.
`DVBTriplet`	The DVB Triplet by which the service is known. This will match the service details inside the transport stream.
`ServiceLocation`	The locations at which the service can be found.
`MaxBitrate`	The peak bitrate (in kbits/s) at which the transport stream carrying the service will operate.
`SI`	Service information about the service carried.
`VideoAttributes`	Each instance of this value specifies a way of coding the video that may be used at some point during the operation of the service. If this element is missing, then the default value of MPEG-2 coded video, operating at MP@ML at a frame rate of 25 Hz shall be used; specifically this shall be the legacy value from TS 101 154 [58]. The format of this type is defined in clause 6.3.5 of TS 102 822-3-1 [60]. The values carried in the href attribute of the Coding element shall be defined by the classification specified in the file VideoCodecCS.xml (and by reference MPEG7 VisualCodingFormatCS.xml) included in ts_102034v010301p0.zip, or, preferably, as defined by TS 102 323 [59].
`AudioAttributes`	Each instance of this value specifies a way of coding the audio that may be used at some point during the operation of the service. If this element is missing, then the default value of MPEG-1 or MPEG-2 layer 2 backwards compatible, mono or stereo shall be used; specifically this shall be the legacy value from TS 101 154 [58]. The format of this type is defined in clause 6.3.5 of TS 102 822-3-1 [60]. The values carried in the href attribute of the Coding element shall be defined by the classification specified in the file AudioCS.xml (and by reference MPEG7 AudioCS.xml) included in ts_102034v010301p0.zip, or, preferably. as defined by TS 102 323 [59].
`ServiceAvailabilty`	Defines the geographical regions in which this service is available or not available.

C.1.3.18 IPServiceList

```
<xsd:complexType name="IPServiceList">
    <xsd:sequence>
        <xsd:element name="ServicesDescriptionLocation" type="dvb:DescriptionLocationBCG"
minOccurs="0" maxOccurs="unbounded"/>
        <xsd:sequence>
            <xsd:element name="SingleService" type="dvb:IPService" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:sequence>
</xsd:complexType>
```

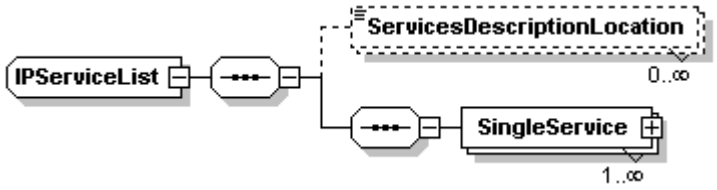


Figure C.10: IPServiceList

This type represents a list of IP services that are grouped together with a single, optional DescriptionLocation.

C.1.3.19 McastType

```
<xsd:complexType name="McastType">
    <xsd:attributeGroup ref="dvb:MulticastAddressAttributes"/>
    <xsd:sequence minOccurs="0">
        <xsd:element name="FECBaseLayer" type="dvb:FECLayerAddressType" maxOccurs="1" />
        <xsd:element name="FECEnhancementLayer" type="dvb:FECLayerAddressType" minOccurs="0"
maxOccurs="unbounded" />
        <xsd:element name="CNAME" type="xsd:string" minOccurs="0"/>
        <xsd:element name="ssrc" type="xsd:unsignedInt" minOccurs="0"/>
        <xsd:element name="RTPRetransmission" type="dvb:RETInfoType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

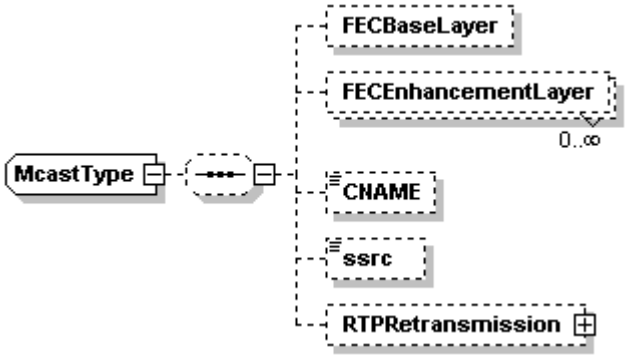


Figure C.11: McastType

This is used to hold a multicast address and optionally AL-FEC layer information and optionally RET information. This supports source specific multicast (SSM) addressing if the Source attribute is specified. Otherwise it supports any source multicast (ASM) addresses.

The CNAME and ssrc fields allow the carriage of values used by the service and may optionally be used to assist an HNED in identifying correct flows, and allocating unique numbers.

C.1.3.20 MosaicDescription

```
<xsd:complexType name="MosaicDescription">
    <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="LogicalCell">
            <xsd:complexType>
```

```
                <xsd:sequence>
                    <xsd:sequence maxOccurs="unbounded">
                        <xsd:element name="ElementaryCell">
                            <xsd:complexType>
                                <xsd:attribute name="CellId" type="dvb:Integer6bit" use="required"/>
                            </xsd:complexType>
                        </xsd:element>
                    </xsd:sequence>
                    <xsd:element name="AudioLink" minOccurs="0" maxOccurs="unbounded">
                        <xsd:complexType>
                            <xsd:attribute name="Language" type="dvb:ISO639-2" use="optional"/>
                            <xsd:attribute name="ComponentTag" type="dvb:Hexadecimal8bit"
use="required"/>
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:choice minOccurs="0">
                        <xsd:element name="TextualId" type="dvb:TextualIdentifier"/>
                        <xsd:element name="DVBTriplet" type="dvb:DVBTriplet"/>
                        <xsd:element name="PackageId">
                            <xsd:complexType>
                                <xsd:simpleContent>
                                    <xsd:extension base="dvb:Hexadecimal16bit">
                                        <xsd:attribute name="Domain" type="dvb:DomainType"
use="optional"/>
                                    </xsd:extension>
                                </xsd:simpleContent>
                            </xsd:complexType>
                        </xsd:element>
                    </xsd:choice>
                </xsd:sequence>
                <xsd:attribute name="CellId" type="dvb:Integer6bit" use="required"/>
                <xsd:attribute name="PresentationInfo" type="dvb:Hexadecimal3bit" use="required"/>
                <xsd:attribute name="LinkageInfo" type="dvb:Hexadecimal8bit" use="required"/>
                <xsd:attribute name="EventId" type="dvb:Hexadecimal16bit" use="optional"/>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="EntryPoint" type="xsd:boolean" default="true"/>
    <xsd:attribute name="HorizontalCellsNumber" type="dvb:Hexadecimal3bit" use="required"/>
    <xsd:attribute name="VerticalCellsNumber" type="dvb:Hexadecimal3bit" use="required"/>
</xsd:complexType>
```

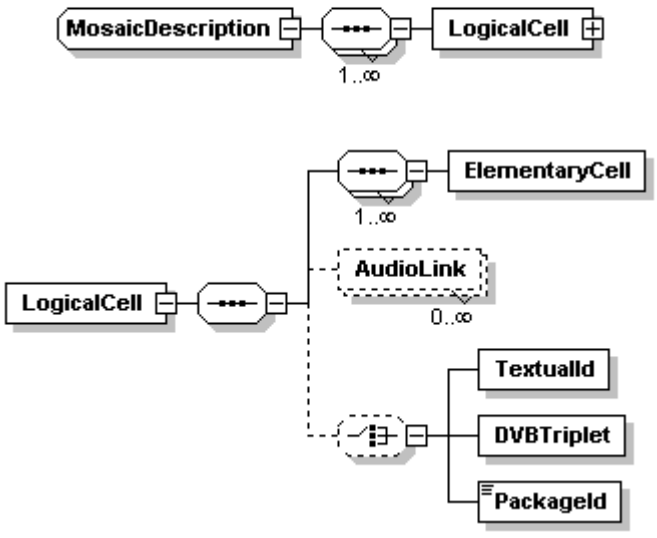


Figure C.12: MosaicDescription

An implementation of the Mosaic descriptor from EN 300 468 [1]. All fields are defined in EN 300 468 [1].

The AudioLink field allows a tag and language to be associated with each logical cell of the mosaic. This enables a different audio stream to be associated with each logical cell.

C.1.3.21 MulticastAddressAttribute

```
<xsd:attributeGroup name="MulticastAddressAttributes">
    <xsd:attributeGroup ref="dvb:BasicMulticastAddressAttributesType"/>
    <xsd:attribute name="Streaming" type="dvb:StreamingType" use="optional"/>
    <xsd:attributeGroup ref="dvb:FECAttributeGroupType"/>
</xsd:attributeGroup>
```

This supports Source Specific Multicast (SSM) addressing if the Source attribute is specified. Otherwise it supports Any Source Multicast (ASM) addresses.

C.1.3.22 MulticastRETType

```
<xsd:complexType name="MulticastRETType">
    <xsd:attribute name="SourceAddress" type="xsd:string" use="optional"/>
    <xsd:attribute name="GroupAddress" type="xsd:string" use="required"/>
    <xsd:attributeGroup ref="dvb:CommonCastRETType"/>
</xsd:complexType>
```

C.1.3.23 MultilingualType

```
<xsd:complexType name="MultilingualType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="Language" type="dvb:ISO639-2" use="required"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

Used to specify an element containing a textual message, which has a Language attribute specifying the language of the string, using the ISO 639-2 [51] three letter language code.

C.1.3.24 OfferingBase

```
<xsd:complexType name="OfferingBase">
    <xsd:attribute name="DomainName" type="dvb:DomainType" use="required"/>
    <xsd:attribute name="Version" type="dvb:Version" use="optional"/>
</xsd:complexType>
```

The base type from which all offerings should be derived. It provides the required Domain Type attribute, and the optional version field required when HTTP protocol is used.

C.1.3.25 OfferingListType

```
<xsd:complexType name="OfferingListType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Push" type="dvb:DVBSTPTransportModeType"/>
        <xsd:element name="Pull">
            <xsd:complexType>
                <xsd:complexContent>
                    <xsd:extension base="dvb:PayloadList">
                        <xsd:attribute name="Location" type="dvb:PullURL" use="required"/>
                    </xsd:extension>
                </xsd:complexContent>
            </xsd:complexType>
        </xsd:element>
    </xsd:choice>
</xsd:complexType>
```

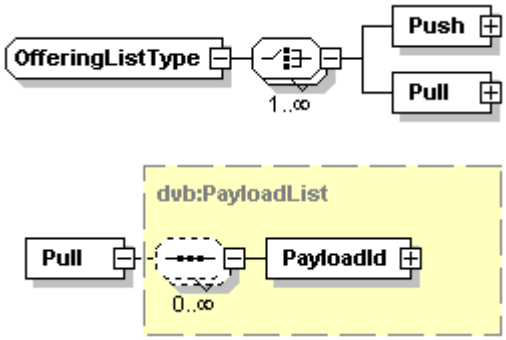


Figure C.13: OfferingListType

This type is used to convey the locations at which an offering can be found. It allows an unlimited list of either push or pull locations at which the specified service or information can be found. Note that the Pull element shall contain Segment Ids and version numbers.

Note the Pull element is deliberately not of type HTTPTransportModeType as there is no defined SOAP support for the SD&S information.

C.1.3.26 Package

```
<xsd:complexType name="Package">
    <xsd:sequence>
        <xsd:element name="PackageName" type="dvb:MultilingualType" maxOccurs="unbounded"/>
        <xsd:element name="CountryAvailability" type="dvb:CountryAvailability" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="PackageDescription" type="dvb:DescriptionLocationBCG" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="Service" type="dvb:PackagedServiceType" maxOccurs="unbounded"/>
        <xsd:element name="PackageReference" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
                <xsd:attribute name="Id" type="dvb:Hexadecimal16bit"/>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="PackageAvailability" type="dvb:ServiceAvailabilityType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="dvb:Hexadecimal16bit" use="required"/>
    <xsd:attribute name="Visible" type="xsd:boolean" use="optional" default="true"/>
</xsd:complexType>
```

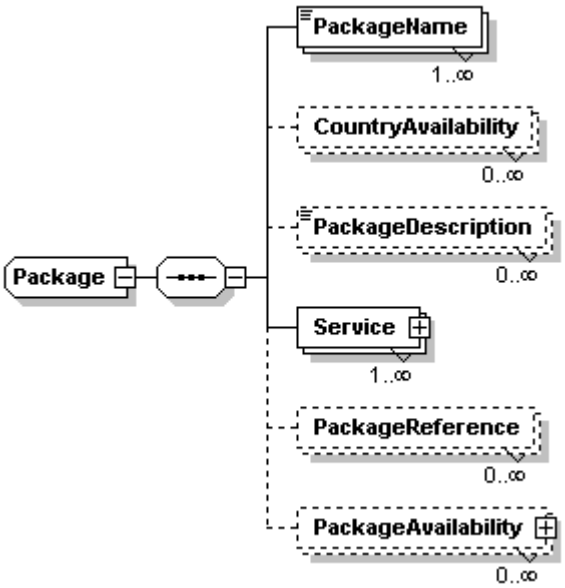


Figure C.14: Package

This provides a means to group services together into a "package" that the SP can offer or refer to as a unit.

The attribute "Id" of a Package is an identifier used to identify a package, and SPs shall ensure that it is unique within the scope of their services.

The attribute "Visible" of a Package is used to indicate if a package should not be displayed to the user and is present simply to provide efficient grouping of common packages.

Table C.2: Package Fields

Name	Definition
PackageName	The textual name of the package.
Service	One or more services which comprise the package
CountryAvailability	The countries within which the package is, or is not, available. This field is deprecated.
PackageDescription	A link to a BCG that provides a description of the content available in the package.
PackageReference	Allows inclusion of packages within other packages.
PackageAvailability	Structure which defines the geographical availability of the package.

C.1.3.27 PackageAvailabilityCountryCodeType

```
<xsd:complexType name="PackageAvailabilityCountryCodeType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="Availability" type="xsd:boolean" default="true"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

C.1.3.28 PackagedServiceType

```
<xsd:complexType name="PackagedServiceType">
    <xsd:sequence>
        <xsd:element name="TextualID" type="dvb:TextualIdentifier"/>
        <xsd:element name="DVBTriplet" type="dvb:DVBTriplet" minOccurs="0"/>
        <xsd:element name="DescriptionLocation" type="dvb:DescriptionLocationBCG" minOccurs="0"/>
        <xsd:element name="LogicalChannelNumber" type="xsd:positiveInteger" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

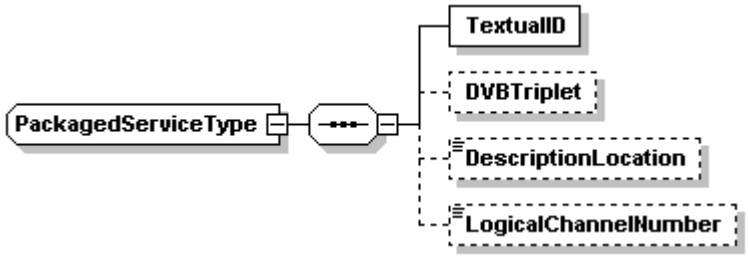


Figure C.15: PackagedServiceType

C.1.3.29 PayloadList

```
<xsd:complexType name="PayloadList">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="PayloadId">
            <xsd:complexType>
                <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="Segment">
                        <xsd:complexType>
                            <xsd:attribute name="Version" type="dvb:Version" use="optional"/>
                            <xsd:attribute name="ID" type="dvb:Hexadecimal16bit" use="required"/>
                        </xsd:complexType>
                    </xsd:element>
                </xsd:sequence>
                <xsd:attribute name="Id" type="dvb:Hexadecimal8bit" use="required"/>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
```

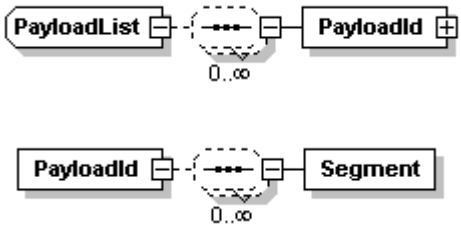


Figure C.16: PayloadList

This type describes a list of payload IDs (as described in clause 5.2.2.1) and optional SegmentIDs (similiarily described in clause 5.2.2.1).

C.1.3.30 RegionalisationOffering

```
<xsd:complexType name="RegionalisationOffering">
    <xsd:complexContent>
        <xsd:extension base="dvb:OfferingBase">
            <xsd:sequence>
                <xsd:element name="Cell" type="dvb:Cell" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

C.1.3.31 ReplacementService

```
<xsd:complexType name="ReplacementService">
    <xsd:choice>
        <xsd:element name="TextualIdentifier" type="dvb:TextualIdentifier"/>
        <xsd:element name="DVBTriplet" type="dvb:DVBTriplet"/>
    </xsd:choice>
    <xsd:attribute name="ReplacementType" type="dvb:Hexadecimal8bit" use="optional" default="5"/>
</xsd:complexType>
```

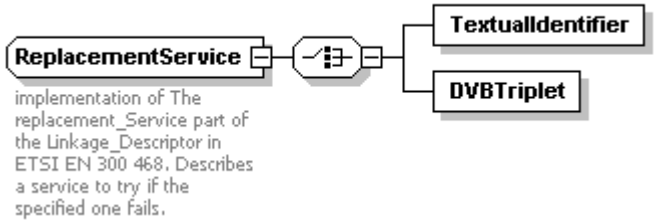


Figure C.17: Replacement Service

This is an XML representation of the replacement service functionality of the Linkage descriptor in EN 300 468 [1]. The service indicated by either the DVB triplet or the textual identifier may be used when the specified service (as derived from the context) fails.

C.1.3.32 RETInfoType

```
<xsd:complexType name="RETInfoType">
    <xsd:sequence>
        <xsd:element name="RTCPReporting" type="dvb:RTCPReportingType"/>
        <xsd:element name="UnicastRET" type="dvb:UnicastRETType" minOccurs="0"/>
        <xsd:element name="MulticastRET" type="dvb:MulticastRETType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

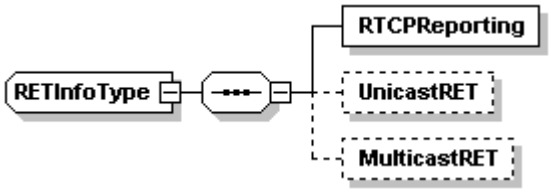


Figure C.18: RETInfoType

C.1.3.33 RMSFUSMulticastAddressType

```
<xsd:complexType name="RMSFUSMulticastAddressType">
    <xsd:attributeGroup ref="dvb:BasicMulticastAddressAttributesType"/>
    <xsd:attribute name="Protocol" use="optional">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="1 SAP"/>
                <xsd:enumeration value="2 DVBSTP"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
```

C.1.3.34 RMSType

```
<xsd:complexType name="RMSType">
    <xsd:sequence>
        <xsd:element name="RMSProvider" type="dvb:MultilingualType" maxOccurs="unbounded"/>
        <xsd:element name="RMSID" type="xsd:decimal" minOccurs="0"/>
        <xsd:element name="Description" type="dvb:MultilingualType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="LogoURI" type="xsd:anyURI" use="optional"/>
    <xsd:attribute name="RMSLocation" type="xsd:anyURI" use="required"/>
</xsd:complexType>
```

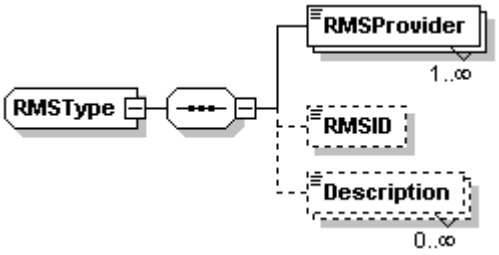


Figure C.19: RMSType

C.1.3.35 RTCPReportingType

```
<xsd:complexType name="RTCPReportingType">
    <xsd:attribute name="DestinationAddress" type="xsd:string" use="required"/>
    <xsd:attribute name="DestinationPort" type="xsd:unsignedShort" use="optional"/>
    <xsd:attribute name="dvb-t-ret" type="xsd:positiveInteger" use="optional"/>
    <xsd:attribute name="rtcp-bandwidth" type="xsd:positiveInteger" use="optional"/>
    <xsd:attribute name="rtcp-rsize" type="xsd:positiveInteger" use="optional"/>
    <xsd:attribute name="trr-int" type="xsd:positiveInteger" use="optional"/>
    <xsd:attribute name="dvb-disable-rtcp-rr" type="xsd:boolean" use="optional" default="false"/>
    <xsd:attribute name="dvb-enable-byte" type="xsd:boolean" use="optional" default="false"/>
    <xsd:attribute name="dvb-t-wait-min" type="xsd:unsignedInt" use="optional" default="0"/>
    <xsd:attribute name="dvb-t-wait-max" type="xsd:unsignedInt" use="optional" default="0"/>
    <xsd:attribute name="dvb-ssrc-bitmask" type="dvb:Hexadecimal32bit" use="optional"
default="ffffffff"/>
    <xsd:attribute name="dvb-rsi-mc-ret" type="xsd:boolean" use="optional"/>
    <xsd:attribute name="dvb-ssrc-upstream-client" type="xsd:positiveInteger" use="optional"/>
</xsd:complexType>
```

C.1.3.36 RTSPURLType

```
<xsd:complexType name="RTSPURLType">
    <xsd:simpleContent>
        <xsd:extension base="dvb:RTSP">
            <xsd:attribute name="RTSPControlURL" type="xsd:anyURI" use="optional"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

This type is used to allow an additional RTSP Control URL to be carried, as discussed in clauses 5 and 6, to assist with AL-FEC and/or RET services.

C.1.3.37 ServiceAvailabilityType

```
<xsd:complexType name="ServiceAvailabilityType">
    <xsd:sequence>
        <xsd:element name="CountryCode" type="dvb:PackageAvailabilityCountryCodeType"/>
        <xsd:element name="Cells" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

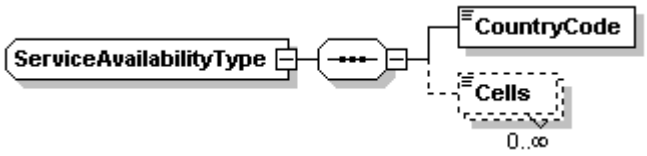


Figure C.20: ServiceAvailabilityType

Table C.3: Service Availability Fields

Name		Definition	Mandatory/Optional
CountryCode		This element indicates the country for which the availability is being defined. This element shall be of the 2-letter format specified in ISO 3166 [50].	M
	@Availability	This flag indicates whether the service is available in the country specified by CountryCode. The default is TRUE. When TRUE, the service is available in the specified country with the exception of those regions identified by Cells. When FALSE, the service is not available in the specified country with the exception of those regions identified by Cells.	O
Cells		A list of string identifiers representing geographical regions in the country identified by CountryCode. The Cells listed represent the exception to the value supplied by the flag, i.e. the negation of the Availability flag applies to any listed cells.	O

C.1.3.38 ServiceLocation

```
<xsd:complexType name="ServiceLocation">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="IPMulticastAddress" type="dvb:McastType"/>
        <xsd:element name="RTSPURL" type="dvb:RTSPURLType"/>
    </xsd:choice>
</xsd:complexType>
```

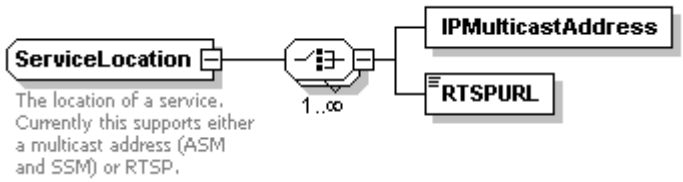


Figure C.21: ServiceLocation

This describes the location(s) at which a service may be found, either a multicast location or via an RTSP server.

C.1.3.39 ServiceProviderType

```
<xsd:complexType name="ServiceProviderType">
    <xsd:sequence>
        <xsd:element name="Name" type="dvb:MultilingualType" maxOccurs="unbounded"/>
        <xsd:element name="Description" type="dvb:MultilingualType" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="Offering" type="dvb:OfferingListType" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="DomainName" type="dvb:DomainType" use="required"/>
    <xsd:attribute name="Version" type="dvb:Version" use="required"/>
    <xsd:attribute name="LogoURI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>
```

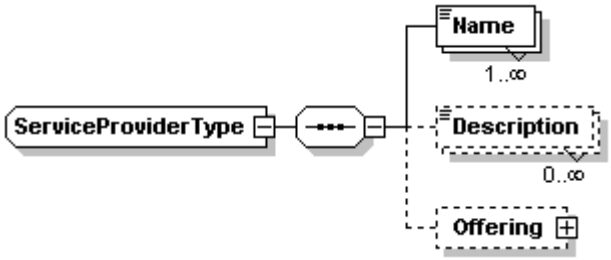


Figure C.22: ServiceProviderType

This type provides details of a SP and the references to the service offerings.

Table C.4: Service provider fields

Name	Definition
ServiceProvider	A service provider consists of:
Name	The textual name of the service provider.
Description	An optional description of the service provider.
Offering	The location of where details of the service providers offering may be found.
@DomainName	The domain name of the service provider.
@Version	The version of the service providers record.
@LogoURI	A URI for a logo for the service provider.

If the element Offering is missing, then the ServiceProvider is not currently providing any services, but simply announcing its presence.

C.1.3.40 SI

```
<xsd:complexType name="SI">
    <xsd:sequence>
        <xsd:element name="Name" type="dvb:MultilingualType" maxOccurs="unbounded"/>
        <xsd:element name="Description" type="dvb:MultilingualType" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="ServiceDescriptionLocation" type="dvb:DescriptionLocationBCG"
minOccurs="0" maxoccurs="unbounded"/>
        <xsd:element name="ContentGenre" type="dvb:Genre" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="CountryAvailability" type="dvb:CountryAvailability" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="ReplacementService" type="dvb:ReplacementService" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="MosaicDescription" type="dvb:MosaicDescription" minOccurs="0"/>
        <xsd:element name="AnnouncementSupport" type="dvb:AnnouncementSupport" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="ServiceType" type="dvb:ServiceType" use="required"/>
    <xsd:attribute name="PrimarySISource" type="dvb:PrimarySISource" use="optional" default="XML"/>
</xsd:complexType>
```

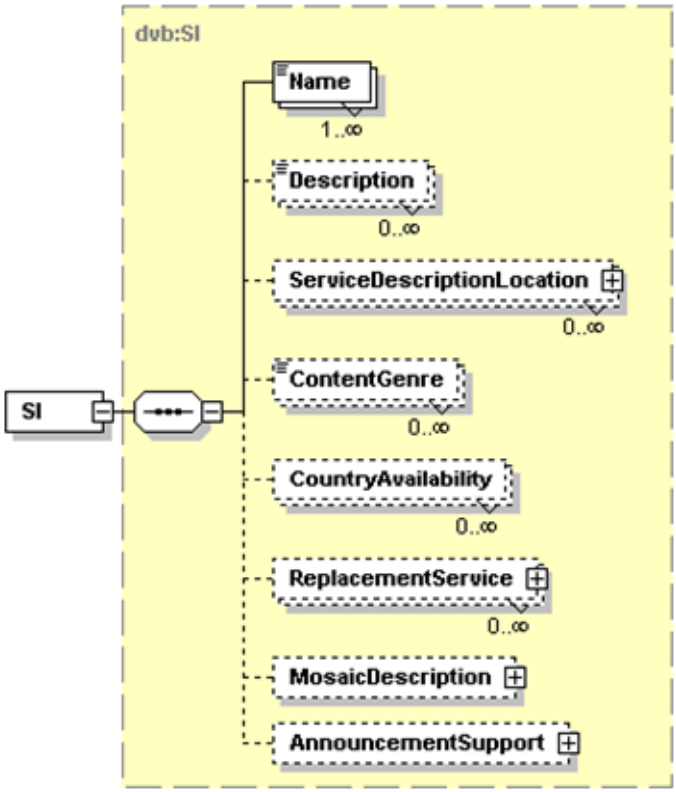


Figure C.23: SI

This type describes the service information traditionally provided in a stream as DVB descriptors.

Table C.5: SI Fields

Name	Definition
Name	The text form of the name by which the service is known to the user.
Description	A textual description of the service.
ContentGenre	The (primary) genre of the service.
CountryAvailability	The list of countries in which the service is, or is not, available.
AnnouncementSupport	The announcements supported by the service, and linkage information as to their location.
ReplacementService	Details the linkage to a service that can be used in case of a failure of the service to which this SI record refers.
MosaicDescription	Details of the services, or service packages, which are displayed in a mosaic stream.
ServiceDescriptionLocation	The identifier(s) of the BCG Record(s) for the BCG Discovery element that

	carries the information on this offering.
@ServiceType	An attribute that is an eight-bit number encoding the type of the service, using traditional DVB values.
@PrimarySISource	An attribute indicating whether the XML record, or SI in the transport stream takes precedence.

C.1.3.41 TextualIdentifier

```
<xsd:complexType name="TextualIdentifier">
    <xsd:attribute name="DomainName" type="dvb:DomainType" use="optional"/>
    <xsd:attribute name="ServiceName" type="dvb:Service" use="required"/>
</xsd:complexType>
```

A service can be identified in a textual fashion. This identifier is comprised of the domain name of the SP and the textual service name. The domain name may be omitted where it can be inferred from the context. The Textual Identifier is the means of uniquely identifying an IP service.

This is an implementation of the textual service identifier, as specified in TS 101 812 [3], clause 14.9.1.

