ETSI TS 102 034 V1.1.1 (2005-03)

Technical Specification

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

Reference
DTS/JTC-DVB-135

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

Important notice

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

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

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

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

Copyright Notification

Contents

Intellectual Property Rights

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

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

Foreword

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

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

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

Founded in September 1993, the DVB Project is a market-led consortium of public and private sector organizations in the television industry. Its aim is to establish the framework for the introduction of MPEG-2 based digital television services. Now 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 provides a first set of standardized specifications for early deployments of DVB services over bi-directional IP networks. The phase 1 work is limited to MPEG-2 DVB services [1] encoded with MPEG-2 technology [63], [64] and encapsulated in MPEG-2 TS [62] 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 the system 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 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 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 MPEG-2 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.
- 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.

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 Segments

The present document does not cover home networking but does not conflict with protocols like UPnP. These protocols can be added to the IPI-1 interface as options.

Clauses 11 and 12 provide respectively specifications for the carriage over Ethernet and IEEE P1394.1 [10].

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_102034v010101p0.zip which accompanies the present document. The DTDs and XML schemas included in the present document are informative.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- 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.
- For a non-specific reference, the latest version applies.

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

[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: ETSI ETR 162 is expected to be replaced with ETSI TR 101 162 (see bibliography).

[4] ETSI TS 101 225 (V1.1.1): "Digital Video Broadcasting (DVB); Home Local Network Specification based on IEEE 1394".

[5] ETSI TS 101 812 (V1.3.1): "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.0.3".

[6] IEEE Std 802-1990: "Overview and Architecture".

[7] IEEE Std 802.1Q-1998: "IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks".

[8] IEEE 802.2-1998: "IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements--Part 2: Logical Link Control".

[9] IEEE 802.3 2000 Edition: "CSMA/CD Access Method and Physical Layer Specifications".

[10] IEEE P1394.1 (Draft1.01): "Draft Standard for High Performance Serial Bus Bridges".

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

[12] IETF RFC 791: "Internet Protocol".

[13] IETF RFC 826: "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: "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 1889: "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 2734: "IPv4 over IEEE 1394".

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

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

[48] IETF RFC 2855: "DHCP for IEEE 1394".

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

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

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

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

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

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

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

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

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

[58] IETF Draft: "Dynamic Configuration of IPv4 Link-Local Addresses".

[59] ISO 3166: "Codes for the representation of names of countries and their subdivisions".

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

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

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

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

[64] ISO/IEC 13818-3 (1997): "Information technology - Generic coding of moving pictures and associated audio information - Part 3: Audio".

[65] 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".

[66] XML, Extensible Markup Language (XML) 1.0 (Second Edition). T. Bray, E. Maler, J. Paoli, and C. M. Sperberg-McQueen W3C Recommendation, October 2000.

[67] XML Schema, W3C Recommendations, 2nd of May 2001.

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): A Content Service Provider acquires/licenses content from Content Providers and packages this into a service.

Delivery Network (DN): the 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, e.g. 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. Systems coding is defined in Part 1 [62]. The real time interface specification is defined in Part 9 [65].

package: collection of DVB services marketed as a single entity

program: (taken from ISO/IEC 13818-1 [62]) A collection of program elements. 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.

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): the 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 [62]

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 [62], all other MPEG-2 and DVB tables are optional

3.2 Abbreviations

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

ABNF	Augmented Backus-Nauer Form
A/V	Audio/Video
BNF	Backus-Nauer Form
CPU	Central Processing Unit
CoD	Content on Demand
CoS	Class of Service
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
EUI	End-system Unique Identifier
HN	Home Network
HNCD	Home Network Connecting Device
HNED	Home Network End Device
HNS	Home Network Segment
HTC	Head-end Time Clock
HTTP	Hyper Text Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ID	IDentifier
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4

IPI	Internet Protocol Infrastructure
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
STC	(MPEG-2) System Time Clock
TCP	Transmission Control Protocol
TLS	Transaction Layer Security
ToS	Type of Service
T-STD	(MPEG-2) Transport Stream System Target Decoder
TV	TeleVision
UDP	User Datagram Protocol
URI	Uniform Resource Indentifier
URL	Uniform Resource Locator
UTF	Unicode Transformation Format
UTC	Coordinated Universal Time
VCR	Video Cassette Recorder
VOD	Video On Demand
WWW	World Wide Web
XML	eXtensible Markup Language

3.3 Notation

3.3.1 Augmented Backus-Nauer Form

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 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

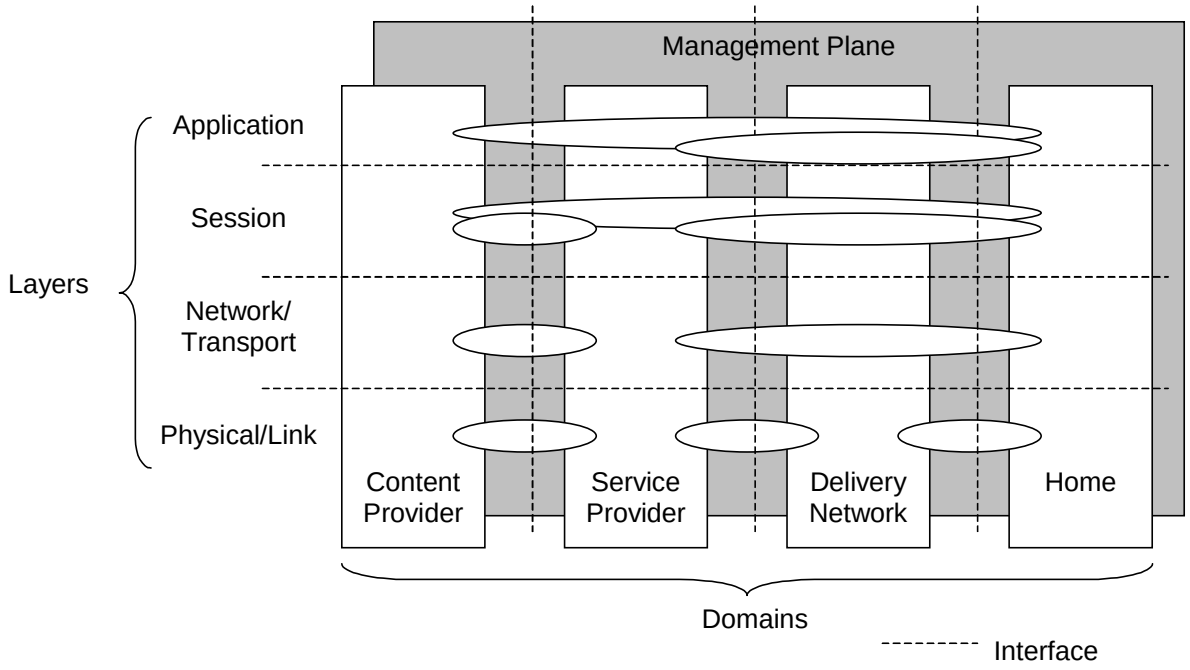


Figure 1: Layer Model

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 Reference Model

The architecture of the DVB-IPI network shall support the following (non-exhaustive) list of possible scenarios taken from TM2456 (see bibliography):

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) Access to the content is independent from the underlying hardware.

   As an example, terminals with different capabilities (e.g. CPU power, display size, storage capacity) may be allowed to access to the same content through the use of transcoding resources, or through the use of device specific resources.

5) Roaming of end users between delivery networks should be possible.

   As an example, the personal environment of a (SOHO) user stored on a home server should be accessible from different external locations. Adequate security aspects need to be taken into account.

Based on these scenarios a reference model for the DVB-IPI home network can be constructed. This reference model is depicted in figure 2.

