Service discovery and delivery protocols for a DVB Home Broadcast system

DVB Document A179

February 2021

DVB®

Contents

Intellectual Property Rights

Essential patents

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 (https://ipr.etsi.org).

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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU) and the European Telecommunications Standards Institute (ETSI), and is now submitted for the combined Public Enquiry and Vote phase of the ETSI standards EN Approval Procedure.

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

The Digital Video Broadcasting Project (DVB) is an industry-led consortium of broadcasters, manufacturers, network operators, software developers, regulatory bodies, content owners and others committed to designing global standards for the delivery of digital television and data services. DVB fosters market driven solutions that meet the needs and economic circumstances of broadcast industry stakeholders and consumers. DVB standards cover all aspects of digital television from transmission through interfacing, conditional access and interactivity for digital video, audio and data. The consortium came together in 1993 to provide global standardization, interoperability and future proof specifications.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", " **need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

DVB-HB intends to enable consumption of traditional broadcast services (i.e., delivered via satellite, terrestrial or cable broadcast networks) by means of in-home networked devices, e.g., smartphones, tablets, Personal Computers, IP-enabled TV sets or Set-Top-Boxes, etc.

1 Scope

The present document defines the interfaces through which a DVB-HB Local Server can redistribute broadcast signals and associated metadata to DVB-HB Clients.

In particular, the present document specifies:

- Mechanisms allowing a DVB-HB Local Server to announce its presence on the LAN and a DVB-HB Client to discover it;
- Mechanisms allowing a DVB-HB Local Server to expose its capabilities in terms of available resources and distributed services;
- Backwards-compatible optional extensions to the SAT>IP specification, improving supported capabilities and network resilience;
- Backwards-compatible extensions to the DVB-I specification, adding support to DVB-HB functionalities.

Additionally, the present document provides informative guidelines on a number of aspects, e.g.:

- Conversion of DVB-SI metadata to DVB-I format;
- Combined use of HTTP and HTTPS in browser-based DVB-HB Clients;
- Encoding and packaging requirements and recommendations;
- Provision of HbbTV applications to DVB-HB Clients.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

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

NOTE 1: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] SAT>IP specification: "SAT>IP Protocol Specification, Version 1.2.2".

NOTE 2: Available at https://www.satip.info/sites/satip/files/resource/satip_specification_version_1_2_2.pdf.

[2] ETSI TS 103 285: "Digital Video Broadcasting (DVB); MPEG-DASH Profile for Transport of ISOBMFF Based DVB Services over IP Based Networks".

[3] ETSI TS 103 770: "Digital Video Broadcasting (DVB); Service Discovery and Programme Metadata for DVB-I".

[4] ETSI EN 300 421: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services".

[5] ETSI EN 302 307-1: "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 1: DVB-S2".

[6] ETSI EN 302 307-2: "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 2: DVB-S2 Extensions (DVB-S2X)".

[7] ETSI EN 300 744: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television".

[8] ETSI EN 302 755: "Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)".

[9] ETSI EN 300 429: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for cable systems".

[10] ETSI EN 302 769: "Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital transmission system for cable systems (DVB-C2)".

[11] ETSI TS 101 154: "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcast and Broadband Applications".

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

[13] ISO/IEC 29341-1-1: "Information technology - UPnP device architecture - Part 1-1: UPnP Device Architecture Version 1.1".

[14] IETF RFC 5261: "An Extensible Markup Language (XML) Patch Operations Framework Utilizing XML Path Language (XPath) Selectors".

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

[16] ETSI TS 102 809: "Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in Hybrid Broadcast/Broadband environments".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE 1: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI EN 303 560: "Digital Video Broadcasting (DVB); TTML subtitling systems".

[i.2] IETF RFC 3550: "RTP: A Transport Protocol for Real-Time Applications".

[i.3] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[i.4] IETF RFC 2818: "HTTP Over TLS".

[i.5] IETF RFC 1918: "Address Allocation for Private Internets".

[i.6] W3C Encrypted Media Extensions: "W3C Recommendation 18 September 2017 (Link to Editor's Draft updated 19 December 2019)".

NOTE 2: Available at https://www.w3.org/TR/encrypted-media.

[i.7] W3C Mixed Content, "Editor’s Draft, 10 September 2020".

NOTE 3: Available at https://w3c.github.io/webappsec-mixed-content/.

[i.8] CA/Browser Forum BR: "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates", Version 1.6.3.

NOTE 4: Available at https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.6.3.pdf.

[i.9] D. Nandakumar, S. Kotecha, K. Sampath, P. Ramachandran, T. Vaughan, "Efficient Multi-Rate HEVC Encoding for Adaptive Streaming," IBC White Paper, Amsterdam, 2016.