C.1.3.42 TransportModeType

```
<xsd:complexType name="TransportModeType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="DVBSTP" type="dvb:DVBSTPTransportModeType"/>
        <xsd:element name="HTTP" type="dvb:HTTPTransportModeType"/>
    </xsd:choice>
</xsd:complexType>
```

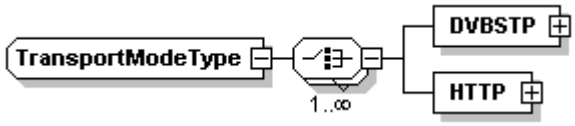


Figure C.24: TransportModeType

This type is used to indicate both the mechanism used to carry BCG information, and the payloadIds and segmentIds of the relevant information.

C.1.3.43 UnicastRETType

```
<xsd:complexType name="UnicastRETType">
    <xsd:attribute name="trr-int" type="xsd:unsignedInt" use="optional"/>
    <xsd:attribute name="DestinationPort-ForRTCPReporting" type="xsd:unsignedInt" use="optional"/>
    <xsd:attribute name="SourcePort" type="xsd:unsignedInt" use="optional"/>
    <xsd:attribute name="RTSPControlURL" type="xsd:anyURI" use="optional"/>
    <xsd:attributeGroup ref="dvb:CommonCastRETType"/>
</xsd:complexType>
```

C.1.4 Element Types

C.1.4.1 BCGOffering

```
<xsd:complexType name="BCGOffering">
    <xsd:complexContent>
        <xsd:extension base="dvb:OfferingBase">
            <xsd:sequence>
                <xsd:element name="BCG" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="Name" type="dvb:MultilingualType"
maxOccurs="unbounded"/>
                            <xsd:element name="Description" type="dvb:MultilingualType"
minOccurs="0" maxOccurs="unbounded"/>
                            <xsd:element name="TransportMode" type="dvb:TransportModeType"/>
                            <xsd:element name="Logo" type="xsd:anyURI" minOccurs="0"/>
```

```
                        <xsd:element name="Type" type="tva:ControlledTermType" minOccurs="0"/>
                        <xsd:element name="TargetProvider" type="dvb:DomainType" minOccurs="0"
maxOccurs="unbounded"/>
                        <xsd:element name="BCGProviderName" type="dvb:MultilingualType"
minOccurs="0" maxOccurs="unbounded"/>
                        <xsd:element name="CDSDownloadSessionDescriptionLocation"
type="dvb:CDSDownloadSessionDescriptionLocationType" minOccurs="0"/>
    </xsd:sequence>
                    <xsd:attribute name="Id" type="tva:TVAIDType" use="required"/>
                    <xsd:attribute name="Version" type="dvb:Version" use="optional"/>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

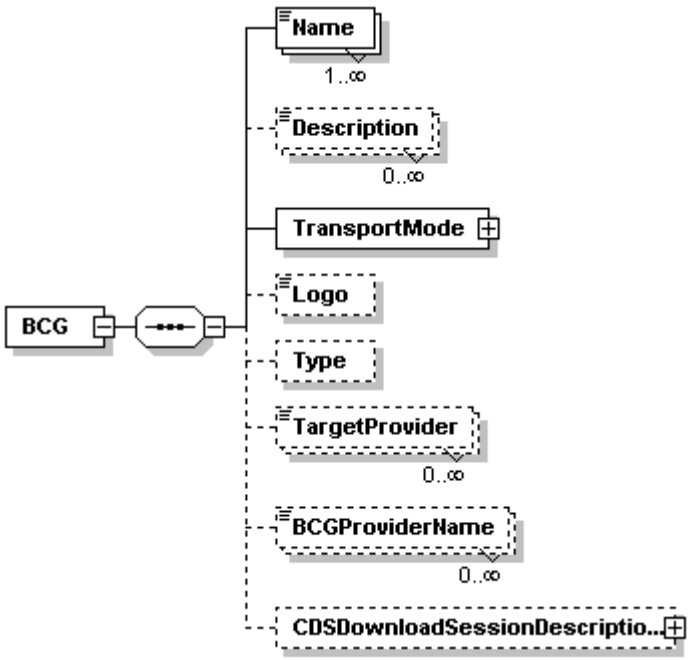


Figure C.25: BCGOffering

This element is used to discover Broadcast Content Guide (BCG) Offerings.

Table C.6: BCG Offering fields

Name	Definition
BCGOffering	A BCG Offering consists of several BCG elements
BCG	A BCG consists of:
@Id	A unique textual identifier for the BCG.
@Version	Version of the BCG.
Name	The name of the BCG.
Description	An optional description of the BCG.
TransportMode	The location of where the BCG offering may be found.
Logo	A URI for a logo for the BCG.
Type	Optional element indicating the type of the BCG. The type is tva:ControlledTermType. The different values of the BCG type are defined in the following extensible ClassificationScheme.
TargetProvider	The SP whose content is described by this BCG (for example Canal+). The domain name shall be the same as a domain name present in the ServiceList.
BCGProviderName	The name of the BCG provider (for example "Telerama"). This field shall be identical to the textual string of the Publisher attribute of the TVAMain element in the BCG metadata.
CDSDownloadSessionDescriptionLocation	The multicast location where details of the CDS download Session Description can be found.

The classification scheme for the Type element is as follows:

```
<ClassificationScheme uri="urn:dvb:metadata:cs:BCGTypeCS:2007">
    <Term termID="1">
        <Name xml:lang="en">Live</Name>
        <Definition xml:lang="en">BCG for live TV programs</Definition>
    </Term>
    <Term termID="2">
        <Name xml:lang="en">CoD</Name>
        <Definition xml:lang="en">BCG for Content on Demand programs </Definition>
    </Term>
    <Term termID="3">
        <Name xml:lang="en">Downloadable Content</Name>
        <Definition xml:lang="en">BCG for downloadable content</Definition>
    </Term>
</ClassificationScheme>
```

This classification scheme is also attached as BCGTypeCS.xml in the file ts_102034v010401p0.zip which accompanies the present document.

C.1.4.2 BroadcastOffering

```
<xsd:complexType name="BroadcastOffering">
    <xsd:complexContent>
        <xsd:extension base="dvb:OfferingBase">
            <xsd:sequence>
                <xsd:element name="ServiceList" type="dvb:IPServiceList" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

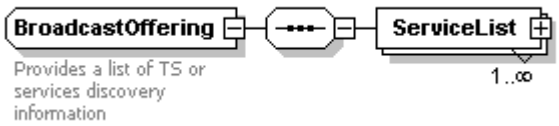


Figure C.26: Broadcast offering

This element is used where the SP is offering a range of "broadcast" services, which are continuously streamed MPEG-2 transport streams. The services provided are grouped in ServiceLists (which may contain only a single service), which is represented by an instantiation of the complex type IPServiceList.

C.1.4.3 CoDOffering

```
<xsd:complexType name="CoDOffering">
    <xsd:complexContent>
        <xsd:extension base="dvb:OfferingBase">
            <xsd:sequence>
                <xsd:element name="Catalogue" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="Name" type="dvb:MultilingualType"
maxOccurs="unbounded"/>
                            <xsd:element name="Description" type="dvb:MultilingualType"
minOccurs="0" maxOccurs="unbounded"/>
                            <xsd:element name="Locator" type="dvb:DescriptionLocation"
maxOccurs="unbounded"/>
                        </xsd:sequence>
                        <xsd:attribute name="Id" type="dvb:Hexadecimal16bit" use="required"/>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

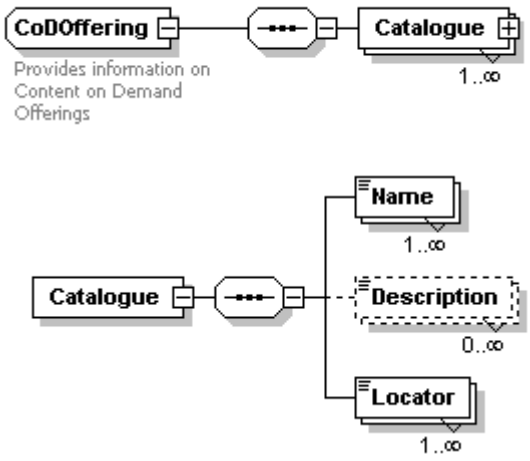


Figure C.27: Content on Demand (CoD)

This element is used where the SP is offering "content on demand" services.

Table C.7: Content on Demand (CoD) catalogue fields

Name	Definition
Catalogue	A catalogue, that consists of:
Name	The name of the catalogue.
Description	A description of the catalogue.
Locator	One or more URI(s) specifying where the catalogue can be found.
@Id	A 16 bit Id used to refer to the catalogue.

Note that use of this element is now deprecated.

C.1.4.4 PackagedServices

```
<xsd:complexType name="PackagedServices">
    <xsd:complexContent>
        <xsd:extension base="dvb:OfferingBase">
            <xsd:sequence>
                <xsd:element name="Package" type="dvb:Package" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

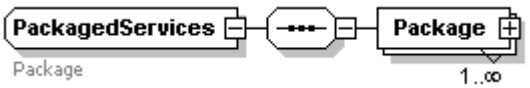


Figure C.28: Packaged services

C.1.4.5 ReferencedServices

```
<xsd:complexType name="ReferencedServices">
    <xsd:complexContent>
        <xsd:extension base="dvb:OfferingBase">
            <xsd:sequence>
                <xsd:element name="ReferencedServiceProvider" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="Service" minOccurs="0" maxOccurs="unbounded">
                                <xsd:complexType>
                                    <xsd:attribute name="Name" type="dvb:Service" use="required"/>
                                </xsd:complexType>
                            </xsd:element>
                        </xsd:sequence>
                        <xsd:attribute name="Domain" type="dvb:DomainType" use="required"/>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

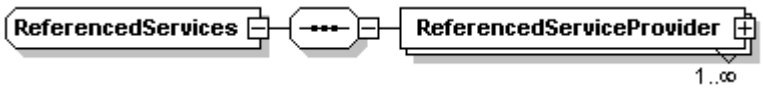


Figure C.29: Referenced services

This provides a means for a SP to list services provided by other SPs from within his own service discovery information.

Table C.8: Referenced services fields

Name	Definition
ReferencedServiceProvider	A group of one or more service from a different SP to which the SP of the current context wishes to refer.
Service	A list of one or more referenced services.
@Name	The name of the each referenced service.
@Domain	The domain component of the textual service identifier of the SP which is referred to.

C.1.4.6 RMSFUSDiscoveryType

```
<xsd:complexType name="RMSFUSDiscoveryType">
    <xsd:complexContent>
        <xsd:extension base="dvb:OfferingBase">
            <xsd:choice>
                <xsd:element name="FUSProvider" type="dvb:FUSType" maxOccurs="unbounded"/>
                <xsd:element name="RMSProvider" type="dvb:RMSType" maxOccurs="unbounded"/>
            </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

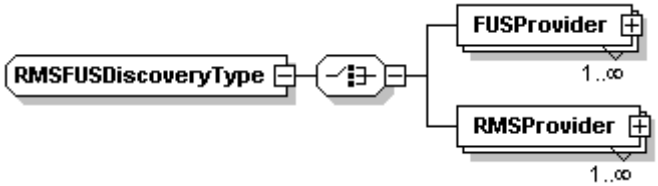


Figure C.30: RMSFUSDiscoveryType

This element is used to indicate the presence of either a FUS or RMS Provider. Full details of the RMS and FUS mechanisms are provided in TS 102 824 [79].

C.1.4.7 ServiceProviderListType

```
<xsd:complexType name="ServiceProviderListType">
    <xsd:sequence>
        <xsd:element name="ServiceProvider" type="dvb:ServiceProviderType" maxOccurs="unbounded">
    </xsd:sequence>
</xsd:complexType>
```

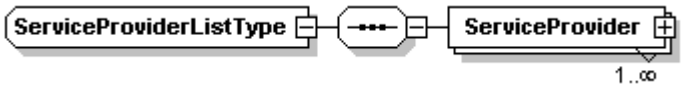


Figure C.31: Service Provider (SP)

This element is used in the first stage of service discovery. It is sent by SPs and is used as a link to their own service discovery information.

An aggregating SP may send multiple ServiceProvider elements in a single document.

ServiceProvider are represented by an instantiation of the complex type ServiceProviderType.

C.1.5 Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="urn:dvb:metadata:iptv:sdns:2008-1"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:dvb="urn:dvb:metadata:iptv:sdns:2008-1"
xmlns:tva="urn:tva:metadata:2005" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:import namespace="urn:tva:metadata:2005" schemaLocation="./tva_metadata_3-1_v131.xsd"/>
    <xsd:element name="ServiceDiscovery">
        <xsd:complexType>
            <xsd:choice>
                <xsd:element name="BroadcastDiscovery" type="dvb:BroadcastOffering"
maxOccurs="unbounded"/>
                <xsd:element name="CoDDiscovery" type="dvb:CoDOffering" maxOccurs="unbounded"/>
                <xsd:element name="ServicesFromOtherSP" type="dvb:ReferencedServices"
maxOccurs="unbounded"/>
                <xsd:element name="PackageDiscovery" type="dvb:PackagedServices"
maxOccurs="unbounded"/>
                <xsd:element name="ServiceProviderDiscovery" type="dvb:ServiceProviderListType"
maxOccurs="unbounded"/>
                <xsd:element name="BCGDiscovery" type="dvb:BCGOffering" maxOccurs="unbounded"/>
                <xsd:element name="RegionalisationDiscovery" type="dvb:RegionalisationOffering"
maxOccurs="unbounded"/>             <xsd:element name="RMSFUSDiscovery"
type="dvb:RMSFUSDiscoveryType" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:choice>
            <xsd:attribute name="Version" type="dvb:Version" use="optional"/>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="CoDAnnounceDescribe">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="ContentDescription" type="tva:BasicContentDescriptionType"/>
                <xsd:element name="FECInfo" type="dvb:FECInfoType" minOccurs="0"/>
                <xsd:element name="RETInfo" type="dvb:RETInfoType" minOccurs="0"/>
            </xsd:sequence>
            <xsd:attribute name="RTSPControlURL" use="optional"/>
            <xsd:attribute name="Streaming" type="dvb:StreamingType" use="optional"/>
        </xsd:complexType>
    </xsd:element>
    </xsd:schema>
```

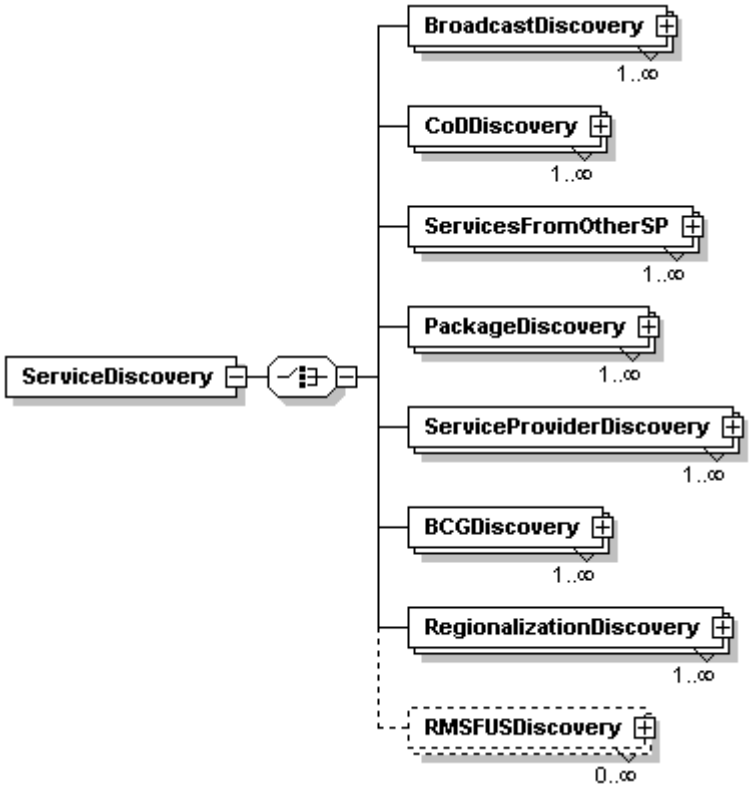


Figure C.32: Service discovery

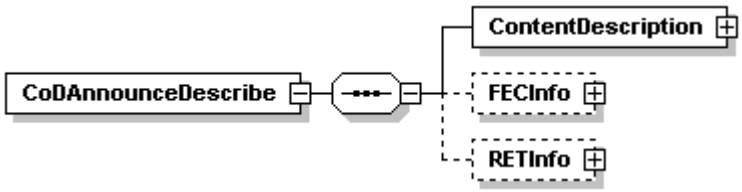


Figure C.33

Figure C.33: CoDAnnounceDescribe (RTSP only). Figure C.31 shows the structure of a service offering. Each service offering shall contain only one of the "Element Types" as described in clause 5.2.6, but may have multiple instances of this type.

The version attribute of the offering is used as described in clauses 5.2.5 and 5.2.6. It is used to carry the version number of the XML document within the XML. Note that for records described in clause 5.2.6 the version number is provided through the OfferingBase type as defined in clause C.1.3.24.

The Version attribute of the root element (ServiceDiscovery) shall be present when XML is delivered via the pull mode (HTTP). It is recommended that the version attribute is not present when the XML is delivered via push mode (multicast) and in this case the value of the missing Version attribute is equal to the Version field of the DVBSTP Segment header.

Figure C.32 shows the root element CoDAnnounceDescribe that shall only be present in documents used as part of the RTSP ANNOUNCE and DESCRIBE methods as outlined in clause 6.

C.1.6 Multicasting SD&S XML documents

Where multicast is used to distribute the service discovery information, the DVBSTP protocol defined in clause 5.4.1 shall be used. The following clauses define how the XML shall be mapped into the protocol.

C.1.6.1 XML records and payload ID

XML records shall be constructed such that each record only contains elements of one of the types from clause C.1.4. The payloadId field of the multicast protocol header shall be set to reflect the type of record contained within the transmitted multicast packets. Thus any XML record shall contain the root element (ServiceDiscovery) which contains only an arbitrary number of BroadcastDiscovery elements, or only an arbitrary number of CoDDiscovery elements, or only an arbitrary number of ServicesFromOtherSP elements, or only an arbitrary number of PackageDiscovery elements, or only an arbitrary number of ServiceProviderDiscovery elements.

C.1.6.2 Segmentation of records

Records containing SP discovery information (i.e. Payload ID 0x01) shall not be segmented when using the "pull mode".

All other cases, the XML records shall be segmented, that is divided up into smaller units, to enable easier processing in the HNED, or variable access times. Note that a record may be divided into a single segment.

Each segment shall contain a complete root element (ServiceDiscovery) which comprises of an integral number of child elements (BroadcastDiscovery, or CoDDiscovery, or ServicesFromOtherSP, or PackageDiscovery, or ServiceProviderDiscovery), as defined in clause C.1.4 (specifically, a segment shall not contain part of a child element). A segment shall not contain more than one type of child element (i.e. it shall be in accordance with clause C.6.1).

Each segment shall be valid and well formed.

Each segment shall have a segment ID that is unique within the scope of the SP and the payload ID. For a shared multicast address the SP shall be signalled by the conditional Provider ID field of the DVB-STP header (see clause 5.4.1). For a multicast address carrying only a single SP, this information is inferred from the multicast address. With HTTP, the SP is included in the request (see clause 5.4.2).

Segment Ids need not be contiguous.

C.2 CDS XML Schemas

The following sub clauses define the various types and elements that are used in the CDS XML schema. The full normative XML schema is available as the file dvb_metadata_iptv_cds_2008-1.xsd in archive ts_102034v010401p0.zip which accompanies the present document.

C.2.1 Namespace

The namespace for the CDS XML schema is urn:dvb:metadata:iptv:cds:2008-1.

C.2.2 Basic schema definitions

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:cds="urn:dvb:metadata:iptv:cds:2008-1" xmlns:sdns="urn:dvb:metadata:iptv:sdns:2008-
1" attributeFormDefault="unqualified" elementFormDefault="qualified" targetNamespace="
urn:dvb:metadata:iptv:cds:2008-1" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import schemaLocation=".\sdns handbook v1 4 r10.xsd" namespace="
urn:dvb:metadata:iptv:sdns:2008-1" />
  <xs:annotation>
    <xs:documentation xml:lang="en">This document defines XML schemas used for DVB IPTV Content
Dwonload Services (CDS)</xs:documentation>
  </xs:annotation>
<xs:annotation>
    <xs:documentation xml:lang="en">XML schema for download session description</xs:documentation>
</xs:annotation>
...

<xs:annotation>
    <xs:documentation xml:lang="en">XML schema for reception reporting</xs:documentation>
</xs:annotation>
...
```

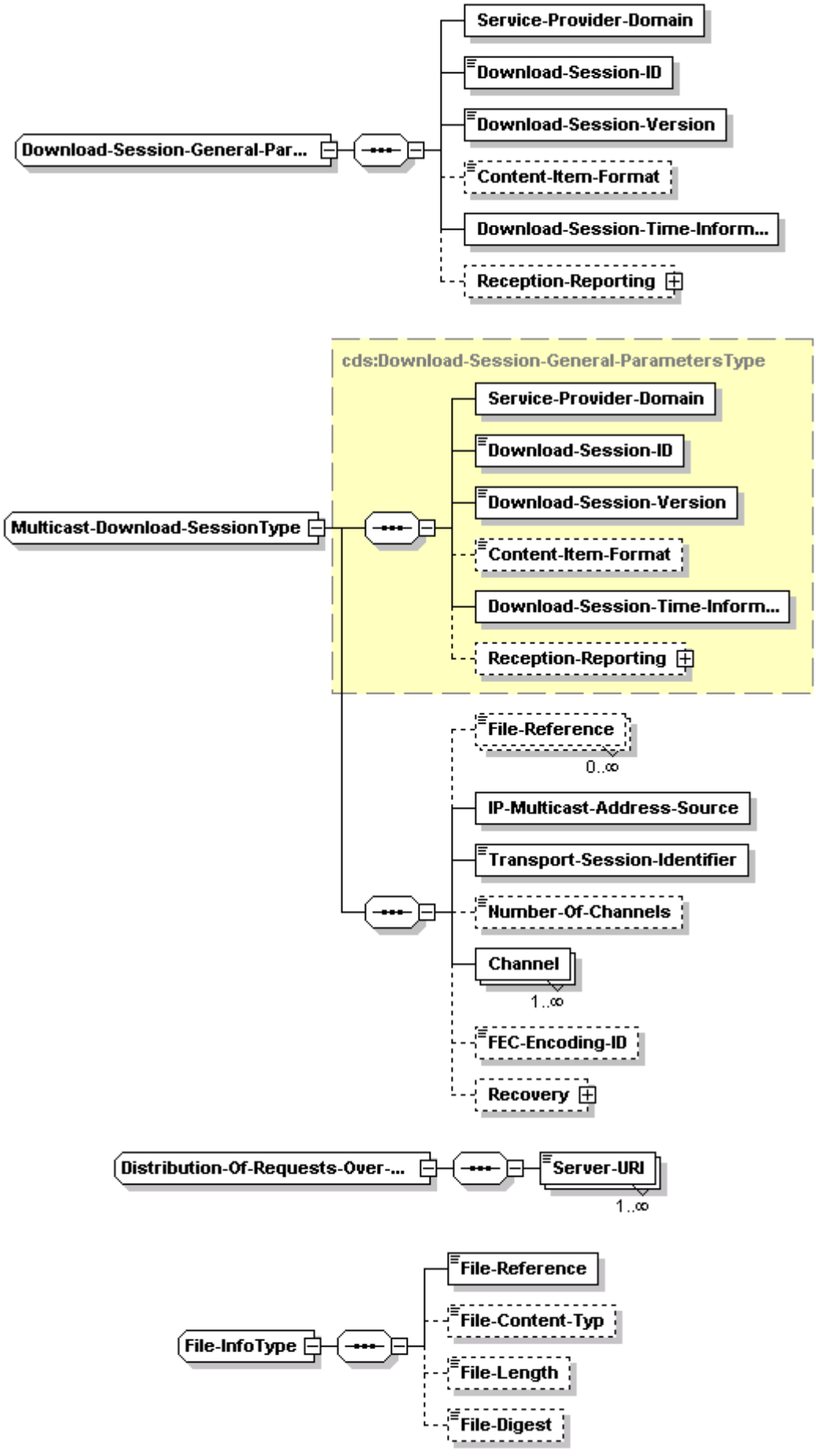
```
</xs:schema>
```

C.2.3 Download session description

```
  <xs:complexType name="Download-Session-General-ParametersType">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="Service-Provider-Domain" type="sdns:DomainType"
/>
      <xs:element minOccurs="1" maxOccurs="1" name="Download-Session-ID"
type="xs:nonNegativeInteger" />
      <xs:element minOccurs="1" maxOccurs="1" name="Download-Session-Version"
type="xs:nonNegativeInteger" />
      <xs:element minOccurs="0" maxOccurs="1" name="Content-Item-Format">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedByte">
            <xs:enumeration value="0" />
            <xs:enumeration value="1" />
            <xs:enumeration value="2" />
            <xs:enumeration value="3" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element minOccurs="1" maxOccurs="1" name="Download-Session-Time-Information">
        <xs:complexType>
          <xs:attribute name="Start-Time" type="xs:dateTime" use="required" />
          <xs:attribute name="Stop-Time" type="xs:dateTime" use="optional" />
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="Reception-Reporting">
        <xs:complexType>
          <xs:complexContent mixed="false">
            <xs:extension base="cds:Distribution-Of-Requests-Over-Time-And-ServersType">
              <xs:attribute default="0" name="Mode" use="optional">
                <xs:simpleType>
                  <xs:restriction base="xs:unsignedByte">
                    <xs:enumeration value="0" />
                    <xs:enumeration value="1" />
                    <xs:enumeration value="2" />
                  </xs:restriction>
                </xs:simpleType>
              </xs:attribute>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Unicast-Download-Session">
    <xs:complexType>
      <xs:complexContent mixed="false">
        <xs:extension base="cds:Download-Session-General-ParametersType">
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="unbounded" name="File-Info">
              <xs:complexType>
                <xs:choice>
                  <xs:element name="File-Server-Info" type="cds:File-Server-InfoType" />
                  <xs:element name="File-Server-Chunk-Info" type="cds:File-Server-Chunk-InfoType" />
                </xs:choice>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="Multicast-Download-SessionType">
    <xs:complexContent mixed="false">
      <xs:extension base="cds:Download-Session-General-ParametersType">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="File-Reference">
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:anyURI" />
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
```

```
          <xs:element minOccurs="1" maxOccurs="1" name="IP-Multicast-Address-Source"
type="sdns:IPOrDomainType" />
          <xs:element minOccurs="1" maxOccurs="1" name="Transport-Session-Identifier">
            <xs:simpleType>
              <xs:restriction base="xs:unsignedLong">
                <xs:maxInclusive value="281474976710655" />
                <xs:minInclusive value="0" />
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element minOccurs="0" default="1" name="Number-Of-Channels">
            <xs:simpleType>
              <xs:restriction base="xs:unsignedByte">
                <xs:maxInclusive value="16" />
                <xs:minInclusive value="1" />
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element maxOccurs="unbounded" name="Channel">
            <xs:complexType>
              <xs:attribute name="IP-Multicast-Address" type="sdns:IPType" use="required" />
              <xs:attribute name="IP-Multicast-Port-Number" type="xs:unsignedShort" use="required"
/>
              <xs:attribute name="Max-Bandwidth-Requirement" type="xs:positiveInteger"
use="optional" />
            </xs:complexType>
          </xs:element>
          <xs:element minOccurs="0" default="0" name="FEC-Encoding-ID" type="xs:unsignedByte" />
          <xs:element minOccurs="0" name="Recovery">
            <xs:complexType>
              <xs:complexContent mixed="false">
                <xs:extension base="cds:Distribution-Of-Requests-Over-Time-And-ServersType">
                  <xs:attribute default="0" name="Mode" use="optional">
                    <xs:simpleType>
                      <xs:restriction base="xs:unsignedByte">
                        <xs:enumeration value="0" />
                        <xs:enumeration value="1" />
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:attribute>
                </xs:extension>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="Carousel-Multicast-Download-Session">
    <xs:complexType>
      <xs:complexContent mixed="false">
        <xs:extension base="cds:Multicast-Download-SessionType" />
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Scheduled-Multicast-Download-Session">
    <xs:complexType>
      <xs:complexContent mixed="false">
        <xs:extension base="cds:Multicast-Download-SessionType">
          <xs:sequence>
            <xs:element minOccurs="0" name="Completion-Poll-Response-Server">
              <xs:complexType>
                <xs:attribute name="IP-Address" type="sdns:IPOrDomainType" use="required" />
                <xs:attribute name="Port-Number" type="xs:unsignedShort" use="required" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="Distribution-Of-Requests-Over-Time-And-ServersType">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="Server-URI" type="xs:anyURI" />
    </xs:sequence>
    <xs:attribute default="0" name="Offset-Time" type="xs:unsignedShort" />
    <xs:attribute default="0" name="Random-Time-Period" type="xs:unsignedShort" />
  </xs:complexType>
```

```
  <xs:complexType name="File-InfoType">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="File-Reference" type="xs:anyURI" />
      <xs:element minOccurs="0" maxOccurs="1" name="File-Content-Typ" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Content-Item-Format">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedByte">
            <xs:enumeration value="0" />
            <xs:enumeration value="1" />
            <xs:enumeration value="2" />
            <xs:enumeration value="3" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="File-Length" type="xs:unsignedLong" />
      <xs:element minOccurs="0" maxOccurs="1" name="File-Digest" type="xs:base64Binary" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="File-Server-InfoType">
    <xs:complexContent mixed="false">
      <xs:extension base="cds:File-InfoType">
        <xs:sequence>
          <xs:element minOccurs="1" maxOccurs="unbounded" name="Server-Info">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="1" maxOccurs="1" name="Server-Base-URI" type="xs:anyURI" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="File-Server-Chunk-InfoType">
    <xs:complexContent mixed="false">
      <xs:extension base="cds:File-InfoType">
        <xs:sequence>
          <xs:element minOccurs="1" maxOccurs="1" name="Chunk-Length" type="xs:unsignedLong" />
          <xs:element minOccurs="0" maxOccurs="unbounded" name="Chunk-Digest">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Digest" type="xs:base64Binary" />
                <xs:element name="Chunk-Number" type="xs:unsignedLong" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element minOccurs="1" maxOccurs="unbounded" name="Server-Info">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="1" maxOccurs="1" name="Server-Base-URI" type="xs:anyURI" />
                <xs:element minOccurs="0" maxOccurs="1" name="Available-Chunk-List"
type="cds:Available-Chunk-ListType" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="Available-Chunk-ListType">
    <xs:list itemType="xs:positiveInteger" />
  </xs:simpleType>
```


