ETSI TS 102 034 V1.3.1 (2007-10)

Technical Specification

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

Reference
RTS/JTC-DVB-206

Keywords
broadcasting, digital, DVB, IP, 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 Services over IP Based Networks"; it is referred to as DVB-IP phase 1.3 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:

- signalling of Logical Channel Numbers; the logical channel number is a number associated with a DVB-IP service indicating the ordering preferred by the Service Provider for presentation in a service list;
- signalling of availability of DVB-IP services based on the region in which the Home Network End Device is located;
- UDP only transport of DVB-IP services;
- optional Application Layer Forward Error Correction for DVB-IP services carried over RTP transport.

As in previous releases of TS 102 034, the DVB-IP phase 1.3 work is limited to DVB services [1] encapsulated in MPEG-2 TS [61] and covers both Live Media Broadcast services (i.e. TV or radio styles) Media Broadcast with Trick Modes and Content on Demand services (CoD) (see also clause 6.2). 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 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-IP 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 Delivery Network Gateway (DNG) per home network segment.
- Clause 9 covers an identification agent for the HNED. This agent provides a simple identification of the HNED to signal its existence to the Network Service Provider if requested.

- Network provisioning. Clause 10 covers an optional network management and provisioning agent. This agent allows a Network Service Provider to automatically provision the end device with additional functionalities. This clause will be deprecated in a future version of the present document when the DVB-IP Remote Management System specification becomes available.

- Discovery of Broadband Content Guides (inc. third party). The Broadband Content Guide itself is provided as a separate specification [71].

- Annex E defines an optional protocol for Application Layer FEC (AL-FEC) protection of streaming media for DVB-IP services carried over RTP transport. This AL-FEC protocol is a layered protocol with a base layer and an enhancement layer. The base layer is a simple packet-based interleaved parity code based on a subset of [76]. The base layer shall be used wherever AL-FEC is used. The enhancement layer is a Raptor code, as defined in [74] and [75] and may optionally be used to provide further packet loss protection.

1.1.2 What is out of the scope

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 [72].

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_102034v010301p0.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.

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

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

2.1 Normative references

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

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

[2] ETSI ETR 211: "Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)".

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

NOTE: ETR 162 is expected to be replaced with TR 101 162 [81].

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

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

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

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

[8] 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".

[9] 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).

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

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

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

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

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

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

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

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

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

[19] IETF RFC 1213: "Management Information Base for Network Management of TCP/IP-based internets: MIB-II".

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

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

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

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

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

[25] IETF RFC 2011: "SNMPv2 Management Information Base for the Internet Protocol using SMIv2".

[26] IETF RFC 2013: "SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2".

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[47] IETF RFC 2863: "The Interfaces Group MIB".

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

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

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

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

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

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

[54] IETF RFC 3203: "DHCP reconfigure extension".

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

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

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

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

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

[60] ISO 8601: "Data elements and interchange formats - Information interchange - Representation of dates and times".

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

[62] ISO/IEC 13818-2 (1996): "Information technology - Generic coding of moving pictures and associated audio information: Video".

[63] ISO/IEC 13818-3 (1998): "Information technology - Generic coding of moving pictures and associated audio information - Part 3: Audio".

[64] 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".

[65] "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.

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

"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.

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

[67] 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".

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

[69] 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 Schema".

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

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

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

[73] UPnP Device Architecture 1.0.

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

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

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

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

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

[79] 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".

2.2 Informative references

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

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

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.

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 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) or storage

NOTE: The user could be a person or a PVR or some other entity.

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

Delivery Network (DN): network connecting the delivery network gateway and service providers

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

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

DVB Service: as defined by DVB, a sequence of programmes under the control of a broadcaster which can be broadcast as part of a schedule

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.

MPEG-2: Refers to ISO/IEC 13818-1

NOTE: Systems coding is defined in ISO/IEC 13818- 1 [61]. The real time interface specification is defined in ISO/IEC 13818- 9 [64].

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 [61].

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.

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-IP services.

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

transport stream: data structure defined in ISO/IEC 13818-1 [61]

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 [61], 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-Nauer Form
ASM	Any Source Multicast
BCG	Broadband Content Guide
BiM	Binary MPEG format for XML
BNF	Backus-Nauer Form
CoD	Content on Demand
CoS	Class of Service
CPU	Central Processing Unit
CSP	Content Service Provider
DHCP	Dynamic Host Configuration Protocol
DNG	Delivery Network Gateway
DNS	Domain Name System
DSCP	Differentiated Services CodePoint
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
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
IPI	Internet Protocol Infrastructure
IPv4	Internet Protocol version 4
ISO	International Organization for Standardization
ISP	Internet Service Provider
LMB	Live Media Broadcast
MAC	Media Access Control
MBwTM	Media Broadcast with Trick Modes
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
NTP	Network Time Protocol
OSI	Open Systems Interconnection
PAT	Program Association Table
PCR	(MPEG-2) Program Clock Reference
PLL	Phased Locked Loop
PMT	Program Map Table
QoS	Quality of Service
RFC	Request For Comments
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SD&S	Service Discovery and Selection
SI	Service Information
SNTP	Simple Network Time Protocol
SOHO	Small Office/Home Office
SP	Service Provider
SSM	Source Specific Multicast
STC	(MPEG-2) System Time Clock
TCP	Transmission Control Protocol
TLS	Transaction Layer Security
ToS	Type of Service
TS	Transport Stream
T-STD	(MPEG-2) Transport Stream System Target Decoder
TV	TeleVision
UDP	User Datagram Protocol
URI	Uniform Resource Indentifier
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