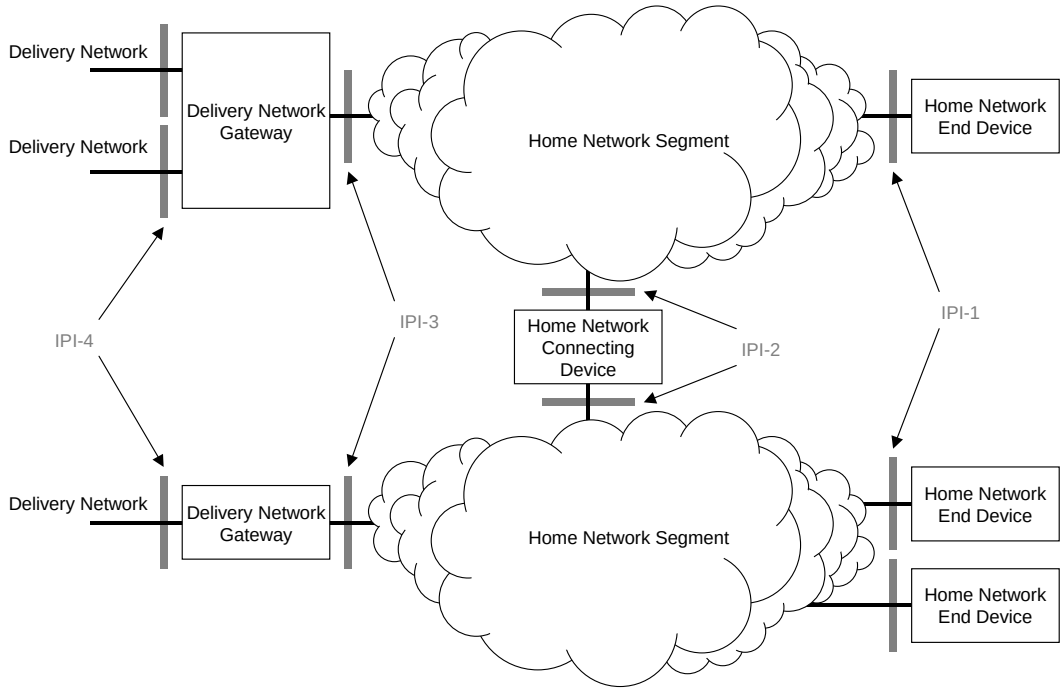


Figure 2: Home Reference Model

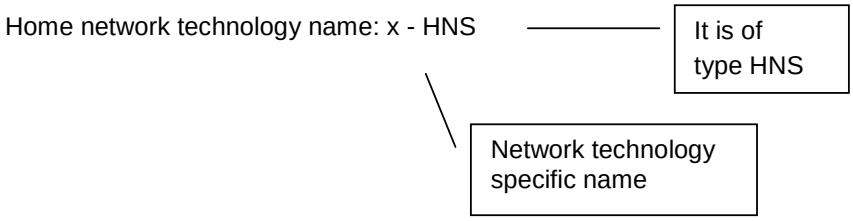
The Home Reference Model, as depicted in figure 2, consists of the Home domain of the Layer Model. Furthermore, it shows the interconnection with the Delivery Network domain. This Home Reference Model shows the elements that can be present in the home and their mutual relation. Based on the fact that 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. The separation of a home network into home network segments does not imply that each segment needs to be an "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.



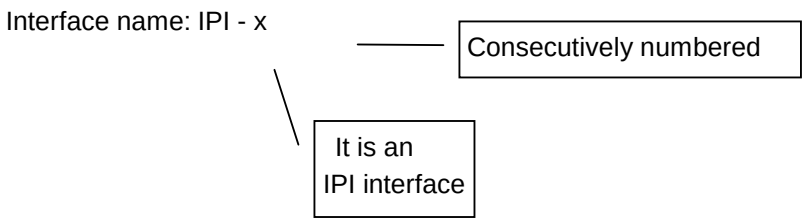
Some examples: Ethernet - HNS, IEEE 1394 - HNS, HiperLAN/2 - HNS.

- Home Network Connecting Device (HNCD): this device, which contains one or more connecting components, connects two or more home network segments with each other and functions as a bridge, router, or gateway.

- 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.

  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 Reference Model can be described by means of interfaces, which are provided in the figure by using the following naming principle:



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.

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. However, for use in future DVB Home Networking, the present document also specifies the Ethernet and IEEE 1394 Home Network Segments as physical layers. They are shown in their correct place, at the bottom of the diagram of figure 3.

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 paragraphs 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 paragraph are shown with a specific fill color.

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 [62]. The resulting MPEG-2 packets are encapsulated in RTP, 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.

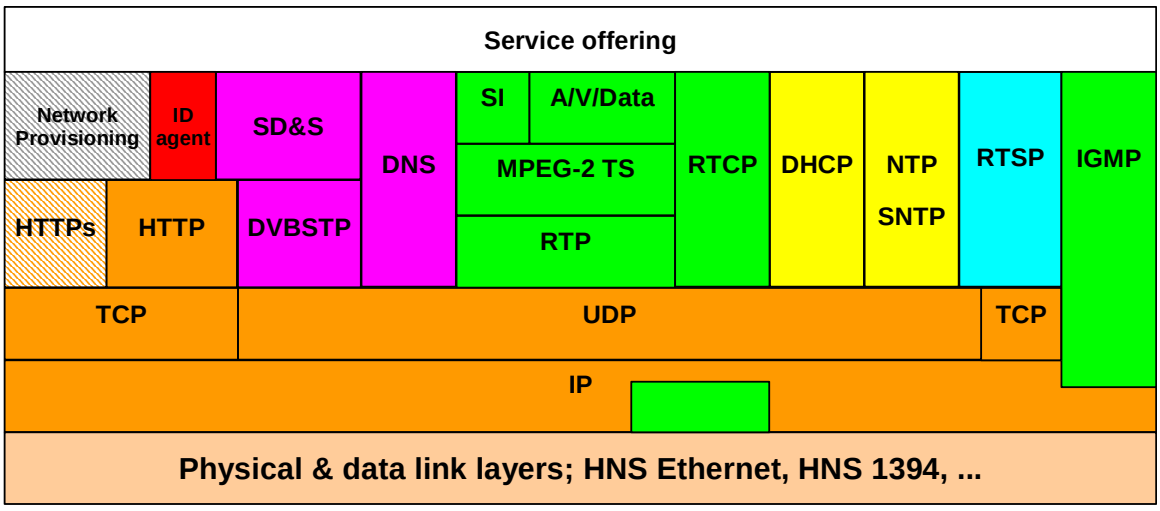


Figure 3: diagram of the protocol stack for DVB-IP services

4.1.4 Quality of Service

Quality of Service shall be enabled using the Differentiated Services approach [38].

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

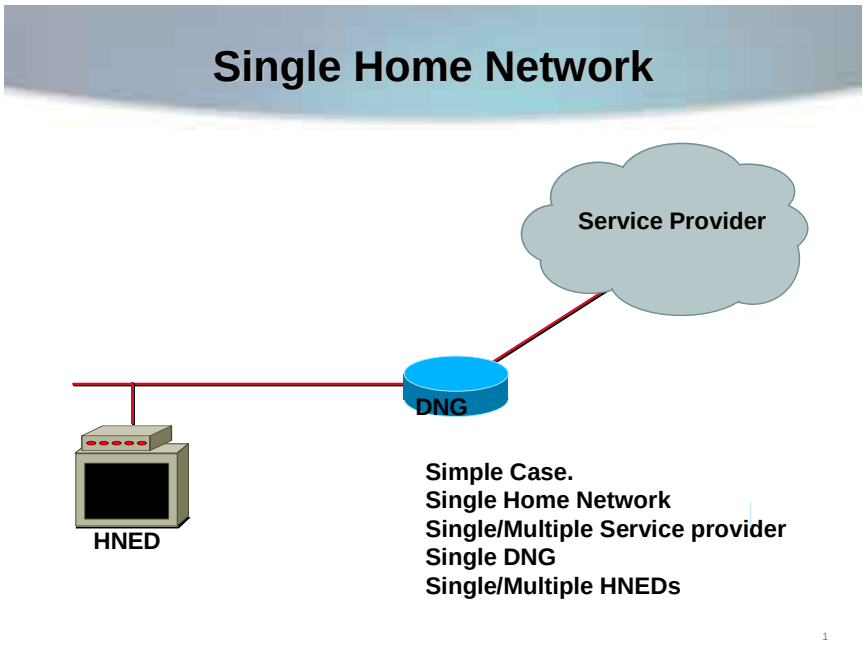


Figure 4: Single Home Network

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

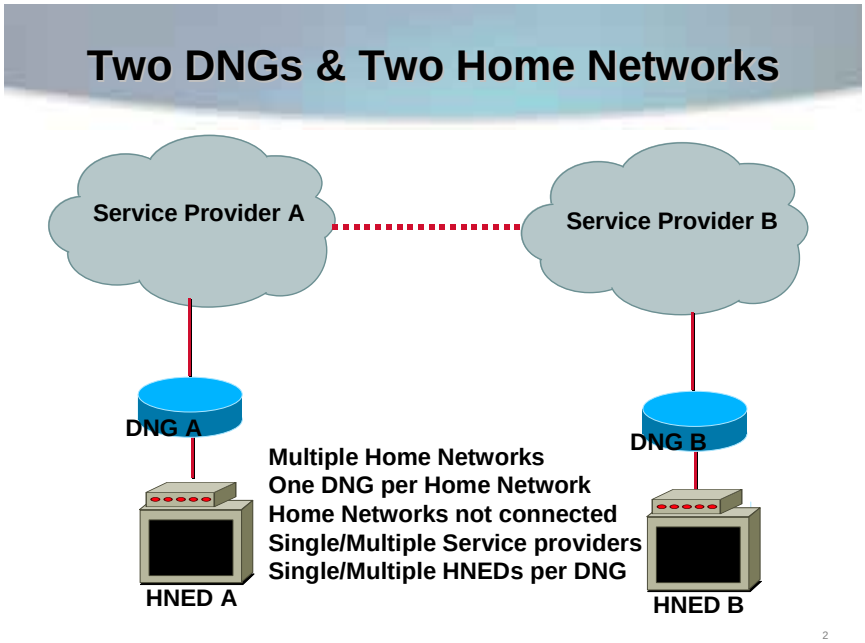


Figure 5: Multiple Delivery Network Gateways

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 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 [66] and the XML schemas [67] 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.