Download-Session-General-Par...
Service-Provider-Domain
Download-Session-ID
Download-Session-Version
Content-Item-Format
Download-Session-Time-Inform...
Reception-Reporting
cds:Download-Session-General-ParametersType
Multicast-Download-SessionType
Service-Provider-Domain
Download-Session-ID
Download-Session-Version
Content-Item-Format
Download-Session-Time-Inform...
Reception-Reporting
File-Reference
0..∞
IP-Multicast-Address-Source
Transport-Session-Identifier
Number-Of-Channels
Channel
1..∞
FEC-Encoding-ID
Recovery
Distribution-Of-Requests-Over-...
Server-URI
1..∞
File-InfoType
File-Reference
File-Content-Typ
File-Length
File-Digest

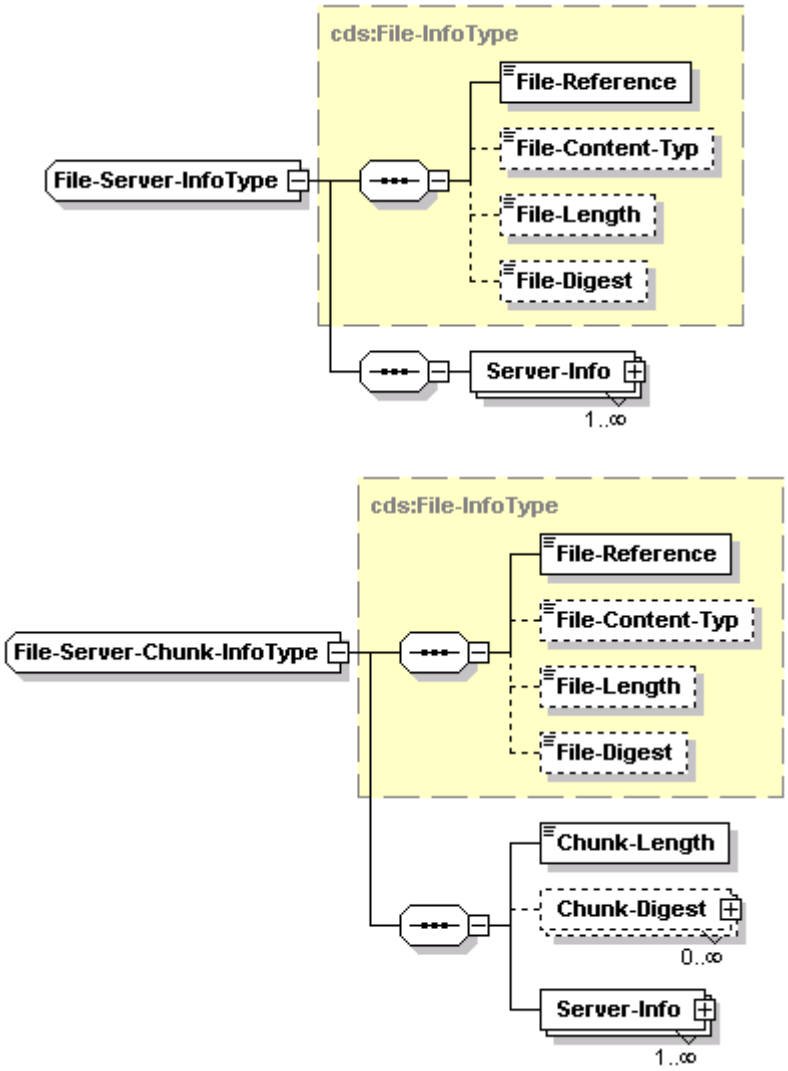


Figure C.34: Download session schema types

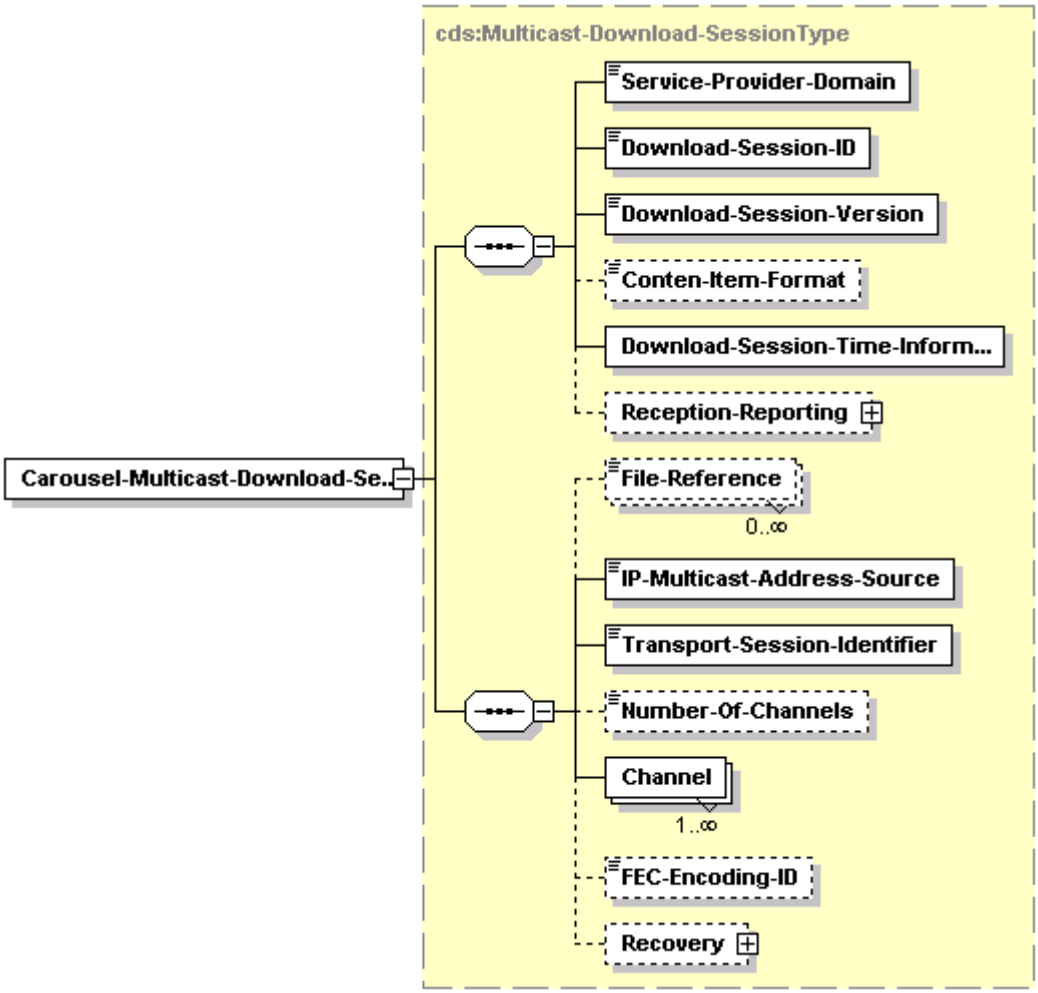


Figure C.35: Carousel Multicast Download session schema

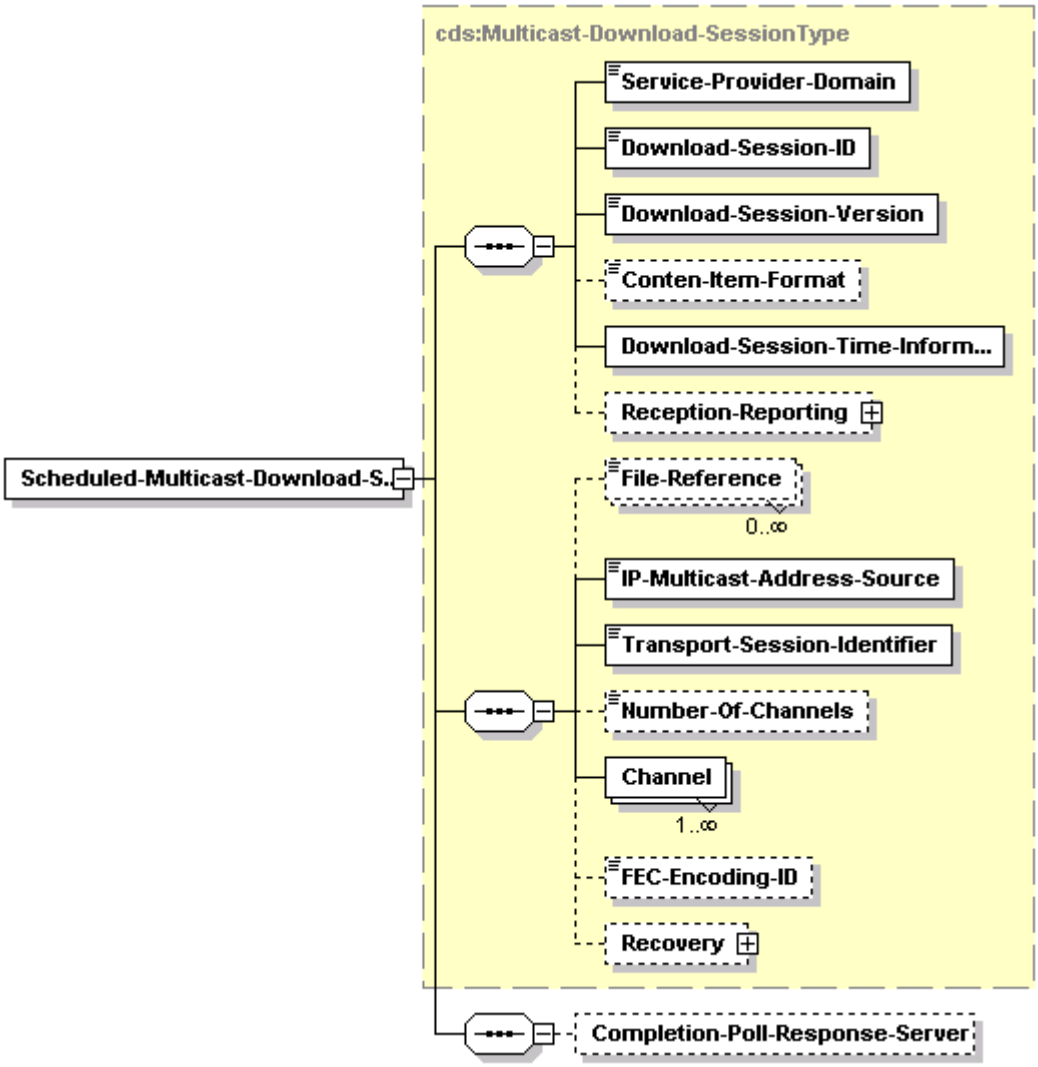


Figure C.36: Scheduled Multicast Download session schema

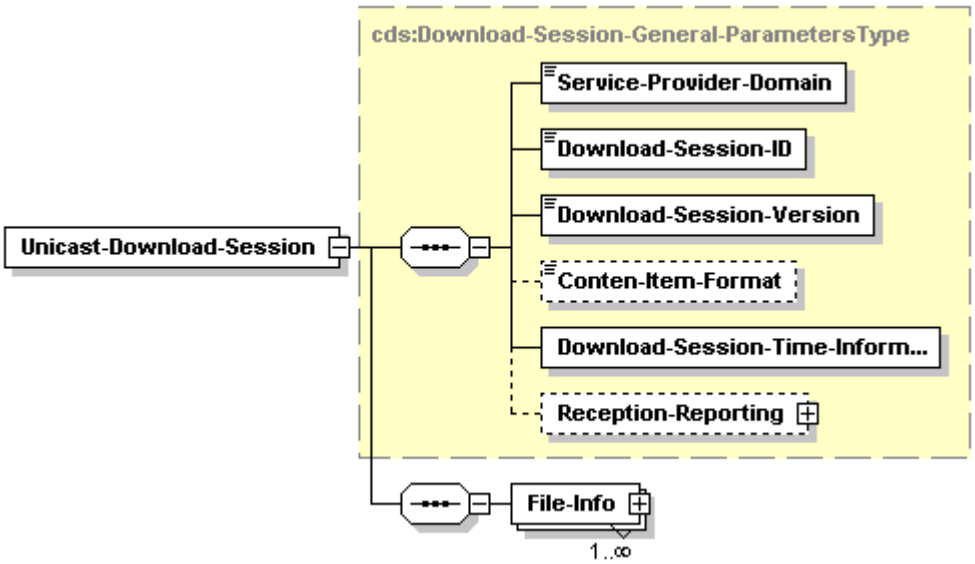


Figure C.37: Unicast Download session schema

C.2.4 Reception reporting message

```
  <xs:element name="Content-Item-Reception-Report">
    <xs:complexType>
      <xs:complexContent mixed="false">
        <xs:extension base="cds:Basic-Reception-Report-Type">
          <xs:sequence minOccurs="1" maxOccurs="1">
            <xs:element minOccurs="1" maxOccurs="unbounded" name="File">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="File-URI" type="xs:anyURI" />
                  <xs:element name="Download-Action">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="download" />
                        <xs:enumeration value="skipped" />
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="File-Reception-Report">
    <xs:complexType>
      <xs:complexContent mixed="false">
        <xs:extension base="cds:Basic-Reception-Report-Type">
          <xs:sequence minOccurs="1" maxOccurs="1">
            <xs:element minOccurs="1" maxOccurs="unbounded" name="File">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="File-URI" type="xs:anyURI" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Chunk-Reception-Report">
    <xs:complexType>
      <xs:complexContent mixed="false">
        <xs:extension base="cds:Basic-Reception-Report-Type">
          <xs:sequence minOccurs="1" maxOccurs="1">
            <xs:element minOccurs="1" maxOccurs="unbounded" name="File">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="File-URI" type="xs:anyURI" />
                  <xs:element minOccurs="1" maxOccurs="unbounded" name="Byte-Range"
type="cds:ByteRange" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="Basic-Reception-Report-Type">
    <xs:sequence>
      <xs:element name="Client-ID" type="xs:string" />
      <xs:element name="Push-Action" type="xs:boolean" />
      <xs:element name="CRID" type="xs:anyURI" />
      <xs:element name="Content-Version" type="xs:unsignedByte" />
      <xs:element name="Service-Provider-Domain" type="sdns:DomainType" />
      <xs:element name="Download-Session_ID" type="xs:nonNegativeInteger" />
      <xs:element name="Download-Session-Version" type="xs:nonNegativeInteger" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ByteRange">
    <xs:attribute name="first-byte-pos" type="xs:unsignedLong" use="required" />
    <xs:attribute name="last-byte-pos" type="xs:unsignedLong" use="required" />
  </xs:complexType>
```

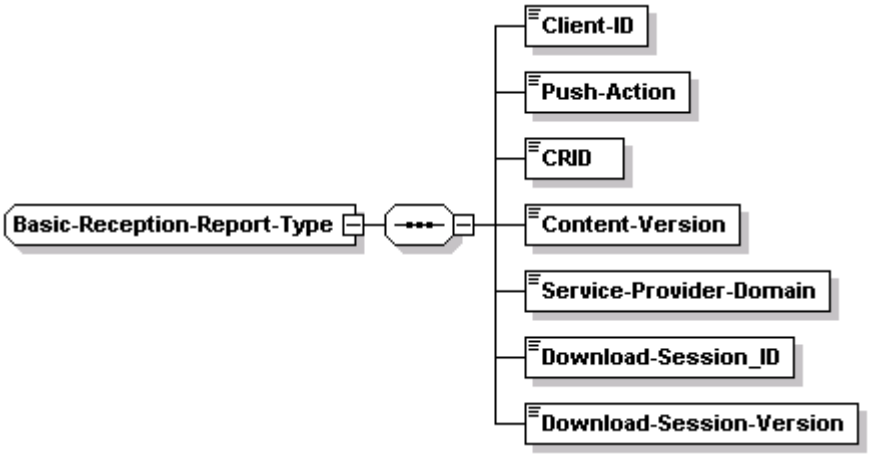
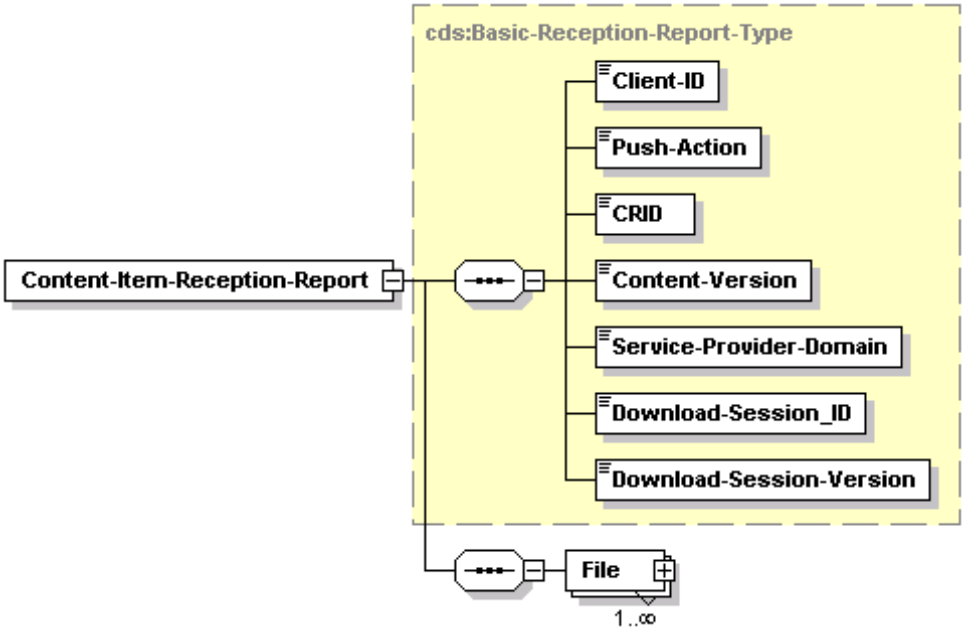
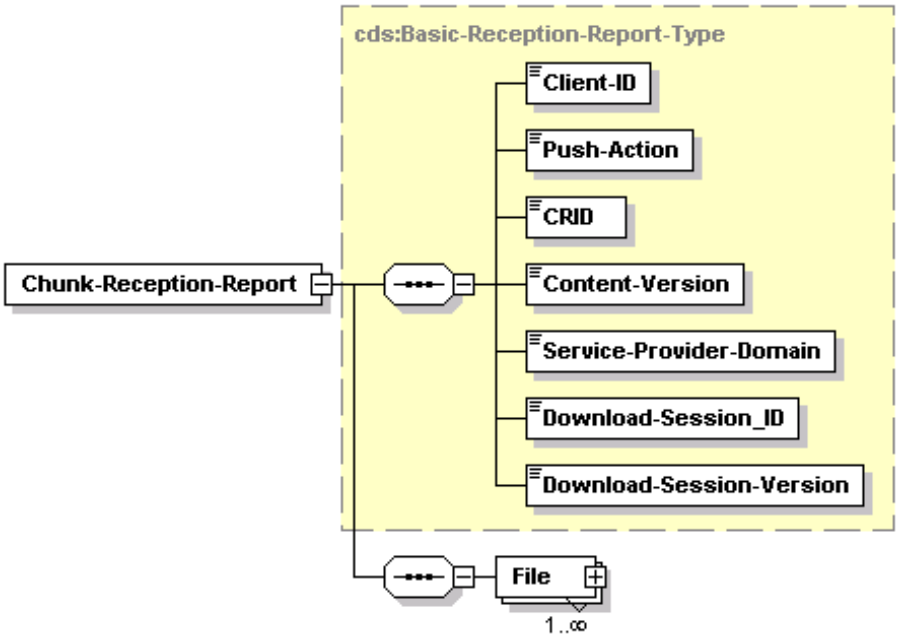


Figure C.38: Reception Report schema types





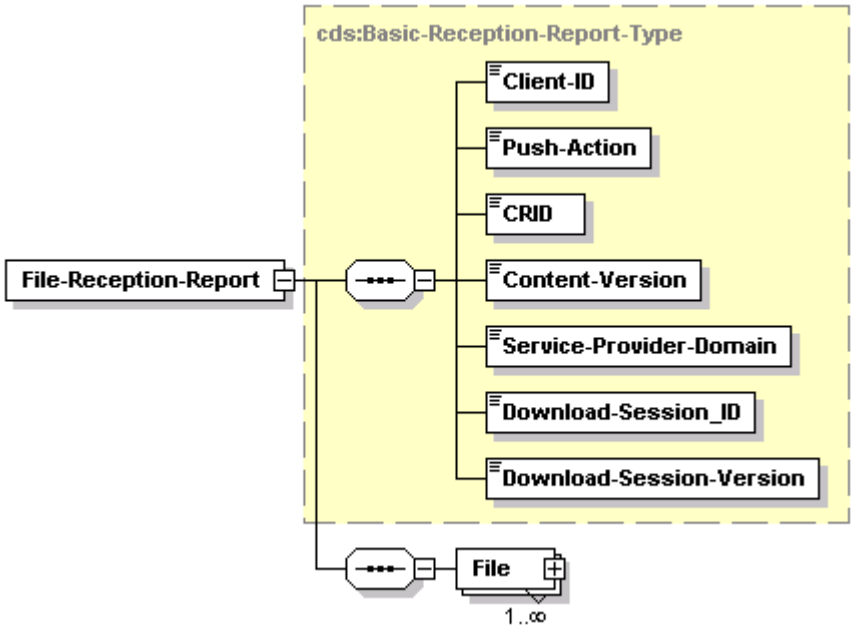


Figure C.39: Reception Report schema

Annex D (informative): Bibliography

- ISO/IEC 15802-3:1998: "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks Common specifications - Part 3: Media Access Control (MAC) Bridges".
- IETF RFC 2597: "Assured Forwarding PHB Group".
- IETF RFC 3246: "An Expedited Forwarding PHB (Per-Hop Behavior)".
- IETF RFC 3454: "Preparation of Internationalized Strings ("stringprep")".
- IETF RFC 1213: "Management Information Base for Network Management of TCP/IP-based internets: MIB-II".
- IETF RFC 2011: "SNMPv2 Management Information Base for the Internet Protocol using SMIv2".
- IETF RFC 2013: "SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2".
- IETF RFC 2863: "The Interfaces Group MIB".
- ISO 8601: "Data elements and interchange formats - Information interchange - Representation of dates and times".
- ISO/IEC 13818-2 (1996): "Information technology - Generic coding of moving pictures and associated audio information: Video".
- ISO/IEC 13818-3 (1998): "Information technology - Generic coding of moving pictures and associated audio information - Part 3: Audio".
- IETF RFC 3208: "PGM Reliable Transport Protocol Specification".
- draft-begen-fecframe-interleaved-fec-scheme-00 (July 2008): "1-D Interleaved Parity FEC Scheme for FEC Framework".
- draft-watson-fecframe-raptor-00 (July 2008): "Raptor FEC Schemes for FECFRAME".

Annex E (normative): Application Layer Forward Error Correction

E.1 Introduction

This annex defines an optional protocol for Application Layer FEC (AL-FEC) protection of streaming media for DVB-IPTV services carried over RTP transport. This AL-FEC protocol is a layered protocol based on a combination of the following two forward error correction codes:

- a simple packet-based interleaved parity code, equivalent to a subset of the code defined in [67];
- the Raptor code, as defined in [65] and [66].

Note that the code defined in [67] is only applicable to the case of media carried within a single RTP flow. In this case, FEC repair packets may be sent in one (or more) layers, the first layer containing packets generated by the interleaved parity code and the optional second and subsequent layers containing packets generated by the Raptor code. Receivers process only packets from the layer or layers they support. A key property of the code defined in the present document is that simultaneous support of multiple layers is possible and FEC packets from these multiple layers can be combined at the receiver to achieve error correction performance which is better than any single layer alone.

Clause E.3 defines the first layer, based on [67].

Clause E.4 defines the subsequent layers, based on [65] and [66].

Clause E.5 describes hybrid decoding procedures which can make use of packets from all layers of the code.

Finally, clause E.6 defines complete FEC protocols for multicast and unicast video with both MPEG-2 Transport Stream encapsulation and direct transport of audio and video over RTP, constructed using the components described in the previous clauses.

E.2 Terms and Acronyms

Table E.1: Terms and Acronyms

Term/Acronym	Definition/Description
Bundle	Collection of Streams (a.k.a. Flows) that are collected into a single Source Block, and used to generate a single stream of Repair Symbols. For example, a low-bitrate audio stream might be bundled with a high-bitrate stream, providing better FEC protection than if it had not been bundled.
Flow	Another term for "Stream", used in the context of Bundles.
Intermediate Block	A block of data derived from the original Source Block data in the case of Raptor Encoder or the combination of Received Source Symbols and Repair Symbols in the case of Raptor Decoder.
Repair Symbol	A Symbol generated by the Raptor Encoder that is derived from Source Symbols.
Source Block	A block of source data over which the Raptor Encoder provides FEC repair information.
Source Symbol	The unit of data from a Source Block. All Source Symbols within a source block are the same size.
FEC	Forward Error Correction.
Encoding Symbol	A source symbol or a repair symbol.
Source Packet Information (SPI)	Information included in a source block related to or from a source packet.

Term/Acronym	Definition/Description
FEC Streaming Configuration Information	Information which controls the operation of the FEC Streaming Framework.
FEC Payload ID	See [48].
Source FEC Payload ID	See [48].
Repair FEC Payload ID	See [48].
FEC Object Transmission Information	See [48].
FEC Encoding ID	See [48].
Content Delivery Protocol	See [48].

E.3 SMPTE 2022-1-based code

SMPTE 2022-1 [67] based coding MAY be applied for streams which meet the requirements of SMPTE 2022-2 [68].

All requirements of [67] and [68] shall apply, with the modifications and exceptions as shown in table E.2. Modifications/exceptions are classified as follows:

(R) Additional requirement (normative).

(E) Exception (normative).

(N) Note (informative).

Table E.2: Modification/exceptions to [67] and [68]

Clause from [67] and [77]	Modification/exception
[67] 7.1 RTP/UDP/IP Layer	(E) The SSRC of the source stream shall be chosen randomly (with collision detection) per the requirements of [21]. RTCP CNAME field SHOULD be used to associate the FEC streams with the source streams.
[67] 7.1 RTP/UDP/IP Layer	(E) The initial value of the sequence number for the source stream shall be random per [21].
[67] 7.1 RTP/UDP/IP Layer	E) The source packets have a zero CC field.
[67] 8 FEC Scheme	(N) The term "FEC Scheme" used here does not have the same meaning as "FEC Scheme" in the present document or in [65].
[67] 8.1 FEC Packet Arrangement	(E) When used with multiple layers, then the L × D block of packets protected by one or more FEC packets SHALL be wholly contained within a single source block of the Raptor code.
[67] 8.1 FEC Packet Arrangement	(E) Only the first (interleaved) FEC stream shall be supported.
[67] 8.3 FEC Packet RTP Header Format	(E) The FEC stream should use the PT number specified in SD&S which defaults to 96, the same value as in the SMPTE 2022 specification.
[67] 8.3 FEC Packet RTP Header Format	(E) The SSRC of the FEC stream shall be 0. RTCP CNAME field should be used to associate the FEC streams with the source streams.
[67] 8.3 FEC Packet RTP Header Format	(E) The initial value of the sequence number for the FEC stream should be random per [21] and it must be one higher than the sequence number in the previously transmitted FEC packet.
[67] 7.1 (or [67] 8.2) FEC buffer Overhead and Latency Implications	(E) The limits defined in this clause SHALL NOT apply. Receivers SHALL support values of L and D within the restrictions L×D <= 400 and L <= 40 (L is the length of burst protection - in packets). Receivers MAY also support values of L and D outside this range.
[67] 8.4 FEC Header Format	(R) The D bit SHALL be set to 0.
[67] 8.4 FEC Header Format	(R) The SNBase ext bits SHALL be ignored by receivers.
[67] 8.5 FEC Traffic Shaping Issues	(E) The requirements of this clause SHALL NOT apply.
[67] 8.6 Reorder Tolerance	(E) The requirement for reordering capability of minimum 10 packets before applying FEC does not apply.
[67] annex B Non Block lAigned FEC Arrangement	(E) The sending arrangement described in this annex SHALL NOT be used.

NOTE: The DVB AL-FEC base layer conforms to [67] with the restrictions/exceptions listed in table E.2.

E.4 Raptor code

E.4.1 Introduction

The FEC Building Block [48] defined by the IETF Reliable Multicast working group describes an approach to the specification of protocols using FEC but separates the definition of the protocol from the specification of the FEC code itself. In the language of the FEC Building Block, separate specifications are provided for "Content Delivery Protocols" and for "FEC Schemes", the former defining the protocols and the latter defining the actual FEC codes. The FEC Building Block describes rules that both kinds of specification shall follow so that they can be used together and so it provides the "glue" between Content Delivery Protocols and FEC Schemes.