3.3 Notations

3.3.1 Augmented Backus-Nauer Form (ABNF)

The present document uses the Augmented Backus-Nauer Form (ABNF) conform to RFC 2234 [31], 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
deviceID        = manufacturer '/' [model] '/' clientID
manufacturer    = name / 'DVB-IPI P1 Generic'
model           = name
clientID        = 1*(HEXDIGIT)
```

3.3.1.2 Core rules

The following set of ABNF core rules derived from [31] 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 [15], RFC 1101 [17] (First mention of labels starting with digits), RFC 1738 [22] (URL), RFC 2181 [30] (Clarifications), RFC 2396 [36] (Including the optional trailing dot), RFC 2486 [40] (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.

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

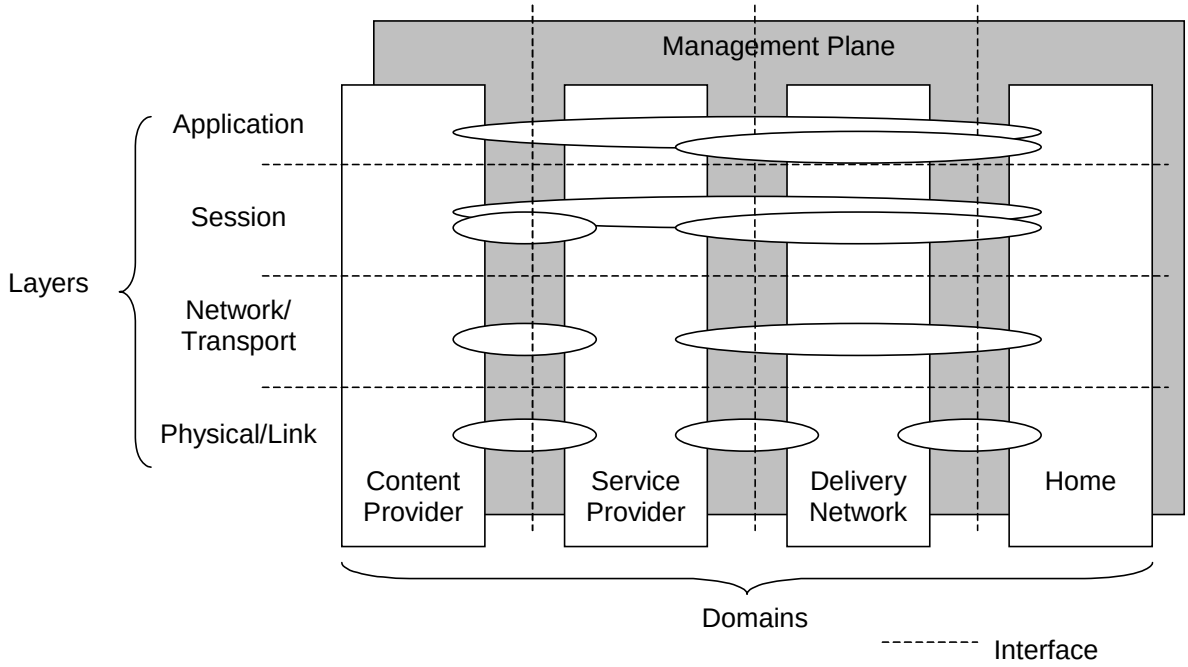

Figure 1: Layer model

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 [80]:

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 [72] In the present document only the delivery of DVB-IP services over broadband delivery networks to DVB-IP HNEDs is defined as shown in figure 2. The advanced home network functionality depicted in the Home Network Reference model [72] will be defined in a separate specification.

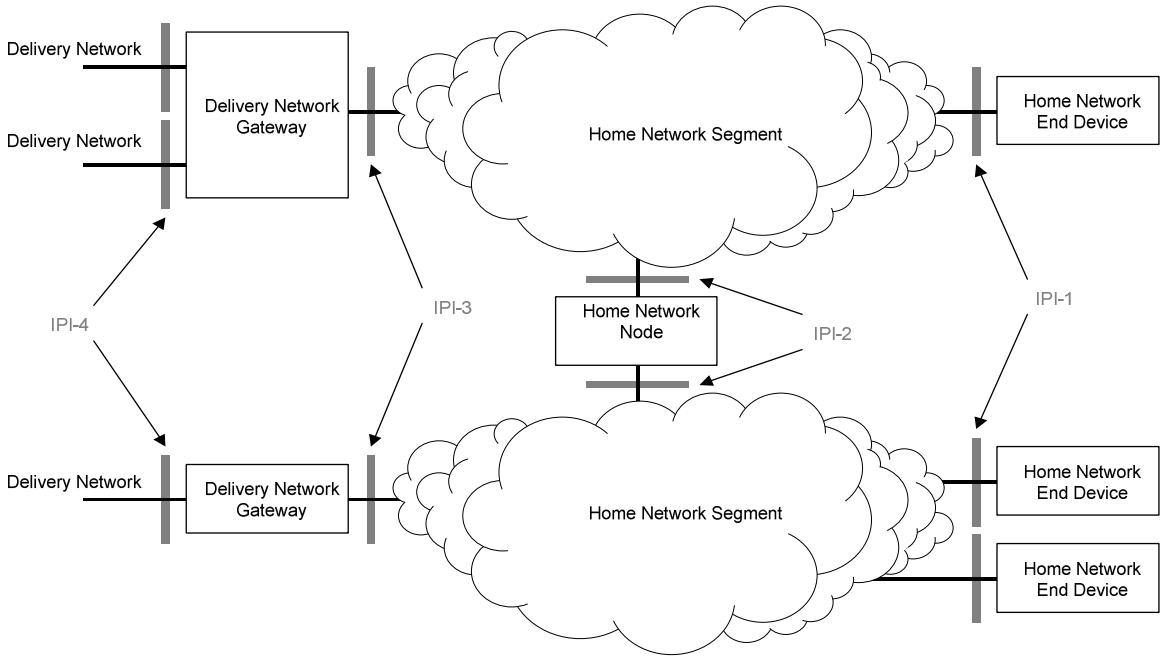

Figure 2: Home Network Reference Model

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.

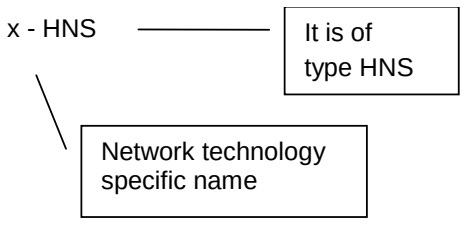

Figure 2a: Home network technology name

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

An Ethernet HNS is based on 100BASE-T as specified in IEEE 802.3 [8].

A Wireless LAN HNS is based on IEEE 802.11 as specified in [9].

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 [5] LANs (below the MAC service boundary) in a bridged format is defined in IEEE 802.1Q [6]. 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 802.1p [10] (see clause 7.4.2).

- 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. This convertion is out of scope of the present document.

- In case the delivery network gateway 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.

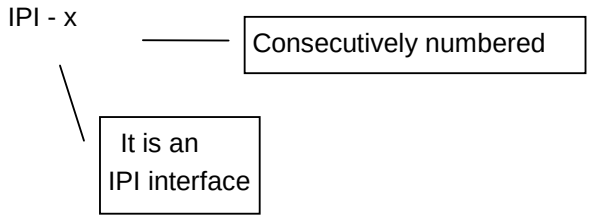

Figure 2b: Interface name

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 [72].

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 [16]. The Address Resolution Protocol as defined in RFC 826 [13] 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 [73] 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-IP 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. The organization of this protocol stack is according to the ISO/OSI layering convention. 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 present document specifies the protocols required for transport of elements of the service offering via IP networking, in principle independent of the physical layers below the IP networking layer.

The HNED is an IP compliant device; on its IPI-1 interface it supports the requirements laid down in RFC 1122 [18]. 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. To enhance clarity, the protocols of each clause are shown with a specific fill colour.

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 control of the delivery of broadcast TV and audio (radio) programs as well as for on-demand delivery. 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 [61]. The resulting MPEG-2 packets are encapsulated directly in UDP or in RTP\UDP, with DSCP packet markings for quality of service. Transport 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.

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.

An identification agent is specified in clause 9. This agent uses HTTP on TCP.

The present document specifies an optional Network Provisioning protocol in clause 10. This protocol is carried on HTTP or secure HTTP over SSL. Since Network Provisioning is optional, HTTPs is also optional on the client interface.

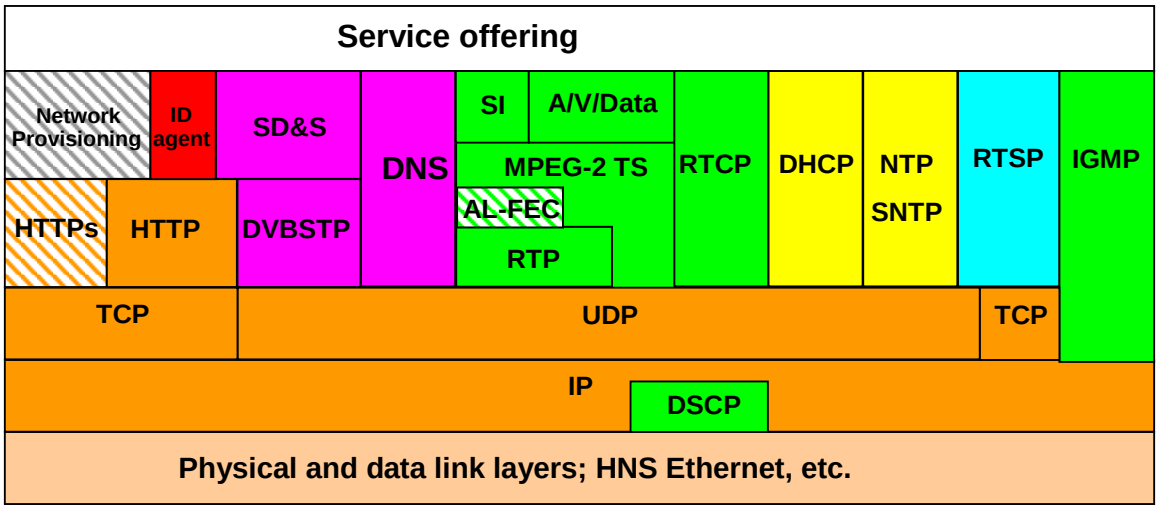

Figure 3: Diagram of the protocol stack for DVB-IP services

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-IPI specifications.

4.2.1 Single delivery network gateway scenario

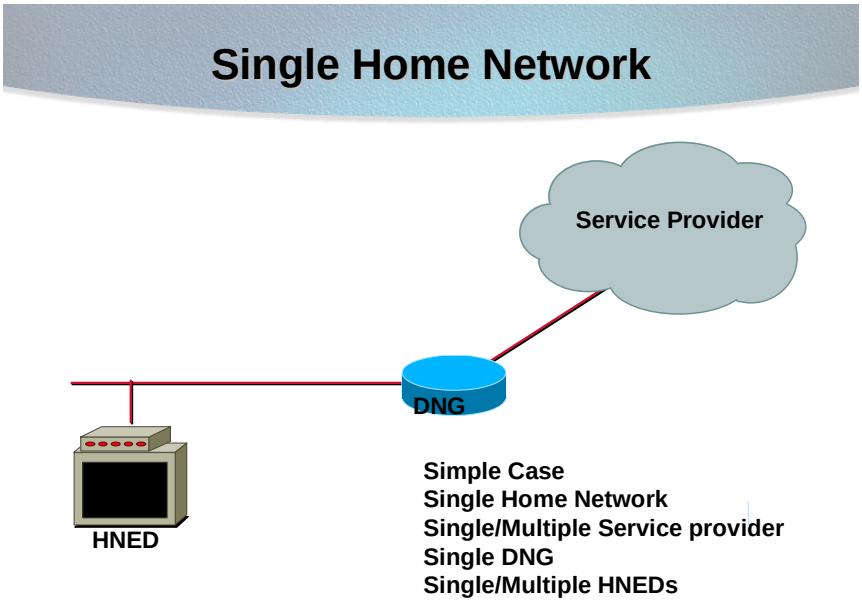

Figure 4: Single home network

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 Delivery Network Gateways (DNG)

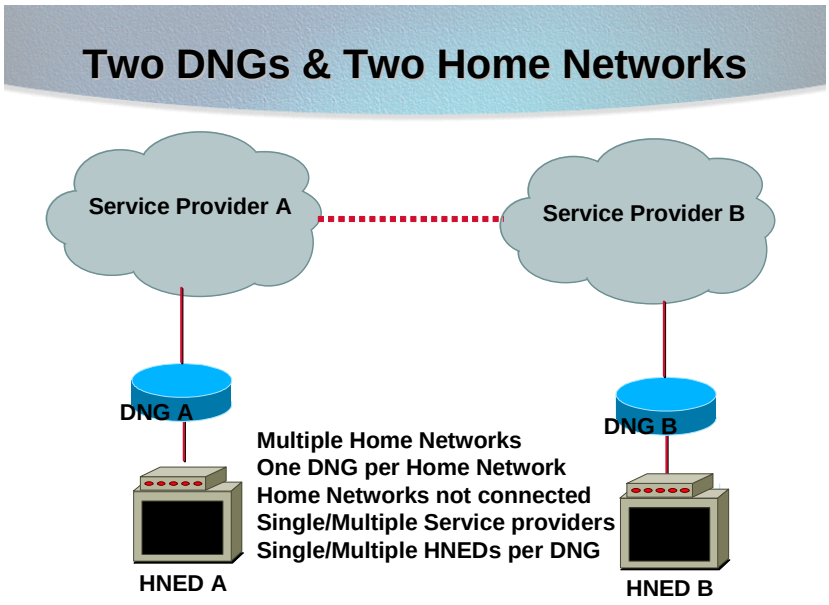

Figure 5: Multiple Delivery Network Gateways (DNG)

In the "Multiple Delivery Network Gateways" scenario the home has multiple Delivery Network Gateways (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 Delivery Network Gateway (DNG) and HNED in One Box

If the Delivery Network Gateways (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-IP 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 and Content on Demand services are both 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.

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 [65] and the XML schemas [66] describing their syntax and grammar are specified in annex C.

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

A Service Provider 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 or more textual identifiers that take 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 service provider) and the service provider's domain name.

The syntax of a textual service identifier is as defined in MHP (clause 14.9 [4]):

<service_name>"."<service_provider_domain_name>

where <**service_name**> is a unique name for the service within the service provider's domain and <**service_provider_domain_name**> is an Internet DNS domain name that the service provider 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 [15].

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

Different types of SD&S information have been identified.

Currently the following types of SD&S information are identified but new ones could be defined as and when needed.

- SD&S information relating to a service provider; and
- four types of SD&S information relating to the service offering of a service provider.

This is to cover the different types of service offering a service provider may have. A Service Provider Offering can be made of services of type Live Media Broadcast ("TS Full SI" or "TS Optional SI") or Content on Demand. The Service Provider can also reference services provided by another service provider 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 service provider may use and give the associated value the payload ID takes.

Table 1: Payload ID values

Payload ID value	SD&S record carried
0x00	Reserved
0x01	Service Provider Discovery Information
0x02	Broadcast Discovery Information
0x03	COD Discovery Information
0x04	Services from other SPs
0x05	Package Discovery Information
0x06	BCG Discovery Information
0x07 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 service provider 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 annex C.

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

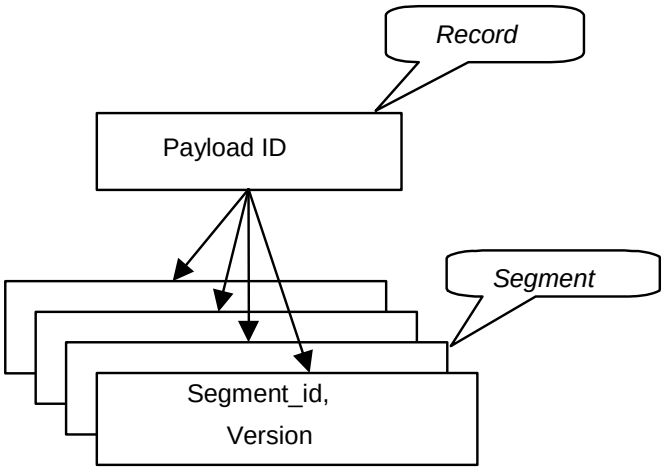

Figure 6: Relationship between records, payload IDs and segments

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 Service Provider 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 service providers offering DVB-IP services over the IP network and continues with the discovery of available services from each service provider. The SD&S data model can be found in the informative 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 Service Providers offering DVB-IP services is done via the acquisition of the Service Provider Discovery Information specified in clause 5.2.5. Service Providers 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.

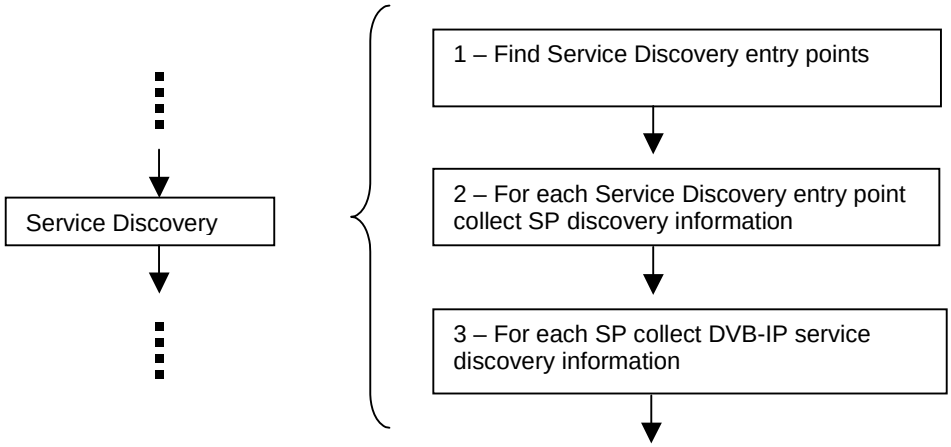

Figure 7: Steps in service discovery

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 [45]. 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 [45]. The DVB organization will maintain the services.dvb.org domain name for service discovery and new service providers 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 [45] 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 [45].
- The HNED can be provided with entry points as part of the configuration data received on a provisioned network. The addresses of the entry points will be specified in the element sdEntry of the configuration DTD see clause 10.2.6.

NOTE: The DNS mechanism as described in RFC 2782 [45] 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 (dvbservdscport) 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) If the Networked Provisioning option is implemented then the SD&S entry point(s) should come from the Configuration DTD element(s) sdEntry. If the element is null then no entry points have been provisioned so the HNED shall go to the next step.

2) 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.

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) 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.

5) If no entry point has been found through the steps above there shall be the option for the user to enter the URL [22] or an IP address and optional portnumber of an entry point manually.

5.2.5 Service Provider discovery information

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

This Service Provider Discovery Record shall be carried in a record containing the information listed in table 2. The Service Provider 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 Service Provider Discovery Information record may aggregate discovery information on several service providers. 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 service providers providing DVB-IP services on the network.

Table 2: Service Provider(s) discovery record

Element / Attribute Name	Element / Attribute Description	Mandated/ Optional/ Conditional
ServiceDiscovery type:	/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	
@DomainName	An internet DNS domain name registered by the Service Provider that uniquely identifies the Service Provider	M
@Version	Version of the Service Provider(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 Service Provider occurs.	M
@LogoURI	Pointer to a Service Provider logo for potential display. The pointer shall be a URI [21].	O
Name	Name of the Service Provider for display in one or more languages; one Service Provider 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 Provider 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 IP Offering Records which describe the offerings that the Service Provider makes available. This element is optional.	O M (see note 1) M (see note 1)
Pull@Location	This URI [21] encodes the location of the DVB IP Offering(s) Records which describe the offerings that the Service Provider makes available.	O
PayloadList type (one entry per payload ID):	/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-IP 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 service provider.	C (see note 2)
Segment@Version	Version number of the segment identified by Segment@ID.	O

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

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-IP 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).

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

5.2.6 DVB-IP service discovery information

5.2.6.1 DVB-IP Offering Record

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

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

Table 3: DVB-IP Offering Record

Element / Attribute Name	Element / Attribute Description	Mandated/ Optional/ Conditional
@DomainName	An internet DNS domain name registered by the Service Provider that uniquely identifies the Service Provider.	M
@Version	Version of the DVB-IP Offering record, the version number shall be incremented every time a change in the DVB-IP Offering record is made.	C (see note)
NOTE: The version number of the DVB-IP 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-IP 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 IP 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-IP 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 Service Provider that uniquely identifies the Service Provider. If this is not present, then the DNS domain name from the DVB IP Offering record is used.	O
TextualIdentifier@ServiceName	A unique host name for the service within the service provider's domain.	M

Element / Attribute Name	Element / Attribute Description	Mandated/ Optional
DVBTriplet@OrigNetId	Identifies the network Id of the originating delivery system.	M
DVBTriplet@TSId	Identifies the Transport Stream.	M
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.	O
ServiceLocation type (one entry per service location):	/BroadcastDiscovery/ServiceList/SingleService/Service Location At least one of IPMulticastAddress or RTSPURL shall be present.	
IPMulticastAddress	Signals the use of IGMP to access the service and provides the multicast address at which the service may be accessed.	O
IPMulticastAddress@Source	Optionally the IP unicast address of the source of the TS may be provided.	O
IPMulticastAddress@Address	Provides the multicast 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
FECBaseLayer	Contains the multicast address and port of the AL-FEC stream. This element shall be present if the FECEnhancementLayer element is present.	O
FECBaseLayer @Address	IP Multicast Address for FEC Base Layer (SMPTE-2022-1 [76]). 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
FECBaseLayer @Source	IP Multicast Source Address for FEC Base Layer (SMPTE-2022-1 [76]). 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
FECBaseLayer @Port	UDP port for FEC Base Layer.	M (see note)
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.	O
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
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
FECEnhancementLayer@Port	UDP port for FEC Enhancement Layer.	M (see note)
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
IPMulticastAddress@FECMaxBlockSizeTime	The maximum transmission duration of any FEC Block (source and repair packets)	O
IPMulticastAddress@FECObjectTransmissionInformation	The FEC Object Transmission Information for the Raptor code. If an FECEnhancementLayer element is included then this element SHALL be included.	O
RTSPURL	Signals the use of RTSP to access the service and provides the URL at which the service may be accessed.	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 [69] and used in TS 102 323 [68]. The classification scheme used for the Coding element shall either be defined by TS 102 323 [68], 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 [67].	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 [69] and used in TS 102 323 [68]. The classification scheme used for the Coding element shall either be defined by TS 102 323 [68], 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 [67].	O
ServiceAvailability	This element provides support for regionalization. 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	This element indicates the country for which the availability is being defined.	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
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 IP 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-IP 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 Service Provider that uniquely identifies the Service Provider. If this is not present, then the DNS domain name from the DVB IP Offering record is used.	O
TextualIdentifier@ServiceName	A unique host name for the service within the service provider"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	O
ServiceLocation type (one entry per service location):	/BroadcastDiscovery/ServiceList/SingleService/Service Location At least one of IPMulticastAddress or RTSPURL shall be present.	
IPMulticastAddress	Signals the use of IGMP to access the service and provides the multicast address 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 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
FECBaseLayer	Contains the multicast address and port of the AL-FEC stream. This element shall be present if the FECEnhancementLayer element is present.	O
FECBaseLayer@ Address	IP Multicast Address for FEC Base Layer (SMPTE-2022-1 [76]). 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
FECBaseLayer@ Source	IP Multicast Source Address for FEC Base Layer (SMPTE-2022-1 [76]). 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
FECBaseLayer@ Port	UDP port for FEC Base Layer.	M (see note)
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.	O
FECEnhancement Layer@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

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 address as the original data.	O
FECEnhancementLayer@Port	UDP port for FEC Enhancement Layer.	M (see note)
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
IPMulticastAddress@FECMaxBlockSizeTime	The maximum transmission duration of any FEC Block (source and repair packets).	O
IPMulticastAddress@FECObjectTransmissionInformation	The FEC Object Transmission Information for the Raptor code If an FECEnhancementLayer element is included then this element SHALL be included.	O
RTSPURL	Signals the use of RTSP to access the service and provides the URL at which the service may be accessed. At least one of IPMulticastAddress or RTSPURL shall be present.	O
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 [69] and used in TS 102 323 [68]. The classification scheme used for the Coding element shall either be defined by TS 102 323 [68], 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 [67].	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 [69] and used in TS 102 323 [68]. The classification scheme used for the Coding element shall either be defined by TS 102 323 [68], 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 [67].	O
ServiceAvailability	This element provides support for regionalization. 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	This element indicates the country for which the availability is being defined.	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
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 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-IP 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 Content on Demand Provider/Server; This Id is allocated by the Service Provider.	M
Name	Name of the Content on Demand 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 Content on Demand general offering catalogue for potential display in one or more languages; one description per language code.	O
Locator	One or more URI [21] where the aggregated content descriptions can be found (catalogue/metadata).	M

An HTTP request on the " Locator" URI [21] 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 Service Provider can reference individual services or a complete offering provided by another service provider. Supplying its textual service identifier references a service. Supplying the service provider's DNS domain name without a service list references an entire service provider's offering. Discovery information relating to a service, or service provider, such as the location of the service will need to be acquired directly from the service provider providing the service, and is not "pointed to" from this record.

The "Services From other Service Providers" Record implements the Services From other Service Providers and linked Service Id, and by inheritance the DVB-IP 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 Service Providers record

Element / Attribute Name	Element / Attribute Description	Mandated/ Optional
ReferencedServices type:	/ServicesFromOtherSP	
@Domain	An internet DNS domain name registered by the referenced Service Provider that uniquely identifies the service provider being referenced.	M
Service@Name	A unique host name for the service within the referenced service provider's domain for each service from the referenced provider. Not required if the entire set of offerings from the service provider 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-IP 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 Service Provider	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 Service Provider that uniquely identifies the Service Provider. If this is omitted the Service Provider Domain Name from the inherited DVB-IP Offering is used.	O
TextualID@ServiceName	A unique host name for the service within the service provider's domain.	M
DVBTriplet	The DVB triplet by which the service may be known.	O
DescriptionLocation	The URI [21] of additional service description provided in the context of a package; this is not required to acquire a service.	O
LogicalChannelNumber	The logical channel number of the service.	O
PackageAvailability	This element provides support for regionalization. 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.	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 service provider, such as the location of the service will need to be acquired directly from the service provider 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 content on demand. A provider discovered through this shall offer a service as described in TS 102 539 [71].

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 Service Provider	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, content on demand, 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 domainName 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
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

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

An HNED is located geographically in a region which is identified by the Service Provider using a string identifier 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 is used, in conjunction with a country code, in the PackageAvailability and ServiceAvailability elements to indicate to the HNED which package and services can be received by the HNED.

The way in which the HNED determines the cell ID and country code for the region where it is located is not defined. This mechanism will be defined in a later release of the present document.

5.3 Service Selection

A 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, service providers 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 [35], 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-IP 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 specifies which methods to use.

5.4 Transport mechanisms

This clause specifies the protocols that are used to transport the Service Provider 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 [44] 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).

5.4.1.1 Syntax

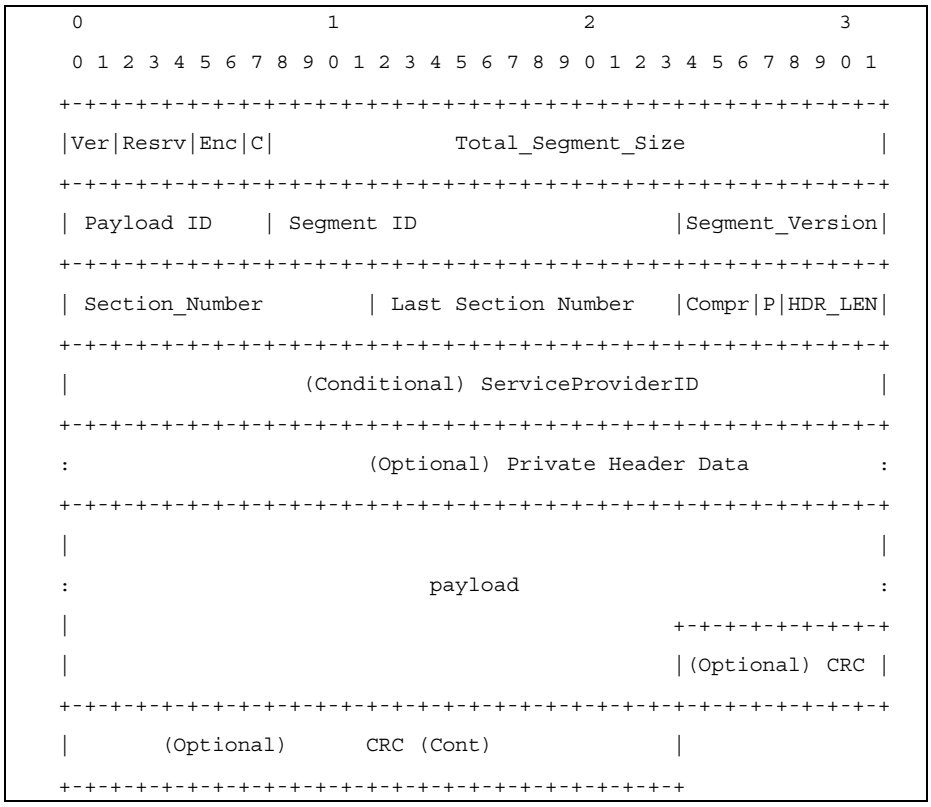

Figure 8: Syntax SD&S multicast delivery protocol

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 payload type (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 10.

Table 10: 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 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 service provider. This number shall be an IPv4 address, as detailed in clause 5.4.1.3. It is the responsibility of the Service Provider 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 Service Provider ID field is mandatory unless the provider knows that no other Service Providers 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 13818-1 [61], 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.

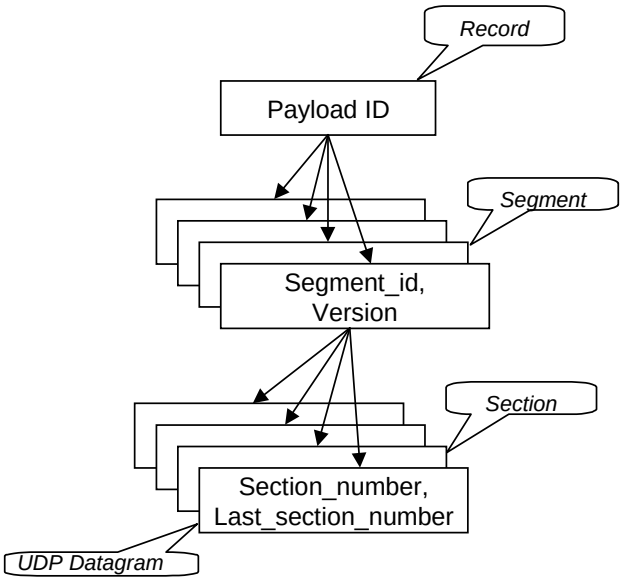

Figure 9: Relationship between records, segments and sections

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, 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 [12]) 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 Service Provider 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 ETR 162 [3] 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 [44] 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 service providers; or
- **service_discovery_request** for a request for discovery information relating to the service offering of a service provider.

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 [44].

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 9.2.

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 [44]).

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

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=DomainName 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 service provider. The request has three mandatory parameters which take the domain name of the service provider, 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 [44] 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 [44]; 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       = OCTET; any hex number from 0x00 to 0xff
SegmentId       = 4*4 HEXDIG;any hex number from 0x0000 to 0xffff
SegmentItem     = SegmentId 0*1('&'VersionNumber)
```