Service Name or Service Id

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 in MHP (specification 1.0.2, section 14.9 [5]).

Both can be used for identifying a service globally 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.

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:

<service_name> "." <service_provider_domain_name>

where <**service_name**> is a unique name for the service within the service provider's domain. 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]. The <**service_provider_domain_name**> is an Internet DNS domain name that the service provider has rights to control.

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".

This textual identifier provides a mechanism for uniquely identifying a service.

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
0x06-0xEF	Reserved
0xF0-0xFF	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.

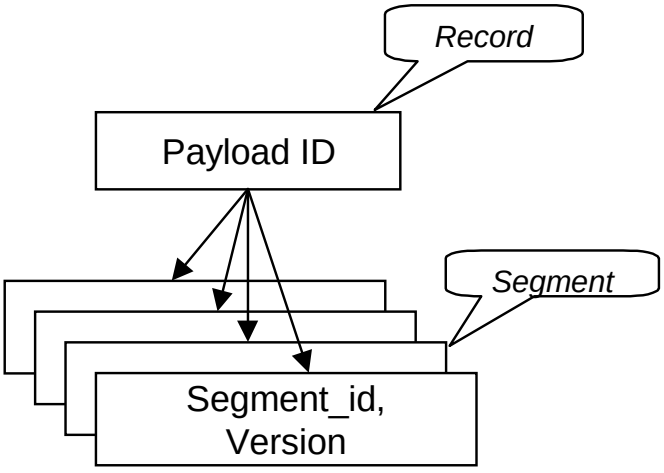


Figure 6: Relationship between Records, Payload Ids and Segments

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.

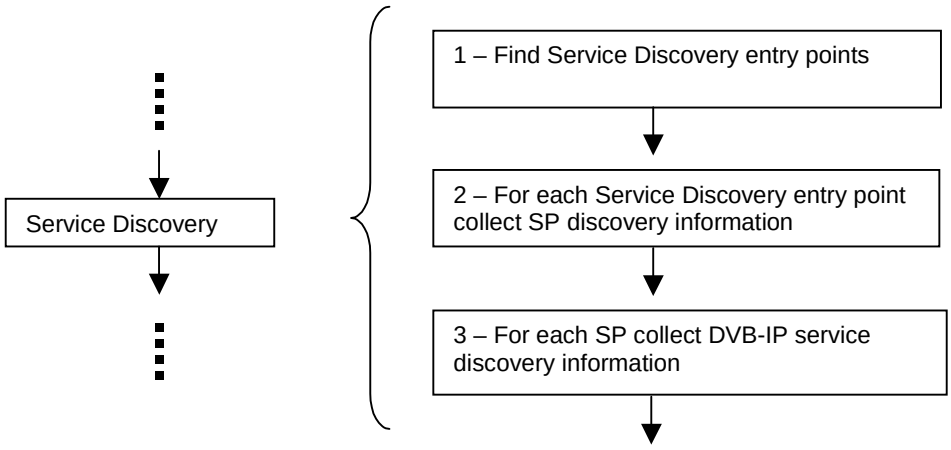


Figure 7: Steps in Service Discovery

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 [46]. 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 [46]. 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 [46] 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 [46].
- The HNED can be provided with entry points as part of the configuration data received on a provisioned network. The adresses of the entry points will be specified in the element sdEntries of the configuration DTD see clause 10.2.6.

NOTE: the DNS mechanism as described in RFC 2782 [46] 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 sdEntries. 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). Both models are supported.

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

Service Provider(s) Discovery Record	Attribute Description	Mandated/ Optional/ Conditional
Service Provider Domain Name	An internet DNS domain name registered by the Service Provider that uniquely identifies the Service Provider	M
Version Number	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
Multilingual Service Provider 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
Multilingual Service Provider Description	Description of the Service Provider for potential display in one or more languages; one description is allowed per language code	O
Service Provider Logo	Pointer to a Service Provider logo for potential display. The pointer shall be a URI [21].	O
Location(s)	This URI [21] encodes the location of the DVB IP Offering(s) Records which describe the offerings that the Service Provider makes available.	O
Payload 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
List of Segment Id(s) and their version number	Indicates which segment carries service discovery information of type payload Id and the version number of the segment.	C see note

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

NOTE: 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 record will not be used, except where it is inherited by one of the subsequent records.

Table 3: DVB-IP Offering Record

DVB IP Offering Record	Attribute Description	Mandated/ Optional/ Conditional
Service Provider Domain Name	An internet DNS domain name registered by the Service Provider that uniquely identifies the Service Provider	M
Version Number	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

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

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").

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 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

"TS Full SI" Broadcast Discovery Record	Attribute Description	Mandated/ Optional
Service(s) Descriptions Location(s)	A URI [21] which conveys the physical address of aggregated service descriptions; this provides additional information on services (metadata, events, etc.) and is not required to access a service.	O
Service(s)	For each service	
Service Identifier	A unique identifier for the service	M
Textual service identifier	A unique identifier for the service; it is a unique DNS host name under a DNS domain controlled by the Service Provider.	M
Service Provider Domain Name	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
Service host name	a unique host name for the service within the service provider's domain	M
DVB Triplet	The DVB service identifier triplet	M
Original Network Id	gives the network Id of the originating delivery system	M
TS Id	Identifies the Transport Stream	M
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
Max Bitrate	Specifies the maximum bitrate of the overall stream carrying the service	O
Service Locator(s)	A locator specifying where the service may be found. Both IGMP and RTSP protocols are supported; at least one of the two protocols will be specified in this field	M
IGMP protocol	Signals the use of IGMP to access the service	O
Address	IP multicast address	M (see note)
Port	Port number	M (see note)
Source Address	IP unicast address of the source of the TS	O
RTSP protocol	Signals the use of RTSP to access the service	O
Textual Service Identifier	A unique identifier for the service; it is a unique DNS host name under a DNS domain controlled by the Service Provider. If this field is not present, then the textual Service Identifier from the current record is used.	O
NOTE: The Mandatory here means that if the Optional IGMP protocol information 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.

Table 5: "TS - Optional SI" Discovery Information

"TS - Optional SI" Discovery Record	Attribute Description	Mandated/ Optional
Service Descriptions Location(s)	A URI [21] which conveys the physical address of aggregated service descriptions; this provides additional information on services (metadata, events, etc.) and is not required to access a service.	O
Service(s)	For each service	
Service Identifier	A unique identifier for the service	M
Textual service identifier	A unique identifier for the service; it is a unique DNS host name under a DNS domain controlled by the Service Provider.	M
Service Provider Domain Name	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
Service host name	a unique host name for the service within the service provider's domain	M
DVB Triplet	The DVB service identifier triplet	M
Original Network Id	gives the network Id of the originating delivery system	M
TS Id	Identifies the Transport Stream	M
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
Multilingual Service 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
Multilingual Service Description	Description of the service for potential display in one or more languages; one description per language code	O
Content Description/genre	Indicates the general 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
Service Type	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
Max Bitrate	Specifies the maximum bitrate of the overall stream carrying the service	O
Country availability	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.	O
Announcement Support descriptor	The announcement support descriptor 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
Service Replacement descriptor	Identifies a service replacement service which may be selected automatically by the IRD when the service being decoded fails.	O
Mosaic Description	The mosaic descriptor 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
Primary SI Source	Indicates which source of service information to give priority (XML record or DVB SI) in case DVB SI tables are present.	O

"TS - Optional SI" Discovery Record	Attribute Description	Mandated/ Optional
Service Description Location(s)	A URI [21] which conveys where additional information on this service (metadata, events, etc.) may be found.	O
Service Locator(s)	A locator specifying where the offering may be found. Both IGMP and RTSP protocols are supported; at least one of the two protocols will be specified in this field	M
IGMP protocol	Signals the use of IGMP to access the service	O
Address	IP multicast address	M (see note)
Port	Port number	M (see note)
Source Address	IP unicast address of the source of the TS	O
RTSP protocol	Signals the use of RTSP to access the service	O
Textual Service Identifier	A unique identifier for the service; it is a unique DNS host name under a DNS domain controlled by the Service Provider. If this field is not present, then the textual Service Identifier from the current record is used.	O
NOTE: The Mandatory here means that if the Optional IGMP protocol 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 TS.