Following this approach, this clause is organized as a number of modular components. These are then combined to form complete protocols suitable for the DVB-IPTV services. These components include:

- An FEC Streaming Framework, equivalent to that defined in TS 126 346 [65], which provides an overall protocol framework for the application of FEC to media streams. This is described in clause E.4.2.
- A number of FEC Schemes, which define protocol components according to the IETF FEC Building Block [48] suitable for various classes of application and which define how the Raptor FEC code is applied for streaming applications. These are defined in clause E.4.3.

Complete protocol specifications for multicast and unicast video with both MPEG-2 Transport Stream encapsulation and direct transport of audio and video encapsulated in RTP are then described in clause E.5. In both cases, the construction is based on the building blocks described above.

E.4.2 FEC Streaming Framework

E.4.2.1 Introduction

This clause defines a framework for the definition of CDPs, in the sense of the FEC Building Block, which provides for FEC protection of streamed data flows over UDP. This clause does not define a complete Content Delivery Protocol, but rather defines only those aspects that are expected to be common to all Content Delivery Protocols that support streaming data over UDP.

The framework defined in this clause is not specific to a single streaming application protocol. The framework provides FEC protection for application protocol flows over UDP and for combined protection of multiple such flows. For example, multiple RTP flows may be protected together with the associated RTCP flows and potentially also other related flows such as security protocol packets.

Content Delivery Protocols which use this framework shall provide for communicating two kinds of information from sender to receiver:

- FEC Streaming Configuration Information.
- FEC Object Transmission Information.

FEC Streaming Configuration Information is information independent of the FEC Scheme being used that is needed by the FEC Streaming Framework, e.g. the definition of the UDP flows that are protected by the FEC Streaming Framework. The FEC Streaming Configuration Information is defined in this clause and the means to transport it (for example with Service Discovery Information) shall be defined by each Content Delivery Protocol.

FEC Object Transmission Information is information which is specific to a particular FEC Scheme. The FEC Object Transmission Information is defined by each FEC Scheme. Content Delivery Protocols shall define a means to transport the FEC Object Transmission Information from sender to receiver.

The architecture outlined above is illustrated in figure E.1.

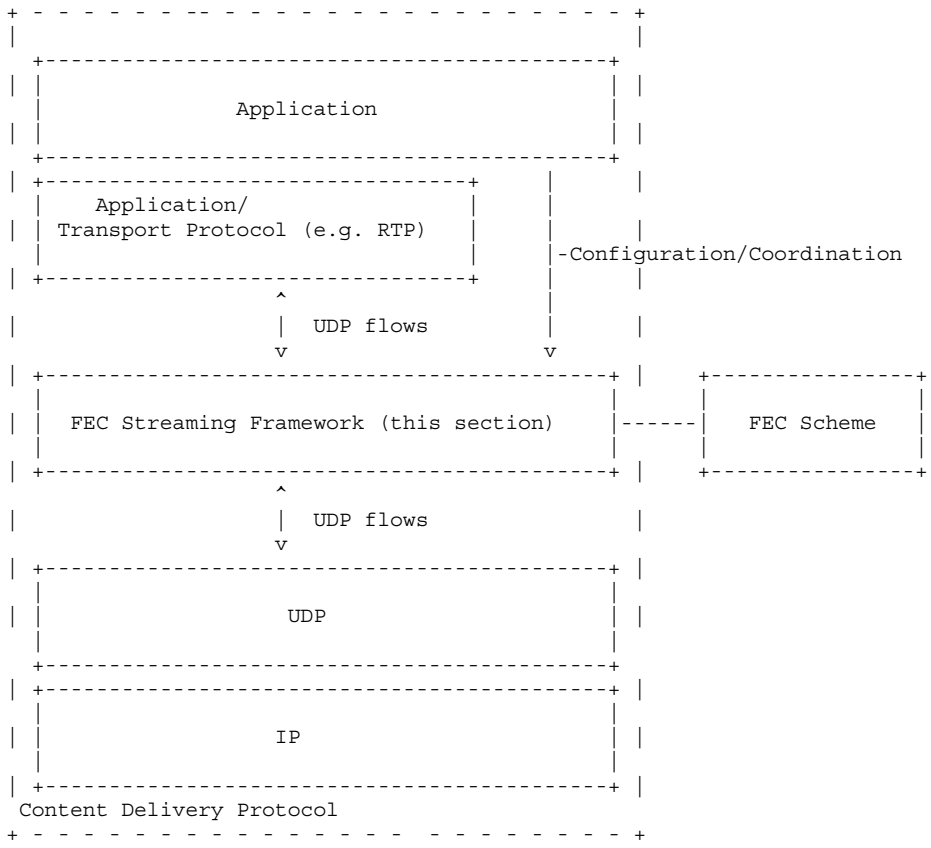


Figure E.1: FEC Streaming Framework Architecture

E.4.2.2 Procedural overview

E.4.2.2.1 General

The mechanism defined in this clause consists of three components:

(i) Construction of a "source block" from source media packets belonging to one or several UDP packet flows. The UDP flows MAY include, for example, RTP and RTCP packets and also other protocols related to the stream.

(ii) Optional extension of source packets to indicate the source block and the position within the source block occupied by the data from and related to the source packet.

(iii) Definition of repair packets, sent over UDP, which can be used by the FEC decoder to reconstruct missing portions of the source block.

The protected data may be from several different UDP flows that are protected jointly. In general, multiple source blocks will be constructed for a stream; each source block is constructed from different sets of source packets. For example, each source block may be constructed from those source packets related to a particular segment of the stream in time.

A receiver supporting this streaming framework SHALL support the packet format for FEC Source packets and SHALL also support the packet format for FEC Repair packets.

This clause does not define how the sender determines which source packets are included in which source blocks. A specific Content Delivery Protocol MAY define this mapping or it MAY be left as implementation dependent at the sender, possibly including some memory constraints at receivers. However, a CDP specification SHALL define how a sender communicates to the receiver the maximum length of time that the sender will allow between a source packet and a repair packet that protects that source packet.

At the sender, the mechanism processes original UDP packets to create:

(i) A stored copy of the original packets in the form of one or more "source block(s)". The source block is a logical block of data to which the FEC code will subsequently be applied. It is constructed by concatenating "Source Packet Information" (SPI) for each source packet. Generally, the SPI for a packet contains a short identifier for the flow the packet belongs to, a length indicator for the packet, the UDP payload and possible padding bytes.

(ii) FEC Source packets for transmission to the receiver.

The FEC Streaming Framework uses the FEC encoder specified by the FEC Scheme in use to generate the desired quantity of repair symbols from a source block. These repair symbols are then sent using the FEC repair packet format to the receiver. The FEC Repair packets are sent to a UDP destination port different from any of the original UDP packets' destination port(s) as indicated by the FEC Streaming Configuration Information.

The receiver recovers original source packets directly from any FEC Source packets received. The receiver also uses the received FEC Source Packets to construct a stored copy of the original packets in the same source block format as constructed at the sender.

If any FEC Source packets related to a given source block have been lost, then this copy of the source block at the receiver will be incomplete. If sufficient FEC source and FEC Repair packets related to that source block have been received, the FEC Framework may use the FEC decoding algorithm defined by the FEC Scheme to recover a (hopefully, but not necessarily, complete) copy of the source block. The SPI for the missing source packets can then be extracted from the completed parts of the source block and used to reconstruct the source packets to be passed to the application.

The receiver of FEC Source packets SHALL be able to identify the source block and the position within the source block occupied by the SPI derived from each packet. This information is known as FEC Source Packet Identification Information and may be communicated in several ways. The FEC Source Packet Identification Information MAY be encoded into a specific field within the FEC Source packet format defined in this Annex, called the Source FEC Payload ID field. The exact contents and format of the Source FEC Payload ID field are defined by the FEC Scheme. Alternatively, the FEC Scheme or CDP MAY define how the FEC Source Packet Identification Information is derived from other fields within the source packets. This clause defines the way that the Source FEC Payload ID field, if used, is appended to source packets to form FEC Source packets.

The receiver of FEC Repair packets SHALL also be able to identify the source block and the relationship between the contained repair data and the original source block. This information is known as FEC Repair Packet Identification information. This information SHALL be encoded into a specific field, the Repair FEC Payload ID field, the contents and format of which are defined by the FEC Scheme.

Any FEC Schemes to be used in conjunction with this framework SHALL be a systematic FEC Scheme and SHALL be based on source blocks. The FEC Scheme MAY define different FEC Payload ID field formats for FEC Source packets and FEC Repair packets.

E.4.2.2.2 Sender Operation

It is assumed that the sender has constructed or received original data packets for the session. These may be RTP, RTCP or other UDP packets. The following operations describe a possible way to generate compliant FEC Source packet and FEC repair packet streams:

1) A source block is constructed as specified in **clause E.4.2.3.2**, by concatenating the SPI for each original source packet. In doing so, the Source FEC Packet Identification Information of the FEC Source packet can be determined and included in the Source FEC Payload ID field, if used. In the SPI the identity of the packet's UDP flow is marked using a short "UDP flow ID", defined in this Annex. The association of UDP flow specifications to UDP flow IDs is defined by the FEC Streaming Configuration Information.

2) The FEC Source packet is constructed according to **clause E.4.2.3.3**. The identity of the original flow is maintained by the source packet through the use of the same UDP ports and IP addresses which have been advertised by the Content Delivery Protocol (for example using DVB Service Discovery), as carrying FEC Source packets generated from an original stream of a particular protocol (e.g. RTP, RTCP, etc.). The FEC Source packet generated is sent according to normal UDP procedures.

3) The FEC encoder generates repair symbols from a source block and the FEC Streaming Framework places these symbols into FEC Repair packets, to be conveyed to the receiver(s). These repair packets are sent using normal UDP procedures to a unique destination port to separate them from any of the source packet flows. The ports to be used for FEC Repair packets are defined in the FEC Streaming Configuration Information.

E.4.2.2.3 Receiver Operation

The following describes a possible receiver algorithm, when receiving an FEC source or repair packet:

1) If an FEC Source packet is received (as indicated by the UDP flow on which was received):

   a) The original source packet is reconstructed by removing the Source FEC Payload ID, if used. The resulting packet MAY be buffered to allow time for the FEC repair.

   b) The Source FEC Packet Identification Information is determined, either from the Source FEC Payload ID, if used, or by other means.

   c) The SPI for the resulting packet is placed into the source block according to the Source FEC Packet Identification Information and the source block format described in **clause E.4.2.3.2**. The IP addresses and UDP ports the packet was received on/sent from are used to determine the UDP flow ID within the SPI.

2) If an FEC Repair packet is received (as indicated by the UDP flow on which it was received), the contained repair symbols are associated with a source block according to the Repair FEC Payload ID.

3) If at least one source packet is missing and at least one repair packet has been received for a source block then FEC decoding may be desirable. The FEC decoder determines if the source block constructed in step 1 plus the associated repair symbols received in step 2 contains enough symbols for decoding of any or all of the missing source symbols in the source block and, if so, performs a decoding operation.

4) Any SPI that was reconstructed during the decoding operation is then used to reconstruct the missing source packets and these are buffered as normal received source packets (see step 1a above).

NOTE: The above procedure may result in a situation in which not all original source packets are recovered.

E.4.2.3 Protocol Specification

E.4.2.3.1 General

This clause specifies the protocol elements for the FEC Streaming Framework. The protocol consists of three components which are described in the following sections:

1) Construction of a source block from source packets. The FEC code will be applied to this source block to produce the repair data.

2) A format for packets containing source data.

3) A format for packets containing repair data.

The operation of the FEC Streaming Framework is governed by certain FEC Streaming Configuration Information. This configuration information is also defined in this clause. A complete protocol specification that uses this framework SHALL specify the means to determine and communicate this information between sender and receiver.

E.4.2.3.2 Structure of Source Block

This clause defines the layout of the source block. A source block consists of the concatenation of SPI for at least one original source UDP packet.

Let:

n	be the number of UDP packets in the source block. n MAY be determined dynamically during the source block construction process.
T	be the source symbol size in bytes. Note: this information is provided by the FEC Scheme as defined in clause E.4.2.3.6.
i	the index to the (i+1)-*th* UDP packet to be added to the source block, $0 <= i < n$.
$R[i]$	denote the number of octets of the UDP payload of the i-th UDP packet.
$l[i]$	be a length indication associated with the i-th UDP packet - the nature of the length indication is defined by the FEC Scheme
$L[i]$	denote two octets representing the value of $l[i]$ in network byte order (high order octet first) of the i-th UDP packet.
$f[i]$	denote an integer "UDP flow ID" identifying the UDP flow from which the i-th packet was taken
$F[i]$	denote a single octet representing the value of $f[i]$
$s[i]$	be the smallest integer such that $s[i]\times T >= (l[i]+3)$. Note $s[i]$ is the length of $SPI[i]$ in units of symbols of size T bytes.
$P[i]$	denote $s[i]\times T-(l[i]+3)$ zero octets.

NOTE: $P[i]$ are padding octets to align the start of each UDP packet with the start of a symbol.

$SPI[i]$	be the concatenation of $F[i]$,$L[i]$, $R[i]$ and $P[i]$.

Then, the source block is constructed by concatenating $SPI[i]$ for i = 0, 1, 2, ... n-1. The source block size, S, is then given by sum $\{s[i]\times T, i=0, ..., n-1\}$.

Source blocks are identified by integer SBNs and symbols within a source block by integer ESIs. This clause does not specify how SBNs are allocated to source blocks. Symbols are numbered consecutively starting from zero within the source block. Each source packet is associated with the ESI of the first symbol containing SPI for that packet. Thus, the ESI value associated with the j-th source packet, ESI[j], is given by:

$$\text{ESI}[j] = 0, \text{ for } j=0 \quad \text{(E.1)}$$

$$\text{ESI}[j] = \text{sum}\{s[i], i=0,...,(j-1)\}, \text{ for } 0 < j < n \quad \text{(E.2)}$$

The Source FEC Packet Identification Information consists of the identity of the source block and the ESI associated with the packet.

A UDP flow is uniquely defined by an IP source and destination address and UDP source and destination port values. The assignment of UDP flow ID values to UDP flows is part of the FEC Streaming Configuration Information.

E.4.2.3.3 Packet format for FEC Source packets

The packet format for FEC Source packets SHALL be used to transport the payload of an original source UDP packet. As depicted in figure E.2, it consists of the original UDP packet, followed, optionally, by the Source FEC Payload ID field, if used.

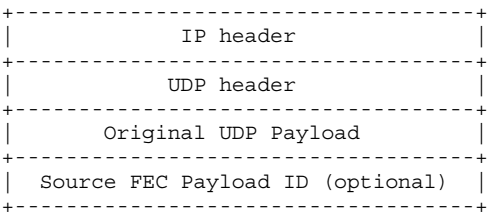


Figure E.2: Structure of FEC Source Packets

The IP and UDP header fields SHALL be identical to those of the original source packet. The Original UDP Payload field SHALL be identical to the UDP payload of the original source packet. The UDP payload of the FEC Source packet SHALL consist of the Original UDP Payload followed by the Source FEC Payload ID field.

The Source FEC Payload ID field, if present, contains information required for the operation of the FEC algorithm, in particular for the derivation of the Source FEC Packet Identification Information. The format of the Source FEC Payload ID and the derivation of the Source FEC Packet Identification Information are defined by the FEC Scheme. Note that the FEC Scheme or CDP may define a means to derive the Source FEC Packet Identification Information from other information in the source packet (for example the RTP Sequence number).In this case the Source FEC Payload ID field described here is not appended to the packet and the Source FEC packet is identical in every way to the original Source packet.

E.4.2.3.4 Packet Format for FEC Repair packets

The packet format for FEC Repair packets is shown in figure E.3. The UDP payload consists of a Repair FEC Payload ID field and one or more repair symbols generated by the FEC encoding process.

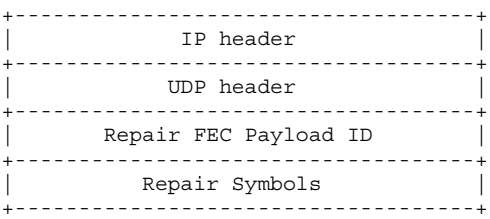


Figure E.3: FEC Repair packet format

The Repair FEC Payload ID field contains information required for the operation of the FEC algorithm. This information is defined by the FEC Scheme. The format of the Repair FEC Payload ID field is defined by the FEC Scheme.

Any number of whole repair symbols may be contained within an FEC Repair packet, subject to packet size restrictions or other restrictions defined by the FEC Scheme. The number of repair symbols within a packet can be determined from the symbol length and the packet length. Partial repair symbols SHALL NOT be included in FEC repair packets.

E.4.2.3.5 FEC Streaming Configuration Information

The FEC Streaming Configuration Information is information that the FEC Streaming Framework needs in order to apply FEC protection to the UDP flows. A complete Content Delivery Protocol specification for streaming that uses the framework specified here SHALL include details of how this information is derived and communicated between sender and receiver.

The FEC Streaming Configuration Information includes identification of a number of UDP packet flows. Each UDP packet flow is uniquely identified by a tuple { Source IP Address, Destination IP Address, Source UDP port, Destination UDP port }.

A single instance of the FEC Streaming Framework provides FEC protection for all packets of a specified set of source UDP packet flows, by means of one or more UDP packet flows containing repair packets. The FEC Streaming Configuration Information includes, for each instance of the FEC Streaming Framework:

1) Identification of the UDP packet flow(s) carrying FEC Repair packets, known as the FEC repair flow(s).

2) For each source UDP packet flow protected by the FEC repair flow(s):

   a) Identification of the UDP packet flow carrying source packets.

   b) An integer identifier, between 0 and 255, for this flow. This identifier SHALL be unique amongst all source UDP packet flows which are protected by the same FEC repair flow.

3) The FEC Scheme that is to be applied.

Multiple instances of the FEC Streaming Framework, with separate and independent FEC Streaming Configuration Information, may be present at a sender or receiver. A single instance of the FEC Streaming Framework protects all packets of all the source UDP packet flows identified in (2) above i.e. all packets on those flows SHALL be FEC Source packets as defined in **clause E.4.2.3.3**. A single source UDP packet flow SHALL NOT be protected by more than one FEC-SF instance.

A single FEC repair flow provides repair packets for a single instance of the FEC-SF. Other packets SHALL NOT be sent within this flow i.e. all packets in the FEC repair flow SHALL be FEC repair packets as defined in **clause E.4.2.3.4** and SHALL relate to the same FEC Streaming Framework instance.

The FEC Streaming Framework SHALL be informed of the symbol size to be used for each source block. This information MAY be included in the FEC Streaming Configuration Information or it MAY be communicated by other means, for example within the FEC Repair Payload ID field. A complete Content Delivery Protocol specification SHALL specify how this information is communicated between sender and receiver.

E.4.2.3.6 FEC Scheme requirements

In order to be used with this framework, an FEC Scheme SHALL:

- adhere to the requirements of RFC 5052 [48];
- be systematic;
- be based on source blocks which are non-overlapping and contiguous within the stream;
- specify how the SBN and ESI associated with a source packet are derived or communicated from sender to receiver (for example, within the Source FEC Payload ID field);
- specify how the symbol length is derived or communicated from sender to receiver (for example, as part of the FEC Object Transmission Information);
- specify how the length indication, $l[i]$, included in the Source Packet Information, is derived from a UDP packet;
- specify how the Source Packet Information length, $s[i]$, is derived from a UDP packet.

E.4.3 FEC Schemes for streaming

E.4.3.1 Raptor FEC Scheme for arbitrary packet flows

This clause defines an FEC Scheme for Raptor protection of arbitrary packet flows over UDP.

E.4.3.1.1 Formats and Codes

E.4.3.1.1.1 FEC Object Transmission Information

This FEC Object Transmission Information elements for this FEC Scheme and their value ranges are as follows:

Maximum Source Block Length

A non-negative integer less than 2^{16}, in units of symbols.

Encoding Symbol Size

A non-negative integer less than 2^{16}, in units of bytes.

An encoding format for this information in a 4 octet field is defined in figure E.4:

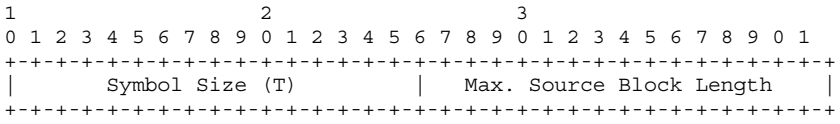


Figure E.4: Encoded Common FEC Object Transmission Information for Raptor FEC Scheme for arbitrary packet flows

E.4.3.1.1.2 FEC Payload ID

E.4.3.1.1.2.1 Source FEC Payload ID

The Source FEC payload ID is composed as follows.

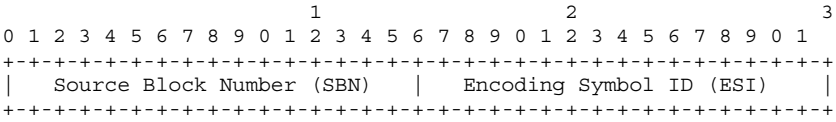


Figure E.5: Source FEC Payload ID format for Raptor FEC Scheme for arbitrary packet flows

Source Block Number (SBN), (16 bits): An integer identifier for the source block that the source data within the packet relates to.

Encoding Symbol ID (ESI), (16 bits): The starting symbol index of the source packet in the source block.

The interpretation of the ESI is defined by the FEC Streaming Framework (see clause E.4.2).

E.4.3.1.1.2.2 Repair FEC Payload ID

The structure of the Repair FEC Payload ID is defined in figure E.6.

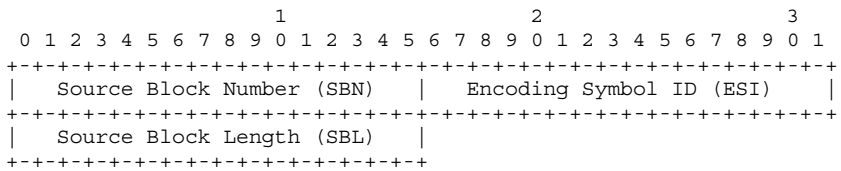


Figure E.6: Repair FEC Payload ID

Source Block Number (SBN), (16 bits): An integer identifier for the source block that the repair symbols within the packet relate to.

Encoding Symbol ID (ESI), (16 bits): Integer identifier for the encoding symbols within the packet.

Source Block Length (SBL), (16 bits): The number of source symbols in the source block.

The interpretation of the SBN, ESI and SBL is defined by the FEC Code Specification.

E.4.3.1.2 Procedures

This FEC Scheme uses the procedures of the framework defined in clause E.4.2.2 to construct a source block to which the FEC code can be applied. The sender SHALL allocate SBNs to source blocks sequentially, wrapping around to zero after SBN 2^{16}-1.

During the construction of the source block as per clause E.4.2.3.2:

- The length indication, $l[i]$, included in the Source Packet Information for each packet shall be the UDP payload length.
- The value of $s[i]$ in the construction of the Source Packet Information for each packet shall be the smallest integer such that $s[i] \times T >= (l[i]+3)$.

E.4.3.1.3 FEC Code specification

The Raptor FEC encoder defined in clause E.7 SHALL be used. The source block passed to the Raptor FEC encoded SHALL consist of the Source Block constructed according to clause E.4.2.3.2 extended with zero or more padding symbols such that the total number of symbols in the source block is equal to the Maximum SBL signaled in the FEC Object Transmission Information. Thus the value of the parameter K used by the FEC encoded is equal to the Maximum SBL for all blocks of the stream. Padding symbols shall consist entirely of bytes set to the value zero.

The symbol size, T, to be used for source block construction and the repair symbol construction are is equal to the Encoding Symbol Size signaled in the FEC Object Transmission Information. The parameter T shall be set such that the number of source symbols in any source block is at most K_{MAX} = 8 192.

The Maximum SBL parameter - and hence the number of symbols used in the FEC Encoding and Decoding operations - SHALL be set to one of the values specified in clause E.7. Recommended derivation of other parameters is presented in clause E.4.3.1.6.

E.4.3.1.4 Encoding packet construction

As described in clause E.4.2.3.4, each repair packet contains the following information:

- Source Block Number (SBN).
- Encoding Symbol ID (ESI).
- Source Block Length (SBL).
- Repair symbol(s).

The number of repair symbols contained within a repair packet is computed from the packet length. The ESI value placed into a repair packet is given by the following formula:

$$ESI_{repair} = I_{repair} + K \quad \text{(E.3)}$$

Where I_{repair} is the index of the repair symbol in the sequence of repair symbols generated according to clause E.7, where the first repair symbol has index 0, the second index 1 etc. and K is the number of source symbols (equal to the Maximum SBL parameter).

The SBL field of the Repair FEC Payload ID field SHALL be set to the number of symbols included in the Source Packet Information of packets associated with the source block i.e. before padding to the Maximum SBL.

E.4.3.1.5 Transport

This clause describes the information exchange between the Raptor encoder/decoder and any transport protocol making use of Raptor forward error correction for streaming.

The Raptor encoder for streaming requires the following information from the transport protocol for each source block:

- The symbol size, *T*, in bytes.
- The number of symbols in the source block, *K*.
- The Source Block Number (SBN).
- The source symbols to be encoded.

The Raptor encoder supplies the transport protocol with encoding packet information consisting, for each repair packet, of:

- Source Block Number (SBN).
- Encoding Symbol ID (ESI).
- Source Block Length (SBL).
- repair symbol(s).

The transport protocol shall communicate this information transparently to the Raptor decoder.

A suitable transport protocol is defined in the present document.

E.4.3.1.6 Example parameters

E.4.3.1.6.1 Parameter derivation algorithm

This clause provides recommendations for the derivation of the transport parameter *T*. This recommendation is based on the following input parameters:

B	the maximum source block size, in bytes. For further explanation, see below.
A	the symbol alignment factor, in bytes, i.e. symbol size *T* is a multiple of *A*.
P	the maximum repair packet payload size (not including Repair FEC Payload ID), in bytes, which SHALL be multiple of *A*.
K_{MAX}	the maximum number of source symbols per source block. As defined in clause E.7, K_{MAX} = 1 281.
K_{MIN}	a minimum target on the number of symbols per source block.
G_{MAX}	a maximum target number of symbols per repair packet.

A requirement on these inputs is that $\text{ceil}(B/P) \leq K_{MAX}$. Based on the above inputs, the transport parameter *T* is calculated as follows:

Let:

$$G = \min\{\text{ceil}(P \cdot K_{MIN}/B), P/A, G_{MAX}\} \quad \text{(E.4)}$$

- the approximate number of symbols per packet

$$T = \text{floor}(P/(A \cdot G)) \cdot A \quad \text{(E.5)}$$

The value of *T* derived above should be considered as a guide to the actual value of *T* used. It may be advantageous to ensure that *T* divides into *P*, or it may be advantageous to set the value of *T* smaller to minimize wastage when full size repair symbols are used to recover partial source symbols at the end of lost source packets (as long as the maximum number of source symbols in a source block does not exceed K_{MAX}). Furthermore, the choice of *T* may depend on the source packet size distribution, e.g., if all source packets are the same size then it is advantageous to choose *T* so that the actual payload size of a repair packet *P"*, where *P"* is a multiple of *T*, is equal to (or as few bytes as possible larger than) the number of bytes each source packet occupies in the source block.

Recommended settings for the input parameters, *A*, K_{MIN} and G_{MAX} are as follows:

$A = 16$ $K_{MIN} = 640$ $G_{MAX} = 10$

E.4.3.1.6.2 Examples

The above algorithm leads to transport parameters as shown in table E.3, assuming the recommended values for *A*, K_{MIN} and G_{MAX} and $P = 1\ 424$.

Table E.3: Example parameters settings

Max source block size *B*	*G*	Symbol size *T*	*G·T*
16 KB	10	128	1 280
32 KB	10	128	1 280
128 KB	7	192	1 344
256 KB	4	352	1 408

E.4.3.2 Raptor FEC Scheme for a single sequenced packet flow

This clause defines an FEC Scheme for FEC protection of a single packet flow in which source packets each carry a unique sequence number. We call such a packet flow a "sequenced flow". A primary example would be FEC protection of an RTP flow containing an MPEG-2 Transport Stream within which all data for the service is multiplexed. In this case the RTP Sequence Numbers can be used to derive the Source FEC Packet Identification Information.

Compared to the FEC Scheme defined in clause E.4.3.1, the primary advantage of this scheme is that it does not modify source packets in any way. As a result this FEC scheme can be used in the presence of legacy equipment which would not recognize source packets which had been modified according to the schemes defined in clause E.4.3.1.

In this FEC Scheme, the role played by the Source FEC Payload ID in the scheme of clause E.4.3.1 is replaced by the sequence number. The sequence numbers of packets within each flow to be protected SHALL be incremented by one for each packet in the stream.

The size of the Source Packet Information within a given Source Block for each packet within a given sequenced flow SHALL be the same and is derived from the size of the FEC Repair packets, which SHALL also all be the same size for a given source block.

E.4.3.2.1 Formats and Codes

E.4.3.2.1.1 FEC Object Transmission Information

See clause E.4.3.1.1.1.

E.4.3.2.1.2 FEC Payload ID

E.4.3.2.1.2.1 Source FEC Payload ID

The Source FEC Payload ID field is not used by this FEC Scheme. Source packets are not modified by this FEC Scheme.

E.4.3.2.1.2.2 Repair FEC Payload ID

The Repair FEC Payload ID for this FEC scheme consists of two parts:

- an optional RTP header for the DVB AL-FEC enhancement layer;
- a Repair FEC Payload ID Field for the DVB AL-FEC enhancement layer.