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

```
VersionNumber = OCTET; any hex number from 0x00 to 0xff
```

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

```
'GET /dvb/sdns/service_discovery?id=DomainName&Payload=02&Segment=0001 HTTP/1.1' CRLF

    'Host: ' host CRLF
```

The host contains the IP address of the service discovery server of the service provider; this address is obtained by resolving the URL contained in the field "PullURL" as documented in clause C.2.15 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 service provider offering or the service provider 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 Service Provider.

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-IP 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-IP offering record.

5.5 Encoding

5.5.1 Introduction

SD&S segments may be encoded with BiM [70]. 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 Service Provider 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 [68] 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 [68]) 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:ipi:sdns:2006. DVBContextPath of additional schemas are specified by the ContextPathCode in [70].

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 [68] (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 [68] is updated with the following values.

Value	Description	EquivalentStartType
0x0030	serviceDiscovery	sdns:ServiceDiscovery type

where: sdns = urn:dvb:ipi:sdns:2006

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 [69]), for decoding string data. This will be signalled in the DecoderInit using the ClassificationScheme "urn:tva:metadata:cs:CodecTypeCS:2004" defined in [79].

6 RTSP Client

6.1 Usage of RTSP in DVB

In this clause the use of the *Real Time Streaming Protocol* (RTSP) [35] for a playback capable HNED is specified.

NOTE: A recording capable HNED is not specified, as DVB-IPI has decided to address playback only for Phase 1.

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 RTSP URL for accessing the RTSP based service in question. 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 URL to access the service. The URL indicates whether the session control is based on RTSP. When this is the case, the HNED shall use RTSP to access the service in question.

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. For example, the server can use the persistent connection to send asynchronously RTSP ANNOUNCE messages (see table 11) to the HNED.

Multimedia streams, encapsulated as described in clause 7, can be transmitted from the RTSP server in either unicast or multicast mode. However, in multicast mode trick mode operation like *paus*e, *fast forward* and similar can obviously 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 11). Or, the HNED can also poll the server with the aid of a DESCRIBE method (see table 11) 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.

The ANNOUNCE and DESCRIBE methods use the XML description as described in clause 5 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: DVB-IPI has decided not to specify security and authentication for 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 profile contains a subset of the methods and headers defined in the RTSP protocol. The relationship between the 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.

6.2.2 Live media broadcast

The Live Media Broadcast 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 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 Content on Demand Profile is that the user cannot initiate it.

6.2.4 Content on demand

The Content on Demand 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.

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.

6.3 RTSP methods

For unicast mode of delivery, table 11 specifies for each profile the RTSP methods to be supported by the IPI-1 interface.

Table 11: RTSP methods for unicast mode

RTSP Method	**Direction: H = HNED; S = Server;**	**IETF**	**DVB Requirement**	
			LMB	**MBwTM and CoD**
ANNOUNCE	H→S	MAY	MAY	MAY
ANNOUNCE	S→H	MAY	**SHOULD**	**SHOULD**
DESCRIBE	H→S	SHOULD	SHOULD	SHOULD
GET_PARAMETER	H→S	MAY	**SHOULD**	**SHOULD**
GET_PARAMETER	S→H	MAY	MAY	MAY
OPTIONS	H→S	SHALL	SHALL	SHALL
OPTIONS	S→H	MAY	MAY	MAY
PAUSE	H→S	SHOULD	**N.A.**	**SHALL**
PLAY	H→S	SHALL	SHALL	SHALL
REDIRECT	S→H	MAY	**SHALL**	**SHALL**
SETUP	H→S	SHALL	SHALL	SHALL
TEARDOWN	H→S	SHALL	SHALL	SHALL
NOTE 1: The column IETF presents the methods required to be supported according to the IETF RTSP specification: RFC 2326 [35]. The DVB requirement columns present the methods required to be supported for each given DVB profile. NOTE 2: The key words in bold indicate where the DVB specification differs from the IETF. NOTE 3: The RTSP methods RECORD and SET_PARAMETER are not supported.				

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 as described in clause 5.2.6 for the broadcast profiles (LMB and MBwTM) and clause 5.2.6.3 for CoD. For the broadcast profiles the ANNOUNCE method shall contain the BroadcastOffering XML complex structure (see clause C.4.2 BroadcastOffering).

The MIME Type in the Content-Type header (see table 13) 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 [52] on XML Media Types. The HNED shall always include `text/xml` in the Accept header.

6.3.1.2 DESCRIBE

The DVB RTSP client is required to support the reception of descriptions in XML format as described in clause 5.2.6 for the broadcast profiles (LMB and MBwTM) and clause 5.2.6.3 for CoD. For the broadcast profiles the DESCRIBE method shall contain the `BroadcastOffering` XML complex structure (see clause 5.2.6).

The MIME Type in the Content-Type header for such a message shall be `text/xml` and the content of the Content-Encoding header and XML description shall be UTF-8. See RFC 3023 [52] on XML Media Types. The HNED shall always include `text/xml` in the Accept header.

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. Parameter 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 12 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 12: 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 13 presents the RTSP header fields that are generated by the HNED and are either mandatory or recommended for the IPI-1 interface.

Table 13: 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	
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` For additional transport parameters to support AL-FEC when RTP streaming is used see clause 6.3.2.2.
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 [35]. 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 13.			

Table 14 presents the RTSP header fields that are supported by the HNED (either mandatory or recommended) on the IPI-1 interface.

Table 14: 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.
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. For additional transport parameters to support AL-FEC when RTP streaming is used see clause 6.3.2.2.
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 [35]. The DVB requirement columns present the response headers required to be supported for each given DVB profile.
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 14.

6.3.2.2 Transport header extensions

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

MP2T/H2221/UDP or RAW/RAW/UDP

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

The following additional options within the Transport header are defined for Application Layer FEC when RTP streaming is used:

dvb_fec_base_address (optional)

> In the multicast case, this option may be included in messages from server to client. It indicates the IP Multicast address on which the base AL-FEC layer may be found. If not included then the base AL-FEC layer is sent on the same multicast address as the source data.

dvb_fec_base_source (optional)

> In the multicast case, this option may be included in messages from server to client. It indicates the IP Multicast source address on which the base AL-FEC layer may be found. If not included then the base AL-FEC layer is sent on the same multicast source address as the source data.

dvb_fec_base_port (optional)

> This option may be included in messages from server to client and from client to server. It indicates the UDP destination port for the base AL-FEC layer. When included in a message from client to server, it indicates that AL-FEC is supported by the client and specifies the destination UDP port that should be used for the AL-FEC base layer. When included in a message from server to client, it indicates the UDP destination port that the server will use for the AL-FEC base layer. In the multicast case, if this option is specified but the dvb_base_fec_address is not, then the AL-FEC layer is assumed to be available on the same multicast address as the main stream. If this header is not specified then AL-FEC is not provided and the dvb_fec_base_address shall not be present. This header shall be present if the dvb_fec_enhance_port header is present.

dvb_fec_enhance_address (optional)

> In the multicast case, this option may be included in messages from server to client. It indicates the IP Multicast address on which an enhancement AL-FEC layer may be found. This option may be repeated to specify multiple enhancement layers.

dvb_fec_enhance_source (optional)

> In the multicast case, this option may be included in messages from server to client. It indicates the IP Multicast source address on which an enhancement AL-FEC layer may be found. This option may be repeated to specify multiple enhancement layers.

dvb_fec_enhance_port (optional)

> This option may be included in messages from server to client and from client to server. It indicates the UDP destination port for the enhancement AL-FEC layer. When included in a message from client to server, it indicates that the AL-FEC ienhancement layer is supported by the client and specifies the destination UDP port that should be used for the AL-FEC enhancement layer. When included in a message from server to client, it indicates the UDP destination port that the server will use for the AL-FEC enhancement layer. In the multicast case, if this option is specified but the dvb_enhance_fec_address is not, then the AL-FEC enhancement layer is assumed to be available on the same multicast address as the main stream. If this header is not specified then AL-FEC enhancement is not provided and the dvb_fec_ enhance _address shall not be present. This header shall only be present if the dvb_fec_base_port header is present.

dvb_fec_max_block_size (optional)

> 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).

dvb_fec_max_block_time (optional)

This indicates the maximum sending duration of any AL-FEC block in milliseconds.

dvb_fec_oti (optional)

This indicates the FEC Object Transmission Information for the Raptor AL-FEC layer(s).

These parameters are defined as an extension to the Transport header as follows (extensions are in bold):

```
Transport           =    "Transport" ":"
                         1\#transport-spec
transport-spec      =    transport-protocol/profile[/lower-transport]
                         *parameter *dvb-parameter
transport-protocol  =    "RTP" | "RAW" | "MP2T"
profile             =    "AVP" | "RAW" | "H2221"
lower-transport     =    "TCP" | "UDP"

dvb-parameter       =    ";" "dvb-fec-base-address" "=" address
                    |    ";" "dvb-fec-base-source" "=" port
                    |    ";" "dvb-fec-base-port" "=" port
                    |    ";" "dvb-fec-enhance-address" "=" address
                    |    ";" "dvb-fec-enhance-source" "=" address
                    |    ";" "dvb-fec-enhance-port" "=" port
                    |    ";" "dvb-fec-max-block-size" "=" 1*5DIGIT
                    |    ";" "dvb-fec-max-block-time" "=" 1*5DIGIT
                    |    ";" "dvb-fec-oti" "=" *BASE64DIGIT

BASE64DIGIT         =    ALPHA | DIGIT | "+" | "/" | "="
```

6.4 Status codes in response to requests

Table 15 lists the RTSP and HTTP status codes that the RTSP enable HNED shall be able to interpret.

Table 15: 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"*
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"*

Status Code	Description
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 [55] is used for all such messages on the IPI-1 interface.

The transport configuration parameters: `destination` and `source` (see table 14) 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 [55]).

NOTE 3: RFC 2326 [35] 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 14.

For multicast mode of delivery, table 16 presents the RTSP methods to be supported by the IPI-1 interface.

Table 16: 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 13 and 14).
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

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 the service. 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.

7.1 Transport stream encapsulation

The present document can be used to encapsulate any TS 101 154 compliant MPEG-2 Transport Stream (MTS) [67], 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 compliant [67], and shall be encapsulated either in RTP (Real-time Transport Protocol) according to RFC 3550 [23] in conjunction with RFC 2250 [34] or directly in UDP (User Datagram Protocol) according to ITU-T Recommendation H.610 [78].

7.1.1 Real-time Transport Protocol (RTP) encapsulation

RTP always uses an even UDP port number (see RFC 3550 [23]). If the end device is supplied with an odd number for use as the RTP port, it should replace this number with the next lower (even) number. The corresponding RTCP stream uses the next higher (odd) port number.

Each IP packet [12] 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.

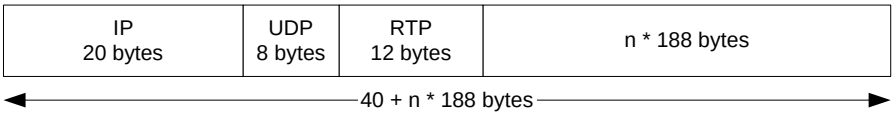

Figure 10: Minimal packet format (IPv4) for RTP encapsulation

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 [8] frame with LLC) or 1 500 bytes (IEEE 802.3 [8] frame without LLC, see IEEE 802.3 [8] and IEEE 802.2 [7]), 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 [12]) 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 [11]).

The RTP header is shown in figure 11.

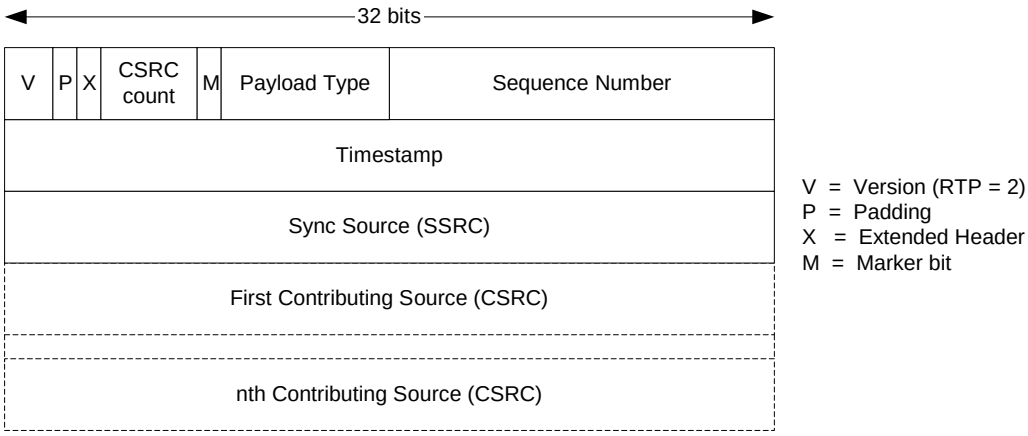

Figure 11: RTP header format

The payload type shall be set to MP2T (33), as specified in RFC 1890 [24].

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 [61].

Other fields are completed as per RFC 3550 [23] and RFC 2250 [34]. 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 [41]) 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.

The associated RTP stream will always use an even UDP port number (see RFC 3550 [23]). The RTCP stream uses the next higher (odd) port number.

RTCP defines two separate message sets. 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). 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 Receiver Reports. This decision is based on scalability. 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 seconds.

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) [20].

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 [78].

Each IP packet [12] 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.

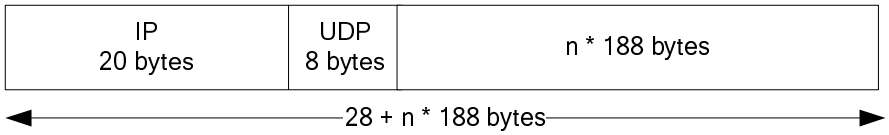

Figure 12: Minimal packet format (IPv4) for UDP encapsulation

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 [8] frame with LLC) or 1 500 bytes (IEEE 802.3 [8] frame without LLC, see IEEE 802.3 [8] and IEEE 802.2 [7]), 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 [12]) requires that all hosts shall be prepared to accept datagrams of at least 576 octets.

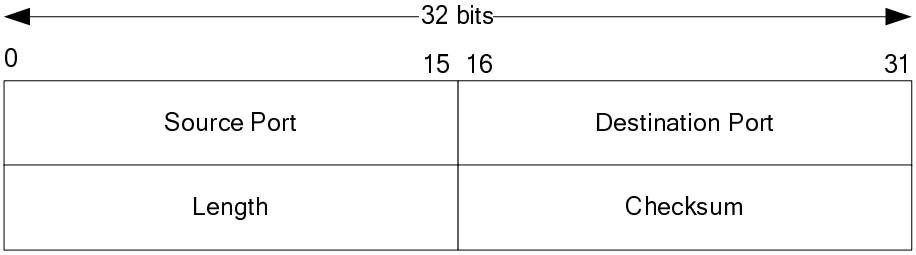

Figure 13: UDP header format

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 [11]).

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 [61] and DVB [1] tables other than those required by TS 101 154 [67] are optional.

TS - optional SI transport streams are 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.