5.2.6.3 Content On Demand Discovery Record

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

Content on Demand Discovery Record	Attribute Description	Mandated/ Optional
Content on Demand Offering(s)	For each Content on Demand Provider/Server	
Content on Demand Provider Id	Identifies a Content on Demand Provider/Server; This Id is allocated by the Service Provider	M
Multilingual Content on Demand Offer/Server 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
Multilingual Content on Demand Offer/Server description	Description of the Content on Demand general offering for potential display in one or more languages; one description per language code	O
Content on Demand Description Location	Address of URI [21] where the aggregated content descriptions can be found (catalogue/metadata).	M

An HTTP request on the "Content on Demand Description Location" 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

Services From other Service Providers Record	Attribute Description	Mandated/ Optional
Service Provider(s)	For each referenced service provider	
Service Provider Domain Name	An internet DNS domain name registered by the referenced Service Provider that uniquely identifies the service provider being referenced	M
Service List	For each service from the referenced provider. Not required if the entire set of offerings from the service provider is referenced.	
Service host name	a unique host name for the service within the referenced service provider's domain	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

Package(s) Discovery Record		Mandated/ Optional
Package(s)	For each package	
Package Identifier	Identifies a package; this ID is allocated by the Service Provider	M
Multilingual Package Name	Name of the package for display in one or more languages; one name per language code	M
Package Description Location(s)	A URI [21] which conveys where additional information on the package may be found.	O
Country availability	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.	O
Service list(s)	List of services forming the package	M
Service Provider Domain Name	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
Service Host Name	a unique host name for the service within the service provider's domain	M
Service Description Location	The URI [21] of additional service description provided in the context of a package; this is not required to acquire a service	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.

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 attribute "Service Locator" in the service discovery records gives all the information required to issue the appropriate IGMP message.

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 attribute "Service Locator" 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.

Content on Demand 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 service. 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

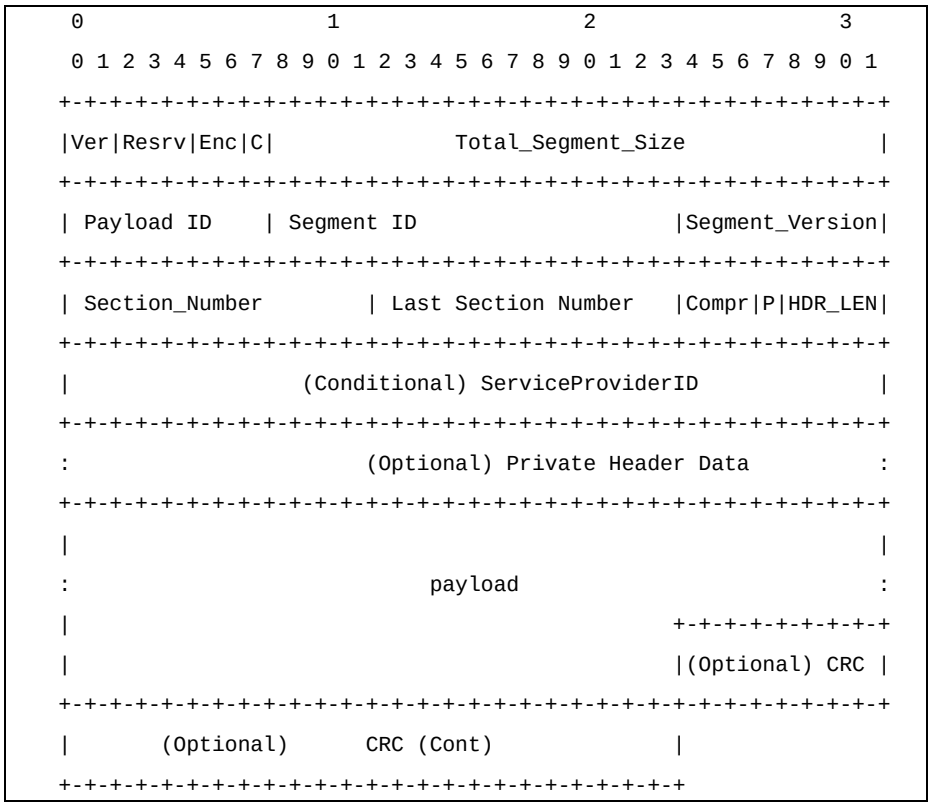


Figure 8: Syntax SD&S multicast delivery protocol

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 9.

Table 9: Compression Values

Compression value	Meaning	Total Segment Size Meaning
000	No Compression	Transmitted Size
001 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 13818-1:2000 [62], 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.

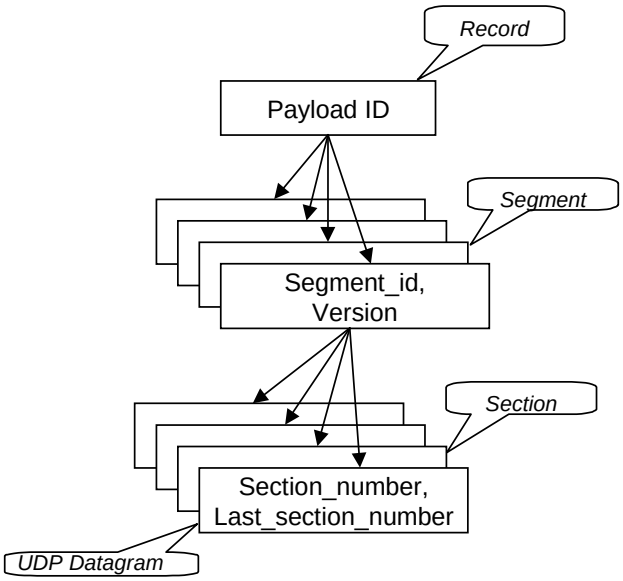


Figure 9: Relationship between Records, Segments and Sections

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 1492, 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 uncompressed and 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.

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

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 dicovery 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.

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 10) to the HNED.

Multimedia streams, which are encapsulated in RTP 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 10). Or, the HNED can also poll the server with the aid of a DESCRIBE method (see table 10) 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 can be delivered in either unicast or multicast mode. 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 can be delivered in multicast mode.

6.3 RTSP Methods

For unicast mode of delivery, table 10 specifies for each profile the RTSP methods to be supported by the IPI-1 interface.

Table 10: 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.4.1).

The MIME Type in the Content-Type header (see table 12) 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 [54] 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 [54] 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 11 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 11: 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 12 presents the RTSP header fields that are generated by the HNED and are either mandatory or recommended for the IPI-1 interface.

Table 12: 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: 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 at least `RTP/AVP/UDP` transport. The following transport configuration parameters should be provided by the HNED to help configuring intermediaries: `unicast`, `multicast` and `client_port`
User-Agent	MAY	**SHOULD**	The following format for the User-Agent header is recommended: `User-Agent = "User-Agent" ":" deviceID " HNED V1.0"` See also clause 3.3. E.g.: `User-Agent: PHILIPS-CE/HN3200/A6743ABCD201 HNED V1.0`

NOTE 1: The column IETF presents the request headers required to be supported according to the IETF RTSP specification: RFC 2326 [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 12.

Table 13 presents the RTSP header fields that are received by the HNED and are either mandatory, or recommended for the IPI-1 interface.

Table 13: 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	
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: 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	At least `RTP/AVP/UDP` transport shall be supported. Furthermore, the HNED should support (and the server is expected to provide) at least the following transport configuration parameters: `unicast`, `multicast`, `destination`, `port`, `client_port`, `source` and `server_port`. These parameters can help intermediaries in forwarding the multimedia stream in question.
Timestamp	MAY	**SHOULD**	
Unsupported	SHALL	SHALL	

NOTE 1: The column IETF presents the response headers required to be supported according to the IETF RTSP specification: RFC 2326 [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 13.

6.4 Status Codes in Response to Requests

Table 14 lists the RTSP and HTTP status codes that the RTSP enable HNED shall be able to interpret.

Table 14: 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"*
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 the IPI-1 interface supports IGMP version 3 [57].

The transport configuration parameters: `destination` and `source` (see table 13) 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 [57]).

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 13.

For multicast mode of delivery, table 15 presents the RTSP methods to be supported by the IPI-1 interface.

Table 15: 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 12 and 13).
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: There is no distinction between unicast and multicast mode for LMB. NOTE 2: The keywords in bold indicate where the DVB specification differs from the IETF. NOTE 3: 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 MPEG-2 Transport Stream (MTS), 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 MPEG-2 transport streams shall be encapsulated in RTP (Real-time Transport Protocol) according to RFC 1889 [23] in conjunction with RFC 2250 [34]. The ad-hoc UDP encapsulation used in some current systems shall not be used.

RTP always uses an even UDP port number (see RFC 1889 [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.