NOTE 5: Available at https://www.ibc.org/efficient-multi-bitrate-hevc-encoding-for-adaptive-streaming/822.article.

[i.10] FFmpeg, "Creating Multiple Outputs".

NOTE 6: Available at https://trac.ffmpeg.org/wiki/Creating%20multiple%20outputs.

[i.11] Android Developers Guides, "Supported media formats", Google.

NOTE 7: Available at https://developer.android.com/guide/topics/media/media-formats.

[i.12] IEEE 802.11: "IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

[i.13] D. Murray, T. Koziniec, M. Dixon and K. Lee, "WiFi multicast streaming using AL-FEC inside the trains of high-speed rails," IEEE international Symposium on Broadband Multimedia Systems and Broadcasting, Seoul, 2012, pp. 1-6.

[i.14] H. Chiao, S. Chang, K. Li, Y. Kuo and M. Tseng, "WiFi multicast streaming using AL-FEC inside the trains of high-speed rails," IEEE international Symposium on Broadband Multimedia Systems and Broadcasting, Seoul, 2012, pp. 1-6.

[i.15] C.A.G.D. Silva and C.M. Pedroso, "MAC-Layer Packet Loss Models for Wi-Fi Networks: A Survey," in IEEE Access, vol. 7, pp. 180512-180531, 2019.

[i.16] Y.J. Liang, J.G. Apostolopoulos and B. Girod, "Analysis of Packet Loss for Compressed Video: Effect of Burst Losses and Correlation Between Error Frames," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 7, pp. 861-874, July 2008.

[i.17] S. Aramvith, Chia-Wen Lin, S. Roy and Ming-Ting Sun, "Wireless video transport using conditional retransmission and low-delay interleaving," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 6, pp. 558-565, June 2002, doi:10.1109/TCSVT.2002.800326.

[i.18] T.K. Sarkar, M.C. Wicks, M. Salazar-Palma, R.J. Bonneau, "A Survey of Various Propagation Models for Mobile Communication," in Smart Antennas, IEEE, 2003, pp. 239-307.

[i.19] Han, Longzhe; Park, Sungjun; Kang, Seung-Seok and In, Hoh, "An Adaptive Cross-Layer FEC Mechanism for Video Transmission over 802.11 WLANs," TIIS, 4, 2010.

[i.20] M. Gast, "802.11 Wireless Networks: The Definitive Guide", Definitive Guide Series, "O'Reilly Media, Inc.", 2005.

[i.21] C. Lin, H. Zhang, C. Shieh and W. Hwang, "Performance analysis of MPEG-4 video stream with FEC error recovery over IEEE 802.11 DCF WLANs," 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), Tsukuba Science City, 2013, pp. 634-640.

[i.22] P. Bucciol, G. Davini, E. Masala, E. Filippi and J. C. De Martin, "Cross-layer perceptual ARQ for H.264 video streaming over 802.11 wireless networks," IEEE Global Telecommunications Conference, 2004. GLOBECOM '04., Dallas, TX, 2004, Vol. 5, pp. 3027-3031, doi:10.1109/GLOCOM.2004.1378908.

[i.23] M. Podolsky, S. McCanne and M. Vetterli, "Soft ARQ for Layered Streaming Media", VLSI Signal Processing. 27, 81-97 (2001), doi:10.1023/A:1008123631453.

[i.24] M. Portoles-Comeras et al, "Modeling and Exploiting the Relation Between Packet Losses and Hidden Traffic," in IEEE Wireless Communications Letters, vol. 2, no. 4, pp. 391-394, August 2013, doi:10.1109/WCL.2013.050613.130159.

[i.25] S. Afzal, V. Testoni, C.E. Rothenberg, P. Kolan and I. Bouazizi, "A Holistic Survey of Wireless Multipath Video Streaming.", arXiv:1906.06184 (2019).

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

DVB-HB Client: A unit capable of connecting to a DVB-HB Local Server and processing and/or rendering content.

NOTE: A DVB-HB Client consumes only one DVB service at a time. Multiple DVB-HB Clients may be hosted on a single DVB-HB Client Device.

DVB-HB Local Server: A unit capable of serving a DVB-HB Client with zero, one or more service lists, and capable of serving DVB-HB Clients with a DVB service from a broadcast it has direct access to.

DVB-I Client: An implementation of the client side of ETSI TS 103 770 [3].

NOTE: This may be integrated into the User Interface (UI) of a device such as a TV set or Set-Top-Box (STB) or part of an app on devices such as mobile phones or tablets.

HTML5: Fifth version of HTML, a markup language used for structuring and presenting content on the World Wide Web.

NOTE: It is a W3C recommendation.