Where transport streams with SI (TS - Full SI) are transported over IP, they shall be compliant with EN 300 468 [1] and ETR 211 [2] 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 [64]) 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 is used according to Annex E then the acceptable IP packet loss rate may be considerably 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 [55].

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.

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, service providers 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

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 [38]. IP packets passing over the IPI-1 interface shall be appropriately marked at the originating source, as described in clause 7.4.1.

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.

7.4.1 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 [37]. Networks compliant with RFC 2474 [37] 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 [37] use a 3-bit field within the ToS to determine precedence.

Within IP networks designed to carry DVB services, the markings detailed in table 17 shall be used. It is recommended that the full DSCP value be used.

Table 17: DSCP markings

Traffic Type	IP DSCP Value	Corresponding IP Precedence
Voice Bearer (see note 1)	0b110000	0b110
Video Bearer (high priority) (see note 2)	0b100010	0b100
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-IP services. NOTE 2: Normal marking for 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.		

7.4.2 Ethernet Priority

The interfaces IPI-1, IPI-2 and IPI-3 on an Ethernet MAC based HNS shall support IEEE 802.1Q [6], with defined user priority classes. The IEEE 802.1p [10] field shall be supported in an IEEE 802.1Q [6] compliant Ethernet frame. The marking shall be based on the DiffServ CodePoint (DSCP) marking method [49] as described in clause 7.4.1.

Table 18: DSCP Values and corresponding Ethernet IEEE 802.1p marking

Traffic type	IP DSCP value	Corresponding IEEE 802.1p 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-IP services.		

For a HNS based on Ethernet MAC these DSCP values are used to map a traffic type onto the corresponding IEEE 802.1p [10] priority codes. Packets shall be marked using the Layer 2 Class of Service (CoS) settings in the User Priority bits of the 802.1p [10] 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 802.1p [10] 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 [6] 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.

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 [28] and RFC 2132 [29]. The protocol consists of a number of messages that have the same fixed format as shown in figure 14.

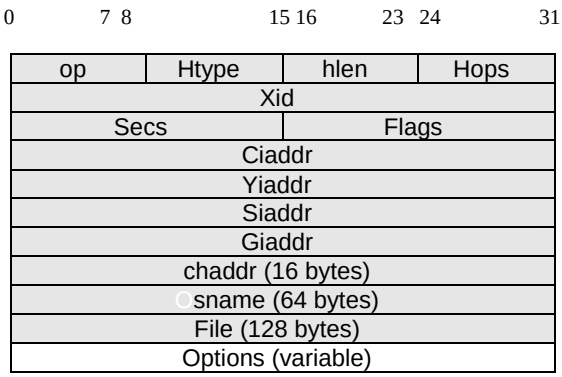

Figure 14: DHCP Format

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 [28] and RFC 2132 [29].

The modifications to allow DHCP client reconfiguration in RFC 3203 [54] (the "FORCERENEW" message) shall be implemented to allow the server to reconfigure the IP address of the HNED as part of Network Provisioning. If the IP address is changed by the server using RFC 3203 [54] (as stated in clause 2.2 of the RFC) then the HNED treats this as the same as initial booting for network provisioning.

NOTE: The HNED reconfiguration may disrupt running services to the HNED.

DHCP requires a client identifier, which is the MAC address in Ethernet or Ethernet like products (RFC 2131 [28]/RFC 2132 [29]). 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 [29] and are detailed in table 19.

Table 19: DHCP options table

Option description	Reference (RFC 2132 [29] unless otherwise stated)	Option number	Support on IPI-1
Pad Option	3.1	0	Mandatory
End Option	3.2	255	Mandatory
Subnet Mask	3.3	1	Mandatory
Time Offset	3.4	2	Not required
Router Option	3.5	3	Mandatory
Time Server Option	3.6	4	Mandatory
Name Server Option	3.7	5	Not required
Domain Name Server Option	3.8	6	Mandatory
Log Server Option	3.9	7	Not required
Cookie Server Option	3.10	8	Not required
LPR Server Option	3.11	9	Not required
Impress Server Option	3.12	10	Not required
Resource Location Server Option	3.13	11	Not required
Host Name Option	3.14	12	Not required
Boot File Size Option	3.15	13	Not required
Merit Dump File	3.16	14	Not required
Domain Name	3.17	15	Mandatory
Swap Server	3.18	16	Not required
Root Path	3.19	17	Not required
Extensions Path	3.20	18	Not required
IP Forwarding Enable/Dizble Option	4.1	19	Not required
Non-Local Source Routing Option	4.2	20	Not required
Policy Filter Option	4.3	21	Not required
Max. Datagram Reassembly Size	4.4	22	Not required
Default IP TTL	4.5	23	Not required
Path MTU Aging Timeout	4.6	24	Not required
Path MTU Plateau Option	4.7	25	Not required
Interface MTU Option	5.1	26	Not required
All Subnets are Local Option	5.2	27	Not required
Broadcast Address Option	5.3	28	Not required
Perform Mask Discovery Option	5.4	29	Not required
Mask Supplier Option	5.5	30	Not required
Perform Router Discovery Option	5.6	31	Not required
Router Solicitation Address Option	5.7	32	Not required
Static Route Option	5.8	33	Not required
Trailer Enacapsulation Option	6.1	34	Not required
ARP Cache Timeout	6.2	35	Not required
Ethernet Encapsulation Option	6.3	36	Not required
TCP Default TTL Option	7.1	37	Not required

Option description	Reference (RFC 2132 [29] unless otherwise stated)	Option number	Support on IPI-1
TCP Keepalive Interval Option	7.2	38	Not required
TCP Keepalive Garbage Option	7.3	39	Not required
Network Information Service Domain Option	8.1	40	Not required
Network Information Servers Option	8.2	41	Not required
Network Time Protocol Servers Options	8.3	42	Mandatory if NTP used
Vendor Specific Info	8.4	43	Not required
NetBIOS over TCP/IP Name Server Option.	8.5	44	Not required
NetBIOS over TCP/IP Datagram distribution server option	8.6	45	Not required
NetBIOS over TCP/IP Node Type Option	8.7	46	Not required (see clause 8.1.1.4.2)
NetBIOS over TCP/IP Scope Option	8.8	47	Not required (see clause 8.1.1.4.2)
X Window System Font Server Option	8.9	48	Not required
X Window System Display Manager Option	8.10	49	Not required
Network Information Service+ Domain Option	8.11	64	Not required
Network Information Service+ Servers Option	8.12	65	Not required
Mobile IP Home Agent Option	8.13	68	Not required
SMTP Server Option	8.14	69	Not required
POP3 Server Option	8.15	70	Not required
NNTP (News) Server Option	8.16	71	Not required
Default WWW Server Option	8.17	72	Not required
Default Finger Server Option	8.18	73	Not required
Default IRC Server Option	8.19	74	Not required
StreetTalk Server Option	8,20	75	Not required
StreetTalk Directory Assistance Server Option	8.21	76	Not required
Requested IP Address	9.1	50	Mandatory
IP Address Lease Time	9.2	51	Mandatory
Option Overload	9.3	52	Mandatory
TFTP Server Name	9.4	66	Not required
Bootfile Name	9.5	67	Not required
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 Not required
Renewal (T1) Time Value	9.11	58	Mandatory
Rebinding (T2) Time Value	9.12	59	Mandatory
Vendor class identifier	9.13	60	Not required
Client-identifier	9.14	61	Mandatory
NDS Servers	RFC 2241/2.0 [32]	85	Not required
NDS Tree Name	RFC 2241/3.0 [32]	86	Not required
User Authentication Protocol List	RFC 2485 [39]	98	Not required
Autoconfigure	RFC 2563/2.0 [42]	116	Mandatory that this option is not implemented
SLP (Service Location Protocol) Directory Agent	RFC 2610/3.0 [43]	78	Not required
SLP Service Scope Option	RFC 2610/4.0 [43]	79	Not required
Name Service Search (Search order)	RFC 2937 [48]	117	Not required
User Class	RFC 3004/4.0 [50]	77	Mandatory
Subnet Selection	RFC 3011/2.0 [51]	118	Mandatory
Relay Agent Information	RFC 3046/2.0 [53]	82	Not required

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. It is not possible for the user to change these class names, however the Network Provisioning process may change the class name. Following are the class IDs currently defined.

The class designator should be:

Table 20: 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 [53]) 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 desired is to use RFC 3927 [57].

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.

8.2 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 [27], 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 [20] 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.

9 Identification Agent for the transport of DVB Services over IP based networks

This clause covers the mandatory identity agent that allows a service provider provisioning system to recognize a Home Network End Devices (HNED).

9.1 Data sent at startup or reset

On startup of the device, the device shall check the DHCP next server "*siaddr"* field. If the "*siaddr*" field is set to 0 or is an invalid IP address then there is no provisioning server and no data shall be sent.

If there is a valid IP address then the following shall be sent according to the congestion avoidance mechanism in clause 9.2 and according to the requirements of version 1.1 of the HTTP specification [44]:

```
'GET /dvb/boot?DeviceID=' deviceId '&Version=' version
    '&RAM=' ram '&Flash=' flash ' HTTP/1.1' CRLF
'Host: ' HOST CRLF

deviceID = manufacturer "/" [model] "/" clientID
ram = 1*(DIGIT)                     ; in KBytes e.g.: 262144 (256 MBytes)
flash = 1*(DIGIT)                   ; in KBytes e.g.: 8192
```

See clause 3.3.

<**deviceId**> is used to identify which HNED is requesting this information:

- <**manufacturer**> is the unique name of the manufacturer. If the manufacturer does not want to use a name then "DVB-IPI P1 Generic" shall be used by default. If the manufacturer does use a name then it shall be the same name used across different models made by the same manufacturer.
- <**model**> is the unique model name of the particular HNED.
- <**clientid**> is the MAC address of the Ethernet interface connected to the network management system.

<**version**> is a vendor defined string which uniquely describes the software image running in the HNED.

<**ram**> is the amount of RAM memory installed in this device. If ram is zero then the HNED contains the model default amount of RAM. This shall be in kilobytes.

<**flash**> is the amount of flash or read-only memory installed in this device. If flash is zero then the HNED contains the model default amount of flash or read-only memory. This shall be in kilobytes.

<**HOST**> is the IP address of the provisioning server, obtained from the DHCP siaddr field.

The HNED should evaluate the message returned from the Event Gateway 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.

After receiving a 200 series success status, the TCP connection is closed (no data is returned). It is then up to the provisioning server and other systems to evaluate what to do next, for example, upgrade the software within the HNED to a Provisioned Profile software load.

9.2 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 Network Service Provider servers.

Each time the HNED attempts to contact the 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.

10 Network provisioning (optional)

10.1 Network management and provisioning agent

The network management and provisioning chapter documents the way the HNED network configuration shall be provisioned and the HNED shall be managed over an IP network if the option is implemented. This clause will specify the protocols and XML DTDs used rather than the rationale. The normative DTDs are supplied as attached files. The DTDs included in the present document are informative.

10.2 HTTP and HTTPS protocol

There are two options dependent on the need for an encrypted connection:

- HTTP [44] Protocol shall be used for all communication in the clear between the HNED and the remote system.
- HTTPS [46] Protocol with TLS [33] shall be used for all encrypted communication between the HNED and the remote system. This is strongly encouraged for any communications of any security related information e.g. the passwords in the GET requests.

There are only two HTTP commands used: GET and POST, which use a standard format, as defined in the next clauses.

10.2.1 Event gateway IP address and turning off network provisioning

The IP address of the event gateway of the network management system is discovered via the DHCP [28] next server "*siaddr"* field returned by the DHCP server. If the DHCP next server "*siaddr"* field is 0 then network provisioning/management events should not occur.

10.2.2 HTTP GET format

When the HNED requests that XML [65] be generated on its behalf, it shall use the format:

```
'GET /dvb/' request '?deviceID=' deviceID
    '&Password=' [password] ' HTTP/1.1' CRLF
    'Host: ' host CRLF

request = 'configure' | 'event' | 'boot'
password = 1*24(VCHAR)
```

See also clause 3.3.

<**request**> is used to identify the specific type of request. This will be:

- **configure** for a request for configuration information.
- **event** for a request for an event.
- **boot** special form of **event** used once after start-up or reboot.

<**deviceId**> is used to identify which HNED is requesting this information:

- <**manufacturer**> is the unique name of the manufacturer. If the manufacturer does not want to use a name then "DVB-IPI P1 Generic" should be used by default. If the manufacturer does use a name then it shall be the same name used across different models made by the same manufacturer.
- <**model**> is the unique model name of the particular HNED.
- <**clientid**> is the MAC address of the Ethernet interface connected to the network management system.

<**password**> is used to authenticate this HNED. If no password has been configured, it may be omitted.

NOTE: The deviceId production will contain the "/" character; the processing of these requests should take this into account.

The results will be delivered via the XML DTD appropriate to the request: A **configure** request will return a configure XML element, a **boot** and an **event** request returns an event XML element.

The GET request may contain other headers conforming to RFC 2616 [44].

EXAMPLE:

```
GET /dvb/boot?DeviceID=Cisco/IP100/010203040506&Password=SomeSecret HTTP/1.1
Host: provisioneer.sp.net
Content-type: text/xml
```

10.2.3 HTTP POST format

Whenever the HNED generates XML without being requested to do so, it will use this format:

```
'POST /dvb/' report '?deviceID=' deviceID
    '&Password=' [password] ' HTTP/1.1' CRLF
    'Host: ' host CRLF
    'Content-type: text/xml; charset=ISO-8859-1' CRLF

report = 'configure' | 'inventory' | 'status' | 'event'
password = 1*24(VCHAR)
```

See clause 3.3.

<**report**> is used to identify a change. This will either be:

- **configure** for a configuration change;
- **inventory** for an inventory change;
- **status** for a status change;
- **event** for an event change.

<**deviceId**> is used to identify which HNED is requesting this information:

- <**manufacturer**> is the unique name of the manufacturer;
- <**model**> is the unique model name of the particular HNED;
- <**clientid**> is the MAC address of the Ethernet interface connected to the network management system.

<**password**> is used to authenticate this HNED. If no password has been configured, it may be omitted.

NOTE: The deviceId production will contain the "/" character; the processing of these requests should take this into account. The data will be delivered via the XML DTD appropriate to the request, for example a **configure** request will return a configure XML element.

The POST Request may contain other headers conforming to RFC 2616 [44].

EXAMPLE:

```
POST /dvb/configure?DeviceID=Cisco/IP100/010203040506&Password=Secret HTTP/1.1
Host: provisioneer.sp.net
Content-type: text/xml
<?xml version="1.0" encoding="UTF-8"?>
<configure action="apply">
    <identifier>030500-1330</identifier>
    <timestamp>2002-09-26 18:35:39 UTC</timestamp>
    <configuration>
        <password>NewSecret</password>
    </configuration>
</configure>
```

10.2.4 Event polling

After boot up, the HNED will issue one special boot form of the HTTP GET event, in order to allow the management system to adjust the default start-up behaviour of the HNED. This shall follow a congestion avoidance mechanism so that a power cut or other failure does not cause a large number of HNEDs to send data at boot up so overloading the Network Service Provider servers.

Each time the HNED attempts to send the special boot form of HTTP GET to the 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.

If the response action is "none" then the HNED will HTTP GET a configuration before it begins the regular event polling, otherwise the HNED follows the actions until the response action is "none" and regular event polling begins.

The HNED shall send an HTTP GET event on a regular basis, with a time as set in configuration XML DTD. The events returned will be processed and another HTTP GET will be issued until the HNED receives an action = "none". The polling interval shall commence from the reception of the HTTP GET with the event action = "none" and after that interval has passed, this process will repeat.

If the network management system does not receive an event in 3 polling intervals, then the network management system shall consider the HNED to be "missing" and react appropriately.

10.2.5 Event XML DTD

When the HNED GETs an event, it shall receive it in the XML [65] format shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- This is the embedded version of the DTD -->

<!DOCTYPE event [
<!ELEMENT identifier (#PCDATA)>
<!ELEMENT timestamp (#PCDATA)>
<!ELEMENT deviceId (#PCDATA)>
<!ELEMENT event (identifier, timestamp?, deviceId, configuration?)>
<!ATTLIST event action (none | configure | inventory | status | update | boot) #REQUIRED>
]>

<!--
The event action attribute is used to indicate the nature of this event.
 A value of "none" indicates that there are no events for this HNED.
 A value of "configure" is used to indicate that the HNED should report its current configuration.
 A value of "inventory" indicates that the HNED should generate an inventory report.
 A value of "status" indicates that the HNED should generate a status report.
 A value of "update" tells the HNED to update its configuration, either using the configuration
element enclosed or, if none was included, to request a configuration normally.
 A value of "boot" indicates that the HNED should initiate a reboot.
```

```
The identifier element is copied to any XML required to further process this event.

An optional ISO 8601 format timestamp [60] may be used to indicate when this XML was generated.