The RTP header shall only be added if the SD&S flag FECEnhancementLayer@TransportProtocol is present for this FEC enhancement layer and the SD&S flag FECEnhancementLayer@TransportProtocol signals RTP/AVP as transport protocol. If the SD&S flag FECEnhancementLayer@TransportProtocol is not present or the SD&S flag FECEnhancementLayer@TransportProtocol signals UDP/FEC, the RTP header shall not be added.

An HNED shall only join/request an AL-FEC enhancement layer if it supports the reception of the signalled Repair FEC Payload ID.

All the fields in the RTP header of DVB AL-FEC packets are used according to RFC 3550 [21], with for some of them further clarification as follows.

Marker bit: The marker bit shall be set 1 for the last protection RTP packet sent for each source block, and otherwise set to 0.

Timestamp: The timestamp rate shall be 10 kHz and shall be set to a time corresponding to the packet's transmission time. The timestamp value has no use in the actual FEC protection process and is only set to a value to produce reasonable resolution for arrival measuring and jitter calculation.

Sequence number: Is set in accordance with RFC 3550 [21]. The sequence number is primarily used to detect losses of the protection RTP packets.

Payload type (PT): If SD&S is used for service discovery, it is dynamically allocated using FECEnhancementLayer@PayloadTypeNumber in the SD&S. If the SD&S attribute FECEnhancementLayer@PayloadTypeNumber is not present the transmitter may use any dynamic payload number between 96 and 128 and the receiver shall ignore this PT field.

SSRC: One SSRC is used per source SSRC. The SSRC used by the protection payload format shall be different the one used by the source RTP packets. The binding of the source SSRC to the repair SSRC shall be performed using the RTCP SDES CNAME, which shall be identical for the two SSRCs.

The Repair FEC Payload ID Field format for this FEC Scheme is shown in figure E.7.

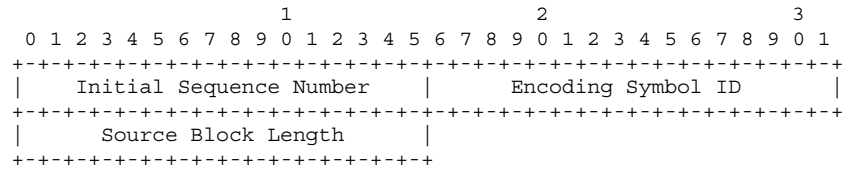


Figure E.7: Repair FEC Payload ID Field format

Initial Sequence Number (Flow *i* ISN) - 16 bits

This field specifies the lowest 16 bits of the sequence number of the first packet to be included in this sub-block. If the sequence numbers are shorter than 16 bits then the received Sequence Number SHALL be logically padded with zero bits to become 16 bits in length respectively.

Encoding Symbol ID (ESI) - 16 bits

This field indicates which repair symbols are contained within this repair packet. The ESI provided is the ESI of the first repair symbol in the packet.

Source Block Length (SBL) - 16 bits

This field specifies the length of the source block in symbols.

E.4.3.2.2 Procedures

This FEC Scheme uses the procedures of the framework defined in clause E.4.2 to construct a source block to which the FEC code can be applied. The sender SHALL allocate SBNs to source blocks sequentially, wrapping around to zero after SBN 2^{16}-1.

During the construction of the source block as per clause E.4.2.3.2:

- The length indication, $l[i]$, included in the Source Packet Information for each packet shall be dependent on the protocol that is carried. Rules for RTP are specified below in clause E.4.3.2.2.3.
- The value of $s[i]$ in the construction of the Source Packet Information for each packet shall be equal to the number of repair symbols placed in each repair packet, which SHALL be the same for all repair packets of a block.

E.4.3.2.2.1 Derivation of Source FEC Packet Identification Information

The Source FEC Packet Identification Information for a source packet is derived from the sequence number of the packet and information received in any Repair FEC packet belonging to this Source Block. Source blocks are identified by the sequence number of the first source packet in the block. This information is signaled in all Repair FEC packets associated with the source block in the ISN field.

The length of the Source Packet Information (in bytes) for source packets within a source block is equal to length of the payload containing encoding symbols of the repair packets (i.e. not including the Repair FEC Payload ID) for that block, which SHALL be the same for all repair packets. The Source Packet Information Length (*SPIL*) in symbols is equal to this length divided by the Encoding Symbol Size (which is signaled in the FEC Object Transmission Information).

The set of source packets which are included in the source block is determined from the ISN and SBL as follows:

Let:

I be the Initial Sequence Number of the source block.

L_P be the Source Packet Information Length in symbols.

L_B be the Source Block Length in symbols.

Then, source packets with sequence numbers from I to $I + L_B/L_P - 1$ inclusive are included in the source block.

Note that if no FEC Repair packets are received then no FEC decoding is possible and it is unnecessary for the receiver to identify the Source FEC Packet Identification Information for the source packets.

The ESI for a packet is derived from the following information:

- The sequence number, N_s, of the packet.
- The Source Packet Information Length for the source block, L_P.
- The Initial Sequence Number of the source block, I.

Then the ESI for packet with sequence number Ns is determined by the following formula:

$$ESI = (Ns - I) \cdot L_P \quad \text{(E.6)}$$

Note that all repair packet associated to a given Source Block SHALL contain the same SBL and ISN.

E.4.3.2.2.2 Derivation of repair packet ESIs

The ESI for a repair packet indicates which repair symbols the packet contains. This is given directly by the ESI field of the Repair FEC Payload ID.

E.4.3.2.2.3 Procedures for RTP flows

In the specific case of RTP packet flows, then the RTP Sequence Number field SHALL be used as the sequence number in the procedures described above.

The length indication included in the Source Packet Information SHALL be the RTP payload length plus the length of the CSRCs, if any, and the RTP padding bytes, if any. Note that this length is always equal to the UDP payload length of the packet, minus 12.

E.4.3.2.3 FEC Code specification

The requirements of clause E.4.3.1 apply.

E.4.3.2.4 Example parameters

E.4.3.2.4.1 Parameter derivation algorithm

It is recommended that the algorithm of clause E.4.3.1.6.1 is used.

In the case of RTP streams carrying MPEG-2 Transport Streams, then the maximum repair packet size should be set to

$$P = \text{ceil}((n \cdot 188 + 15)/A) \cdot A \qquad \text{(E.7)}$$

Where n is the nominal number of 188 byte TS packets per IP Source packet.

The maximum source block size is determined by application configuration at the sender.

E.4.3.2.4.2 Examples

The above algorithm leads to transport parameters for MPEG-2 Transport Streams as shown in table E.4, assuming the recommended values for A, K_{MIN} and G_{MAX}.

Table E.4: Example parameters settings

Maximum packets per protection period	Nominal TS packets per IP packet	Maximum Packet Size, *P*	Maximum Source Block Size, *B*	*G*	Symbol size *T*
100	7	1 344	134 400	7	192
200	7	1 344	268 800	4	336
300	7	1 344	403 200	3	672
400	7	1 344	537 600	2	672

E.5 FEC decoder

E.5.1 Decoder requirements (normative)

E.5.1.1 Minimum decoder requirements

FEC decoders that are compliant to this annex shall support processing of the SMPTE 2022-1 [67] packets. This means that whenever:

1) an SMPTE 2022-1 FEC packet has been received; **and**

2) all but one of the media packets protected by this FEC packet have been received within the previous *max-block-size* source packets and/or within a time window beginning *max-block-size-time* before the current time; **and**

3) the time at which the remaining media packet is useful to the media decoder has not passed,

then, the SMPTE 2022-1 decoding operation shall be applied and the resulting recovered packet passed to the media decoder.

The above requirement applies independently of the arrival time or order of the packets involved.

NOTE: The parameters *max-block-size* and *max-block-size-time* are part of the FEC Configuration Information and are discussed further in clause E.6.

E.5.1.2 Enhanced decoder requirements

FEC decoders may additionally support Raptor FEC packets. In this case, if a receiver receives a mathematically sufficient set of encoding packets (which may include both SMPTE 2022-1 FEC packets and Raptor FEC packets) for reconstruction of a source block within the previous *max-block-size* source packets and/or within a time window beginning *max-block-size-time* before the current time then the decoder shall recover the entire source block. Note that the example decoder procedures described in clause E.5.2 fulfil this requirement and thus a decoder is compliant to this Annex only if it can successfully decode given any set of packets with which the example decoder can also decode.

E.5.2 Hybrid decoding procedures (informative)

E.5.2.1 Outline

In the case that a receiver receives FEC repair packets from multiple layers, including packets generated according to the codes of both clauses E.3 and E.4, then combined decoding may be provided. This clause outlines procedures which may be followed to achieve this.

Combined decoding proceeds in 3 steps:

Step 1: SMPTE 2022-1 decoding

In this step, the packets encoded according to SMPTE 2022-1 [67], together with the received source packets, are processed as usual to recover zero of more source packets.

Step 2: Raptor decoding

In this step, if source packets are still missing, then packets encoded according to Raptor, together with the received source packets and any source packets which were recovered in Step 1, are processed using standard Raptor decoding procedures (for example as described in [66]) to recover zero of more source packets.

Step 3: Hybrid decoding

In this step, if source packets are still missing, then remaining (unprocessed) SMPTE 2022-1 [67] packets are converted to a form in which they can be added to the Raptor decoding process, and Raptor decoding is then continued.

Conversion of SMPTE 2022-1 packets and their use in Raptor decoding are described in the following clauses.

E.5.2.2 Conversion of SMPTE 2022-1 packets

The objective of this conversion operation of SMPTE 2022-1 packets is to convert them into a form such that they can be included in the Raptor decoding process. According to SMPTE 2022-1, each FEC packet is constructed by applying a protection operation, based on the exclusive OR operation (XOR), to a number, *D*, of the source packets (the "protected packets"). The UDP payload of the SMPTE 2022-1 packet contains the following data:

- An RTP header for the SMPTE 2022-1 packet
- An FEC header containing:
  - The Length Recovery field, which is the XOR of the unsigned network-ordered 16-bit representation of the lengths of the protected packets in bytes minus 12 (for the fixed RTP header), i.e., the sum of the lengths of all the following if present: the CSRC list, header extension, RTP payload, and RTP padding
  - The PT Recovery field, which is the XOR of the Payload Type (PT) fields in the RTP headers of the protected packets.
  - The Timestamp Recovery field, which is the XOR of the Timestamp fields in the RTP headers of the protected packets.
- The XOR of the CSRC list, header extension, RTP payload and RTP padding of the protected packets.

After the SMPTE 2022-1 [67] decoding, if all missing source packets associated with an SMPTE 2022-1 [67] FEC packet have been recovered, it is not necessary to perform the conversion operation for that SMPTE 2022-1 [67] FEC packet. However, if there are still remaining unrecovered protected packets, a conversion operation is needed for each such SMPTE 2022-1 [67] FEC packet. The conversion is achieved by concatenating the following fields to form a "virtual" Raptor repair packet payload i.e. the virtual Source Packet Information that includes in addition to the virtual payload also the fields for the UDP flow ID, the length indication field, and padding:

- A single zero byte.
- A two byte length indication, which is equal to the XOR of the unsigned network-ordered 16-bit representation of the lengths of the unrecovered protected packets in bytes minus 12 (for the fixed RTP header). This is equal to the XOR of the Length Recovery field in the SMPTE 2022-1 [67] FEC header and the 16-bit representation of the lengths of the received protected packets in bytes minus 12.
- A two-bit field, which is equal to the XOR of the RTP Version fields of the unrecovered protected packets. This is equal to zero if the number of unrecovered protected packets is even and 2 otherwise.
- A seven (7) bit field, equal to the XOR of the RTP Padding (P), Extension (X), CSRC Count (CC) and Marker (M) fields of the unrecovered protected packets. This is equal to the XOR of the concatenated P, X, CC and M fields in the SMPTE 2022-1 RTP header and the concatenated P, X, CC and M fields of the received protected packets.
- A seven (7) bit field equal to the XOR of the RTP PT fields of the unrecovered protected packets. This is equal to the XOR of the PT Recovery field in the SMPTE 2022-1 FEC header and the PT fields of the received protected packets.
- A 16-bit field equal to the XOR of the RTP Sequence Number fields of the unrecovered protected packets. The Sequence Numbers of the unrecovered protected packets can be explicitly calculated based on the *SNbase*, *offset* and *NA* fields of the SMPTE 2022-1FEC header.
- A 32-bit field equal to the XOR of the RTP Timestamp (TS) fields of the unrecovered protected packets. This is equal to the XOR of the TS Recovery field in the SMPTE 2022-1 FEC header and the TS fields of the received protected packets.

- A 32-bit field equal to the XOR of the RTP SSRC fields of the unrecovered protected packets. This is equal to zero if the number of unrecovered protected packets is even and equal to the SSRC of the stream otherwise.
- The XOR of the CSRC lists, header extensions, RTP payloads and RTP paddings of the unrecovered protected packets,. This is equal to the XOR of all the bits in the SMPTE 2022-1 FEC packet except its RTP and FEC headers and all the bits in the received protected packets except their 12-byte fixed RTP headers (padded as necessary). Note that a header extension and CSRC list are never present in an SMPTE 2022-1 FEC packet, independent of the values of the X and CC fields in its RTP header.
- A number of zero-valued padding bytes, such that the total length of the "virtual" repair packet payload is equal to the length of the other Raptor repair packet payloads (which are all required to be the same according to clause E.4.3.2.5).

The resulting "virtual" repair packet payload is then equal to the XOR of the Source Packet Information of the unrecovered protected packets.

E.5.2.3 Extension of Raptor decoding

A possible Raptor decoding algorithm is described in clause C.7 of [66] in terms of a Gaussian Elimination process upon a matrix **A**. If decoding is not possible without use of the SMPTE 2022-1 [67] packets, then this decoding process will fail during the second phase described in clause C.7 of [66]. At this point, the matrix **A** has less than L non-zero rows (Note, the symbol L here denotes the number of intermediate symbols of the Raptor code as defined in [66], not the L value associated with the SMPTE 2022-1 [67] packets).

Let G be the number of symbols per packet (which can be calculated as the Raptor repair packet payload size divided by the symbol size). Then each "virtual" Raptor repair packet constructed above consists of exactly G new symbols, each of which is the XOR of exactly N_s source symbols (which we call the "unrecovered protected symbols"), where N_s is the number of unrecovered protected packets associated with the SMPTE 2022-1 [67] FEC packet from which the "virtual" Raptor repair packet was constructed.

For each such new symbol, a new row is added to the decoding matrix **A**. This row is constructed as follows:

- The row is initialized to zero.
- For each of the N_s unrecovered protected symbols, the *LTEnc* generator is used to determine the set of intermediate symbols whose sum is equal to the unrecovered protected symbol. For each such intermediate symbol a "1" is XORed into the appropriate position of the new row.

Phase two of the decoding process is then continued with these additional rows and symbols.

E.6 FEC Content Delivery Protocols

This clause defines several complete FEC Content Delivery Protocols, making use of the components defined in the foregoing clauses.

E.6.1 Multicast MPEG-2 Transport Stream over RTP

This clause defines a Content Delivery Protocol for FEC protected multicast delivery of MPEG-2 Transport Streams over RTP.

E.6.1.1 Control protocols

FEC Configuration information SHALL be delivered using the DVB Service Discovery mechanisms as described in clause 5. The DVB Broadcast Discovery record MAY contain the multicast address(es) and port(s) for one or more FEC layers. Receivers may choose which layers to join depending on capability and local configuration.

When the Raptor layer is provided, the Flow ID within the Source Packet Information for the MPEG-2 TS flow SHALL be zero.

E.6.1.2 Transport protocol

The MPEG-2 Transport Stream shall be transported according to clause 7.1.1.

FEC protection of the MPEG-2 Transport Stream MAY be provided according to clauses E.3 and E.4. When a Raptor layer is provided, the FEC Scheme defined in E.4.3.2 SHALL be used.

E.6.2 Unicast MPEG-2 Transport Stream over RTP

This clause defines a Content Delivery Protocol for FEC protected unicast delivery of MPEG-2 Transport Streams over RTP.

E.6.2.1 Control protocols

The receiver shall indicate in the Transport header of the RTSP SETUP request which FEC layers are requested by supplying port numbers that should be used for the FEC repair packets. Only requested FEC layers shall be sent to the receiver.

The server may supply the FEC parameters *max-block-size*, *max-block-size-time* and *FEC Object Transmission Information* in the Transport header of the RTSP SETUP response.

The Flow ID for the MPEG-2 TS flow SHALL be zero.

E.6.2.2 Transport protocol

The MPEG-2 Transport Stream shall be transported according to clause 7.1.1.

FEC protection of the MPEG-2 Transport Stream MAY be provided according to clauses E.3 and E.4 above. When a Raptor layer is provided, the FEC Scheme defined in E.4.3.2 SHALL be used.

E.6.3 Generic multicast video (informative)

This clause defines a Content Delivery Protocol for FEC protected multicast delivery of arbitrary audio/video streams (for example H.264 encapsulated in RTP).

E.6.3.1 Control protocols

FEC Configuration information SHALL be delivered using the DVB Service Discovery mechanisms as described in clause 5. The DVB Broadcast Discovery record MAY contain the multicast address(es) and port(s) for one or more FEC layers. Receivers may choose which layers to join depending on capability and local configuration.

E.6.3.2 Transport protocols

The audio/video stream is assumed to be carried by one or more UDP flows. FEC protection of these UDP flows MAY be provided using the procedures of clause E.4.2.2 and in particular the FEC Scheme defined in clause E.4.3.1.

E.6.4 Generic unicast video (informative)

This clause defines a Content Delivery Protocol for FEC protected unicast delivery of arbitrary audio/video streams (for example H.264 encapsulated in RTP). This clause is provided to describe how FEC can be applied to future extensions to the DVB-IPTV Handbook which address direct encapsulation of audio/video streams in RTP.

E.6.4.1 Control protocols

The receiver shall indicate in the Transport header of the RTSP SETUP request which FEC layers are requested by supplying port numbers that should be used for the FEC repair packets. Only requested FEC layers shall be sent to the receiver.

The server may supply the FEC parameters *max-block-size*, *max-block-size-time* and *FEC Object Transmission Information* in the Transport header of the RTSP SETUP response.

E.6.4.2 Transport protocols

The audio/video stream is assumed to be carried by one or more UDP flows. FEC protection of these UDP flows MAY be provided using the procedures of clause E.3 and in particular the FEC Scheme defined in clause E.4.

E.6.5 MIME Types definitions for AL-FEC

Two MIME media subtypes have been registered with IANA (http://www.iana.org/) for DVB-IPTV AL-FEC:

- "application/vnd.dvb.iptv.alfec-base" to identify the Base layer;
- "application/vnd.dvb.iptv.alfec-enhancement" to identify the Enhancement layer(s).

These two MIME types shall be used when the description of DVB-IPTV services using FEC layers require MIME type usage (e.g. with SDP).

E.7 Raptor explicit encoding sequences

The Raptor code defined in this annex is defined in terms of explicit encoding operation sequences which shall be applied to generate repair symbols from source symbols.

NOTE: The FEC code which results from these encoding sequences is identical to that generated by the procedures described in annex C of [66]. As a results, the example decoder procedures described in [66] may be used.

The Maximum Source Block Size used with the FEC Schemes defined in clause E.4.3 SHALL be one of the following values:

- 101, 120, 148, 164, 212, 237, 297, 371, 450, 560, 680, 842, 1 031, 1 139 and 1 281.

Explicit encoding operation sequences are provided for each of the block sizes indicated above, supporting highly efficient implementation of encoders for the Raptor code for these block sizes.

This clause describes the notation used for the encoding sequences. The encoding sequences are provided as text files attached to the present document.

Each text file consists of two parts, a "pre-coding" section and a "repair symbol encoding" section. The two sections of the file are separated by a blank line.

The encoding sequence assumes that the data to be encoded is stored in a (virtual) block of memory. Each virtual memory location stores a complete symbol. At the start of the process, the source symbols are assumed to be stored consecutively in memory locations 0 to K-1 inclusive, where K is the block size.

Additional working memory locations are required to be available up to and including memory location L-1, where L is given in the following table for each value of K. Note that the L value here is exactly the value of L calculated according to annex C of [66]. The additional working memory shall be initialized to zero.

K	L
101	127
120	149
148	181
164	197
212	251
237	277
297	337
371	419
450	499
560	613
680	739
842	907
1 031	1 103
1 139	1 213
1 281	1 361

Figure E.8: Total memory requirement in symbols (*L*) for different block sizes

Each line of the "pre-coding" section of the text file consists of a series of memory location indices (in decimal notation), separated by spaces and each optionally preceded by the character ">". Each line is interpreted as follows:

Let:

A be a working register which stores one symbol.

n be the number of memory location entries on the line.

m_i be the ith entry of the line, for $i = 0, \ldots, n-1$.

$C[x]$ be the symbol at memory location x.

0 be the zero symbol (all bits are zero).

$\oplus$ be the bitwise exclusive OR operation.

The following algorithm should be followed for each line in sequence:

A := **0**

FOR i = 0 to n-1

IF m_i is preceded by ">" THEN

$C[m_i] := C[m_i] \oplus A$

ELSE

$A := A \oplus C[m_i]$

ENDIF

Each line of the "repair symbol encoding" section of the file lists the memory locations which shall be XORed together to produce a repair symbol, the first line providing the list for the repair symbol with ESI *K*, the second for the repair symbol with ESI *K*+1 etc.

For example, when included within the pre-coding section of the file, the line:

4 8 3 5 > 7 6 > 10

Would result in the following symbol assignments:

$C[7] := C[7] \oplus C[4] \oplus C[8] \oplus C[3] \oplus C[5]$

and:

$C[10] := C[10] \oplus C[6] \oplus C[4] \oplus C[8] \oplus C[3] \oplus C[5]$

Annex F (normative): RTP Retransmission Solution

F.1 Introduction

This annex defines an option for an HNED to provide immediate feedback (FB) using RTCP and then an RET server to retransmit the missing packets. This mechanism can be used instead of the optional AL-FEC solution or in combination with an AL-FEC solution, to provide for protection against packet loss of RTP streams. It can be used both for unicast services (CoD and Broadcast TV service with trick mode) and multicast (Live Media Broadcast service).

F.2 Terms and Acronyms

See clauses 3.1 and 3.2 of the present document.

F.3 Retransmission (RET) architecture

The Retransmission (RET) architecture clause describes how RTP retransmission works for unicast and multicast services, as defined in other parts of this document.

F.3.1 RET for CoD/MBwTM service

The simplest retransmission architecture is for unicast CoD and Media Broadcast with Trick Mode (CoD/MBwTM), as shown in figure F.1. This architecture contains two components: an HNED with an RTP client for media and an RTP client for repair and a CoD/MBwTM streamer with an RTP media and RTP RET server function. If the HNED uses SSRC multiplexing then it is the same RTP client for both media and repair. The RET server may not be in the same physical box as the CoD/MBwTM server. The Retransmission repair consists of 3 steps:

1) The CoD/MBwTM data stream is unicast over RTP (1, in the figure) with a packet loss.

2) Upon RTP packet loss detection by the HNED, the RET client in the HNED sends an RTCP FB message to the RET server (2).

3) The server responds to the RTCP FB message by retransmitting the requested packet (called a RET packet) over RTP to the RET client (3).

The unicast RTP RET packets can be regarded as a separate RTP stream which is one-to-one associated with the original unicast RTP stream composed of the original packets. This is important when you consider the concept of RTP sessions, as described in [21], which are each distinguished by a full, separate space of SSRC identifiers. If the RTP source coincides with the RTP retransmission server, as shown in figure F.1. then the original and retransmission streams are combined into a single RTP session with different SSRC identifiers only if the same transport addresses are used.

DVB RET recommends the use of SSRC multiplexing for a CoD/MBwTM stream and associated RET stream, resulting in a single RTP session. If session multiplexing is used the DVB RET server may use an SSRC different from the SSRC in the original stream. RFC 4588 [86] mandates that for session multiplexing the SSRC of both RTP streams be the same. However for monitoring purposes, such as the performance of the retransmission server, DVB RET Servers may use a different SSRC, so that a distinction at RTP level of original and repair media can be made using the SSRC.

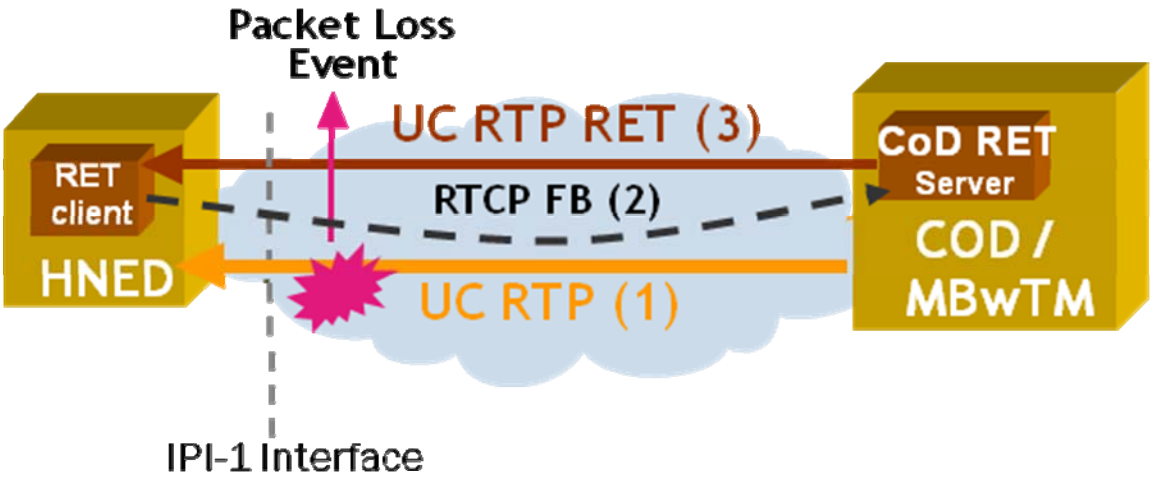


Figure F.1: RET Architecture and messaging for CoD/MBwTM services overview

F.3.2 RET for LMB service

RTP retransmission for Live Media Broadcast (LMB) is more complex than for CoD/MBwTM because the media stream is multicast whilst the repair stream can be multicast or unicast. Figures F.2 and F.3 show the elements and communication flows involved in a RET architecture for LMB service. In this architecture, there can be several DVB LMB RET servers which handle unicast RTCP (FB) messaging from the RET clients, typically each server acting as an RTCP target for a subset of the HNEDs. RET enabled LMB services use SSM multicast with unicast feedback as discussed in [i.5].

A packet loss event may take place anywhere in the network: Downstream from the DVB LMB RET server (see figure F.2) and upstream from the DVB LMB RET server (see figure F.3). The packet loss event upstream of the DVB LMB RET servers can be repaired by adding a retransmission client to the server and use the same retransmission mechanism to repair the packet loss (see figure F.3). Note that the scope of the present document is the IPI-1 interface and hence communication between DVB LMB RET servers and any retransmission server positioned deeper upstream in the network is not addressed in this document.

There are two ways for repairing the packet loss event for LMB services:

- Unicast (see figure F.2).
- Multicast (see figure F.3).

The unicast repair method works similarly to the CoD/MBwTM case. The HNED detects packet loss, transmits a FB message, and the response is a unicast RET packet. The SSRC used by the LMB RET server may be different from the SSRC of the original MC RTP session.

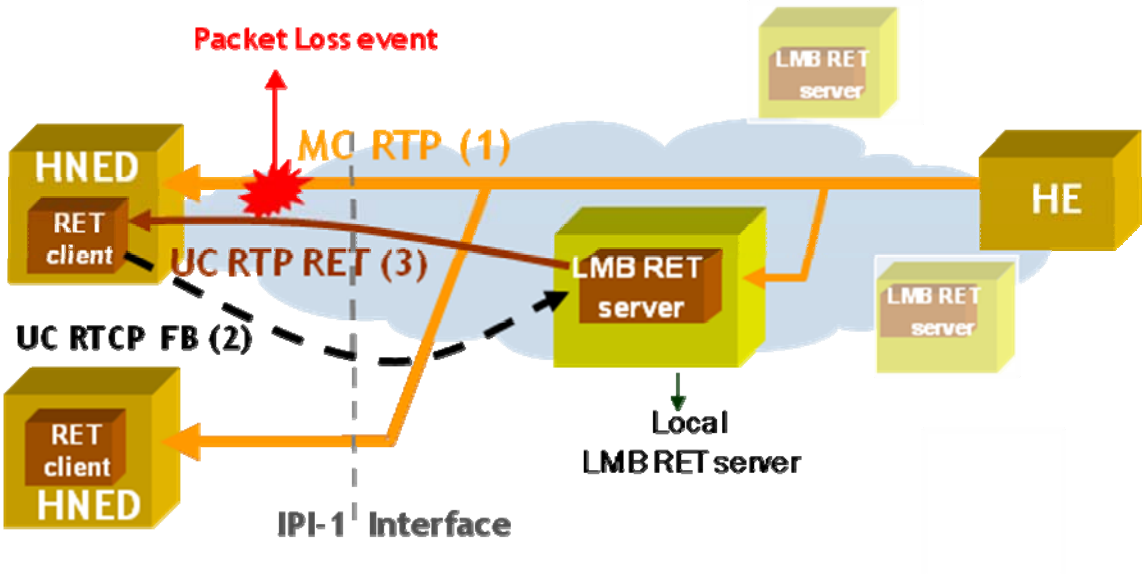


Figure F.2: RET Architecture and messaging for LMB services: unicast retransmission

Figure F.3 shows the multicast repair method for LMB services based on a packet loss event that has taken place upstream of the DVB LMB RET server, impacting many HNEDs. The LMB RET server does not need the RTCP FB message from the HNEDs to know about the packet loss as the LMB RET server itself can detect this packet loss. The LMB RET server, having detected the loss, requests the impacted HNEDs to not transmit the FB message by sending itself an RTCP Feed Forward (FF) message over multicast (step 2 in figure F.3) so preventing "NACK storms." The RET packet, once it has been made available to the LMB RET server, is then sent to the impacted HNEDs (step 3) also over multicast. This mechanism assumes that the RET clients respect some minimum waiting time between packet loss detection and RET requesting, either through randomization of waiting times (see [85]) or a more advanced mechanism as explained in clause F.7.2.