MPEG-2: the ISO/IEC 13818 set of standards

NOTE: Systems coding is defined in part 1, video coding is defined in part 2, and audio coding is defined in part 3 of ISO/IEC 13818.

QR code: A two-dimensional machine-readable optical barcode that contains information about the item to which it is attached.

3.2 Symbols

Void.

3.3 Abbreviations

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

ACFEC	Adaptive Cross-layer Forward Error Correction
AIT	Application Information Table
AL-FEC	Application Layer - Forward Error Correction
AP	Access Point
API	Application Programming Interface
ARP	Address Resolution Protocol
ARQ	Automatic Repeat reQuest
AVC	Advanced Video Coding
BMFF	Base Media File Format
CA	Certificate Authority
CAS	Conditional Access System
CDN	Content Delivery Network
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSV	Comma Separated Values
DASH	Dynamic Adaptive Streaming over HTTP

DDNS	Distributed Domain Name Server
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
DOM	Document Object Model
DSM-CC	Digital Storage Media - Command and Control
DTH	Direct To Home
DVB	Digital Video Broadcasting
DVB-C	DVB Cable Framing and Modulation
DVB-C2	DVB Cable Framing and Modulation, Second Generation
DVB-HB	DVB Home Broadcast
DVB-I	DVB Internet
DVB-IPTV	DVB Internet Protocol TeleVision
DVB-S	DVB Satellite Framing and Modulation
DVB-S2	DVB Satellite Framing and Modulation, Second Generation
DVB-S2X	DVB Satellite Framing and Modulation, Second Generation Extensions
DVB-T	DVB Terrestrial Framing and Modulation
DVB-T2	DVB Terrestrial Framing and Modulation, Second Generation
EIT	Event Information Table
EME	Encrypted Media Extensions
FEC	Forward Error Correction
FQDN	Fully Qualified Domain Name
GOP	Group Of Pictures
HbbTV®	Hybrid Broadcast Broadband TeleVision
HEVC	High Efficiency Video Coding
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IEEE	Institute of Electrical and Electronic Engineers
IP	Internet Protocol
IRD	Integrated Receiver Decoder
ISO	International Organization for Standardization
IT	Information Technology
LAN	Local Area Network
LCN	Logical Channel Numbering
LNB	Low Noise Block
M3U	MPEG audio layer 3 Uniform resource locator
MAC	Media Access Control
MCS	Modulation and Coding Scheme
MPD	Media Presentation Description
MPEG	Moving Pictures Expert Group
MPTS	Multi-Programme Transport Stream
OCR	Optical Character Recognition
OSCP	Online Certificate Status Protocol
OSI	Open Systems Interconnection
PC	Personal Computer
PFR	Playable Frame Rate
PID	Packet IDentifier
PMT	Program Map Table
PSNR	Peak Signal to Noise Ratio
PVR	Personal Video Recorder
QEF	Quasi Error Free
QoE	Quality of Experience
QoS	Quality of Service
RAP	Random Access Point
RF	Radio Frequency
RTCP	RTP Control Protocol
RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
SAT>IP®	SATellite over Internet Protocol
SD	Standard Definition (Video)
SDP	Session Description Protocol
SDT	Service Description Table

SI	Service Information
SPTS	Single Programme Transport Stream
SSDP	Simple Service Description Protocol
STB	Set Top Box
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TS	DVB Transport Stream
TTML	Timed Text Markup Language
TV	TeleVision
UDP	User Datagram Protocol
UI	User Interface
UPnP	Universal Plug and Play
URL	Uniform Resource Locator
URN	Uniform Resource Name
USB	Universal Serial Bus
UUID	Universally Unique IDentifier
W3C	World Wide Web Consortium
Wi-Fi	WIreless FIdelity
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language

4 Concepts and overview

The DVB Home Broadcast (DVB-HB) concept enables client devices to access and consume broadcast content that has been retransmitted, through Internet Protocol (IP) means, by a local server located in the same IP subnetwork such as a Local Area Network (LAN).

Client devices, including those which cannot access linear broadcast services directly (for example because they do not have a tuner, like smartphones and tablets, or because they are only equipped, e.g., with a terrestrial tuner and not satellite), can receive them from a DVB-HB Local Server instead of via the Internet, as those services are already available at full quality at the antenna home plug.

From a commercial point of view, this allows:

- Broadcasters to guarantee the desired picture quality to all users even when they are using an IP-connected device, and at the same time to reduce their Content Delivery Network (CDN) distribution costs;
- Telecom operators to reduce risks of network congestion in case of traffic peaks due to popular live events;
- Users to consume live TeleVision (TV) services on their IP-connected devices with a high and steady picture quality, even in digital-divide areas, where broadband connectivity is not yet optimal;
- Designers of in-building network infrastructures (e.g., for hospitality, campus, etc.) to easily integrate TV distribution with other Information Technology (IT) services.