The deviceId element is used to further identify this device.
-->

<!-- This is an example of using the DTD above -->
<event action="none">
    <identifier>030500-1330</identifier>
    <timestamp>2002-09-26 18:35:40 UTC</timestamp>
    <deviceId>Cisco/IP100/010203040506</deviceId>
</event>
```

10.2.6 Configuration XML DTD

When configuration data is sent to the HNED for processing, it shall use this format. The HNED shall also use this format to report its currently running configuration, either in response to a "configure" event or to report that the configuration has been changed by some other means. When the HNED reports configuration using this format, the action attribute shall not be used.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- This is the embedded version of the DTD -->

<!DOCTYPE configure [
<!ELEMENT identifier (#PCDATA)>
<!ELEMENT timestamp (#PCDATA)>
<!ELEMENT errorMessage (#PCDATA)>
<!ELEMENT deviceId (#PCDATA)>
<!-- Allows the SP to change the automatically generated deviceId -->
<!ELEMENT timezone (#PCDATA)>
<!-- ISO 8601 format timezone (e.g. +0100 in France) -->
<!ELEMENT country (#PCDATA)>
<!-- E.164 country code (e.g. 44 in England) -->
<!ELEMENT hostname (#PCDATA)>
<!ELEMENT interval (#PCDATA)>
<!-- Period between event polls (see clause 10.2.4 Event Polling) -->
<!ELEMENT sdEntry (#PCDATA)>
<!-- Service Discovery entry points -->

<!-- For each non-wireless interface, the following parameters can be configured -->
<!ELEMENT dhcpClientId (#PCDATA)>
<!-- The client-id of the DHCP client -->
<!ELEMENT dhcpHwAddress (#PCDATA)>
<!-- The hardware address of the DHCP client -->
<!ELEMENT dhcpUserClass (#PCDATA)>
<!-- The user-class used by the DHCP client -->
<!ELEMENT interface (dhcpClientId?, dhcpHwAddress?, dhcpUserClass?)>
<!ATTLIST interface name CDATA #REQUIRED>

<!-- In addition, the following parameters can be configured on wireless interfaces -->
<!ELEMENT dhcpClientName (#PCDATA)>
<!-- Wireless client name -->
<!ELEMENT ssid (#PCDATA)>
<!ATTLIST ssid index CDATA #REQUIRED>
<!-- index range is 1-3 (there can be up to 3 of these) -->
<!ELEMENT mode (#PCDATA)>
<!-- AdHoc or Infrastructure -->
<!ELEMENT channel (#PCDATA)>
<!-- Channel number 1-11 (limited by national regulations) -->
<!ELEMENT power (#PCDATA)>
<!-- Transmit power, 0-100mW (limited by national regulations) -->
<!ELEMENT headers (#PCDATA)>
<!-- Radio Headers, short or long -->
<!ELEMENT encrypt (#PCDATA)>
<!-- WEP or none -->
<!ELEMENT wepKey (#PCDATA)>
<!-- Hex string, index range is 1-4, length is 40 or 128 (bits) -->
<!ATTLIST wepKey index CDATA #REQUIRED length CDATA #REQUIRED >
<!ELEMENT wpaKey (#PCDATA)>
<!-- Hex string, length is 40 or 128 bits -->
<!ATTLIST wpaKey length CDATA #REQUIRED >
<!ELEMENT mixedCells (#PCDATA)>
<!-- Access point authentication -->
<!ELEMENT authType (#PCDATA)>
```

```
<!-- None, WEP-open, WEP-shared, LEAP, EAP-SIM, EAP-TLS -->
<!ELEMENT username (#PCDATA)>
<!ELEMENT password (#PCDATA)>
<!ELEMENT domain (#PCDATA)>
<!ELEMENT certificate (#PCDATA)>
<!-- The TLS certificate to use for authentication -->
<!ELEMENT apAuth (authType?, username?, password?, domain?, certificate?)>
<!ELEMENT wireless (dhcpClientId?, dhcpHwAddress?, dhcpUserClass?, dhcpClientName?, ssid*, mode?,
channel?, power?, headers?, encrypt?, wepKey*, wpaKey?, mixedCells, apAuth?)>
<!ATTLIST wireless name CDATA #REQUIRED>

<!-- Access List Configuration
The order in which an accessEntry is added to an accessList is important. When an accessList is
evaluated, each accessEntry within that list is evaluated in order, from the first to the last,
until a match is found. Once that match is found, each remaining accessEntry in the accessList will
be ignored. -->
<!ELEMENT action (#PCDATA)>
<!-- permit, drop or deny -->
<!ELEMENT protocol (#PCDATA)>
<!-- 0-255 or one of the following keywords:
ip, gre, icmp, igmp, ip, ipinip, nos, pim, tcp, or udp -->
<!ELEMENT srcAddress (#PCDATA)>
<!-- source IP address (e.g. 32.1.0.0) -->
<!ELEMENT srcWild (#PCDATA)>
<!-- source wildcard mask (e.g. 0.0.255.255) -->
<!ELEMENT srcPort (#PCDATA)>
<!-- source port range (e.g. 0-65535 for any port)
either a single port number or a range of two -->
<!ELEMENT dstAddress (#PCDATA)>
<!-- destination IP address (e.g. 32.1.0.0) -->
<!ELEMENT dstWild (#PCDATA)>
<!-- destination wildcard mask (e.g. 0.0.255.255) -->
<!ELEMENT dstPort (#PCDATA)>
<!-- destination port range (e.g. 0-65535 for any port)
either a single port number or a range of two -->
<!ELEMENT accessEntry (action, protocol, srcAddress, srcWild, srcPort?, dstAddress?, dstWild? ,
dstPort?)>
<!ELEMENT accessList (accessEntry*)>
<!ATTLIST accessList name CDATA #REQUIRED >

<!-- VPN Tunnel Configuration -->
<!ELEMENT peer (#PCDATA)>
<!-- IP address or DNS name and port number of the IPsec peer -->
<!ELEMENT natIp (#PCDATA)>
<!-- IP address or DNS name and port number if using NAT -->
<!ELEMENT voice (natIp?, port*)>

<!ELEMENT accessListName (#PCDATA)>
<!-- Name of the list to use to control which packets are to be encrypted into the tunnel -->
<!ELEMENT authentication (#PCDATA)>
<!-- esp-sha, esp-md5-hmac, ... -->
<!ELEMENT keyExchange (#PCDATA)>
<!-- izkmp, ... -->
<!ELEMENT sharedKey (#PCDATA)>
<!-- key to use for authentication, if required -->
<!ELEMENT encryption (#PCDATA)>
<!-- esp-des, esp-3des, ... -->
<!ELEMENT tunnel (peer, accessListName, authentication, keyExchange?, certificate?, sharedKey?,
encryption)>
<!ATTLIST tunnel name CDATA #REQUIRED>

<!-- Remote Control Configuration -->
<!ELEMENT rcType (#PCDATA)>
<!-- Type of remote control: (RC5, RC6, NRC17, ...) -->
<!ELEMENT button (#PCDATA)>
<!-- Button number on the remote control -->
<!ELEMENT code (#PCDATA)>
<!-- Program code to be used by the button -->
<!ELEMENT speed (#PCDATA)>
<!-- Programming speed: slow, medium or fast -->
<!ELEMENT remoteControl (rcType, button, code, speed)>
<!ATTLIST remoteControl name CDATA #REQUIRED>

<!-- Simple voice messaging over IP configuration using SIP. -->
<!-- For each individual voice device -->
<!ELEMENT uid (#PCDATA)>
<!-- The identifier used to register this device with the SIP registrar -->
<!ELEMENT registrar (#PCDATA)>
```

```
<!-- The IP address or DNS name of the SIP registrar -->
<!ELEMENT ttl (#PCDATA)>
<!-- The TTL to use for all SIP and RTP packets -->
<!ELEMENT renew (#PCDATA)>
<!-- How often in seconds SIP registration is renewed -->
<!ELEMENT proxy (#PCDATA)>
<!-- The IP address or DNS name of the SIP proxy server -->
<!ELEMENT rtpPort (#PCDATA)>
<!-- Port the HNED uses to receive RTP packets if NAT is used -->
<!ELEMENT tos (#PCDATA)>
<!-- The TOS bits to use on all voice packets (e.g. 5) range 0-5 -->
<!ELEMENT rxcodec (#PCDATA)>
<!-- A list of CODECs to use to encode received analog data. CODECs include G.711, G.729, G.723 and
GSM. Each CODEC is followed by a to indicate a-law or u for u-law (e.g. G711a, G729a) -->
<!ELEMENT txcodec (#PCDATA)>
<!-- A list of CODECs to use to decode received Voice data. CODECs include G.711, G.729, G.723 and
GSM. Each CODEC is followed by a to indicate a-law or u for u-law (e.g. G711u, G729u, G723u) -->
<!ELEMENT port (uid?, username?, password?, registrar?, ttl?, renew?, proxy?, rtpPort?, tos?,
rxcodec?, txcodec?)>
<!ATTLIST port name CDATA #REQUIRED>
<!-- The name of this voice device (Voice 0) -->

<!ELEMENT config (#PCDATA)>
<!ELEMENT otherEntity (config*)>
<!ATTLIST otherEntity name CDATA #REQUIRED>
<!-- A name that uniquely identifies this entity (i.e. "Gizmo") -->

<!ELEMENT configuration (deviceId?, timezone?, country?, hostname?, interval?, sdEntry?, interface*,
wireless*, accessList*, tunnel*, remoteControl?, voice*, otherEntity*)>

<!ELEMENT configure (identifier, timestamp?, errorMessage?, configuration?)>
<!ATTLIST configure action (wait | apply | save) #IMPLIED>
]>

<!-- This is an example of using the DTD above -->
<configure action="apply">
 <identifier>030500-1330</identifier>
 <timestamp>2002-09-26 18:35:39 UTC</timestamp>
 <configuration>
  <hostname>ThisIsMySTB</hostname>
  <interval>00:05:00</interval>
  <otherEntity name="Gizmo1">
   <config>$MUMBLE=Fritz,$SOME=thing,$RETRY=3</config>
  </otherEntity>
  <otherEntity name="Gizmo2">
   <config>TFTP://32.1.2.3/full/path/filename.ext</config>
  </otherEntity>
 </configuration>
</configure>
```

10.2.7 Failure XML DTD

When the HNED needs to report an error that occurred during the processing of XML it received, it shall use this format.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- This is the embedded version of the DTD -->

<!DOCTYPE failure [
<!ELEMENT identifier (#PCDATA)>
<!ELEMENT timestamp (#PCDATA)>
<!ELEMENT deviceId (#PCDATA)>
<!ELEMENT errorMessage (#PCDATA)>
<!--
errorMessage = errorCode "-" [errorItem] "-" errorText
errorCode = [ "+" | "-" ] DIGIT DIGIT DIGIT
errorItem = TAG
errorText = *255(ALPHA|DIGIT)

Example:    123-timestamp-Value is not valid (0000-13-32 25:61:61 XXX)
            (Note: each sub-field within this timestamp is illegal)
-->
<!ELEMENT failure (identifier, timestamp?, deviceId, errorMessage)>
]>
```

```
<!--
The identifier element is copied from the XML that was being processed.

An optional ISO 8601 format timestamp may be used to indicate when this XML was generated.

The deviceId element is used to further identify this device.

The errorMessage element will contain a text string describing the problem.
-->

<!-- This is an example of using the DTD above -->
<failure>
    <identifier>030500-1330</identifier>
    <timestamp>2002-09-26 18:35:40 UTC</timestamp>
    <deviceId>Cisco/IP100/010203040506</deviceId>
    <errorMessage>123--Incomplete command</errorMessage>
</failure>
```

10.2.8 Success XML DTD

When the HNED processes XML it received correctly, it shall use this format to report success.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- This is the embedded version of the DTD -->

<!DOCTYPE success [
<!ELEMENT identifier (#PCDATA)>
<!ELEMENT timestamp (#PCDATA)>
<!ELEMENT deviceId (#PCDATA)>
<!ELEMENT success (identifier, timestamp?, deviceId)>
]>

<!--
The identifier element is copied from the XML that was being processed.

An optional ISO 8601 format timestamp may be used to indicate when this XML was generated.

The deviceId element is used to further identify this device.
-->

<!-- This is an example of using the DTD above -->
<success>
    <identifier>030500-1330</identifier>
    <timestamp>2002-09-26 18:35:40 UTC</timestamp>
    <deviceId>Cisco/IP100/010203040506</deviceId>
</success>
```

10.2.9 Inventory XML DTD

Inventory information shall be delivered using this format, either in response to an "inventory" event, to report an inventory change (SIM card inserted) or upon boot up ensure that any hardware changes which occurred during the power down are reported.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- This is the embedded version of the DTD -->

<!DOCTYPE inventory [
<!ELEMENT identifier (#PCDATA)>
<!ELEMENT timestamp (#PCDATA)>
<!ELEMENT deviceId (#PCDATA)>
<!ELEMENT vendor (#PCDATA)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT serial (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT speed (#PCDATA)>
<!ELEMENT size (#PCDATA)>
<!ELEMENT version (#PCDATA)>
<!ELEMENT checksum (#PCDATA)>
<!ELEMENT hardware (vendor, model, serial?, name?, type, speed?, size?)>
<!ELEMENT software (name, version, size, checksum)>
<!ELEMENT inventory (identifier, timestamp?, deviceId, hardware*, software*)>
```

```
<!ATTLIST inventory report (full | add | change | remove) #REQUIRED>
]>

<!--
The inventory element shall be used to report on the current contents of the HNED.

The report attribute is used to designate if this XML represents the
 A value of "full" represents all components of this box
 A value of "add" represents only the component(s) added
 A value of "remove" represents only the component(s) removed
 A value of "change" represents only the component(s) that changed

The identifier element is copied from the XML that was being processed.

An optional ISO 8601 format timestamp may be used to indicate when this XML was generated.

The deviceId element is used to further identify this device.

The hardware element is used to describe the attributes of the hardware components of this HNED.

The software element is used to describe the attributes of the software components of this HNED.

Within the hardware element, an element for each interface will be included. This will include the
name of the device (which coincides with the configuration) and the type of the device. This will
optionally (as applicable) indicate the speed in bps of that interface (e.g. 100000000 for 100 Mbps
ethernet) and/or the size in bytes (e.g. 90000000000 for a 90GB disk drive).

The software element is used to report on the name, version, size and checksum of each software
image currently running on the HNED.
-->