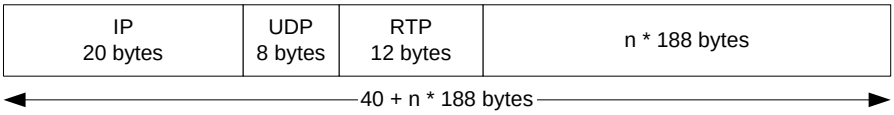


Figure 10: Minimal Packet Format (IPv4)

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 frame with LLC) or 1 500 bytes (IEEE 802.3 frame without LLC, see IEEE 802.3 [9] and IEEE 802.2 [8]), 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.

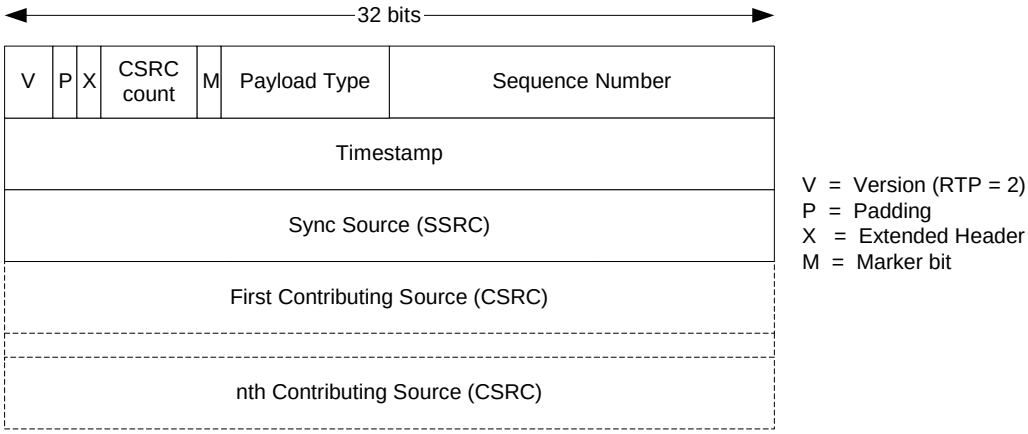


Figure 11: RTP Header Format

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 [62].

Other fields are completed as per RFC 1889 [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 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 1889 [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 support 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 support Sender Reports. CSPs are recommended to send Sender Reports to enable receivers 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.

A receiver 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 Embedded Service Information (SI)

All transport streams shall comply with the MPEG-2 system specification [62] and as such include the following program specific information tables:

- Program Association Table (PAT);
- Program Map Table (PMT).

These tables shall be used in accordance with the DVB SI specification [1] and usage guidelines [2].

For transport streams with optional SI (TS - optional SI) , all other MPEG-2 [62] and DVB [1] tables 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 receivers.

7.2.1 Mandatory constraints

7.2.1.1 RTP Packet Jitter

MAXIMUM 40 ms peak-to-peak

RTP packet jitter is defined as the variation in delay between the source of the RTP 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 receiver shall comply with the MPEG-2 Real Time Interface Specification (ISO/IEC 13818-9 [65]) with t_jitter = 20 ms.

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}.

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 RTP 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 RTP 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 [57].

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 end device 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 end device 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. End devices 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 all IPI interfaces shall be appropriately marked, as described in the next clause. The network operator is responsible for managing the traffic to and from the HNED to provide the required QoS.

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 16 shall be used. It is recommended that the full DSCP value be used.

Table 16: DSCP Markings

Traffic Type	IP DSCP Value	Corresponding IP Precedence
Voice Bearer (see note 1)	0b101110	0b101
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.		

8 IP Address Allocation & 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 12.

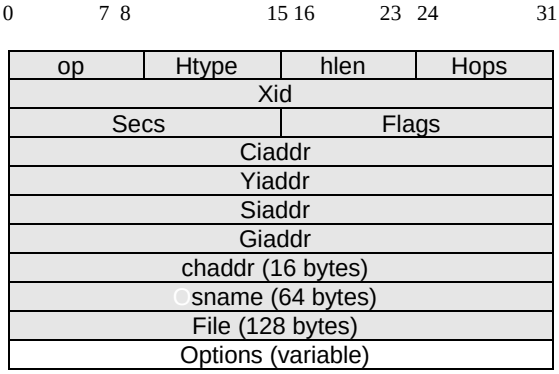


Figure 12: DHCP Format

The messages contain a variable size options part that allows the message to carry additional information other than the IP address. We will divide 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 [56] (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 [56] (as stated in 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]). If IEEE 1394 is being used then the messages should be modified according to RFC 2855 [48] and EUI-64 used. 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 17.

Table 17: 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/Disable 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
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

Option Description	Reference (RFC 2132 [29] unless otherwise stated)	Option Number	Support on IPI-1
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 378bytes, 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 [50]	117	Not required
User Class	RFC 3004/4.0 [52]	77	Mandatory
Subnet Selection	RFC 3011/2.0 [53]	118	Mandatory
Relay Agent Information	RFC 3046/2.0 [55]	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. 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 18: 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 [55]) 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 the IETF Zero Configuration Link Local Addressing scheme [58], however, this is still an Internet draft at the time of writing (January 2004).

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, however, only one of these services needs to be implemented by the HNED.

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

The real time clock 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).

8.3 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).