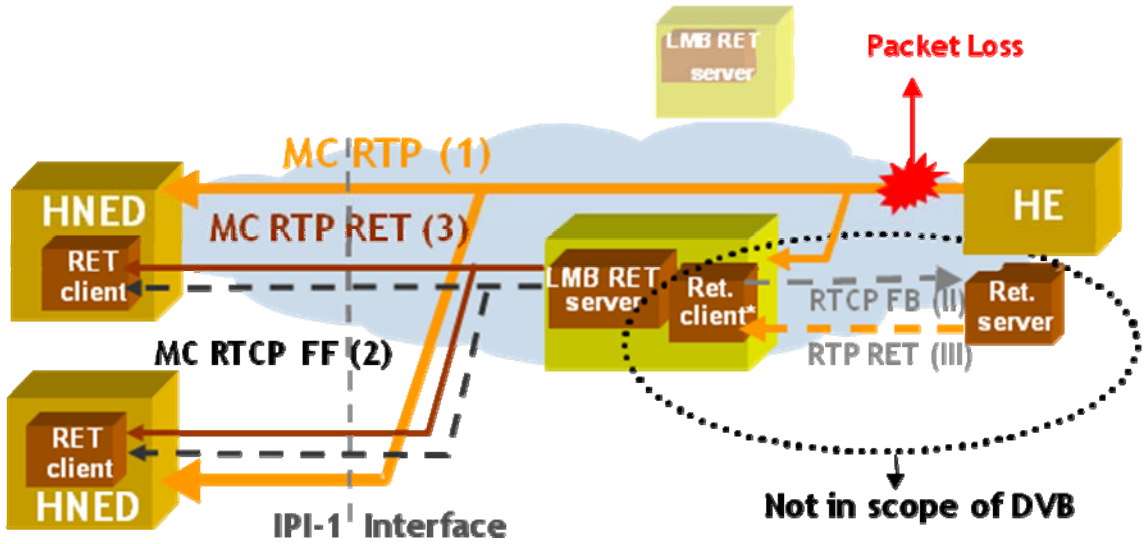


**Figure F.3: RET Architecture and messaging for LMB services:
MC retransmission and MC NACK suppression**

The LMB RET server does not usually coincide with the headend (HE) acting as the RTP media source, and retransmissions can occur both unicast (default) and multicast. This results at IPI-1 in an original multicast RTP session plus 2 associated RTP Retransmission sessions, which gives 2 RTP sessions if only unicast repair is used and 3 if both multicast and unicast repair are used.

The Multicast RTP RET session differs from the original Multicast RTP session because of a different source IP (the IP address of the LMB RET server), a different destination port and/or group Multicast IP address. The default retransmission repair being unicast, all the unicast and optional multicast repair parameters including availability shall be configured or signaled to the RET client. The multicast repair session functions just like any other SSM stream, for example, using the standard IGMPv3 "join" to get the MC RTP RET stream. The SSRC used by the LMB RET server may be different from the SSRC of the original MC session so it is permissible that the LMB RET server does not follow the behaviour defined for session-multiplexing in RFC 4588 [86].

F.3.2.1 RTP Sessions for the RET Enabled LMB service

Whenever a RET-enabled HNED connects to a RET-enabled LMB service, along with the establishment of an RTP session for the original multicast stream, a new unicast RTP RET session should be autonomously established.

The "media source SSRC" and/or the destination transport address of the unicast RTCP FB packets issued by the HNEDs indicate to the LMB RET server from which original RTP stream a packet is requested for retransmission. This results in, for a service with unicast and multicast repair, *N + 2* RTP sessions where *N* is the number of serviced RET clients. Specifically the *N + 2* RTP sessions are:

1) The original RTP MC session, which is a source-specific multicast (SSM) with the headend being the source, in which the DVB LMB RET server acts both as receiver (receiving the MC RTP stream) and as an RTCP target (receiving the unicast RTCP reports of the HNEDs).

2) *N* x unicast RET RTP sessions (one for each RET-enabled HNED): the LMB RET server acts as RTP sender/source.

3) (optional) A source-specific multicast RET RTP session where the LMB RET server acts as the source. This source-specific multicast may also be used by the LMB RET server for distributing the RTCP Feed Forward messages for NACK suppression in the original multicast RTP session (see clause F.5.2).

F.4 RTCP signaling by RET-enabled HNEDs

The HNED's RTCP reports for RET enabled LMB services are always transmitted in unicast towards the DVB LMB RET server that shall also be the RTCP target [i.5]. The LMB RET server must never return the reports towards the HNEDs except when the LMB RET server forwards the RTCP FF message (see clause F.5.2).

The following clauses describe the RTCP reports that can be transmitted by an HNED and shall be supported by a RET-enabled HNED.

F.4.1 RTCP FB message

The protocol that is used by an HNED for requesting retransmission is RTCP. The format of the Generic NACK transport layer FB message as specified in [85] will be used for packet retransmission requests, as shown in figures F.4 and F.5. The HNED shall support the timing rules set forth in [85].

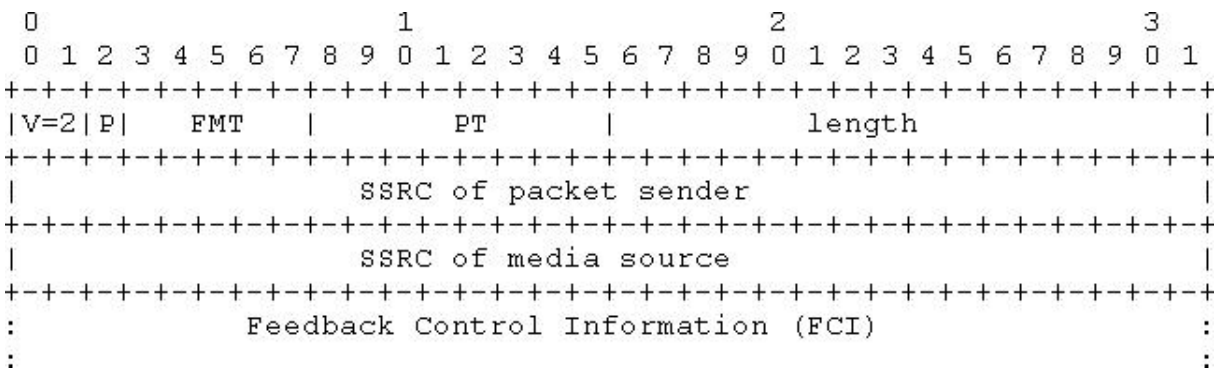


Figure F.4: Packet format for RTCP FB message

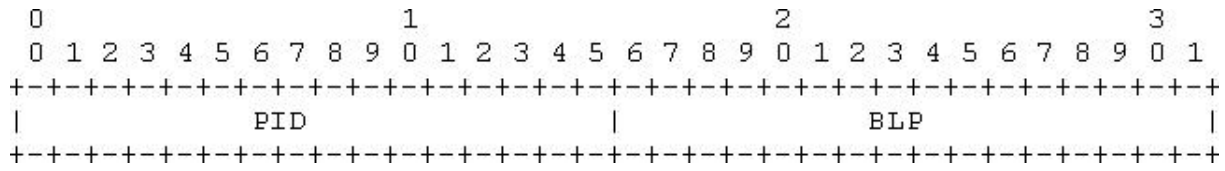


Figure F.5: Format of the Generic NACK put in the FCI field in the RTCP FB message

The PT in the RTCP FB message is filled in as "RTPFB" (value 205) and the Feedback Message Type (FMT) field in the RTCP FB message must be 1.

The Feedback Control Information (FCI) field in the RTCP FB message contains at least one Generic NACK and may contain more than one.

The format of the generic NACK is shown in figure F.5. The Packet ID (PID) field refers to the sequence number of the first reported lost packet. The bitmask of following lost packets (BLP) field allows for reporting losses of any of the 16 RTP packets immediately following the RTP packet indicated by the PID (see [85] for a thorough definition). This allows a single Generic NACK to report up to 17 (consecutive) packet losses.

The RTCP messages issued by the HNED are always unicast towards the network, even for LMB. The transport address for the RTCP messaging (destination IP address and UDP destination port) is obtained by the HNED through the configuration methods as explained in clause F.6.

The "SSRC of packet sender" in the RTCP FB message has the SSRC value used by the HNED in the original RTP session for LMB services. The "SSRC of media source" is the SSRC of the headend used for the LMB service for which the RTCP Feedback message reports packet loss.

Note: When multiple LMB services have the same media source SSRC, for example when LMB streams are sourced from different LMB SPs, the "SSRC of media source" in the RTCP Feedback message might not uniquely define the LMB service. This means that the HNED needs to be configured with different transport addresses for the unicast RTCP FB packets associated with these LMB services. Once configured, the combination of the "SSRC of media source" and the RTP RET session transport address of the IP/UDP/RTCP FB packet uniquely defines the LMB service for which the RTCP Feedback message reports packet loss**.**

F.4.2 RTCP RR, RTCP SDES and RTCP BYE packets

A RET-enabled HNED shall support the RTCP Source Description (SDES) packet, RTCP Receiver Reports (RR), and RTCP BYE packet as defined by the formats in [21].

F.4.2.1 RTCP SDES Packet

The SDES packet shall only contain the CNAME item with the format following [21].

The SDES packet can be sent by the RET-enabled HNED in two different RTP sessions: in the original RTP session (multicast or unicast), and in the unicast RTP RET session. The CNAME, identifying the HNED shall be identical for the two RTP sessions.

The support by the HNED for the SDES packet is mandatory for the original MC RTP session. An HNED, for the RTP RET session, may support sending RTCP, and when it is enabled to do so it must always send the RR combined with an SDES as a compound message (see [85], clause 3.1).

F.4.2.2 RTCP RR Packet

The RR contains the reception quality data of an RTP session as defined by [21].

The RET-enabled HNED shall support sending RR packets that contain reception quality data on the RTP packets received in the original RTP session, both for LMB and CoD services. This RR reporting may be disabled with the attribute dvb-disable-rtcp-rr.

The RET-enabled HNED may support sending RR that contain reception quality data on unicast RTP RET packets. The HNED is not allowed to send RTCP packets for a RET stream before RTCP packets for the original session have been sent. If this were allowed the LMB RET server could not distinguish between RTCP packets for RET and the original session if the headend and the LMB RET server use the same SSRC.

When an HNED sends RTCP reporting data on both the original and RET packets, it may do so by including the RET RR and original RR into the same compound RTCP report or using two separate sessions. When including the RET RR and original RR into the same compound RTCP report, the HNED must use one SSRC but when using a separate session the HNED must use different SSRCs for the RTCP reports in the two sessions. The use of a new SSRC when using a separate session prevents an SSRC collision as the RTCP target sees two RTCP packets with the same SSRC coming from different source transport addresses, with the result that the RTCP target would ignore one of them

F.4.2.3 RTCP BYE packet

When a participant wishes to leave a RTP session, a BYE packet shall be transmitted to inform the other participants of this event as defined in [21]. A BYE packet shall be sent on an SSRC collision.

A BYE packet shall only be sent for LMB services by the HNED if it has been configured through the parameter "dvb-enable-bye" and when configured should only be sent if RTCP FB or SDES/RR have already been exchanged by the HNED in the original RTP session with the RTCP Target.

F.4.3 RTCP messaging types

An HNED shall be able to support both compound and non-compound RTCP FB packets for the original RTP session. A compound RTCP packet must contain an SDES, an RR and FB messages (see [85]), while a non-compound (reduced size) RTCP packet contains only RTCP FB messages (see [i.6]). The use of both types of RTCP packets enables the split between longer term statistical information and retransmission repair information. [i.6] provides more information on reduced size RTCP messaging

SD&S or RTSP configuration determines the message type when requesting retransmissions. When the HNED uses non-compound RTCP FB messages, the frequency of sending compound packets for statistical reporting is governed by [21]. DVB recommends that for compound reporting for statistical purposes, this frequency be once every 5 s.

The BYE packet for the original RTP session, when used, is always sent out in compound format containing at least one SDES and the BYE packet, as per [21]. The RR must also be included if RR reporting is enabled. Note that in the present document, a compound RTCP BYE packet must contain the RR packet.

RTCP compound statistical reporting (comprising SDES +RR) may also be disabled through configuration with the "dvb-disable-rtcp-rr" attribute.

For the RTP RET session non-compound reporting by the HNED is forbidden.

F.5 RTCP signaling towards RET-enabled HNEDs

The following clauses describe the RTCP packets that shall be supported by a RET-enabled HNED.

F.5.1 The RTCP SDES/SR packets

An HNED shall be able to receive RTCP Sender Report (SR) and RTCP SDES packets in the original multicast RTP session from the RTP multicast source.

A DVB LMB/CoD RET server may send RTCP SR and RTCP SDES packets in the unicast RTP RET session.

A DVB LMB RET server may send RTCP SR and RTCP SDES packets in the multicast RTP RET session.

F.5.2 The RTCP Feed Forward (FF) message (LMB service only)

The RTCP FF message is an RTCP FB message relayed by the LMB RET server downstream. The RTCP FB message originated either at an "upstream RTP client" or at immediate reporting HNEDs:

- An upstream RTP client means the client is located upstream from the LMB RET server so it is able to detect upstream packet losses impacting all HNEDs serviced by the LMB RET server and receiving the LMB service. figure F.3 shows such an upstream client.
- Immediate reporting HNEDs is a subset of HNEDs that are configured to report missing packets immediately, thus reporting before all the other RET-enabled HNEDs serviced by the same LMB RET server. Their SSRCs are distinguishable from the SSRC of non-immediate reporting HNEDs (see clause F.7.2).

The relayed messages are now called RTCP FF messages because they are multicast back to the HNEDs in the opposite direction to the RTCP FB messages issued by the HNEDs. The sending of the RTCP FF message to the HNEDs assists in preventing or reducing NACK storms. The operation of the LMB RET Server, when handling such a RTCP FB message originating from an upstream client or a set of immediate reporting HNEDs, is that of an RTP translator that "simply forwards RTCP packets unmodified" as described in clause 7.2 of [21].

The typical event sequence when a RTCP FF message is relayed by the RTCP target/LMB RET server is:

- There is a packet loss event on an upstream link or on a downstream aggregated link impacting several HNEDs serviced by the LMB RET server.
- An RTCP FB message is sent by an upstream RTP client and/or by all immediate reporting HNEDs receiving the LMB service, indicating packet loss to the RTCP target/LMB RET server.
- **Upstream packet loss:** The LMB RET server, acting as RTP translator, translates the transport address of RTCP FB message and forwards it as an RTCP FF message in the MC RTP RET session. The LMB RET server discards any RTCP FB messages received by immediate reporting HNEDs.
- **Downstream packet loss event impacting all or a large subset of HNEDs:** The RTCP target will get RTCP FB messages of all immediate reporting HNEDs, but it needs to forward only one FB message as FF message in the MC RTP RET session.
- The LMB RET server transmits in the MC RTP RET session the RET packet(s) as indicated by the RTCP FF message. Note that in the case of upstream packet loss, this is only done once the missing packets are available at the LMB RET server.

The different logical functionalities: upstream client, translator and LMB RET server may or may not be implemented in the same physical entity. The SSRC of the upstream client is signaled through the parameter "dvb-ssrc-upstream-client."

The format of the RTCP FF messages is the RTCP Generic NACK transport layer FB message as specified in [85]. The feedback control information field in the RTCP FF message must contain at least one NACK and may contain more than one.

The "packet sender SSRC" field in the FF message is the "upstream-client" SSRC or the SSRC of an immediate reporting HNED. The "media source SSRC" field is the same as found in the originating RTCP FB message and will have the SSRC value as in the original RTP MC packets.

The RTCP FF message is sent from the LMB RET server anticipating the loss detection of a certain packet by the HNED. The LMB RET server shall send this message in the MC RTP RET session.

When the HNED receives this message, it should not send out an RTCP FB message asking for retransmission of those packets as indicated in the received RTCP FF message. If, after time "dvb-t-ret" (see clause F.7) as defined in SD&S, not all the expected RET packets have been received then the HNED may send an RTCP FB message.

F.5.3 The RTCP Receiver Summary Information (RSI) packets(LMB service only)

The RTCP RSI packets are RTCP packets that signal information towards the HNEDs relating to the original MC RTP session. In general, these packets convey information which otherwise could be retrieved by the HNEDs if multicast distribution of HNED's RTCP packets were allowed in the original MC RTP session.

The RTCP RSI packet has the format as defined in figure F.6.

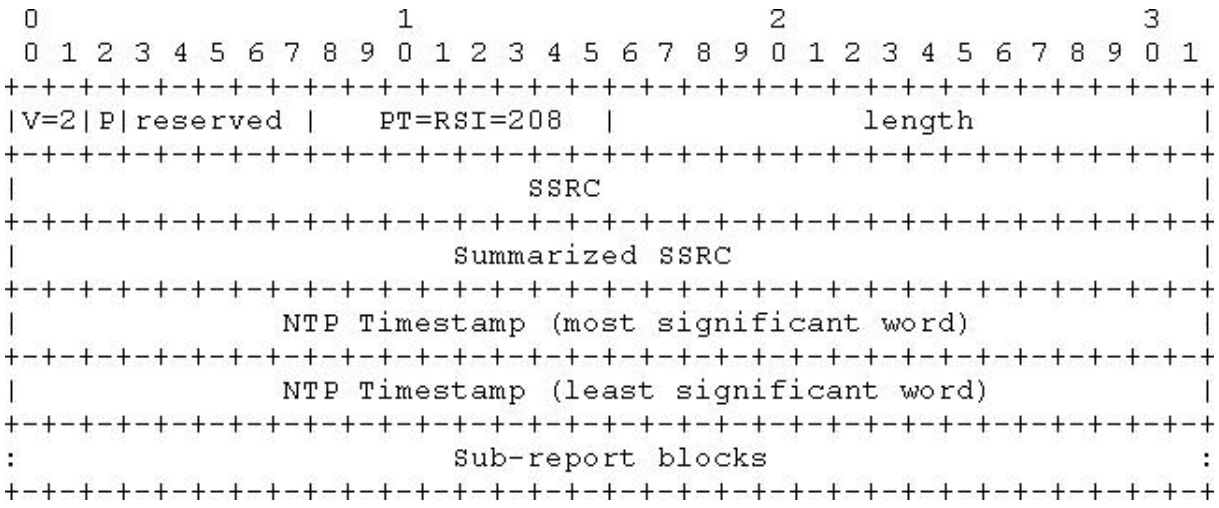


Figure F.6: Format of RSI packet

- Length: 16 bits: the length of the RTCP packet in 32-bit words minus one, including the header and any padding.
- SSRC: 32 bits: The SSRC, which equals the SSRC in the original RTP MC stream packets
- Summarized SSRC: 32 bits: The SSRC (of the Media Sender) of which this report contains a summary; this is generally the same value as found in the previous SSRC field.
- Timestamp: 64 bits: Indicates the wallclock time when this report was sent. Wallclock time (absolute date and time) is represented using the timestamp format of the Network Time Protocol (NTP), which is in seconds relative to 0h UTC on 1 January 1900. Its format is similar to the time stamp in the RTCP sender reports. The timestamp value is used to enable detection of duplicate packets, reordering and to provide a chronological profile of the feedback reports.

Four sub-reports are defined in the present document, which the RET-enabled HNED shall be able to parse and interpret. The sub-reports all share the following format:

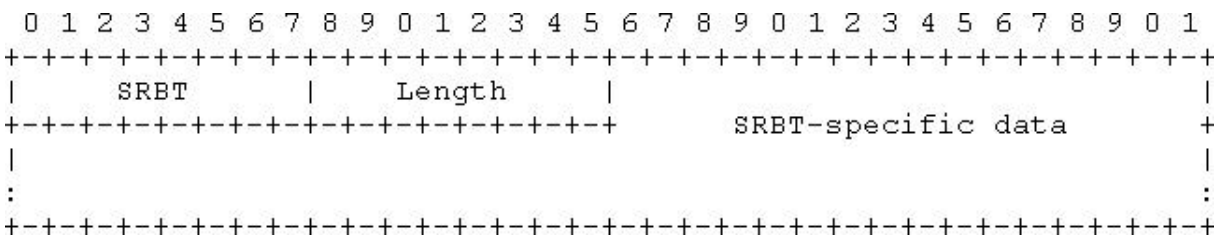


Figure F.7: Generic Format of RSI Packet

- SRBT: 8 bits: Sub-Report Block Type. The sub-report block type identifier. The values for the sub-report block types can be:
    - 0: IP4 Unicast Feedback Address; this is the address of the LMB RET server to which the RTCP (FB) messages shall be addressed.
    - 1: IP6 Unicast Feedback Address. This is not supported in DVB.
    - 2: DNS name for Unicast Feedback: this is the address of the LMB RET server, being the RTCP target to which the RTCP (FB) messages shall be addressed.
    - 8: SSRC collision list.
    - 11: "Receiver Bandwidth".
- Length: 8 bits: The length of the sub-report in 32-bit words.
- SRBT-specific data: <Length*4 - 2> octets: This field contains type-specific information based upon the SRBT value.

The sub-report block type 0,1 or 2 has the following format.

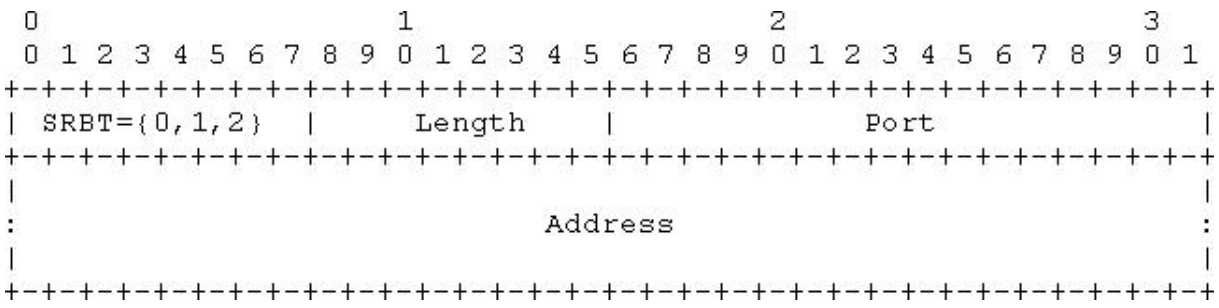


Figure F.8: Format of RSI Packet with sub-report block type equal to 0,1 or 2

- Length: 8 bits: The length of the sub-report block in 32-bit words. This is 2 for an IPv4 address (Total length = 4 + 4 = 2*4 octets) and since DVB only supports IP v4, this is currently the only valid value for an IP address. For a DNS name, the length field indicates the number of 32-bit words making up the string plus 1 byte and any additional padding required to reach the next word boundary.
- Port: 2 octets: The port number to which the HNEDs send RTCP reports.
- Address: 4 octets (IPv4), or n octets (DNS name). The address to which receivers send RTCP reports. A DNS name is an arbitrary length string that is padded with null bytes to the next 32 bit boundary. The string SHALL be UTF-8 encoded.

The sub-report type 8 has the following format.

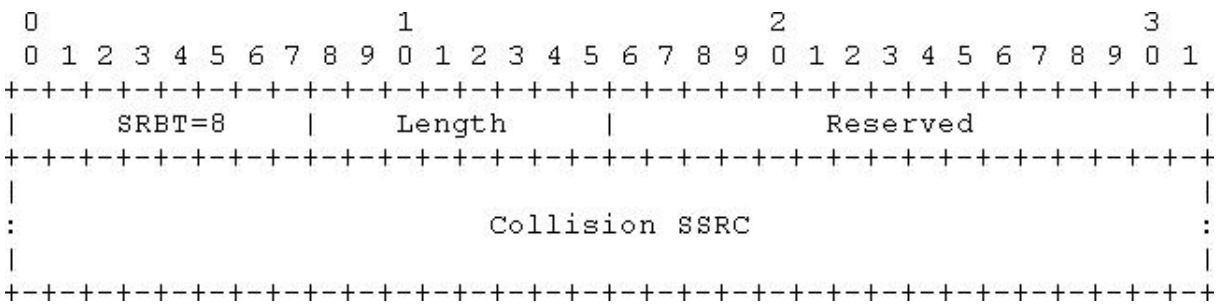


Figure F.9: Format of RSI Packet with sub-report block type equal to 8

- Length: 8 bits: The length of the sub-report block in 32-bit words
- The collision SSRC sub-report provides a mechanism to report SSRC collisions amongst the group. In the event that a non-unique SSRC is discovered based on the tuple [SSRC,CNAME], the collision is reported in this block and the affected nodes shall reselect their respective SSRC identifiers. The "Collision SSRC" is a n × 32 bits field; this field contains a list of SSRCs that are duplicated within the group.

This sub-report is used when multiple HNEDs independently choose the same SSRC for the LMB original RTP session. This SSRC collision is not detected by the HNEDs, rather by the network elements such as the RTCP target, by looking at the (SSRC, CNAME) and/or the transport addresses used by the HNEDs. The "SSRCcollision list " RSI packet signals such an SSRC collision and advertises the HNED SSRCs for which a collision is detected. When an HNED receives such a packet, and it recognizes its own SSRC in the "collision SSRC" list then it shall adapt its SSRC, only sending an RTCP BYE packet when RTCP BYEs are enabled in SD&S.

In case the HNEDs are configured not to use the SDES (including CNAME) packets for their RTCP reporting in the original MC RTP session, SSRC collision may also be detected based on the tuple [SSRC, transport address].

The sub-report type 11 has the following format:

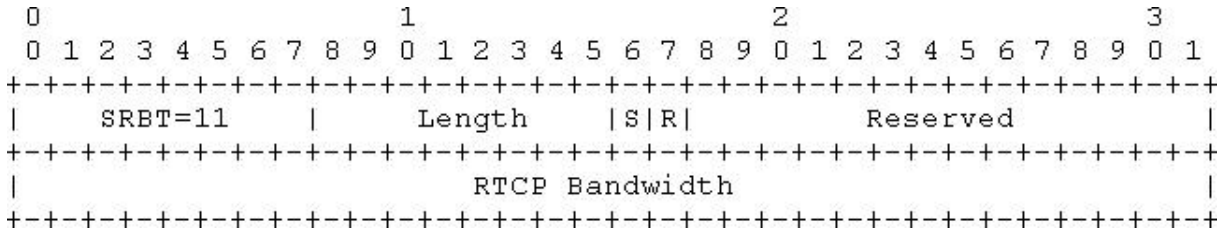


Figure F.10: Format of RSI packet with sub-report block type equal to 11

- Length: 8 bits: The length of the sub-report block in 32-bit words.
- Sender (S): 1 bit: Shall be set to 0.
- Receivers (R): 1 bit.

  Shall be set to 1 and explicitly indicates that the contained bandwidth value applies to each RET-enabled HNED.

- Reserved: 14 bits.

  Shall be set to zero upon transmission and ignored upon reception.

- RTCP Bandwidth: 32 bits.

This field indicates the maximum bandwidth allocated per HNED for sending RTCP data relating to the original RTP session. This is a fixed-point value with the binary point in between the second and third bytes. The value represents the bandwidth allocation per-receiver in kilobits per second with values in the range 0 <= BW < 65536.

The information in the RSI packets always relates to the original multicast RTP session, but they can be distributed either in the original multicast RTP session or in the multicast RTP RET session (with a source address of the LMB RET server address), but never in both multicast RTP sessions. By default the RSI messages are transmitted in the original multicast RTP session. When they are transmitted in the multicast RTP RET session, this is indicated by means of the "dvb-rsi-mc-ret" attribute.

The configuration of both the RTCP target/LMB RET server address and the RTCP bandwidth use SD&S Broadcast Discovery Records as well as RSI. The LMB RET server address may also be signaled via DHCP. If the RSI packets are received, the HNED shall use the information in these packets so overriding SD&S or DHCP.

F.6 Retransmission Format and RTP Retransmission Session SSRC and transport address

F.6.1 Retransmission Format

The HNED shall support RET packets with a 2-byte RET payload format header containing the original sequence number as defined in RFC 4588 [86]. This format increases the IP packet size in two bytes and this should be considered when packetizing original media to avoid IP level fragmentation.

The sequence number in the RTP header of the retransmission packet shall follow the standard definition of being one higher than the sequence number of the preceding packet sent in the retransmission stream. The retransmission packet timestamp in the RTP header shall be set to the timestamp of the original packet. The PT in the RTP header of the RET packets is dynamically assigned and configured in the HNED. The RTP payload section in the RET packet is identical to the original packet.

Optionally a different format for the RET packet may be used where the retransmission RTP packet is an identical copy of the original RTP packet with all fields in the RTP header having identical values, including the SSRC and RTP PT field. The support for this RET format by the HNED is optional. When this RET format is used, session multiplexing for the original RTP and RTP RET stream shall be used with different transport addresses. This format shall be explicitly configured in the HNED via SD&S or RTSP with the XML "dvb-original-copy-ret" attribute.