<!-- This is an example of using the DTD above -->
<inventory report="full">
    <identifier>030500-1330</identifier>
    <timestamp>2002-09-26 18:35:39 UTC</timestamp>
    <deviceId>Cisco/IP100/010203040506</deviceId>
    <hardware>
        <vendor>stbRus</vendor>
        <model>XYZ-2000</model>
        <serial>123456789</serial>
        <type>chassis</type>
    </hardware>
    <hardware>
        <vendor>ramCo</vendor>
        <model>NV-100M</model>
        <type>nvram</type>
        <size>65536</size>
    </hardware>
    <hardware>
        <vendor>ethCo</vendor>
        <model>USB-100</model>
        <serial>123456789</serial>
        <name>eth0</name>
        <type>ethernet</type>
        <speed>100000000</speed>
    </hardware>
    <software>
        <name>stbOS</name>
        <version>1.0.0</version>
        <size>32768</size>
        <checksum>12345</checksum>
    </software>
</inventory>
```

10.2.10 Status XML DTD

Status information shall be delivered using this format, either in response to a "status" event or to report a status change (such as a new IP address delivered via DHCP).

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- This is the embedded version of the DTD -->

<!DOCTYPE status [
<!ELEMENT identifier (#PCDATA)>
<!ELEMENT timestamp (#PCDATA)>
```

```
<!ELEMENT deviceId (#PCDATA)>
<!ELEMENT clock EMPTY>
<!ATTLIST clock sync (lost | ntp | sntp) #REQUIRED>

<!ELEMENT ipAddress (#PCDATA)>
<!ELEMENT ipMask (#PCDATA)>
<!ELEMENT ipGateway (#PCDATA)>
<!ELEMENT timeServer (#PCDATA)>
<!--    The IP address or DNS name of time server(s)
    Up to three of these may be reported-->
<!ELEMENT ntpServer (#PCDATA)>
<!--    The IP address or DNS name of NTP server(s)
    Up to three of these may be reported-->
<!ELEMENT domain (#PCDATA)>
<!ELEMENT dnsServer (#PCDATA)>
<!--    The IP address of the DNS server(s)
    Up to three of these may be reported-->
<!ELEMENT dhcpServer (#PCDATA)>
<!ELEMENT leaseRenew (#PCDATA)>

<!--Selected item from RFC 1213 (mib-2) [19]-->
<!ELEMENT sysUpTime (#PCDATA)>

<!--Selected items from RFC 2863 (interfaces) [47]-->
<!ELEMENT ifDescr (#PCDATA)>
<!ELEMENT ifType (#PCDATA)>
<!ELEMENT ifMtu (#PCDATA)>
<!ELEMENT ifSpeed (#PCDATA)>
<!ELEMENT ifPhysAddress (#PCDATA)>
<!ELEMENT ifAdminStatus (#PCDATA)>
<!ELEMENT ifOperStatus (#PCDATA)>
<!ELEMENT ifLastChange (#PCDATA)>
<!ELEMENT ifInOctets (#PCDATA)>
<!ELEMENT ifInUcastPkts (#PCDATA)>
<!ELEMENT ifInDiscards (#PCDATA)>
<!ELEMENT ifInErrors (#PCDATA)>
<!ELEMENT ifInUnknownProtos (#PCDATA)>
<!ELEMENT ifOutOctets (#PCDATA)>
<!ELEMENT ifOutUcastPkts (#PCDATA)>
<!ELEMENT ifOutDiscards (#PCDATA)>
<!ELEMENT ifOutErrors (#PCDATA)>

<!--Selected items from RFC 2011 (ip) [25]-->
<!ELEMENT ipForwarding (#PCDATA)>
<!ELEMENT ipDefaultTTL (#PCDATA)>
<!ELEMENT ipInReceives (#PCDATA)>
<!ELEMENT ipInHdrErrors (#PCDATA)>
<!ELEMENT ipInAddrErrors (#PCDATA)>
<!ELEMENT ipForwDatagrams (#PCDATA)>
<!ELEMENT ipInUnknownProtos (#PCDATA)>
<!ELEMENT ipInDiscards (#PCDATA)>
<!ELEMENT ipInDelivers (#PCDATA)>
<!ELEMENT ipOutRequests (#PCDATA)>
<!ELEMENT ipOutDiscards (#PCDATA)>
<!ELEMENT ipOutNoRoutes (#PCDATA)>
<!ELEMENT ipReasmTimeout (#PCDATA)>
<!ELEMENT ipReasmReqds (#PCDATA)>
<!ELEMENT ipReasmOKs (#PCDATA)>
<!ELEMENT ipReasmFails (#PCDATA)>
<!ELEMENT ipFragOKs (#PCDATA)>
<!ELEMENT ipFragFails (#PCDATA)>
<!ELEMENT ipFragCreates (#PCDATA)>
<!ELEMENT ipNetToMediaPhyAddress (#PCDATA)>
<!ELEMENT ipNetToMediaNetAddress (#PCDATA)>
<!ELEMENT ipNetToMediaType (#PCDATA)>

<!--Selected items from RFC 2011 (icmp) [25]-->
<!ELEMENT icmpInMsgs (#PCDATA)>
<!ELEMENT icmpInErrors (#PCDATA)>
<!ELEMENT icmpInDestUnreachs (#PCDATA)>
<!ELEMENT icmpInTimeExcds (#PCDATA)>
<!ELEMENT icmpInParmProbs (#PCDATA)>
<!ELEMENT icmpInSrcQuenchs (#PCDATA)>
<!ELEMENT icmpInRedirects (#PCDATA)>
<!ELEMENT icmpInEchos (#PCDATA)>
<!ELEMENT icmpInEchosReps (#PCDATA)>
<!ELEMENT icmpInTimestamps (#PCDATA)>
<!ELEMENT icmpInTimestampsReps (#PCDATA)>
```

```
<!ELEMENT icmpInAddrMasks (#PCDATA)>
<!ELEMENT icmpInAddrMaskReps (#PCDATA)>
<!ELEMENT icmpOutMsgs (#PCDATA)>
<!ELEMENT icmpOutErrors (#PCDATA)>
<!ELEMENT icmpOutDestUnreachs (#PCDATA)>
<!ELEMENT icmpOutTimeExcds (#PCDATA)>
<!ELEMENT icmpOutParmProbs (#PCDATA)>
<!ELEMENT icmpOutSrcQuenchs (#PCDATA)>
<!ELEMENT icmpOutRedirects (#PCDATA)>
<!ELEMENT icmpOutEchos (#PCDATA)>
<!ELEMENT icmpOutEchosReps (#PCDATA)>
<!ELEMENT icmpOutTimestamps (#PCDATA)>
<!ELEMENT icmpOutTimestampReps (#PCDATA)>
<!ELEMENT icmpOutAddrMasks (#PCDATA)>
<!ELEMENT icmpOutAddrMaskReps (#PCDATA)>

<!--Selected items from RFC 2013 (udp)  [26]-->
<!ELEMENT udpInDatagrams (#PCDATA)>
<!ELEMENT udpNoPorts (#PCDATA)>
<!ELEMENT udpInErrors (#PCDATA)>
<!ELEMENT udpOutDatagrams (#PCDATA)>
<!ELEMENT udpLocalAddress (#PCDATA)>
<!ELEMENT udpLocalPort (#PCDATA)>
<!ELEMENT udpEntry (udpLocalAddress, udpLocalPort)>
<!ATTLIST udpEntry index CDATA #REQUIRED>
<!ELEMENT udpTable (udpEntry*)>

<!--Selected items from RFC 2863 (ifMIB)[47]-->
<!ELEMENT ifInMulticastPkts (#PCDATA)>
<!ELEMENT ifInBroadcastPkts (#PCDATA)>
<!ELEMENT ifOutMulticastPkts (#PCDATA)>
<!ELEMENT ifOutBroadcastPkts (#PCDATA)>
<!ELEMENT ifLinkUpDownTrapEnable (#PCDATA)>
<!ELEMENT ifHighSpeed (#PCDATA)>
<!ELEMENT ifPromiscuousMode (#PCDATA)>
<!ELEMENT ifConnectorPresent (#PCDATA)>
<!ELEMENT ifAlias (#PCDATA)>
<!ELEMENT ifCounterDiscontinuityTime (#PCDATA)>

<!ELEMENT freeSpace (#PCDATA)>
<!--    The current amount of free space, in bytes, on the HNED non-volatile store -->
<!ELEMENT display (#PCDATA)>
<!--    The current contents of the LCD display on the HNED (if any) -->
<!ELEMENT temperature (#PCDATA)>
<!--    The current temperature of the HNED (if instrumented) -->

<!ELEMENT snmp (sysUpTime?, ifDescr?, ifType?, ifMtu?, ifSpeed?, ifPhysAddress?, ifAdminStatus?,
ifOperStatus?, ifLastChange?, ifInOctets?, ifInUcastPkts?, ifInDiscards?, ifInErrors?,
ifInUnknownProtos?, ifOutOctets?, ifOutUcastPkts?, ifOutDiscards?, ifOutErrors?, ipForwarding?,
ipDefaultTTL?, ipInReceives?, ipInHdrErrors?, ipInAddrErrors?, ipForwDatagrams?,
ipInUnknownProtos?, ipInDiscards?, ipInDelivers?, ipOutRequests?, ipOutDiscards?,
ipOutNoRoutes?, ipReasmTimeout?, ipReasmReqds?, ipReasmOKs?, ipReasmFails?, ipFragOKs?,
ipFragFails?, ipFragCreates?, ipNetToMediaPhyAddress?, ipNetToMediaNetAddress?,
ipNetToMediaType?, icmpInMsgs?, icmpInErrors?, icmpInDestUnreachs?, icmpInTimeExcds?,
icmpInParmProbs?, icmpInSrcQuenchs?, icmpInRedirects?, icmpInEchos?, icmpInEchosReps?,
icmpInTimestamps?, icmpInTimestampsReps?, icmpInAddrMasks?, icmpInAddrMaskReps?, icmpOutMsgs?,
icmpOutErrors?, icmpOutDestUnreachs?, icmpOutTimeExcds?, icmpOutParmProbs?,
icmpOutSrcQuenchs?, icmpOutRedirects?, icmpOutEchos?, icmpOutEchosReps?, icmpOutTimestamps?,
icmpOutTimestampReps?, icmpOutAddrMasks?, icmpOutAddrMaskReps?, udpInDatagrams?, udpNoPorts?,
udpInErrors?, udpOutDatagrams?, udpTable?, udpLocalAddress?, udpLocalPort?, ifInMulticastPkts?,
ifInBroadcastPkts?, ifOutMulticastPkts?, ifOutBroadcastPkts?, ifLinkUpDownTrapEnable?,
ifHighSpeed?, ifPromiscuousMode?, ifConnectorPresent?, ifAlias?, ifCounterDiscontinuityTime?)>

<!ELEMENT dhcp (ipAddress?, ipMask?, ipGateway?, timeServer*, ntpServer*, domain?, dnsServer*,
dhcpServer?, leaseRenew?)>
<!ELEMENT interface (dhcp, snmp?)>
<!ATTLIST interface name CDATA #REQUIRED>
<!ELEMENT status (identifier, timestamp?, deviceId, clock?, interface*, freeSpace?, display?,
temperature?)>
]>

<!--
The status element is used to report on the current status of the HNED.

The identifier element is copied from the XML that was being processed.

An optional ISO 8601 format timestamp may be used to indicate when this XML was generated.
```

```
The deviceId element is used to further identify this device.

The clock element is used to describe the synchronization status of the clock of this HNED.

The dhcp element is used to describe the status of the dhcp client(s).
Within the dhcp element, an element for each interface may be included.
This will include the name of the device (which coincides with the configuration) and the parameters
most recently received from the DHCP server for that interface.
-->

<!-- This is an example of using the DTD above -->
<status>
    <identifier>030500-1330</identifier>
    <timestamp>2002-09-26 18:35:41 UTC</timestamp>
    <deviceId>Cisco/IP100/010203040506</deviceId>
    <clock sync="sntp"/>
    <interface name="eth0">
        <dhcp>
            <ipAddress>1.2.3.4</ipAddress>
            <ipMask>255.255.255.0</ipMask>
        </dhcp>
    </interface>
</status>
```

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 [34] 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 [61]. 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:

$$R(i) = \frac{(i' - i'') \times 27\ MHz}{PCR(k) - PCR(k-1)} \qquad \text{(A.2)}$$

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 [61]).

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 [64]. 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 [34], 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.

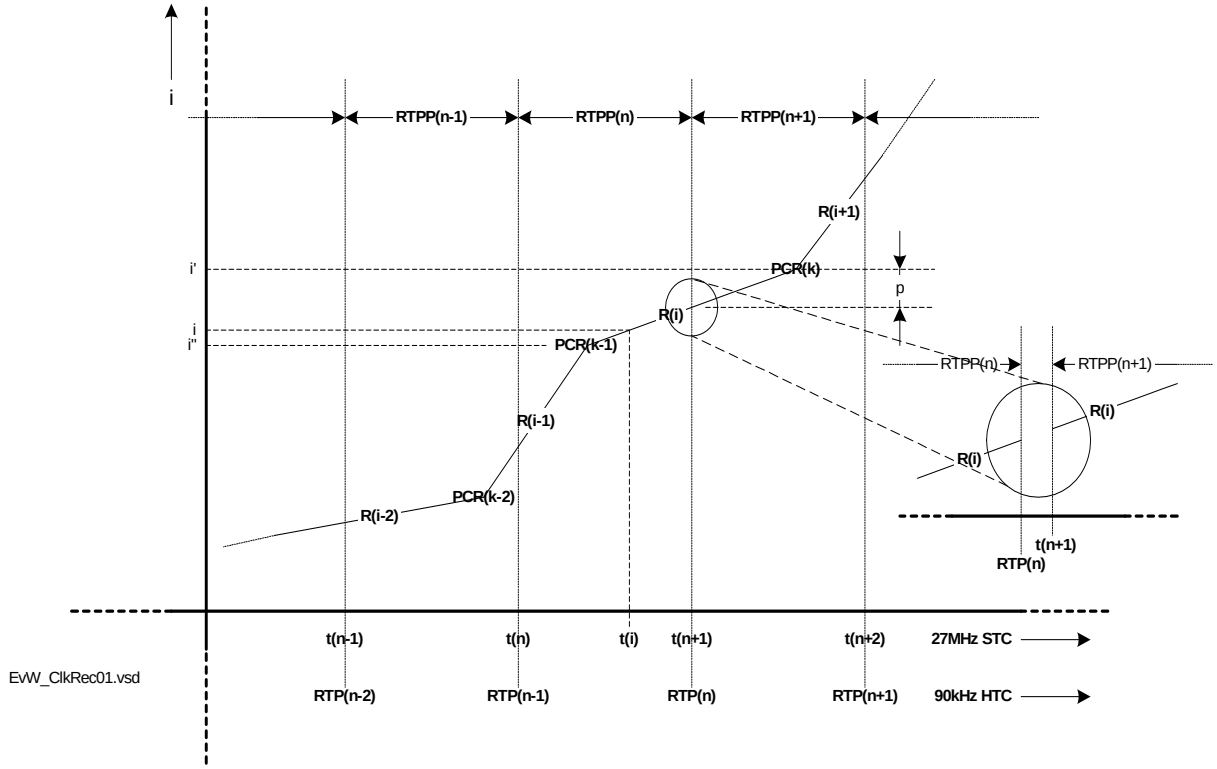

Figure A.1: Timing

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:

$$t(n+1) = \frac{PCR(k)}{27\ MHz} - \frac{p}{R(i)} \qquad (A.3)$$

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 [64].
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-IP service discovery model.

The boxes in bold are the components required to establish the list of DVB-IP services available from different Service Providers.

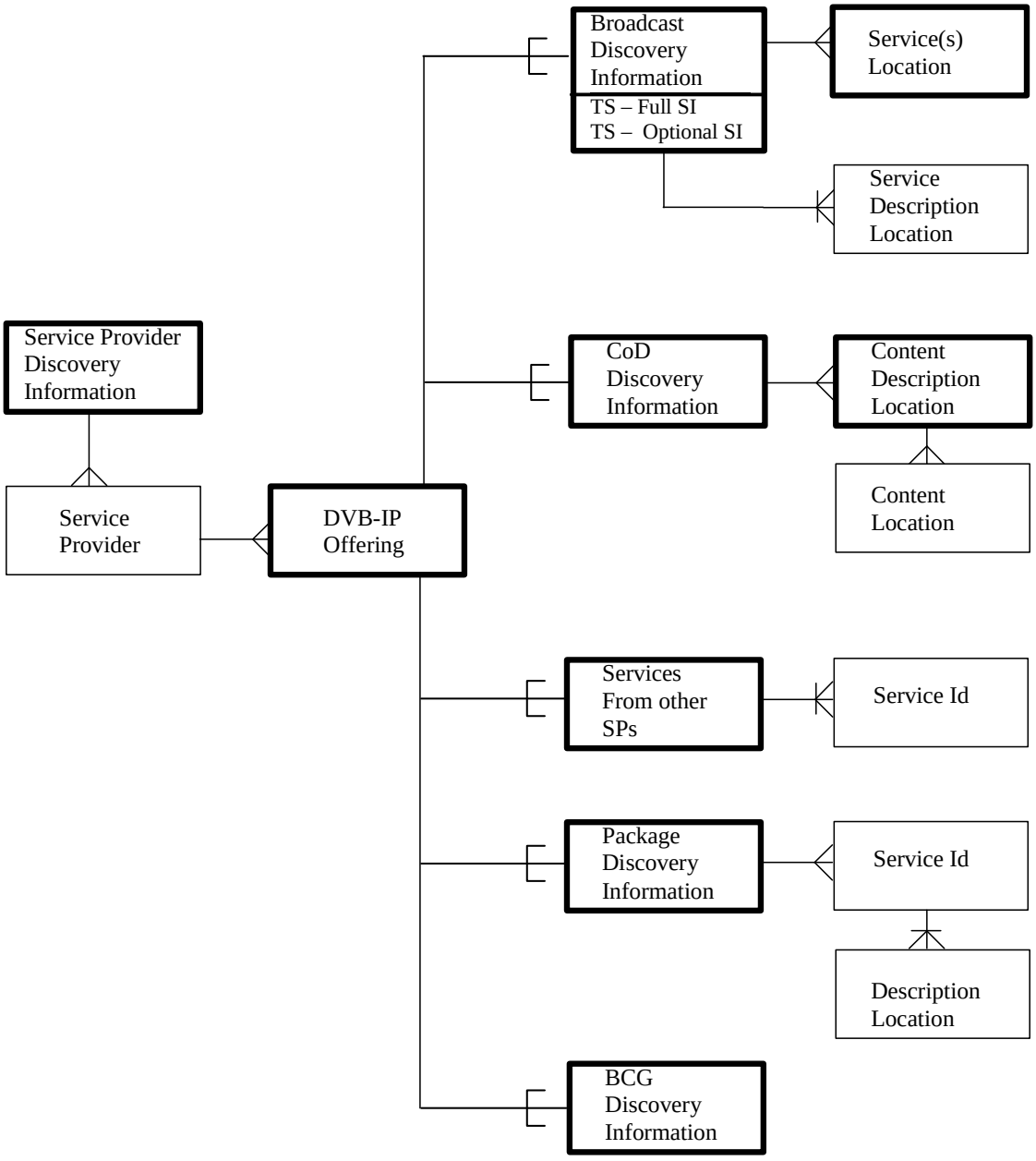

Figure B.1: Proposed data model for DVB-IP service discovery information

The Service Provider Discovery Information enables the discovery of Service Providers offering DVB-IP services. Service Providers publish their offer via the service discovery information. A Service Provider offer is made of services of type broadcast or content on demand.

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 Service Providers that would like to describe their Content on Demand 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 Service Providers 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-IP Service Providers operating on the network, then in a second stage the list of DVB-IP 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 XML schemas

The following clauses work through the various types and elements that are used in the XML schema which represents the service discovery information. The full normative XML schema is available as the file sdns3r7.xsd in archive ts_102034v010301p0.zip) which accompanies the present document.

C.1.1 Namespace

The namespace the service discovery schema is urn:dvb:ipi:sdns:2006.

C.2 Simple types

C.2.1 DescriptionLocation

```
<xsd:simpleType name="DescriptionLocation">
    <xsd:restriction base="xsd:anyURI"/>
</xsd:simpleType>
```

A URI that specifies the location of further information.

C.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 [14].

C.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.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.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.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.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.2.8 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.2.9 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.2.10 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.2.11 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 [58].

C.2.12 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 [59].

C.2.13 OrigNetId

```
<xsd:simpleType name="OrigNetId">
    <xsd:restriction base="xsd:unsignedShort"/>
</xsd:simpleType>
```

The original_network_id, as defined in ETR 162 [3], which also specifies the management of this number space. This value shall be in decimal.

C.2.14 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.2.15 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.2.16 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.2.17 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 [4], clause 14.9. It is recommended that this follows the rules for an internet DNS name as specified in RFC 1035 [15] and subsequent updates.

C.2.18 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.2.19 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 72.