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 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, or the EUI-64 identifier of the IEEE 1394 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 [47] 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 [66] 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 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, or the EUI-64 identifier of the IEEE 1394 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, or the EUI-64 identifier of the IEEE 1394 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 [66] 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 [61] 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)>
<!-- isakmp, ... -->
<!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) [49]-->
<!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)[49]-->
<!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>
```

11 Ethernet Home Network Segment

11.1 Topology of an Ethernet home network segment

Based on the description of a Home Network Segment (HNS) in the DVB-IP architecture (see clause 4), the 100BASE-T Ethernet HNS is based on a star architecture, with use of unshielded twisted pair (UTP) cabling to connect between nodes. For an architecture overview see [6]. Note that use of 10BASE-T is not recommended, due to potential EMC problems and the requirement not to interfere with any existing licensed radio services.

The description of the Home Reference Model in the DVB-IP architecture (clause 4), furthermore introduces a Home Network Connecting Device (HNCD). An HNCD can act as a bridge, router or gateway and connects HNSs with each other. The 100BASE-T Ethernet HNS may be connected via a HNCD to another DVB HNS e.g. IEEE 1394.

The specification for a HNCD that interconnects IEEE 802 LANs (below the MAC service boundary) in a bridged format is defined in IEEE 802.1Q [7]. 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 HNCD shall provide support for QoS via IEEE 802.1p (see clause 11.3).

An example configuration allows for a 100BASE-T Ethernet HNS to connect a Delivery Network Gateway (DNG) to Ethernet based Home Network End Devices (HNEDs). The DNG may present a single 100BASE-T interface to the Ethernet HNS (in this case a HNCD in the form of an external hub or switch is required to connect multiple terminals), or the DNG may provide multiple 100BASE-T interfaces (in hub or switch format). To provide guaranteed QoS it is recommended that a switched configuration is used. Note that the DNG may provide a bridged or routed connection.

11.1.1 The Ethernet Layer

Ethernet 100BASE-T is specified in IEEE 802.3u [9]. The 802.3 MAC layer shall be used as defined in IEEE 802.3 [9]. The Link layer specified in IEEE 802.2 [8] shall be used.

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.

11.1.2 Ethernet Physical Layer

HNEDs connected to the 100BASE-T Ethernet HNS shall support the IEEE 802 100BASE-TX Ethernet physical layer as defined in [9]. RJ45 Ethernet sockets shall be used.

11.2 Carriage of IP-based traffic

Within the context of the architectural framework clause 4, all IP-based traffic shall be carried transparently over a 100BASE-T Ethernet network. Therefore, the interfaces IPI-1, IPI-2 and IPI-3 on a 100BASE-T Ethernet 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 addressing of Home Network End Devices (HNED) on a 100BASE-T Ethernet network DHCP shall be supported. Each HNED should be uniquely identified by its MAC address (48 bit Ethernet address).

All other IP-based functionality required to carry IP traffic over a 100BASE-T HNS, via an HNCD connecting two 100BASE-T HNSs, or via an HNCD connecting a 100BASE-T HNS with another type of HNS can be found in clause 8.

11.3 QoS

The interfaces IPI-1, IPI-2 and IPI-3 on the Ethernet 100BASE-T HNS shall support IEEE 802.1Q [7], with defined user priority classes. The IEEE 802.1p field shall be supported in an IEEE 802.1Q [7] compliant Ethernet frame. The marking shall be based on the DiffServ CodePoint (DSCP) marking method [51] as described in clause 7.4.1.

Table 19: 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)	0b101110	0b101
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 100BASE-T Ethernet these DSCP values are used to map a traffic type onto the corresponding IEEE 802.1p priority codes. Packets shall be marked using the Layer 2 Class of Service (CoS) settings in the User Priority bits of the 802.1p 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 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 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. It is not as easy to map the DSCP field to the user-priority field, as the DSCP is 6 bits in length and the user-priority field is only 3 bits in length. Therefore the IP precedence portion of the DSCP field cannot be copied into the user-priority field. Instead the DSCP field shall be tested for values that match the DSCP value shown in Column 2. If the DSCP value does not match any of the values shown in Column 2, the packet shall be marked with a user-priority value of 0.

12 IEEE 1394 Home Network Segment

12.1 Topology of an IEEE 1394 home network segment

Based on the description of a Home Network Segment (HNS) in the architectural framework clause 4, an IEEE 1394 HNS shall consist of one IEEE 1394 bus. The physical topology of an IEEE 1394 bus is a tree structure.

The description of the Home Reference Model in the architectural framework clause 4, furthermore introduces a so-called Home Network Connecting Device (HNCD). An HNCD can act as a bridge, router or gateway and connects HNSs with each other. Based on this, an HNCD connecting two IEEE 1394 buses with each other shall contain a bridge component based on P1394.1 [10], the IEEE 1394 bus bridge specification.

The general specification of the IEEE 1394 media and connectors, physical layer, and data link layer, can be found in clauses 3.1 through 3.3 of TS 101 225 [4].

12.2 Carriage of IP-based traffic

Within the context of the architectural framework clause 4, all IP-based traffic shall be carried transparently over an IEEE 1394 network. Therefore the interfaces IPI-1, IPI-2, and IPI-3 on an IEEE 1394 - HNS, shall comply to RFC 2734 [45], the IPv4 over IEEE 1394 specification.

For the addressing of devices on an IEEE 1394 network, some DHCP fields shall be filled in in a specific way, due to the fact that IEEE 1394 uses a different link layer addressing method than conventional IEEE 802/Ethernet. Therefore, all the IP-based traffic shall also comply to RFC 2855 [48]. This means that this IETF specification shall be supported on the interfaces IPI-1, IPI-2, and IPI-3 in case of an IEEE 1394 HNS.

All other IP-based functionality required to carry IP traffic over an IEEE 1394 HNS, via an HNCD connecting two IEEE 1394 buses, or via an HNCD connecting an IEEE 1394 bus with another type of HNS can be found in clause 8.

12.3 QoS

Because it is expected that the IP traffic transported on a network will consist of types with varying importance, priority levels shall be given to the different traffic types. Therefore all traffic on the interfaces IPI-1, IPI-2, and IPI-3 shall be marked to ensure consistency. The marking shall be done as described in clause 7.4.1.

Table 20: DSCP Values and corresponding IEEE 1394 transaction code

Traffic type	IP DSCP value	Corresponding IEEE 1394 transaction code
Voice bearer (see note)	0b101110	0x0A
Video bearer(high priority)	0b100010	0x0A
Video bearer (lower priority)	0b100100	0x0A
Voice and video signalling	0b011010	0x01
Best effort data	0b000000	0x01
NOTE: The voice bearer is listed here to ensure that there is no interference with DVB-IP services		

For a HNS based on IEEE 1394 these DSCP values shall be used to map a traffic type on the corresponding IEEE 1394 transaction code (see table 20). The IEEE 1394 transaction codes are described in RFC 2734 [45].

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 [62]. 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):

$$t(i) = \frac{PCR(k-1)}{27MHz} + \frac{i - i''}{R(i)} \qquad \text{(Eq. 1)}$$

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:

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

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 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 2. (see also ISO/IEC 13818-1 [62]).

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 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 [65]. 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.

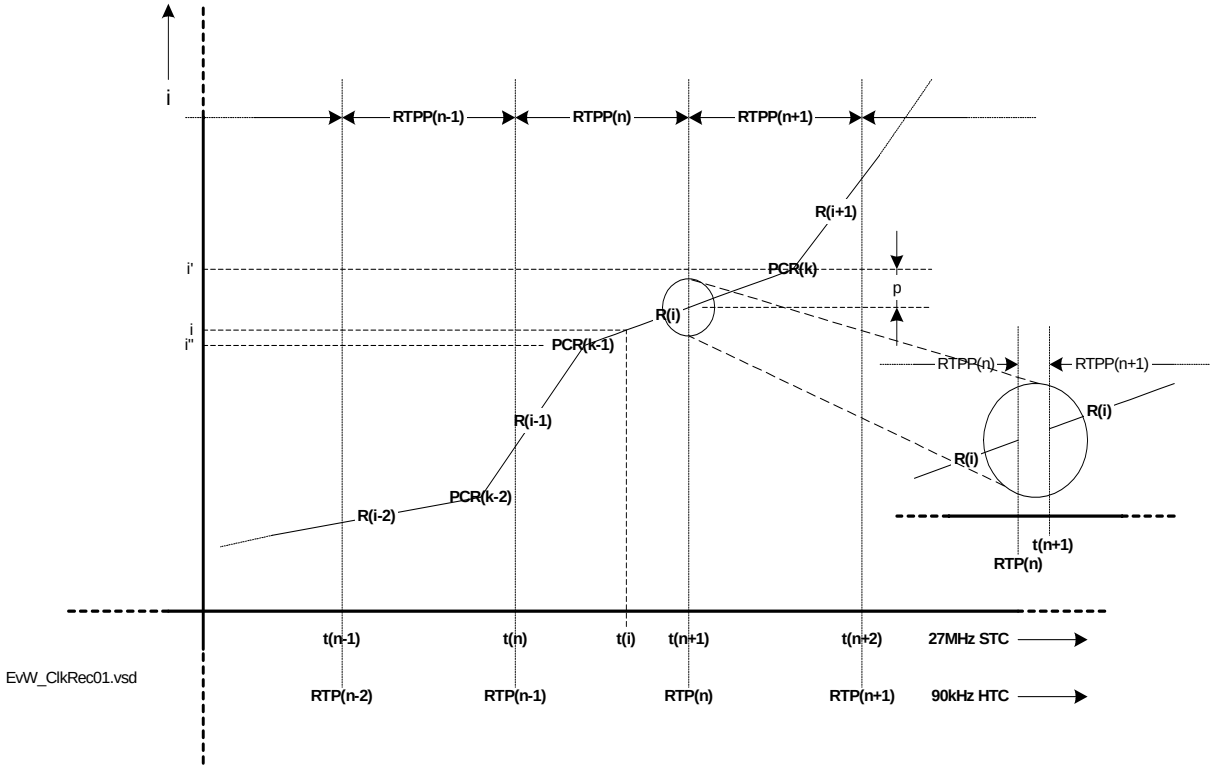


Figure A.1: Timing

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:

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

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 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:

$$PCR(k) \cong RTP(n) + 90kHz \times \frac{(p+1)}{R(i)} \qquad \text{(Eq. 4)}$$

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 [65].
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.

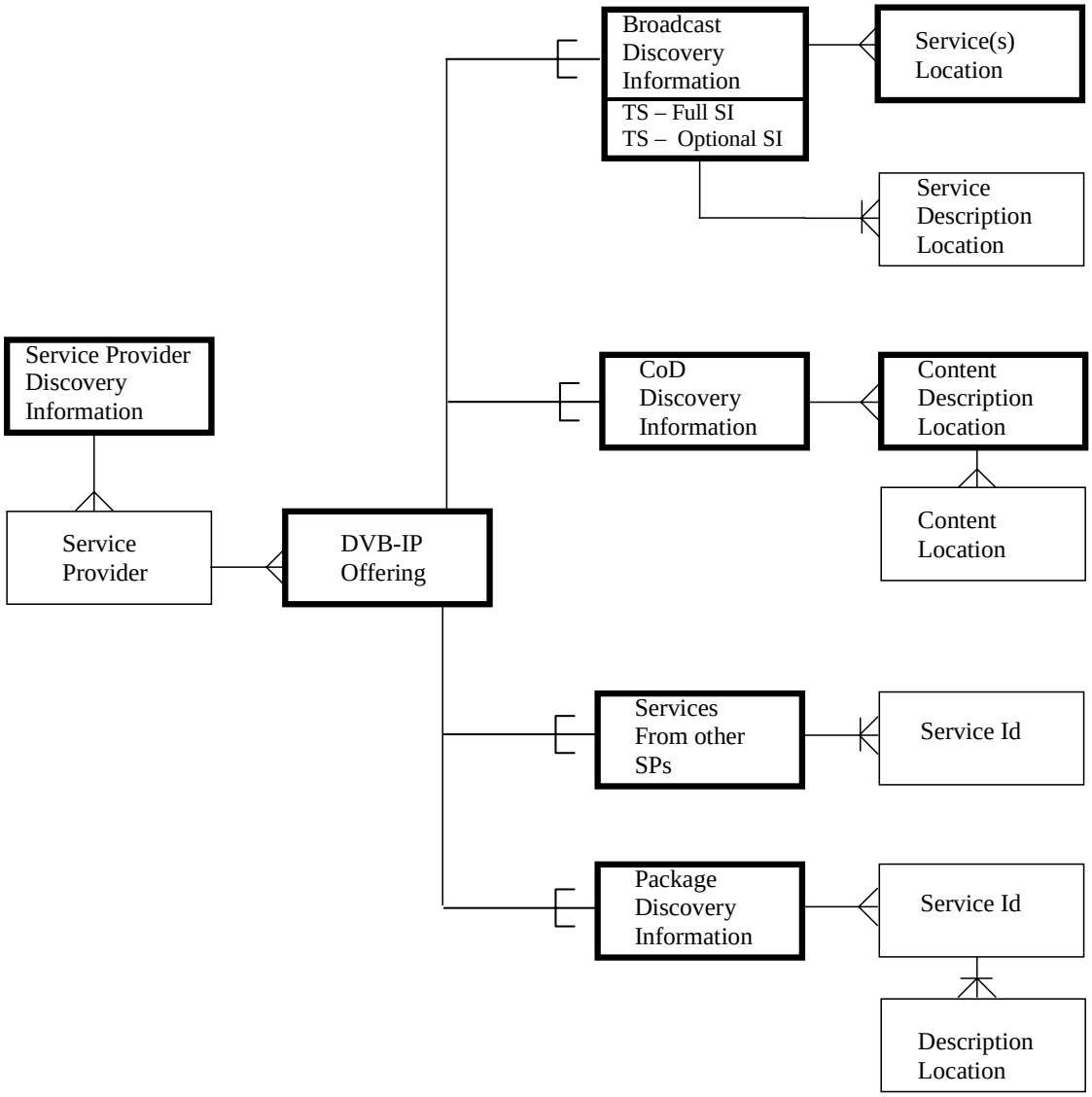


Figure B.1: Proposed data model for DVB-IP service discovery information

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 sections 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 in archive ts_102034v010101p0.zip) which accompanies the present document.

C.1.1 Namespace

The namespace the service discovery schema is urn:dvb:ipisdns:2003.

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 [59].

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 [60].

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="http://.*/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 [5], clause 14.9.1. 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 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.2.21 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>
```