When the identical copy, non RFC 4588 [86] compliant, format is used the packet loss and jitter statistics in the RTCP Receiver Report will not reflect the original stream characteristics. This means that when RTCP RRs are used:

- The HNED that requests the retransmission of an RTP packet using an RTCP FB message shall count that packet as a lost packet in RTCP Receiver Reports, even if the RTP packet is subsequently received.
- The HNED which requests the retransmission of an RTP packet using an RTCP FB message shall not include that packet when computing the value for the "interarrival jitter" field in RTCP Receiver Reports, even if the RTP packet is subsequently received.

NOTE: These rules are the same as defined in [i.10].

In addition the following rules are added for RET enabled LMB services to take into account any RTCP FF messages transmitted by the LMB RET server:

- The HNED that received an RTCP FF message, indicating the loss of an RTP packet, shall count that packet as a lost packet in RTCP Receiver Reports, even if the RTP packet is subsequently received.
- The HNED that received an RTCP FF message, indicating the loss of an RTP packet, shall not include that packet when computing the value for the "inter-arrival jitter" field in RTCP Receiver Reports, even if the RTP packet is subsequently received.

F.6.2 Some Observations on Retransmission Transport Addresses and SSRC Identifiers

F.6.2.1 Unicast services (CoD and MBwTM)

The IP address and port numbers of the original RTP and retransmitted RTP packets are recommended to be identical, resulting in a single RTP session.

The SSRC of the RTP retransmission packets shall be different from the SSRC of the RTP original packets, if the same transport parameters are used for the mandatory RET format as defined in [86]. This allows the HNED to distinguish among RET and original packets based on the SSRC.

NOTE: The "identical copy" RET format does not allow any distinction if the same transport parameters are used.

F.6.2.2 LMB service

When using unicast RET repair for LMB, the SSRC of the RTP retransmission packet can be different from the SSRC of the MC RTP original packets. This is recommended in cases where it is required to distinguish original from RET packets, for example for network monitoring.

When an HNED sends RTCP packets in a RTP RET session, it shall always use an SSRC different from the one it uses for RTCP reporting in the original RTP session.

NOTE: One can overcome typical NAT arrangements like "port restricted cone" (see [87]) and avoid opening an additional "pinhole" in the firewall for the RET RTP and RTCP packets transmitted by the LMB RET server by:

- Using the same port and address for both the destination port/address of the unicast RET RTP packets and source port/address of the RTCP FB messages.
- Use the same port/address for the source port/address of the unicast RET packets and the destination port/address that the HNED uses for its RTCP reporting.
- Multiplex the RET RTP and RTCP on the same port as per [i.7].

In the case that the LMB RET server sends the RTP and RTCP packets in the RTP RET session to the same destination transport address, the combination of the Marker value and expected PT value in the RET RTP packet header will be different from the possible Packet Type values in the RTCP packets. This allows the HNED to distinguish between incoming RTP and RTCP packets in the unicast RTP RET session.

When sent over multicast, the multicast group address of the multicast RET packets may be identical to the one of the original multicast packets, but using a different source IP address.

The source IP address of the MC RTP RET packets may be different from the source IP address of the unicast RTP RET packets so allowing the LMB RET server for unicast repair to be different from LMB RET server for multicast repair. The SSRC of the multicast RET RTP stream can be different from the SSRC of the multicast original RTP stream.

F.7 Retransmission Requesting Behavior of RET-enabled HNED

A RET-enabled HNED will buffer incoming RTP packets over a certain time duration, providing time for packet loss detection, RET requesting and packet repair. The buffering delay is a fixed value and is determined based on some RET timing parameters which are described in this clause separately for CoD/MBwTM and for LMB services.

If an RTP packet is received, which may be a RET packet or an original packet, which carries a payload which was **already** received in a RET packet or in an original packet, than this RTP packet shall be discarded by the HNED. This behavior of the HNED mirrors the behavior of an RTP receiver that shall drop duplicate received RTP packets as defined in [21].

F.7.1 CoD/MBwTM RET (requesting) Timing Parameters

The RET timing parameters for CoD/MBwTM as signaled via RTSP are:

rtx-time

"rtx-time" is the amount of time a packet is available for retransmissions in milliseconds. This is defined in [86]. This value represents a meaningful maximum for the buffering delay as a result of RET.

dvb-t-ret

A RET-enabled HNED may issue multiple retransmission requests for the same packet loss event in the original RTP session, to take into account loss of RET packets or RTCP FB messages.

"dvb-t-ret" is defined as the minimum time in milliseconds a receiver should wait for a requested repair packet per retransmission request before issuing another retransmission request for the same packet(s). This time period has as starting point the sending of the retransmission request. This parameter is optional but the HNED must be able to support this parameter if signaled. If the parameter is not signaled the HNED must then choose an appropriate delay time with which failed retransmissions are retried. This delay time may be based on observing the time that elapses between sending out an RTCP FB for a single packet loss and the time of reception of the requested RET packet at the HNED and could be dynamically adapted (see [86], clause 6.3).

- "rtx-time" will determine how many retransmission attempts the HNED may perform, taking into account the "dvb-t-ret" parameter.
- "dvb-t-ret" is determined by round trip time (RTT) between the HNED and RET server and the RET server performance in processing the HNED generated RTCP FB messages. This could be set to a maximum anticipated value under normal circumstances.

Note that, the following relationship shall be valid:

- "rtx-time" > "dvb-t-ret" > average RTT.

The RET buffer delay shall be chosen by the HNED between:

- minimum value= "dvb-t-ret".
- maximum value= "rtx-time".

An HNED must restrict the maximum bandwidth that is allowed for the RTCP reporting. For this, two parameters are defined:

trr-int

"trr-int" as defined by RFC 4585 [85] defines the minimum interval between two regular full compound RTCP packets in milliseconds for the RTP session. If "trr-int" is not specified, a default value of 0 is assumed.

Full compound RTCP packets include Receiver Reports. RTCP packets containing only FB messages are not subject to the "trr-int" restriction.

rtcp-bandwidth

The "rtcp-bandwidth" XML parameter indicates the maximum amount of bandwidth that can be consumed by the HNED for its RTCP reporting. The default value is 5 % of the original stream bandwidth. If SDP is used, this parameter can be signaled using the "b=RR:<bandwidth-value>" bandwidth modifier as described in RFC 3556 [84].

F.7.2 LMB RET (requesting) Timing Parameters

The RET timing parameters for RET-enabled LMB as distributed by SD&S include those defined in the previous clause F.7.1.

For CoD, a RET-enabled HNED may always transmit an RTCP FB message immediately upon packet loss detection, provided the "rtcp-bandwidth" value is not violated. For LMB, two parameters are defined that determine the window in which the HNED may issue an RTCP FB message upon packet loss detection: "dvb-t-wait-min" and "dvb-t-wait-max". This results in a more scalable LMB RET solution for those cases where a single multicast packet loss event impacts many HNEDs causing NACK storms. An additional bitmask "dvb-ssrc-bitmask" can be configured at the HNED that makes an HNED an immediate reporter by setting the "dvb-t-wait-min" and "dvb-t-wait-max" parameter values to zero.

The RET-enabled HNED shall behave as follows when requesting retransmission for LMB services:

When a packet loss is detected, the HNED sends out a RET request after waiting for an interval time which is randomly selected between "dvb-t-wait-min" and "dvb-t-wait-max." This time interval is called the "waiting time."

"dvb-t-wait-min," "dvb-t-wait-max" and "dvb-ssrc-bitmask" are configured with SD&S. "dvb-t-wait-min" can be zero and "dvb-t-wait-max" can be set equal to "dvb-t-wait-min" and "dvb-t-wait-max" >= "dvb-t-wait-min." The default value for all of these parameters is zero.

The following applies when SD&S signals "dvb-t-wait-min" with a value different from zero:

The HNED shall not send the RTCP FB message after the waiting period has elapsed with "dvb-t-wait-min" <= waiting period <= "dvb-t-wait-max" if one of the following events occurs during the waiting time:

- An RTCP FF message is received from the LMB RET server advertising the packet loss detected by the HNED.
- The missing original packet(s) is/are received.
- RET packet(s) associated with the missing RTP packet(s) is/are received.

However, if an RTCP FF message was received and "dvb-t-ret" has elapsed since the reception of the first RTCP FF message then the HNED may issue an RTCP FB message, provided all of the following conditions are met:

- No additional RTCP FF message arrived reporting the same packet loss as the first received RTCP FF message.
- The expected RET packets have not arrived.
- The original missing packets have not arrived.

As long as RTCP FF messages are received from the LMB RET server reporting the same loss, the HNED shall not send an RTCP FB message.

If an RTCP FF message was received by the HNED prior to the packet loss detection by the HNED, the HNED shall wait for at least "dvb-t-ret" starting from the reception of the FF message before issuing itself an RTCP FB message, if still required.

For non-zero "dvb-t-wait-min" values signaled by SD&S, a 32-bit wide **bitmask** (signaled as dvb-ssrc-bitmask) may also be used to cope with packet loss occurring downstream of the LMB RET server affecting several HNEDs serviced by the server.

The "dvb-ssrc-bitmask" makes a small subset of HNEDs immediate reporters, and at least one RTCP FB message issued by these HNEDs may be translated and forwarded by the LMB RET server as an RTCP FF message downstream over the RET SSM to all HNEDs (see clause F.5.2).

If the "dvb-ssrc-bitmask" is provided by SD&S, the RET-enabled HNED shall compare, on a bit per bit basis, its own 32-bit SSRC which it uses in the original multicast RTP session with the 32-bit SSRC carried in the original MC RTP stream, on those bit positions that have the value 1 in the bit mask. If the two SSRC identifiers have the same values on all the non-zero "dvb-ssrc bitmask" bit positions, the HNED is an immediate reporter, which means that its "dvb-t-wait-min" and "dvb-t-wait-max" equal zero, and hence overrule any SD&S signaled "dvb-t-wait-min" and "dvb-t-wait-max" parameter values.

As the HNEDs select their SSRC randomly, on average the ratio of immediate reporting HNEDs to all RET-enabled HNEDs is (1/ (2^N)) with N the number of "1"s in the bit mask (0 <= N <= 32). For example, if there are two non-zero values in the bit mask, this means that on average 1 out of 4 HNEDs will be an immediate reporter.

If the "dvb-t-wait-min" parameter is configured to be zero or not provisioned in SD&S then the LMB RET server should issue no RTCP FF messages. When packet loss detection occurs, the HNED shall send out an RTCP FB message after a time interval randomly selected between 0 and "dvb-t-wait-max."

F.8 Configuration method and configuration parameters

RET-enabled LMB use SD&S to configure the RET client while CoD uses RTSP to configure the RET client. The exception to this is the initial IP address of the LMB RET servers that can be configured in three different ways:

1) DHCP RTP Retransmission Server Address option

   DHCP should be used at start up to get a list of IP addresses of LMB RET servers as described in clause 8.1.1.10. These IP addresses are the same for all LMB services. The servers shall be in the order of priority from first to last server to connect to. The method for connecting to the server and assuring its operation is vendor specific.

2) SD&S

   SD&S may also contain LMB RET server addresses which can be specified per LMB service. These addresses overrule the LMB RET server address obtained from DHCP for the specific LMB service where SD&S contains a value.

3) RSI messages

   The RSI messages with sub report block type equal to 0,1 or 2 that may be distributed in the RET SSM group to signal the new address of an LMB RET server. The LMB RET server address signaled in an RSI is only valid for a specific SSM group (LMB service), being the original SSM group associated with the RET SSM. The LMB RET server address signaled through RSI takes precedence over the LMB RET server address(es) that may be configured via SD&S for that specific service, and also takes precedence over the LMB RET server address(es) that may be configured via DHCP.

If SD&S records are updated with new LMB RET server addresses after an RSI message then the new SD&S values for the LMB RET server will take preference over addresses in the RSI message.

Note that LMB RET server addresses signaled via DHCP or RSI can be different for different access service regions as they can be distributed locally via the DHCP server or via the operational LMB RET server (RSI).

F.9 QoS Priority settings

The RTP RET packets take over video bearer priority of corresponding original RTP packets (which is DSCP 0b100010 or 0b100100). All RTCP packets issued by the HNED have voice/video signaling priority setting (DSCP 0b011010). The RTCP packets that are transmitted to the HNEDs are considered to be of the video bearer traffic type with appropriate priority setting.

F.10 DVB RET and AL-FEC services combined

The DVB Application Layer FEC and the DVB Application Layer RET are services that both protect for packet losses. They are defined separately and there is no dependency between the two solutions.

If a SP would like to use both packet loss recovery methods for the same LMB and/or CoD/LMBwTM service the FEC and RET protection services may be combined.

The RET mechanism as explained in this document, is then applied only to the original RTP data streams. RET on FEC repair streams shall never be used so the RTCP FB messages always identify a packet missing from the original RTP stream rather than the FEC repair stream.

F.11 Mapping of DVB-specific RET attributes and parameters in SDP

"dvb-t-ret" and "dvb-disable-rtcp-rr" may be included in the SDP description both for LMB and for CoD services.

"dvb-t-ret" can only be specified as media level parameter in the SDP description in the m-line associated with the original RTP packet flow.

"dvb-disable-rtcp-rr" is a session or media level attribute both for the original RTP and the unicast retransmission RTP flows.

For RET-enabled LMB services the following media level parameters may be included in the SDP file in the m-line associated with the original RTP session: "dvb-t-wait-min" , "dvb-t-wait-max", "dvb-ssrc-bitmask" , "dvb-ssrc-upstream-client" ,"dvb-rsi-mc-ret" and "dvb-enable-bye".

Examples of SDP descriptions for a RET-enabled CoD and for a RET-enabled LMB RTP session- including the DVB RET parameters/attribute defined in this annex- will be provided in the guidelines document TS 102 542 v1.3.1 [i.10].

Annex G (normative): CDS Related Information

G.1 CDS Related Extensions to Other Specifications

CDS Announcement requires extension of the BCG as well as extension of TVA. The BCG OnDemandProgram Type and the on-demand decomposed binary locator are extended in order to differentiate between streaming and download modes and with content download specific information. A new BCG type *PushDownloadType* is introduced. Relevant specifications are expected to be updated in their next releases. To provide a consistent CDS specification in the present document, these extensions are collected in this clause.

G.1.1 Usage and Extensions of OnDemandProgramType for pull download service

The OnDemandProgramType as defined in TS 102 822-3-1 [60], clause 6.4.2 may be used to announce CDS-based delivery of content items in pull download service mode. This clause defines the usage and extensions for this type in order to support CDSs.

The XML syntax for the extended OnDemandProgramType is defined in clause G.1.1.

G.1.1.1 Delivery Mode Extension

In order to indicate the different modes of content delivery a new attribute is introduced as an extension to the OnDemandProgramType: the DeliveryMode attribute signals the delivery mode, namely streaming or download.

DeliveryMode	This attribute indicates the delivery mode. It can have the values "streaming" or "download". If the attribute is not provided streaming delivery is assumed.

If the DeliveryMode is not present, or the DeliveryMode signals "streaming", then a streaming delivery service as specified in TS 102 539 [62], clause 6.7 is defined.

If the DeliveryMode is present and it signals "download", a pull download service mode is defined. The other attributes of the OnDemandProgramType shall be interpreted as defined in the following clauses.

G.1.1.2 Usage of TVA OnDemandProgramType attributes for CDS pull download

If the DeliveryMode signals the download mode, then the OnDemandProgramType attributes shall be used as specified in this clause.

All attributes that are defined in TS 102 822-3-1 [60], clause 6.4.2 for the OnDemandProgramType are applicable for CDSs.

The following attributes have a specific usage for delivery mode "download":

ProgramURL	This element specifies a URI for the content download session. This URI can be a unicast or a multicast URI to download session description in SDP or XML format (see clause 10.3.2).
StartOfAvailability	This element specifies the time and date from which on the content item is available for download. If this parameter is not provided the content is already available for download and the value of "now" shall be assumed.
EndOfAvailability	This element specifies the time and date from which the content is no longer available for download. If not present, then the value of "indefinitely" shall be assumed.
NOTE: In case of scheduled multicast download start and end of availability may be the same indicating that a HNED shall join the multicast session only at that specified date and time.	

G.1.1.3 Content Version Number Extension

The ContentVersion attribute allows to signal updated versions of a downloadable content item. A new version of a content item may for example be issued in case of errors in the files of the content item that prevent the correct play out of the content item.

ContentVersion	The attribute indicates the version of the downloadable content. The version number counts from 0 to 255 with an overflow from 255 to 0.
NOTE: This content version number is not intended to be used to signal changed versions of the content item itself (e.g. new, removed or updated scenes). Such changes will result in a new content item with a different CRID.	

If the DeliveryMode signals the download mode, the attribute shall be used as defined in clause 10.6.6 of the present document.

If the DeliveryMode signals the streaming mode, then the ContentVersion attribute shall be ignored by the HNED.

If the parameter is not provided a content version number of 0 shall be assumed.

G.1.1.4 Expiry Time Extension

If the DeliveryMode signals the download mode the ExpiryTime attribute allows the SP to define an expiry time for the downloaded content item. A HNED SHALL automatically remove the content item from the CDS HNED storage at the expiry time (see clause 10.7).

ExpiryTime	The attribute defines the time when the content item expires and shall be removed from the CDS HNED storage.

If the DeliveryMode signals the streaming mode, then the ExpiryTime attribute shall be ignored by the HNED.

If the parameter is not provided no expiry time shall apply to the downloaded content item.

G.1.1.5 Early Play Out Indication Extension

If the DeliveryMode signals the download mode the EarlyPlayout attribute allows to indicate if the play out of the content item can start while the download is still ongoing, i.e. before the content item is completely available on the CDS HNED.

EarlyPlayout	The attribute indicates if the play out of the content item can start while the download is still ongoing. If EarlyPlayout is "true" play out MAY start while the content item is downloaded. If EarlyPlayout is "false" play out SHALL NOT start before the content item is completely downloaded.

If the DeliveryMode signals the streaming mode, then the EarlyPlayout attribute shall be ignored by the HNED.

If the parameter is not provided it shall be assumed as "false".

G.1.1.6 Extended OnDemandProgramType XML Schema

```
<annotation>
  <documentation xml:lang="en">Extended OnDemandProgramType for TM-IPI CDS</documentation>
</annotation>

<simpleType name="DeliveryModeType">
  <restriction base="string">
    <enumeration value="streaming" />
    <enumeration value="download" />
  </restriction>
</simpleType>

<complexType name="OnDemandProgramType">
  <complexContent mixed="false">
    <extension base="tva:ProgramLocationType">
      <sequence>
        <element minOccurs="0" name="PublishedDuration" type="duration" />
        <element minOccurs="0" name="StartOfAvailability" type="dateTime" />
        <element minOccurs="0" name="EndOfAvailability" type="dateTime" />
        <element minOccurs="0" name="FirstAvailability" type="tva:FlagType" />
        <element minOccurs="0" name="LastAvailability" type="tva:FlagType" />
        <element minOccurs="0" name="ImmediateViewing" type="tva:FlagType" />
        <element minOccurs="0" maxOccurs="1" name="DeliveryMode" type="tva:DeliveryModeType" />
        <element minOccurs="0" maxOccurs="1" name="ContentVersion" type="unsignedByte" />
        <element minOccurs="0" maxOccurs="1" name="ExpiryTime" type="dateTime" />
        <element minOccurs="0" maxOccurs="1" name="EarlyPlayout" type="tva:FlagType" />
      </sequence>
      <attributeGroup ref="tva:fragmentIdentification" />
      <attribute name="metadataOriginIDRef" type="tva:TVAIDRefType" use="optional" />
      <attribute ref="xml:lang" use="optional" />
      <attribute name="serviceIDRef" type="tva:TVAIDRefType" use="optional" />
    </extension>
  </complexContent>
</complexType>
```

G.1.2 PushDownloadType for CDS push download service

G.1.2.1 Background and Semantics

A new type PushDownloadType is added to TV Anytime, TS 102 822-3-1 [60]. The PushDownloadType initiates download and storage of the referenced content item to the CDS HNED storage. Subject to any filtering criteria that may be applied by the HNED, the CDS HNED SHALL autonomously join the announced download session and download the content item to its local storage for any content item that is announced by the PushAction to be stored. Metadata related to this content item may be provided together with the content data as part of the download (see clause 10.4). The PushDownloadType is part of the instance description metadata and based on the ProgramLocationType defined in TS 102 822-3-1 [60], clause 6.4.2. The extension of the ProgramLocationType is provided in clause G.1.3.

Table G.1: PushDownloadType

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
Program	TVA CRID reference type defined in TS 102 822-3-1 [60], clause 6.4.2.	M
ProgramURL	URI defined in TS 102 822-3-1 [60], clause 6.4.2. This element specifies a locator for the content download session description for the content item. The locator can be a unicast or a multicast URI to a download session description in SDP or XML format (see clause 10.3.2).	O
InstanceMetadataID	TVA instance metadata ID type as defined in TS 102 822-3-1 [60], clause 6.4.2.	O
InstanceDescription	TVA instance description type as defined in TS 102 822-3-1 [60], clause 6.4.2.; can contain information on the content item (e.g. title, genre, AV attributes)	O
PublishedDuration	The advertised duration of the programme as defined in TS 102 822-3-1 [60], clause 6.4.2.	O

Element/Attribute Name	Element/Attribute Description	Mandated/ Optional
StartOfAvailability	Time and date from which on the content item is available for download. If this parameter is not provided the content is already available for download and the value of "now" shall be assumed.	O
EndOfAvailability	Time and date from which the content is no longer available for download. If not present, then the value of "indefinitely" shall be assumed.	O
ContentVersion	The attribute indicates the version of the downloadable content. The version number counts from 0 to 255 with an overflow from 255 to 0 (see clause G.1.1.3).	O
ExpiryTime	The attribute defines the time when the content item expires and shall be removed from the CDS HNED storage (see clause G.1.1.4).	O

G.1.2.2 PushDownloadType XML Schema

```
<annotation>
  <documentation xml:lang="en">New PushDownloadType for TM-IPI CDS</documentation>
</annotation>

<complexType name="PushDownloadType">
  <complexContent mixed="false">
    <extension base="tva:ProgramLocationType">
      <sequence>
        <element minOccurs="0" name="PublishedDuration" type="duration" />
        <element minOccurs="0" name="StartOfAvailability" type="dateTime" />
        <element minOccurs="0" name="EndOfAvailability" type="dateTime" />
        <element minOccurs="0" maxOccurs="1" name="ContentVersion" type="unsignedByte" />
        <element minOccurs="0" maxOccurs="1" name="ExpiryTime" type="dateTime" />
      </sequence>
      <attributeGroup ref="tva:fragmentIdentification" />
      <attribute name="metadataOriginIDRef" type="tva:TVAIDRefType" use="optional" />
      <attribute ref="xml:lang" use="optional" />
      <attribute name="serviceIDRef" type="tva:TVAIDRefType" use="optional" />
    </extension>
  </complexContent>
```

G.1.3 Extended ProgramLocationTableType

The ProgramLocationType as defined in TS 102 822-3-1 [60], clause 6.7.1 is extended in order to include the PushDownloadType as a valid program location.

G.1.3.1 PushDownloadProgram Extension

The PushDownloadProgram includes CDS push download announcements into the BCG program location table. The attribute is of the type PushDownloadType.

PushDownload	A list of content items pushed to the user device at the request and under the control of the SP

G.1.3.2 Extended ProgramLocationTableType XML Schema

```
  <annotation>
    <documentation xml:lang="en">Extended ProgramLocationType for TM-IPI CDS</documentation>
  </annotation>

  <complexType name="ProgramLocationTableType">
    <sequence>
      <element minOccurs="0" maxOccurs="unbounded" name="Schedule" type="tva:ScheduleType" />
      <element minOccurs="0" maxOccurs="unbounded" name="BroadcastEvent"
type="tva:BroadcastEventType" />
      <element minOccurs="0" maxOccurs="unbounded" name="OnDemandProgram"
type="tva:OnDemandProgramType" />
      <element minOccurs="0" maxOccurs="unbounded" name="OnDemandService"
type="tva:OnDemandServiceType" />
      <element minOccurs="0" maxOccurs="unbounded" name="PushDownload" type="tva:PushDownloadType"
/>
    </sequence>
    <attribute name="metadataOriginIDRef" type="tva:TVAIDRefType" use="optional" />
    <attribute ref="xml:lang" use="optional" />
  </complexType>
```

G.1.4 Extended On-demand decomposed binary locator

An Extended On-demand decomposed binary locator is introduced in order to provide the necessary parameters for pull download services. The Extended On-demand decomposed binary locator is based on the On-demand decomposed binary locator defined in TS 102 323 [59], clause 7.3.2.3.5.

The locator format for the Extended On-demand decomposed binary locator is 0x04. The syntax of the Extended On-demand decomposed binary locator is a superset of the On-demand decomposed binary locator to include CDS-based service modes.

The syntax of the Extended On-demand decomposed binary locator is defined in table G.2.

Table G.2: Extended On-demand decomposed binary locator

Syntax	No. of bits	Identifier
extended_on-demand_decomposed_binary_locator() {		
reserved	6	uimsbf
availability_start_date	9	uimsbf
availability_end_date	9	uimsbf
availability_start_time	16	uimsbf
availability_end_time	16	uimsbf
content_version	8	uimsbf
expiry_time	16	uimsbf
expiry_date	9	uimsbf
reserved	1	uimsbf
delivery_mode	1	bslbf
Early_playout	1	bslbf
URI_length	12	uimsbf
for (i=0; i<URI_length; i++) {		
URI_byte	8	uimsbf
}		
}		

delivery_mode: The delivery mode for the content. The supported modes are:

0: Streaming

1: Download

The same semantics as specified in clause G.1.1.1 shall apply.

If the delivery_mode is 0, i.e. streaming mode, then the semantics for the fields shall be identical to the ones specified in TS 102 323 [59], clause 7.3.2.3.5. The **content_version**, **early_playout**, **expiry_date** and **expiry_time** fields shall be set to 0 and shall be ignored by the receiver.

If the delivery_mode is 1, i.e. download mode, then the following semantics apply:

availability_start_date: The first date on which the on-demand content pointed to by this locator becomes available download delivery. This field uses Universal Co-ordinated Time (UTC) as the time reference. It shall be encoded as the number of days from the beginning of the year indicated by the year_offset field in the enclosing structure. The value zero indicates the 1st of January of that year.

NOTE 1: The size of this field allows the encoded date to extend into the year following that encoded in the year_offset field.

availability_end_date: The first date on which the content pointed to by this locator is no longer available for download delivery. This field uses Universal Co-ordinated Time (UTC) as the time reference. It shall be encoded as the number of days from the beginning of the year indicated by the year_offset field in the enclosing structure. The value zero indicates the 1st of January of that year.

NOTE 2: The size of this field allows the encoded date to extend into the year following that encoded in the year_offset field.

availability_start_time: The time at which the on-demand content pointed to by this locator becomes available for download delivery. This field uses UTC as the time reference. This is encoded as the number of 2 s periods since midnight.

availability_end_time: The first time at which the on-demand content pointed to by this locator is no longer available for download delivery. This field uses UTC as the time reference. This is encoded as the number of 2 s periods since midnight.

NOTE 3: The duration of the on-demand content is signaled elsewhere in the TV-Anytime metadata and therefore does not need to be encoded in the locator.

content_version: The version of the downloadable content. The version number counts from 0 to 255 with an overflow from 255 to 0. The receiver shall interpret this field in the same way as the ContentVersion attribute specified in clause G.1.1.

early_playout: This flag indicates if the content play out can start while the download is still ongoing.

early_playout = "1": play out MAY start before the content item is completely downloaded.

early_playout = "0": play out SHALL NOT start before the content items is completely downloaded

expiry_date: The date on which the on-demand content pointed to by this locator expires and shall be deleted from the local storage of the HNED (see clause 10.7). This field uses Universal Co-ordinated Time (UTC) as the time reference. It shall be encoded as the number of days from the beginning of the year indicated by the year_offset field in the enclosing structure. The value zero indicates the 1st of January of that year.

NOTE 4: The size of this field allows the encoded date to extend into the year following that encoded in the year_offset field.

expiry_time: The time at which the on-demand content pointed to by this locator expires and shall be deleted from the local storage of the HNED (see clause 10.7). This field uses UTC as the time reference. This is encoded as the number of 2-second periods since midnight.

URI_length: The number of URI_bytes present in the following field.

URI_bytes: A sequence of bytes representing a URI compliant string. The string shall include a valid URI scheme at the start. The URI points to a download session description according to clause 10.3.2.

G.1.5 ProgramURL and Locator URIs for files located on CDS HNED storage

Media files being part of content items located on the CDS HNED storage (e.g. after a successful push download) shall be referenced from the BCG metadata (e.g. Program URL of OnDemandProgramType) and locators (e.g. URI locator, URI of Extended On-demand decomposed binary locator) by using the URI scheme defined in clause 10.3.3.

G.2 SDP syntax

This annex defines the SDP syntax of the CDS session descriptions parameters defined in clause 10.5.3. The SDP syntax is defined based on RFC 4566 [76]. CDS specific usage of the standard SDP parameters defined in RFC 4566 [76] and new CDS specific parameters are defined.

For unicast download sessions the SDP grouping framework as defined in RFC 3388 [82]

is used if alternative server locations for a file have to be defined. A new CDS specific semantic for groups is introduced. FLUTE specific SDP parameters for multicast download sessions are based on the definitions in TS 102 472 [66], clause 6.1.3.

G.2.1 SDP message structure

A single SDP message contains the description for a single CDS download session of a content item, which includes one or more files to be downloaded. The SDP message provides all the necessary information to locate and download these files. A CDS download session is completed when all the referenced files have been downloaded.