C.2.20 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.2.21 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.2.22 Version

```
<xsd:simpleType name="Version">
    <xsd:restriction base="xsd:integer">
        <xsd:minInclusive value="0"/>
        <xsd:maxInclusive value="255"/>
    </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 decimal.

C.3 Complex types and attribute groups

C.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>
```

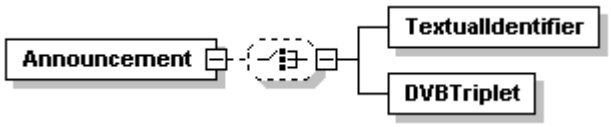

Figure C.1: AnnouncementSupport

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.3.2 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.3.3 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.3.4 DVBSTPTransportModeType

```
<xsd:complexType name="DVBSTPTransportModeType">
    <xsd:complexContent>
        <xsd:extension base="dvb:PayloadList">
            <xsd:attributeGroup ref="dvb:MulticastAddressAttributes"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

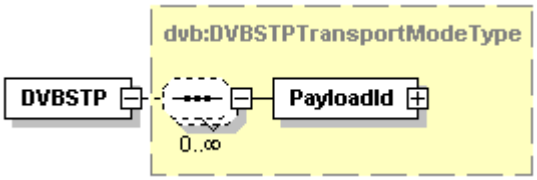

Figure C.2: DVBSTPTransportModeType

C.3.5 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.3.6 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="required"/>
</xsd:complexType>
```

C.3.7 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" default="false"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

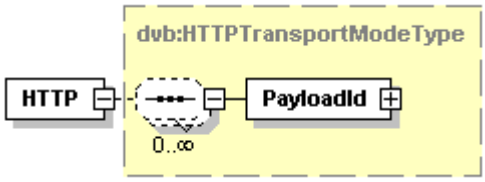

Figure C.3: HTTPTransportModeType

C.3.8 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:ServiceAvailability" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
```

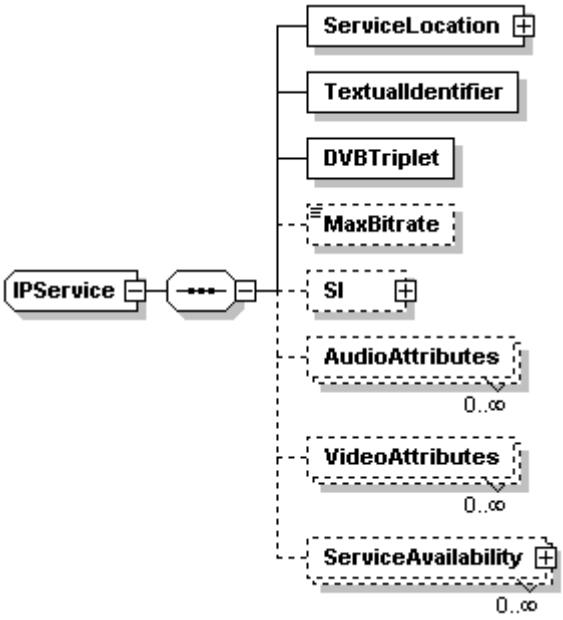

Figure C.4: IPService

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 [67]. The format of this type is defined in clause 6.3.5 of TS 102 822-3-1 [69]. 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 [68].
`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 [67]. The format of this type is defined in clause 6.3.5 of TS 102 822-3-1 [69]. 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 [68].
`ServiceAvailabilty`	Defines the geographical regions in which this service is available or not available.

C.3.9 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>
```

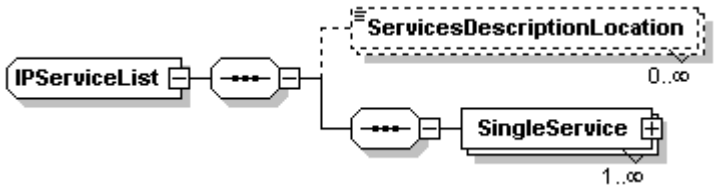

Figure C.5: IPServiceList

This type represents a list of IP services that are grouped together with a single, optional DescriptionLocation.

Note that an instantiation of this type is also used within the RTSP specification as the data returned by the DESCRIBE and ANNOUNCE messages for a multicast service (see clause 6.3).

C.3.10 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:sequence>
</xsd:complexType>
```

This is used to hold a multicast address and optionally AL-FEC layer information. This supports source specific multicast (SSM) addressing if the Source attribute is specified. Otherwise it supports any source multicast (ASM) addresses.

C.3.11 MulticastAddressAttribute

```
<xsd:attributeGroup name="MulticastAddressAttributes">
    <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:attribute name="Streaming" type="dvb:StreamingType" use="optional"/>
    <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 supports Source Specific Multicast (SSM) addressing if the Source attribute is specified. Otherwise it supports Any Source Multicast (ASM) addresses.

C.3.12 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>
```

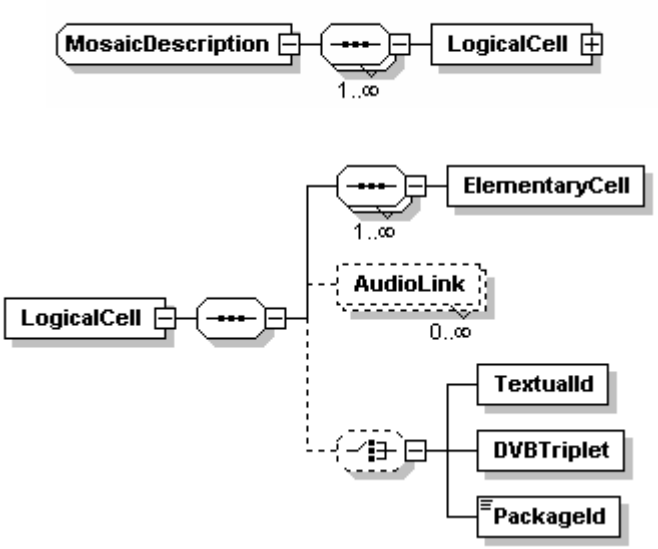

Figure C.6: MosaicDescription

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.3.13 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 [59] three letter language code.

C.3.14 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 ist used.

C.3.15 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>
```

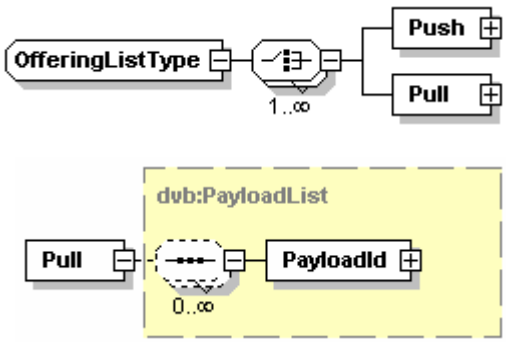

Figure C.7: OfferingListType

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.3.16 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>
```

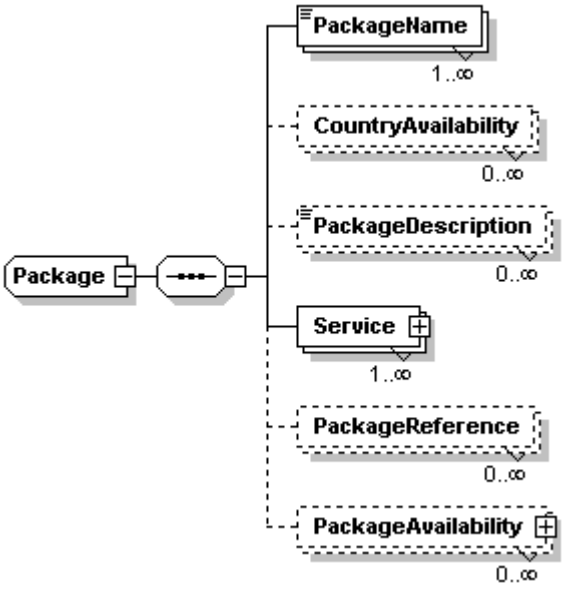

Figure C.8: Package

This provides a means to group services together into a "package" that the service provider can offer or refer to as a unit.

The attribute "Id" of a Package is an identifier used to identify a package, and service providers 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.3.17 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.3.18 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>
```

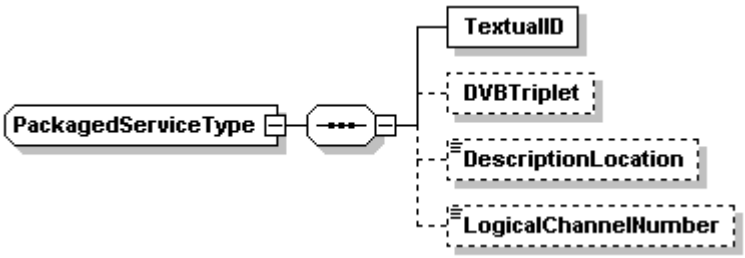

Figure C.9: PackagedServiceType

C.3.19 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>
```

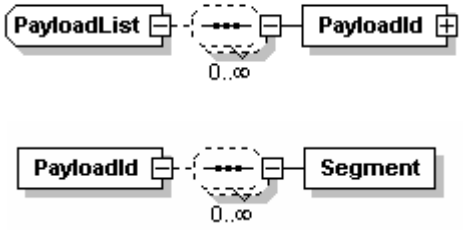

Figure C.10: PayloadList

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.3.20 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>
```

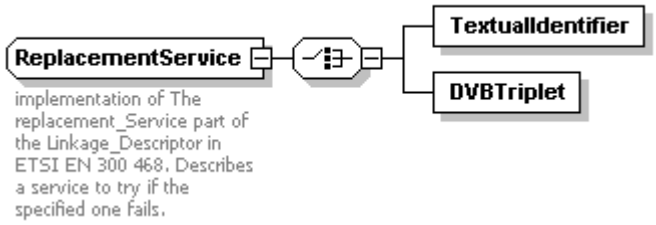

Figure C.11: Replacement Service

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.3.21 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>
```

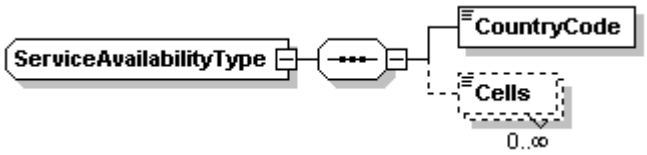

Figure C.12: ServiceAvailabilityType

Table C.3: Service Availability Fields

Name		Definition	Mandatory/Optional
CountryCode		This element indicates the country for which the availability is being defined.	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.3.22 ServiceLocation

```
<xsd:complexType name="ServiceLocation">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="IPMulticastAddress" type="dvb:McastType"/>
        <xsd:element name="RTSPURL" type="dvb:RTSP"/>
    </xsd:choice>
</xsd:complexType>
```

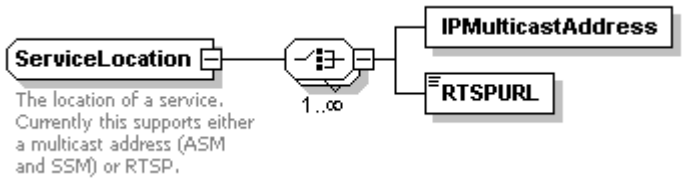

Figure C.13: ServiceLocation

This describes the location(s) at which a service may be found, either a multicast location or via an RTSP server.

C.3.23 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>
```

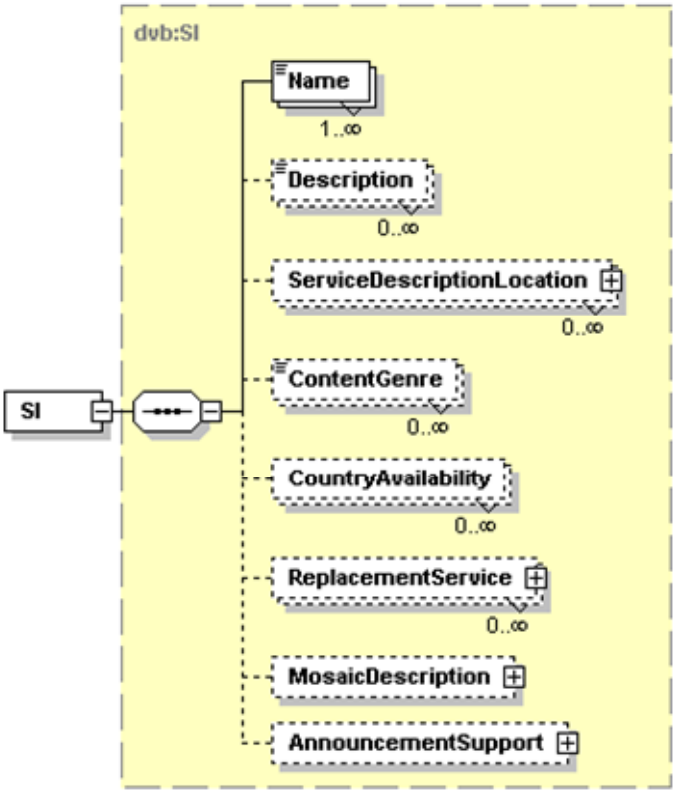

Figure C.14: SI

This type describes the service information traditionally provided in a stream as DVB descriptors.

Table C.4: 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.3.24 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 service provider 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 [4], clause 14.9.1.

C.3.25 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>
```

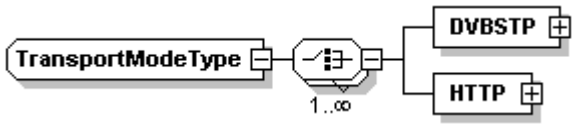

Figure C.15: TransportModeType

This type is used to indicate both the mechanism used to carry BCG information, and the payloadIds and segmentIds of the relevant information.

C.4 Element Types

C.4.1 BCGDiscovery

```
<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: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>
```

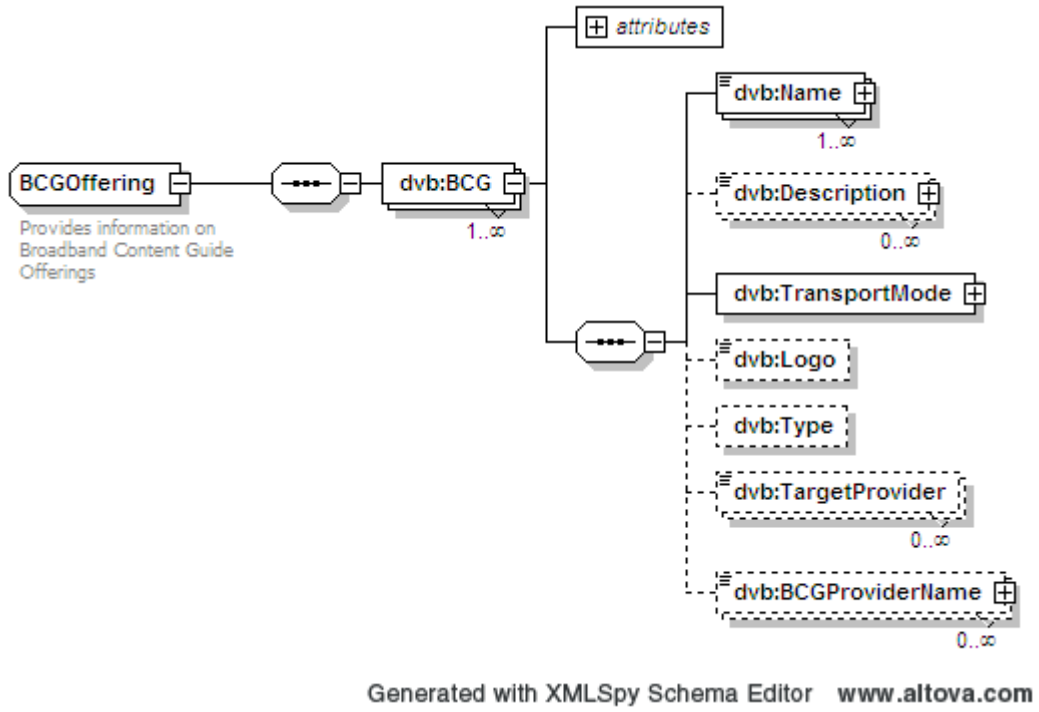

Figure C.16: BCGOffering

This element is used to discover BCG (Broadcast Content Guide) Offerings.

Table C.5: 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 service provider whose content is described by this BCG (for example Canal+). The domainName shall be the same as a domain name present in the ServiceList.

Name	Definition
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.

The classification scheme for the Type element is as follows:

```
<ClassificationScheme uri="urn:dvb:ipi:sdns:cs:BCGTypeCS:2006">
    <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>
```

C.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>
```

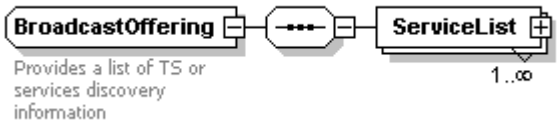

Figure C.17: Broadcast offering

This element is used where the service provider 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.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>
```

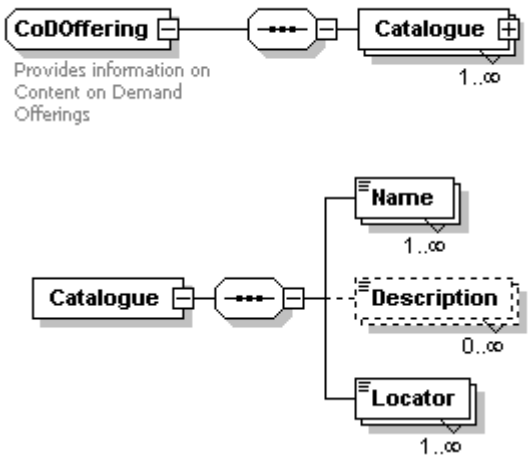

Figure C.18: Content on Demand

This element is used where the service provider is offering "content on demand" services.

Table C.6: Content on demand 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.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>
```