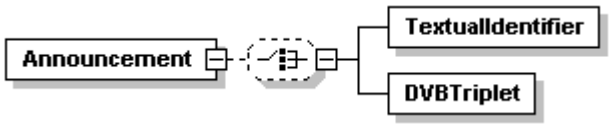


Figure C.1: AnnouncementSupport

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 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.3.4 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:sequence>
</xsd:complexType>
```

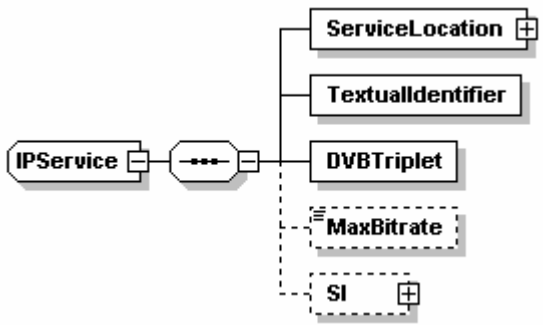


Figure C.2: IPService

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 at which the transport stream carrying the service will operate.
SI	Service information about the service carried.

C.3.5 IPServiceList

```
<xsd:complexType name="IPServiceList">
    <xsd:sequence>
        <xsd:element name="ServicesDescriptionLocation" type="dvb:DescriptionLocation" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:sequence>
            <xsd:element name="SingleService" type="dvb:IPService" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:sequence>
</xsd:complexType>
```

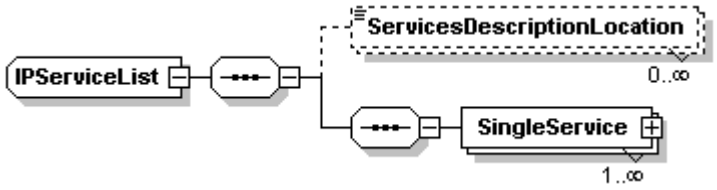


Figure C.3: IPServiceList

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.3.6 McastType

```
<xsd:complexType name="McastType">
    <xsd:attributeGroup ref="dvb:MulticastAddressAttributes"/>
</xsd:complexType>
```

This is used to hold a multicast address. This supports source specific multicast (SSM) addressing if the Source attribute is specified. Otherwise it supports any source multicast (ASM) addresses.

C.3.7 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:attributeGroup>
```

This supports source specific multicast (SSM) addressing if the Source attribute is specified. Otherwise it supports any source multicast (ASM) addresses.

C.3.8 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: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>
```

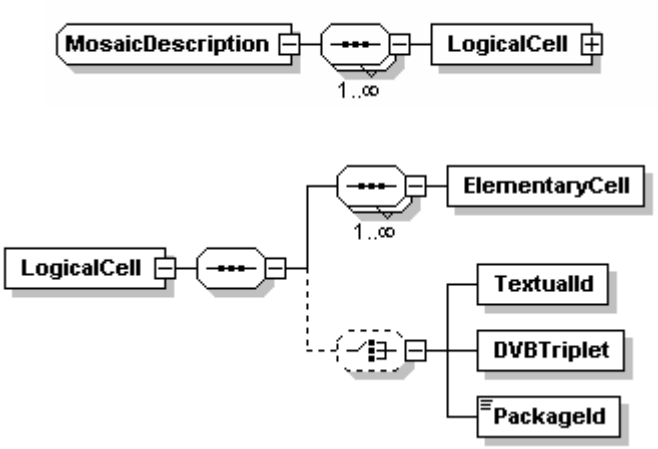


Figure C.4: MosaicDescription

An implementation of the Mosaic descriptor from EN 300 468 [1]. All fields are defined in EN 300 468 [1].

C.3.9 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 [60] three letter language code.

C.3.10 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.3.11 OfferingListType

```
<xsd:complexType name="OfferingListType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Push">
            <xsd:complexType>
                <xsd:complexContent>
                    <xsd:extension base="dvb:PayloadList">
                        <xsd:attributeGroup ref="dvb:MulticastAddressAttributes"/>
                    </xsd:extension>
                </xsd:complexContent>
            </xsd:complexType>
        </xsd:element>
        <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>
```

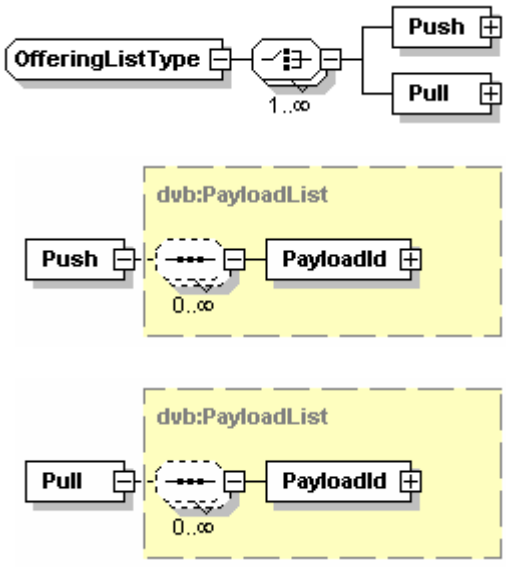


Figure C.5: OfferingListType

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.

C.3.12 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>
```

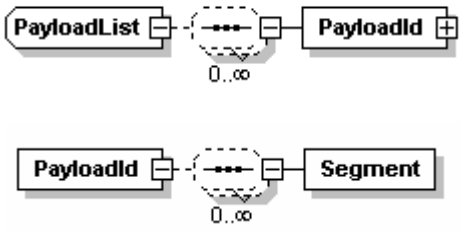


Figure C.6: PayloadList

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.3.13 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>
```

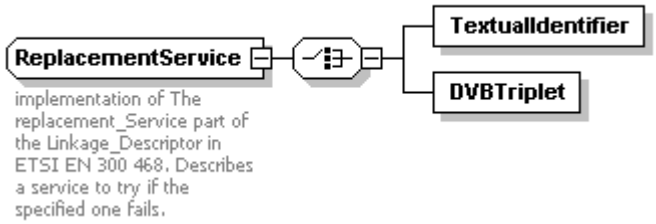


Figure C.7: Replacement Service

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.3.14 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>
```