The SDP starts with a session level section followed by one or more media descriptions as defined below. The order of the lines shall be used as defined in RFC 4566 [76].

G.2.2 General parameters

Clause 10.5.3 defines general parameters that have to be supported for CDS. They apply to any type of CDS download session. This clause defines how they are mapped to the standard SDP parameters in table G.3. In addition new CDS specific SDP parameters are defined.

Table G.3: CDS usage of standard SDP parameters

SDP Line	RFC 4566 [76] attribute definition	DVB CDS usage
Protocol Version	v=0	Mandatory as in RFC 4566 [76]
Origin	o=<username> <sess-id> <sess-version> <nettype> <addrtype> <unicast-address>	mandatory <username> is set to the Service-Provider-ID <sess-id> is set to the Download-Session-ID <sess-version> is set to the Download-Session-Version <nettype> is set to "IN" <addrtype> is set to "IP4" (see note) <unicast-address> as in RFC 4566 [76]
Session Name	s=<session-name>	Mandatory as in RFC 4566 [76]
Session Description	i=<session-description>	optional as in RFC 4566 [76]
URI	u=<URI>	not used
Email Address	e=<email-address>	not used
Phone Number	p=<phone-number>	not used
Connection Data	c=<nettype> <addrtype> <connection address>	mandatory <nettype> is set to "IN" <addrtype> is set to "IP4" (see note) <connection-address> see unicast and multicast download parameters for specific usage
Bandwidth	b=<bwtype>:<bandwidth>	optional see multicast download parameters below for specific

SDP Line	RFC 4566 [76] attribute definition	DVB CDS usage
		usage
Timing	t=<start-time> <stop-time>	mandatory The parameter provides the Download-Session-Time-Information.
Repeat Times	r= <repeat interval> <active duration> <offsets from start-time>	not used
Time Zone	z=<adjustment time> <offset> <adjustment time> <offset>	not used
Encryption Keys	k=<method>:<encryption key>	not used
Attributes	a=<attribute>:<value>	None of the standard attributes defined in RFC 4566 [76] are used. Grouping specific attributes as defined in RFC 3388 [82] are used. For CDS specific attributes see below
Media description	m=<media> <port> <proto> <fmt> ...	mandatory see unicast and multicast download parameters for specific usage
NOTE: The current version of the present document supports only IP version 4.		

NOTE: The session name and session description can be freely defined within the scope of their definitions in RFC 4566 [76].

G.2.3.1 SP domain, download session ID and download session version

The SP name, download session ID and download session version shall be provided at the session level.

The <username> field of the "o=" line is used for the Service-Provider-Domain.

The <sess-id> field of the "o=" line is used for the Download-Session-ID.

The <sess-version> field of the "o=" line is used for the Download-Session-Version.

The usage of the attribute is:

```
o=<Service-Provider-Domain> <Download-Session-ID> <Download-Session-Version> IN IP4 <unicast-address>

<unicast-address>:  IPv4 address or fully qualified domain name of the server that generated the session
```

G.2.3.2 Content item format

One content item format parameter should to be defined at the session level. The Content-item-format is specified by the following syntax:

```
Content-item-format="a=x-dvb-cds-content-item-format:" c-format

c-format="0"|"1"|"2"|"3"

0: Defined by Content-Type of first file in the list of files in the content item description

1: MPEG-2 Transport Stream

2: MPEG-2 Transport Stream with associated BCG metadata

3: MPEG-2 Transport Stream encapsulated in DVB File Format
```

The usage of the attribute is:

```
a=x-dvb-cds-content-item-format:<Content-Item-Format>

<Content-Item-Format>: format of the content item
```

G.2.3.3 Download session mode

Exactly one download session mode parameter shall be defined at the session level. The Download-Session-Mode is specified by the following syntax:

```
Download-session-mode="a=x-dvb-cds-mode:" d-mode

d-mode="SMD"|"CMD"|"UD"

SMD: Scheduled Multicast Download

CMD: Carousel Multicast Download

UD: Unicast Download
```

The usage of the attribute is:

```
a=x-dvb-cds-mode:< Download-Session-Mode>

< Download-Session-Mode>: mode of download session
```

G.2.3.4 Download session time information

Download session time information shall be provided at the session level. The "t=" line is used to provide Download-Session-Time-Information.

The usage of the attributes is as defined in RFC 4566 [76].

```
t=<Download-Session-Time-Information@Start-Time> [<Download-Session-Time-Information@End-Time>]
```

G.2.3.5 Reception reporting server

One or more reception reporting servers can be provided at the session level. The Reception-Reporting-Server-URI is specified by the following syntax:

```
Reception-Reporting-Server-URI="a=x-dvb-cds-rr-server:" uri

uri=Uniform Resource Identifier as defined in RFC 3986 [80]
```

The usage of the attribute is:

```
a=x-dvb-cds-rr-server:<Reception-Reporting-Server-URI>
```

One "a=x-dvb-cds-rr-server:" line shall be provided per reception reporting server.

G.2.3.6 Reception reporting mode

Reception reporting mode can be provided at the session level. It is specified by the following syntax:

```
Reception-Reporting-Mode="a=x-dvb-cds-rr-mode:" rr-mode

rr-mode="0"|"1"|"2"

0: Content item reporting only

1: Content item and file reporting

2: Content item, file and chunk reporting
```

The usage of the attribute is:

```
a=x-dvb-cds-rr-mode:<Reception-Reporting-Mode>
```

G.2.3.7 Reception reporting offset time and random time period

Reception reporting offset time and random time period can be provided at the session level. They are specified by the following syntax:

```
Reception-Reporting-Time="a=x-dvb-cds-rr-time:" dtime [SP dtime]

dtime= integer representing time in milliseconds
```

The usage of the attribute is:

```
a=x-dvb-cds-rr-time:<Reception-Reporting-Offset-Time> <Reception-Reporting-Random-Time-Period>
```

G.2.4 Unicast download parameters

This clause defines the SDP parameters used for the description of a unicast content download session. First the SDP syntax for the unicast specific CDS download session parameters as defined in clause 10.5.3 is provided. The use of grouping is defined in clause G.2.4.8. The structure of a SDP message for a multicast download session is defined in clause G.2.4.9.

G.2.4.1 File Reference

The files that are downloaded are defined by their <path-absolute> *relative reference*

The file reference absolute is specified by the following syntax:

```
File-Reference="a=x-dvb-cds-file-reference:" Path-Absolute

Path-Absolute= <path-p-absolute> relative reference as defined in clause 10.5.2
```

The attribute shall be provided for each file in the media description. In case grouping is used the attribute is defined in the primary media description (see clause G.2.4.8).

The usage of the attribute is:

```
a=x-dvb-cds-file-reference:<File-Reference>
```

G.2.4.2 File Length

The length of a file is specified by the following syntax:

```
File-Length="a=x-dvb-cds-file-length:" Length

Length = integer defining the length in bytes
```

In case of single server file download the attribute may be provided for the file in the media description. In case of multiple server file download the attribute shall be provided for the file in the media description. In case grouping is used the attribute is defined in the primary media description (see clause G.2.4.8)

The usage of the attribute is:

```
a=x-dvb-cds-file-length:<File-Length>
```

G.2.4.3 File Digest

The MD5 digest of a file is specified by the following syntax:

```
File-Digest="a=x-dvb-cds-file-digest:" digest

digest = base64
```

The attribute may be provided for the file in the media description. In case grouping is used the attribute is defined in the primary media description (see clause G.2.4.8).

The usage of the attribute is:

```
a=x-dvb-cds-file-digest:<File-Digest>
```

G.2.4.4 Chunk Length

The common length of the chunks of a file is specified by the following syntax:

```
Chunk-Length="a=x-dvb-cds-chunk-length:" Length

Length = integer defining the length in bytes
```

In case of multiple server file download the attribute shall be provided for the file in the media description. The attribute is not used for single server file download. The attribute is defined in the primary media description (see clause G.2.4.8).

The usage of the attribute is:

```
a=x-dvb-cds-chunk-length:<Chunk-Length>
```

G.2.4.5 Chunk Digest

The MD5 digest of a chunk and its number (position within the file counted from 1 to n) is specified by the following syntax:

```
Chunk-Digest="a=x-dvb-cds-chunk-digest:" int digest

digest-list = digest 1*[SP digest]
```

In case of multiple server file download the attribute may be provided for each chunk of the file in the media description. The attribute is not used for single server file download. The attribute is defined in the primary media description (see clause G.2.4.8).

The usage of the attribute is:

```
a=x-dvb-cds-chunk-digest:<Chunk-Number> <Chunk-Digest>
```

G.2.4.6 Server Base URI and File Content Type

The base URI of a server location where the file is available shall be provided by c-line and the m-line.

The <connection address> filed of the c-line shall be set to the server address part of the base URI (IPv4 address or fully qualified domain name). The <nettype> field of the c-line shall be set fixed to "IN". The <addrtype> field of the c-line shall be set fixed to "IP4" as only IPv4 is support by this version of the specification.

The <port> field of the m-line shall be set to the optional port part of the base URI. In case the port is not provided it is set to the HTTP default port of 80. The <proto> field of the m-line shall be set fixed to "TCP/HTTP". The <media> field shall be set to the main media type of the content type of the File as defined (e.g. video, application). The <fmt> field shall be set to the subtype of the content type of the download file as defined in clause 10.4.2 (e.g. mp2t, xml). In case the mime type is provided the fields shall be set to "*",

The attributes shall be provided for each file in the media description. In case several server locations are provided for a file grouping shall be used as defined in clause G.2.4.8.

The usage of the attributes is:

```
c=IN IP4 <Server-Base-URI@Address>

m=<File-Content-Type@MainTyp> <Server-Base-URI@Port> TCP/HTTP <File-Content-Type@Subtype>
```

G.2.4.7 Available Chunk List

The list of available chunks of a file at a server location is specified by the following syntax:

```
Available-Chunk-List="a=x-dvb-cds-available-chunk-list:" chunk-list

chunk-list = chunk|chunk-range 1*[SP chunk|chunk-range]

chunk-range = chunk "-" chunk

Chunk = integer
```

In case of multiple server file download the attribute may be provided for each server location. In case it is not provided all chunks of the file are available at that server location. The attribute is not used for single server file download. The attribute is defined in each media description.

The usage of the attribute is:

```
a=x-dvb-cds-available-chunk-list:<Available-Chunk-List>
```

G.2.4.8 Grouping of media lines

The grouping framework as defined in RFC 3388 [82] shall be used in order to define alternative or multiple server locations for a file that has to be downloaded. For each server location a media description shall be used and shall be identified by the mid-attribute as defined in RFC 3388 [82]. The media descriptions that belong to a single file are grouped together using the group-attribute as defined in RFC 3388 [82]. A new semantic "X-DVB-CDS-AS" (alternative server) is defined for the group-attribute. It indicates that the media descriptions of a group define alternative server locations. This can be alternative servers for a single server download or locations of file chunks for multiple server downloads.

The first media description of a group is the primary media description and contains parameters that apply for the whole group (e.g. File-Absolute-Path).

If at least one file has alternative server locations grouping has to be used for all files in the SDP message.

Each group is defined by a group attribute at the session level. The group attributes lists the media descriptions that are members of the group.

```
a=group:X-DVB-CDS-AS 1*<id>
id: token
```

The mid-attribute has to be defined for each media description if grouping is used. It provides a unique identifier for each media description.

```
a=mid:<id>
id: token
```

G.2.4.9 SDP message structure for unicast download session

Each file that has to be downloaded within the session is defined by a media description or a group of media descriptions. Grouping is used in case alternative server locations are provided for a single server file download or for a multiple server file download. File absolute path, mime type, file length, file digest, file chunk length and file chunk digest information is provided per file in the media description. In case or grouping they are provided in the primary media description. Server base URI and available chunk list information is provide per media description.

A unicast download SDP starts with session level fields in the order listed below.

```
v=0
o=<Service-Provider-Domain> <Download-Session-ID> <Download-Session-Version> IN IP4 <unicast-
address>
s=<session name>
i=<session description> (optional)
a=x-dvb-cds-content-item-format:<Content-Item-Format>
a=x-dvb-cds-mode:UD
a=x-dvb-cds-rr-server:<Reception-Reporting-Server-URI> (0..n)
a=group:X-DVC-CDS-AS 1*<id> (0..n)
t=<Download-Session-Time-Information@Start-Time> [<Download-Session-Time-Information@End-Time>]
```

The session level information is followed by one or more media descriptions, one for each file and server location, with fields in the following order:

Media description without grouping and primary media description of a group:

```
m=<File-Content-Type@MainTyp> <Server-Base-URI@Port> TCP/HTTP <File-Content-Type@Subtype>
a=x-dvb-cds-file-reference:<File-Reference>
c=IN IP4 <Server-Base-URI@Address>
a=mid:<id> (in case of grouping)
a=x-dvb-cds-file-length:<File-Length> (optional for single server file download)
a=x-dvb-cds-file-digest:<File-Digest> (optional)
a=x-dvb-cds-file-chunk-length:<File-Chunk-Length> (not used for single server file download)
a=x-dvb-cds-file-chunk-digest:<File-Chunk-Digest> (optional for multiple server file download)
a=x-dvb-cds-available-chunk-list:<Available-Chunk-List> (optional for multiple server file download)
```

All other media descriptions of a group:

```
m=<File-Content-Type@MainTyp> <Server-Base-URI@Port> TCP/HTTP <File-Content-Type@Subtype>
c=IN IP4 <Server-Base-URI@Address>
a=mid:<id>
a=x-dvb-cds-available-chunk-list:<Available-Chunk-List> (optional)
```

Example of a unicast download session description with a single server file download of two files with no alternative server locations and no reception reporting:

```
v=0
o=provider.org 1234 1 IN IP4 135.27.66.45
s= Example1
i= Example session 1
a=x-dvb-cds-content-item-format:1
a=x-dvb-cds-mode:UD
t= 3034423619 3042462419
m=video 80 TCP/HTTP mp2t
c=IN IP4 server.provider.org
a=x-dvb-cds-file-reference:/content/video1.mp2t
a=x-dvb-cds-file-length:160607052
a=x-dvb-cds-file-digest: Q2hlY2sgSW50ZWdyaXR5IQ==
m=application 80 TCP/HTTP xml
c=IN IP4 server.provider.org
a=x-dvb-cds-file-reference:/content/meta1.xml
```

Example of a unicast download session description with a multiple server and a single server file download with two alternative server locations and reception reporting:

```
v=0
o=provider.org 1240 1 IN IP4 135.27.66.45
s= Example2
i= Example session 2
a=x-dvb-cds-content-item-format:1
a=x-dvb-cds-mode:UD
a=x-dvb-cds-rr-server:rr1.provider.org
a=x-dvb-cds-rr-server:rr2.provider.org
a=group:X-DVB-CDS-AS 1 2
a=group:X-DVB-CDS-AS 3 4
t= 3034423619 3042462419
m=video 80 TCP/HTTP mp2t
c=IN IP4 server1.provider.org
a=mid:1
a=x-dvb-cds-file-reference:/content/video1.mp2t
a=x-dvb-cds-file-length:160607052
a=x-dvb-cds-file-digest: Q2hlY2sgSW50ZWdyaXR5IQ==
a=x-dvb-cds-file-chunk-length:50000000
a=x-dvb-cds-file-chunk-digest: We14fgTT5DSwqGH44fGzr3== aa3f4GHj76fGHCB86AqwDD==
asdd23EsWQ65DFVmLkkJS8== ASq34fDD5gRGdSSw34D214==
a=x-dvb-cds-available-chunk-list:1-2 3
m=video 80 TCP/HTTP mp2t
a=mid:2
c=IN IP4 server2.provider.org
a=x-dvb-cds-available-chunk-list:1 3 4
m=application 80 TCP/HTTP xml
c=IN IP4 server3.provider.org
a=mid:3
a=x-dvb-cds-file-reference:/content/meta1.xml
m=application 80 TCP/HTTP xml
c=IN IP4 server4.provider.org
a=mid:4
```

G.2.5 Multicast download parameters

This clause defines the SDP parameters used for the description of a multicast content download session. The same SDP structure is used for scheduled and carousel multicast download. FLUTE is used for multicast download and the SDP FLUTE parameters defined in TS 102 472 [66], clause 6.1.3 are used for the description of the FLUTE specific information. First the SDP syntax for the multicast specific CDS download session parameters as defined in clause 10.5.3 is provided. The structure of a SDP message for a multicast download session is defined in clause G.2.5.13.

G.2.5.1 File Reference

For each file within the Flute session that has to be downloaded the <path-absolute> *relative reference* (see clause10.5.2) of the file can be provided at the session level. If non such parameter is provided all files of the Flute session have to be downloaded to the HNED. The same "`a=x-dvb-cds-file-REFERENCE:`" attribute as defined in clause G.2.4.1 for the unicast file reference is used.

Note, that in contrast to the definition in clause G.2.4.1 the line is used at the session level and all files indicated by such lines have to be downloaded.

The usage of the attribute is:

```
a=x-dvb-cds-file-reference:<File-Reference>
```

G.2.5.2 Multicast channel source address

The source address of the FLUTE multicast channels is provided at the session level by the "a=source-filter" attribute as defined in TS 102 472 [66], clause 6.1.3.1.1.

```
a=source-filter: incl IN IP4 * <IP-Source-Address>
```

G.2.5.3 Transport Session Identifier

The transport session identifier is provided at the session level by the "a=flute-tsi:" attribute as defined in TS 102 472 [66], clause 6.1.3.1.4.

```
a=flute-tsi:<Transport-Session-Identifier>
```

G.2.5.4 FEC Encoding ID

The FEC encoding ID is provided at the session level by the "a= FEC-declaration:" attribute as defined in TS 102 472 [66], clause 6.1.3.1.6. The fec-inst-id parameter is not used. The fec-ref parameter is set to "0" as the "a=fec:" line per media is not supported.

If the FEC encoding ID is not provided the compact no-code FEC scheme is assumed.

```
a=FEC-declaration:0 <FEC-Encoding-ID>
```

G.2.5.5 Numbers of channels

The number of channels is provided at the session level by the "a=flute-ch:" attribute as defined in TS 102 472 [66], clause 6.1.3.1.2.

```
a=flute-ch:<Number-Of-Channels>
```

G.2.5.6 Multicast Address

The address of each multicast channel is provided by a c-line per media description. The <nettype> and <addrtype> fields of the c-line are fixed values as this version of the specification supports only IPv4.

c=IN IP4 <IP-Multicast-Address>

G.2.5.7 Multicast Port Number

The port number of each multicast channel is provided by a m-line per media description. The <media>, <proto> and <fmt> field of the m-line are fixed Flute specific values.

```
m=application <IP-Multicast-Port-Number> flute/udp *
```

G.2.5.8 Maximum bandwidth

The maximum bandwidth used by each multicast channel can be provided by a b-line per media description. The TIAS bandwidth modifier as defined in RFC 3890 [81] shall be used.

```
b=TIAS <Max-Bandwidth>
```

G.2.5.9 Completion poll response server address and port number

Completion poll information is provided at the session level. The completion poll response server address and port number are provided by the following syntax:

```
Completion-Poll-Server="a=x-dvb-cds-cp-server:" nettype SP addrtype SP unicast-address SP port
    nettype=network type; fixed value "IN" as only IP networks are supported by the current
        specification
    addrtype=address type; fixed value "IP4" as only IPv4 is supported by the current
    specification
    unicast-address=IP address of completion poll response server
port=server port for completion poll response
```

The usage of the attribute is

```
a=x-dvb-cds-cp-server:IN IP4 <Completion-Poll-Response-Server-Address> <Completion-Poll-Response-
Server-Port-Number>
```

G.2.5.10 Recovery server base URI

One or more recovery server base URIs can be provided at the session level. The recovery server base URI is provided by the following syntax:

```
Recovery-Server="a=x-dvb-cds-rec-server:" uri
    uri= Uniform Resource Identifier as defined in RFC 3986 [80]
```

The usage of the attribute is:

```
a=x-dvb-cds-rec-server:<Recovery-Server-Base-URI>
```

One "a=x-dvb-cds-recovery-server:" line shall be provided per recovery server.

G.2.5.11 Recovery mode

Recovery mode can be provided at the session level. It is provided by the following syntax:

```
Recovery-Mode="a=x-dvb-cds-rec-mode:" rec-mode
    rec-mode="0"|"1"
    0: CDS file repair mode
    1: IPDC like file repair mode
```

The usage of the attribute is

```
a=x-dvb-cds-rec-mode:<Recovery-Mode>
```

G.2.5.12 Recovery offset time and random time period

Recovery offset time and random time period can be provided at the session level. They are specified by the following syntax:

```
Recovery-Time="a=x-dvb-cds-rec-time:" dtime [SP dtime]
    dtime= integer representing time in milliseconds
```

The usage of the attribute is:

```
a=x-dvb-cds-rec-time:<Recovery-Offset-Time> <Recovery-Random-Time-Period>
```

G.2.5.13 SDP message structure for multicast download session

The list of files that have to be downloaded from the FLUTE session as part of the CDS session is provided by "a=x-dvb-cds-file:" lines at the session level, one line per file. If this information is not provided all files of the FLUTE session have to be downloaded.

Each multicast channel of the FLUTE session is defined by a media description which starts with an "m=" line and is terminated by either the next "m=" line or by the end of the session description.

A multicast download SDP starts with session level fields in the order listed below.

```
v=0
o=<Service-Provider-Domain> <Download-Session-ID> <Download-Session-Version> IN IP4 <unicast-
address>
s=<session name>
i=<session description> (optional)
a=x-dvb-cds-content-item-format:<Content-Item-Format>
a=x-dvb-cds-mode:CMD|SMD
a=x-dvb-cds-rr-server:<Reception-Reporting-Server-URI> (0..n)
a=x-dvb-cds-rr-time:<Reception-Reporting-Offset-Time> <Reception-Reporting-Random-Time-Period>
(optional)
a=source-filter: incl IN IP4 * <IP-Source-Address>
a=flute-tsi:<Transport-Session-Identifier>
a=FEC-declaration:0 <FEC-Encoding-ID> (optional)
a=flute-ch:<Number-Of-Channels>
a=x-dvb-cds-cp-server:IN IP4 <Completion-Poll-Response-Server-Address> <Completion-Poll-Response-
Server-Port-Number> (optional)
a=x-dvb-cds-rec-server:<Recovery-Server-Base-URI> (0..n)
a=x-dvb-cds-rec-mode:<Recovery-Mode> (optional)
a=x-dvb-cds-rec-time:<Recovery-Offset-Time> <Recovery-Random-Time-Period> (optional)
a=x-dvb-cds-file-reference:<File-Reference> (0..n)
t=<Download-Session-Time-Information@Start-Time> [<Download-Session-Time-Information@End-Time>]
```

The session level information is followed by one or more media descriptions, one for each multicast channel, with fields in the following order:

```
m=application <IP-Multicast-Port-Number> flute/udp *
c=IN IP4 <IP-Multicast-Address>
b=TIAS <Max-Bandwidth> (optional)
```

Example of a carousel multicast download session with 2 FLUTE channels without reception reporting and file repair. All files of the FLUTE sessions are downloaded:

```
v=0
o=provider.org 5678 2 IN IP4 135.27.66.45
s= Example3
i= Example session 3
a=x-dvb-cds-content-item-format:0
a=x-dvb-cds-mode:CMD
a=source-filter: incl IN IP4 * 135.27.66.40
a=flute-tsi:1234765
a=flute-ch:2
t= 3034423619 3042462419
m=application 1200 flute/udp *
c=IN IP4 227.124.5.3
b=TIAS 500
m=application 1200 flute/udp *
c=IN IP4 227.124.5.4
b=TIAS 1000
```

Example of a scheduled multicast download session with 3 FLUTE channels with Raptor FEC, reception reporting and file repair. The listed files are downloaded:

```
v=0
o=provider.org 6123 1 IN IP4 135.27.66.45
s= Example4
i= Example session 4
a=x-dvb-cds-content-item-format:1
a=x-dvb-cds-mode:SMD
a=x-dvb-cds-rr-server:rr1.provider.org
a=x-dvb-cds-rr-server:rr2.provider.org
a=x-dvb-cds-rr-time:100 50
a=source-filter: incl IN IP4 * 135.27.66.40
a=flute-tsi:123456
a=FEC-declaration:0 1
a=flute-ch:3
a=x-dvb-cds-cp-server:IN IP4 135.27.60.4 300
a=x-dvb-cds-rec-server:rec1.provider.org
a=x-dvb-cds-rec-mode:0
a=x-dvb-cds-rec-time:200 30
a=x-dvb-cds-file-reference:/content/video1.mp2t
a=x-dvb-cds-file-reference:/content/meta1.xml
t= 3034423619
m=application 1200 flute/udp *
c=IN IP4 227.124.6.3
b=TIAS 500
m=application 1200 flute/udp *
c=IN IP4 227.124.6.4
b=TIAS 1000
m=application 1200 flute/udp *
c=IN IP4 227.124.6.5
b=TIAS 1000
```

G.3 DVB-MCAST URI scheme

The DVB-MCAST URI scheme is defined to identify resources provided via an IP multicast channel. It provides a means to locate the multicast channel carrying the resource and also to specify information concerning the application layer transport protocol which will be used to carry the data over that multicast channel (e.g. SAP, DVBSTP).

Clause G.3.1 defines the basic scheme. Clauses G.3.2 and G.3.3 define the specific extensions and usage of the scheme for referencing download session descriptions as defined in clause 10.5 using the DVBSTP and SAP protocols.

Additional application layer transport protocols specific extensions and usage might be defined in the future or in other specifications that make use of the DVB-MCAST URI scheme.

G.3.1 Basic DVB-MCAST URI scheme

The basic DVB-MCAST URI scheme defined in this clause provides the client with the information required to join an IP multicast channel. Only the minimum set of parameters required by a multicast connection protocol like IGMP (RFC 3376 [47]) are included in the scheme. By optionally providing the type of the application layer transport protocol, the client will be able to provide the data from the multicast channel to the appropriate application. The scheme might be extended for application layer transport protocol specific usage.

The basic DVB-MCAST URI scheme is defined as follows:

```
'dvb-mcast://' [ src-host '@' ] mcast-addr ':' port ['?payload=' PayloadID]

src-host            = source host (for source specific multicast)
mcast-addr          = multicast address
port                = port
PayloadID           = payload-type
payload-type        = "sap" | "dvbstp"
```

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

The src-host is an optional syntax element referring to the unicast IP address of the source of the multicast data. This is only meaningful in case Source Specific Multicast (SSM) as defined in RFC 4607 [105] is supported.

G.3.2 DVB-MCAST URI scheme for DVBSTP

The basic DVB-MCAST URI scheme defined in G.3.1 is extended in order to reference DVBSTP protocol specific elements, namely specific SPs, PTs and segments transported. The DVBSTP SP ID, Payload ID and Segment ID are defined as part of the Query component of the URI as they provide non-hierarchical information to locate a specific segment distributed on the DVBSTP multicast channel.

Note that a session version number is not provided in the URI scheme as always the latest version of a segment distributed over the multicast channel shall be used.

In order to reference a specific session description within the XML segment the CDS Download Session ID can be provided within the fragment part of the URI. The fragment syntax is not specific to the DVBSTP delivery but to the delivered media, the CDS XML session description in this case.

The DVB-MCAST URI scheme for DVBSTP is defined as follows:

```
'dvb-mcast://' [ src-host '@' ] mcast-addr ':' port '?payload=dvbstp' ['&service-provider='
ServiceProviderID] ['&dvbstp-payload=' DVBSTPPayloadID] ['&segment=' SegmentID] ['#? dvb-cds-
session-id=' Download-Session-ID]

src-host            = source host (for source specific multicast)
mcast-addr          = multicast address
port                = port
ServiceProviderID   = IPv4 address
DVBSTPPayloadId     = 2*2 HEXDIG; any hex number from 0x00 to 0xff
SegmentId           = 4*4 HEXDIG; any hex number from 0x0000 to 0xffff
Download-Session-ID = DecimalString
```

In order to access the specified resource the device has to join the multicast group provided by the mcast-addr and port and optional src-host in the URI. It compares all the parameters provided in the query component of the URI against the corresponding DVBSTP protocol fields and extracts all the segments that match. The parameters in the query component of the URI are optional. In case a parameter is not provided the corresponding field in the DVBSTP protocol will not be used for the comparison.

In case a Download Session ID is provided in the fragment component of the URI, the device has to search all the extracted segments for the session description with the specific Download Session ID.

G.3.3 DVB-MCAST URI scheme for SAP

The basic DVB-MCAST URI scheme defined in G.3.1 is extended in order to reference SDP data provided via the SAP protocol. The PT is set to 'sap'. No further information has to be provided in the query part of the URI as no further payload identification is provided by SAP.

In order to reference a specific session description within the SDP information the CDS Download Session ID can be provided within the fragment part of the URI. The fragment syntax is not specific to the SAP delivery but to the delivered media, the SDP session description in this case.

The DVB-MCAST URI scheme for SAP is:

```
'dvb-mcast://' [ src-host '@' ] mcast-addr ':' port '?payload=sap' ['#? sdp-session-id=' Download-
Session-ID]

Download-Session-ID   = unsigned Integer
```

In case a Download Session ID is provided in the fragment component of the URI, the device has to search all the SDP information delivered over the multicast channel for the session description with the specific Download Session ID.

History

Document history		
V1.1.1	March 2005	Publication
V1.2.1	September 2006	Publication
V1.3.1	October 2007	Publication
V1.4.1	August 2009	Publication