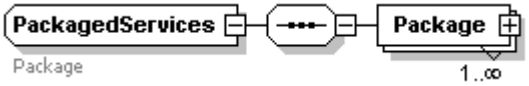

Figure C.19: Packaged services

C.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>
```

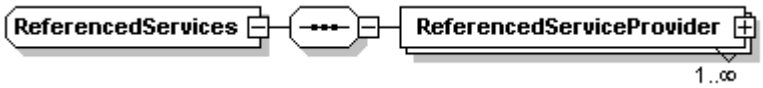

Figure C.20: Referenced services

This provides a means for a service provider to list services provided by other service providers from within his own service discovery information.

Table C.7: Referenced services fields

Name	Definition
ReferencedServiceProvider	A group of one or more service from a different service provider to which the service provider 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 service provider which is referred to.

C.4.6 ServiceProvider

```
<xsd:complexType name="ServiceProvider">
    <xsd:sequence>
        <xsd:element name="ServiceProvider" 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="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>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
```

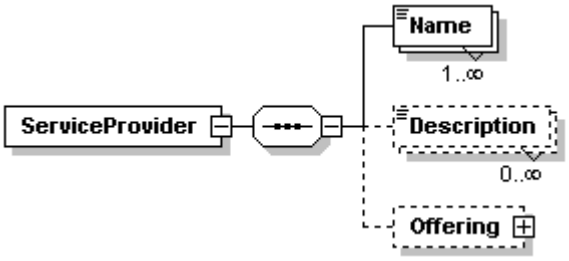

Figure C.21: Service Provider

This element is used in the first stage of service discovery. It is sent by service providers and is used as a link to their own service discovery information.

An aggregating service provide may send multiple ServiceProvider elements in a single document.

If the element Offering is missing, then the ServiceProvider is not currently providing any services, but simply announcing its presence.

Table C.8: 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.

C.5 Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="urn:dvb:ipi:sdns:2006" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:dvb="urn:dvb:ipi:sdns:2006" 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:ServiceProvider"
maxOccurs="unbounded"/>
                <xsd:element name="BCGDiscovery" type="dvb:BCGOffering" maxOccurs="unbounded"/>
            </xsd:choice>
            <xsd:attribute name="Version" type="dvb:Version" use="optional"/>
        </xsd:complexType>
    </xsd:element>
……
</xsd:schema>
```

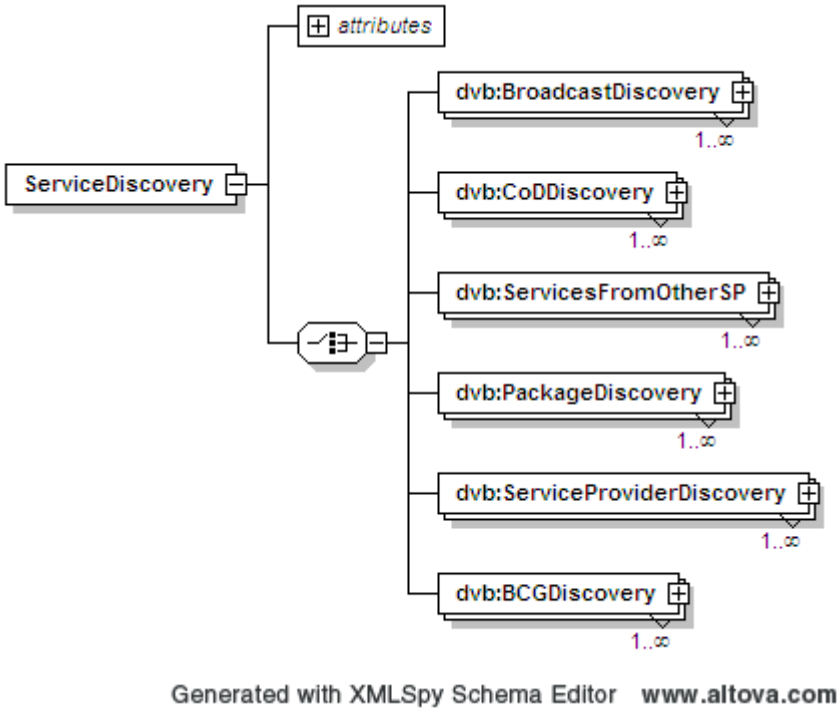

Figure C.22: Service discovery

Figure C.22 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 is used as described in clause 5.2.6. It is used to carry the version number of the XML document within the XML. Note that this is distinct from the version number carried in the ServiceProviderDiscovery record.

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).

C.6 Multicasting XML documents

Where multicast is used to distribute the service discovery information, the protocol defined in clause 5.4.1 shall be used. The following clauses define how the XML shall be mapped into the protocol.

C.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.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.6.2 Segmentation of records

Records containing service provider 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.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 service provider and the payload ID. For a shared multicast address the service provider 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 service provider, this information is inferred from the multicast address. With HTTP, the service provider is included in the request (see clause 5.4.2).

Segment Ids need not be contiguous.

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")".

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-IP 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 [76];
- the Raptor code, as defined in [74] and [75].

Note that the code defined in [76] 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 this specification 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 [76].

Clause E.4 defines the subsequent layers, based on [74] and [75].

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.
FEC Streaming Configuration Information	Information which controls the operation of the FEC Streaming Framework.
FEC Payload ID	See [56].
Source FEC Payload ID	See [56].
Repair FEC Payload ID	See [56].
FEC Object Transmission Information	See [56].
FEC Encoding ID	See [56].
Content Delivery Protocol	See [56].

E.3 SMPTE 2022-1-based code

SMPTE 2022-1 [76] based coding MAY be applied for streams which meet the requirements of [77].

All requirements of [76] and [77] 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 [76] and [77]

Clause from [76] and [79]	Modification/exception
[76] 6.1 RTP/UDP/IP layer	(R) The SSRC SHALL have the same value in every packet of the stream
[76] 7 FEC scheme	(N) The term "FEC Scheme" used here does not have the same meaning as "FEC Scheme" in the present document or in [74].
[76] 7.1 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.
[76] 7.1 FEC packet arrangement	(E) Only the first FEC stream shall be supported
[76] 6.1 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 burst size). Receivers MAY also support values of L and D outside this range.
[76] 7.4 FEC header format	(R) The D bit SHALL be set to 0
[76] 7.4 FEC header format	(R) The SNBase ext bits SHALL NOT be used
[76] 7.5 FEC traffic shaping issues	(E) The requirements of this clause SHALL NOT apply.
Annex B Non block aligned FEC arrangement	(E) The sending arrangement described in this Annex SHALL NOT be used.

E.4 Raptor code

E.4.1 Introduction

The FEC Building Block [56] 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-IP services. These components include:

- An FEC Streaming Framework, equivalent to that defined in [74], 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 [56] 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.

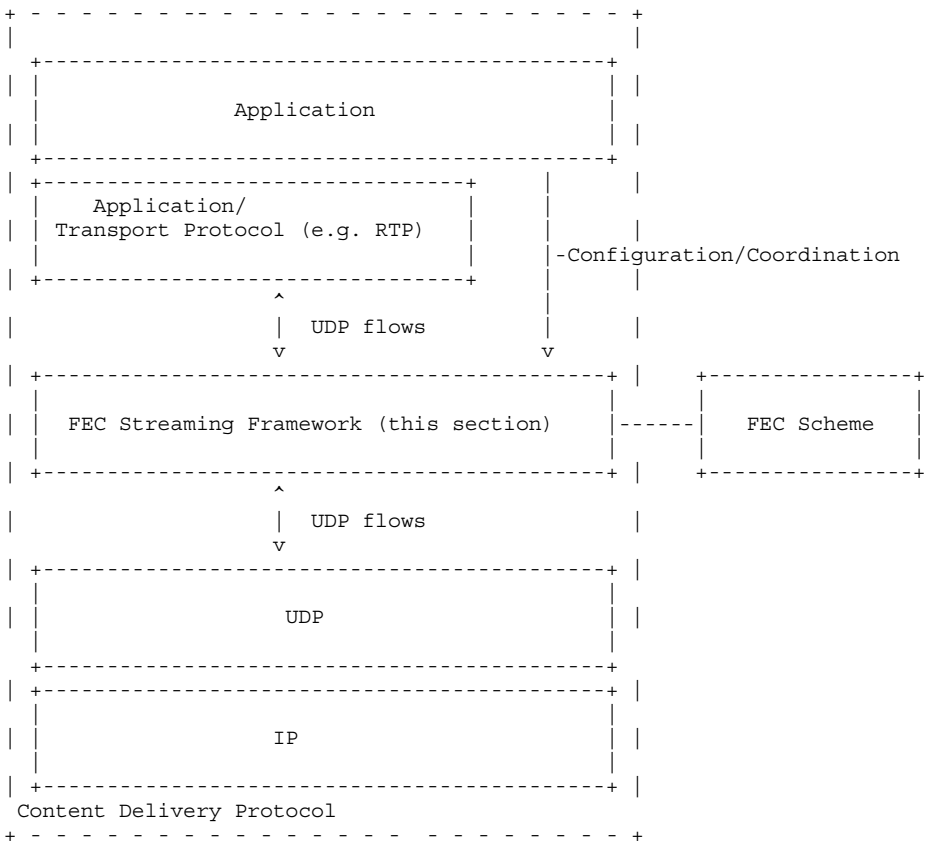

Figure E.1: FEC Streaming Framework Architecture

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*]×*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*]×*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*]×*T*, *i*=0, ..., *n*-1}.

Source blocks are identified by integer Source Block Numbers and symbols within a source block by integer Encoding Symbol IDs. This clause does not specify how Source Block Numbers are allocated to source blocks. Symbols are numbered consecutively starting from zero within the source block. Each source packet is associated with the Encoding Symbol ID of the first symbol containing SPI for that packet. Thus, the Encoding Symbol ID value associated with the *j*-th source packet, ESI[*j*], is given by:

$$ESI[j] = 0, \text{ for } j=0 \qquad (E.1)$$

$$ESI[j] = sum\{s[i], i=0,...,(j-1)\}, \text{ for } 0 < j < n \quad (E.2)$$

The Source FEC Packet Identification Information consists of the identity of the source block and the Encoding Symbol ID 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.

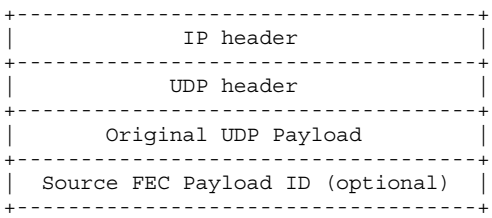

Figure E.2: Structure of FEC Source Packets

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.

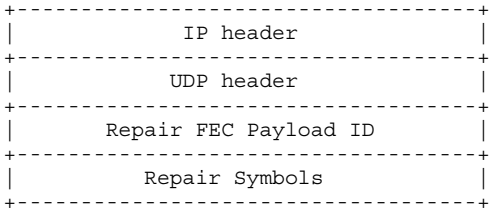

Figure E.3: FEC Repair packet format

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):

   d) a) Identification of the UDP packet flow carrying source packets.

   e) 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 [56];
- be systematic;
- be based on source blocks which are non-overlapping and contiguous within the stream;
- specify how the Source Block Number and Encoding Symbol ID 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:

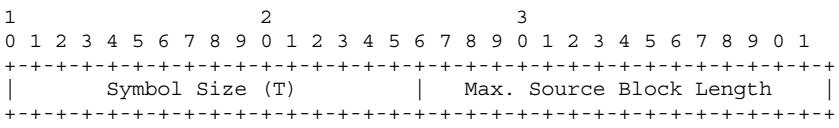

Figure E.4: Encoded Common FEC Object Transmission Information for Raptor FEC Scheme for arbitrary packet flows

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:

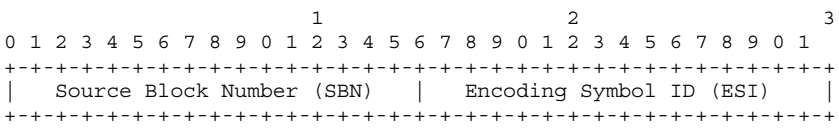

Figure E.5: Source FEC Payload ID format for Raptor FEC Scheme for arbitrary packet flows

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 Encoding Symbol Identifier 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:

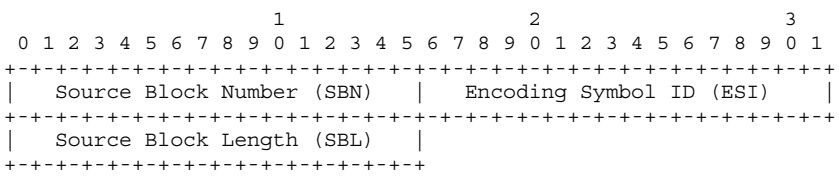

Figure E.6: Repair FEC Payload ID

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 Source Block Number, Encoding Symbol Identifier and Source Block Length 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 Source Block Numbers to source blocks sequentially, wrapping around to zero after Source Block Number 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 Source Block Length signaled in the FEC Object Transmission Information. Thus the value of the parameter K used by the FEC encoded is equal to the Maximum Source Block Length 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 Source Block Length 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 Source Block Length parameter).

The Source Block Length 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 Source Block Length.

E.4.3.1.5 Transport

This sub-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 this specification.

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 below, 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 format for this FEC Scheme is shown in figure E.7.

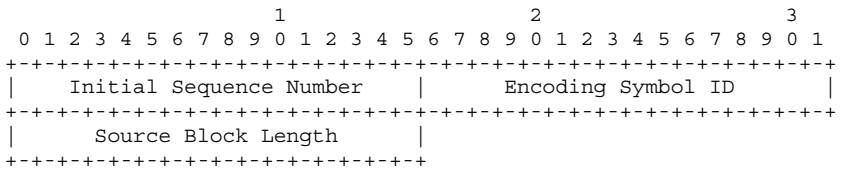

Figure E.7: Repair FEC Payload ID format

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 Source Block Numbers to source blocks sequentially, wrapping around to zero after Source Block Number 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 Initial Sequence Number 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 Initial Sequence Number (ISN) and Source Block Length (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 Encoding Symbol ID 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 Encoding Symbol ID 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 Source Block Length and Initial Sequence Number.

E.4.3.2.2.2 Derivation of repair packet Encoding Symbol IDs

The Encoding Symbol ID for a repair packet indicates which repair symbols the packet contains. This is given directly by the Encoding Symbol ID 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

$$P = \text{ceil}((n \cdot 188 + 15)/A) \cdot A \quad \text{(E.7)}$$

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 below, 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 [76] 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 [76], 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 [75]) 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 [76] 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 (amongst other fields):

- An FEC header containing:
  - The Length Recovery field, which is the XOR of the RTP payload lengths of the protected packets.
  - The XOR of the Payload Type (PT) fields of the RTP headers of the protected packets.
  - The XOR of the Timestamp (TS) fields of the RTP headers of the protected packets.
- The XOR of the RTP payloads of the protected packets.

In the first step of the conversion operation, the fields of each received or recovered source packet protected by a received SMPTE 2022-1 FEC packet, are XORed into the corresponding fields of the FEC packet. After this operation, the fields of the FEC packet are each equal to the XOR of the corresponding fields of the remaining (unrecovered) protected packets (which we call the "unrecovered protected packets").

In the second step of the conversion operation, for each remaining SMPTE 2022-1 FEC packet, the following fields are concatenated to form a "virtual" Raptor repair packet payload:

- A single zero byte.
- A two byte length indication, which is equal to the XOR of the RTP payload lengths of the unrecovered protected packets, taken directly from the SMPTE 2022-1 FEC packet.
- 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.
- Seven (7) zero bits, corresponding to the XOR of the RTP Padding (P), Extension (X), CSRC Count (CC) and Marker (M) bits of the unrecovered protected packets, which are all required to be zero according to SMPTE 2022-1.
- A seven (7) bit field equal to the XOR of the RTP Payload Type (PT) fields of the unrecovered protected packets (taken directly from the corresponding field of the SMPTE 2022-1 FEC header).
- 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* values of the FEC header of the FEC packet as per SMPTE 2022-1.
- A 32-bit field equal to the XOR of the RTP Timestamp (TS) fields of the unrecovered protected packets (taken directly from the corresponding field of the SMPTE 2022-1 FEC header).

- 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 RTP payloads of the unrecovered protected packets, taken directly from the remainder of the SMPTE 2022-1 FEC packet.
- 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 required to all 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 [75] in terms of a Gaussian Elimination process upon a matrix **A**. If decoding is not possible without use of the SMPTE 2022-1 packets, then this decoding process will fail during the second phase described in clause C.7 of [75]. 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 [75], not the L value associated with the SMPTE 2022-1 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 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 or MPEG-2 TS encapsulated in UDP).

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-IP 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.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 [75]. As a results, the example decoder procedures described in [75] 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, 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 this specification.

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 [75]. 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] ⊕ A
    ELSE
        A := A ⊕ 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:

$$C[10] := C[10] \oplus C[6] \oplus C[4] \oplus C[8] \oplus C[3] \oplus C[5]$$

History

Document history		
V1.1.1	March 2005	Publication
V1.2.1	September 2006	Publication
V1.3.1	October 2007	Publication