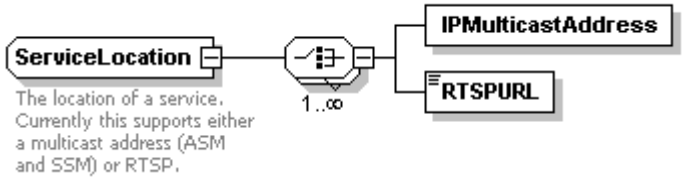


Figure C.8: ServiceLocation

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

C.3.15 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:DescriptionLocation"
minOccurs="0"/>
        <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>
```

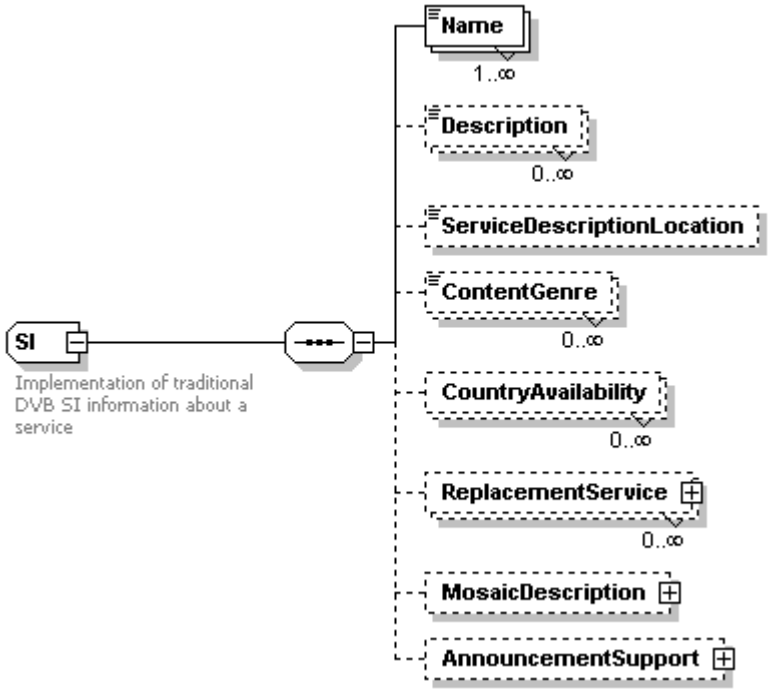


Figure C.9: SI

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

Table C.2: 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	A URI at which further information, on the service this SI record is associated, may be found.
ServiceType	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.3.16 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 provide 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 [5], clause 14.9.1.

C.4 Element Types

C.4.1 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>
```

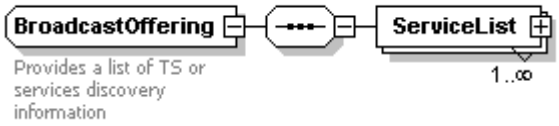


Figure C.10: Broadcast Offering

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.4.2 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>
```

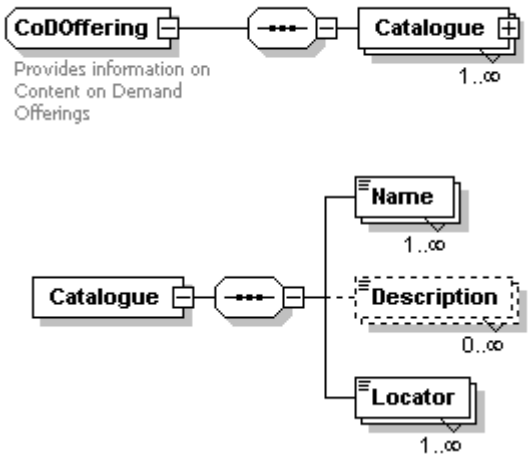


Figure C.11: Content on Demand

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

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.

C.4.3 PackagedServices

```
<xsd:complexType name="PackagedServices">
    <xsd:complexContent>
        <xsd:extension base="dvb:OfferingBase">
            <xsd:sequence>
                <xsd:element name="Package" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="PackageName" type="dvb:MultilingualType"
maxOccurs="unbounded"/>
                            <xsd:element name="Service" maxOccurs="unbounded">
                                <xsd:complexType>
                                    <xsd:sequence>
                                        <xsd:element name="TextualID" type="dvb:TextualIdentifier"/>
                                        <xsd:element name="DescriptionLocation"
type="dvb:DescriptionLocation" minOccurs="0"/>
                                    </xsd:sequence>
                                </xsd:complexType>
                            </xsd:element>
                            <xsd:element name="CountryAvailability" type="dvb:CountryAvailability"
minOccurs="0" maxOccurs="unbounded"/>
                            <xsd:element name="PackageDescription" type="dvb:DescriptionLocation"
minOccurs="0" 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>
```

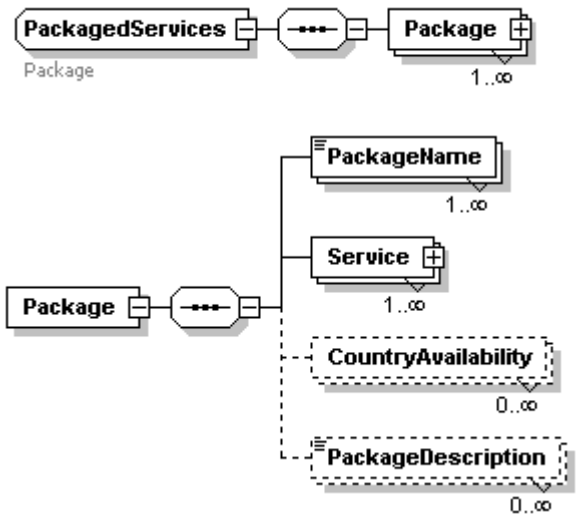


Figure C.12: Packaged Services

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.

Name	Definition
PackageName	The textual name of the package.
Service	One or more services which comprise the package, which contains:
TextualIdentifier	The textual identifier by which the service is known.
DescriptionLocation	An optional location for the description of the service.
CountryAvailability	The countries within which the package is, or is not, available.
PackageDescription	An optional textual description of the service.

C.4.4 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>
```

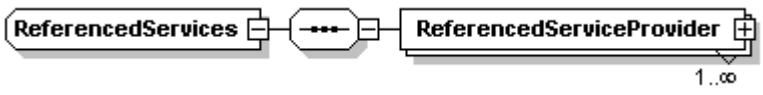


Figure C.13: Referenced Services

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

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.4.5 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>
```

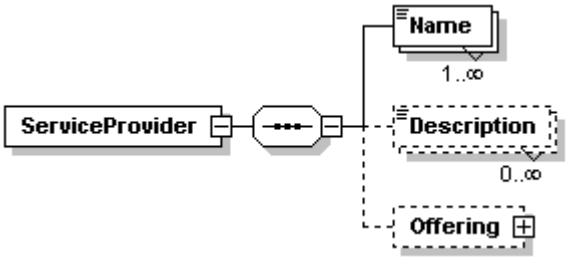


Figure C.14: Service Provider

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.

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:ipisdns:2003" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:dvb="urn:dvb:ipisdns:2003" elementFormDefault="qualified">
    <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:choice>
            <xsd:attribute name="Version" type="dvb:Version" use="optional"/>
        </xsd:complexType>
    </xsd:element>
……
</xsd:schema>
```

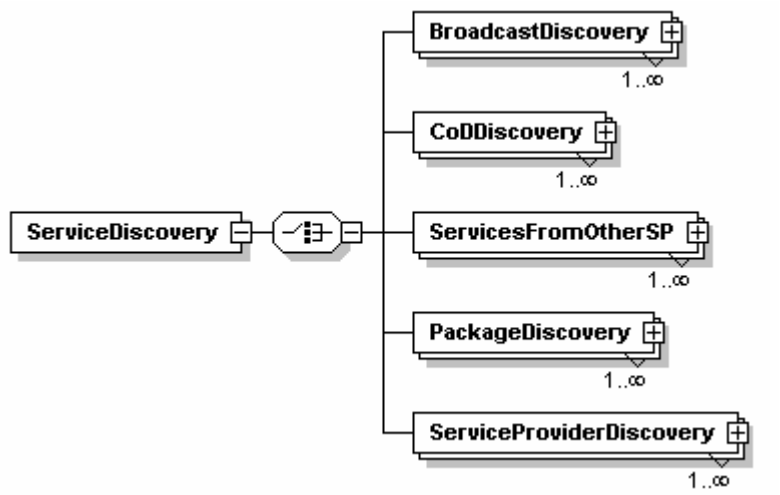


Figure C.15: Service Discovery

Figure C.15 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

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

IEEE 802.11 Handbook: "A Designer's Companion".

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")".

XML Schema Part 0: "Primer" http://www.w3.org/TR/2001/REC-xmlschema-0-20010502.

XML Schema Part 1: "Structures" http://www.w3.org/TR/2001/REC-xmlschema-1-20010502.

XML Schema Part 2: "Datatypes" http://www.w3.org/TR/2001/REC-xmlschema-2-20010502.

TM2456: "Commercial Requirements for Multimedia Services over Broadband IP in a DVB Context" (CM255r4), March 2000.

History

Document history		
V1.1.1	March 2005	Publication