Two general functions can be identified for a typical DVB-HB Local Server:

- Announcement on the LAN and exposure of device capabilities, including signalling of available services,
- Tuning to a Radio Frequency (RF) signal upon request of a DVB-HB Client and subsequent redistribution of the content via IP network, possibly after audio/video transcoding to match the capabilities of the device hosting the DVB-HB Client.

Four general functions can be identified for a typical DVB-HB Client:

- Presentation to an end-user of a catalogue of the available services,
- Signalling exchanges with a DVB-HB Local Server (possibly with the aid of an external repository),
- Reception of content from a DVB-HB Local Server,
- Display and/or processing of the content received.

In order to address all significant deployment scenarios, the present document includes two Profiles, named *Profile A* and *Profile B*:

Profile A

Based on the SAT>IP specification [1], extending it with a number of additional optional features, foreseeing redistribution of the selected TV services as DVB Transport Stream (TS) over IP.

Profile B

Targeting compatibility with DVB-I Clients, foreseeing redistribution of the selected TV services as Moving Pictures Expert Group (MPEG) Dynamic Adaptive Streaming over HTTP (DASH) according to ETSI TS 103 285 [2] together with service discovery metadata according to ETSI TS 103 770 [3]. Profile B also supports browser-based client applications.

5 DVB-HB Reference architecture

5.1 Introduction

The relationships between the logical functions in the reference architecture are identified by named reference points.

In a practical deployment, each of these is realised by a concrete interface and conveys information between the relevant functions using a specific protocol.

5.2 Reference architecture diagram

Figure 1 summarises the simplified DVB-HB reference architecture, showing the reference points and the logical functions. Data plane interactions are depicted using solid lines. Control plane interactions are depicted using dotted lines. Interactions that lie within the scope of the present document are depicted as black lines with a reference point name. Those beyond the scope of the present document (but relevant to the functional architecture and described by means of informative guidelines) are shown with grey lines.

The architecture is then illustrated with more details in figure 2 and figure 3 for Profiles A and B respectively. In these figures, logical functions are depicted as named boxes and these may be nested in cases where a high-level function is composed of several subfunctions. Optional functions are represented with dotted blocks.

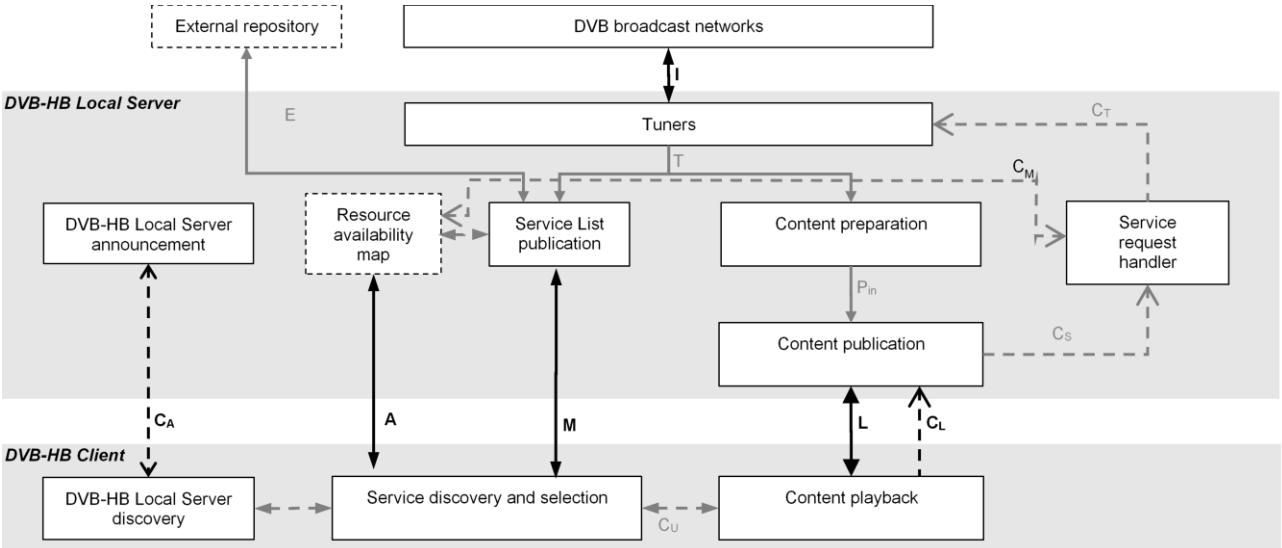

Figure 1: Simplified reference architecture (Profiles A and B)

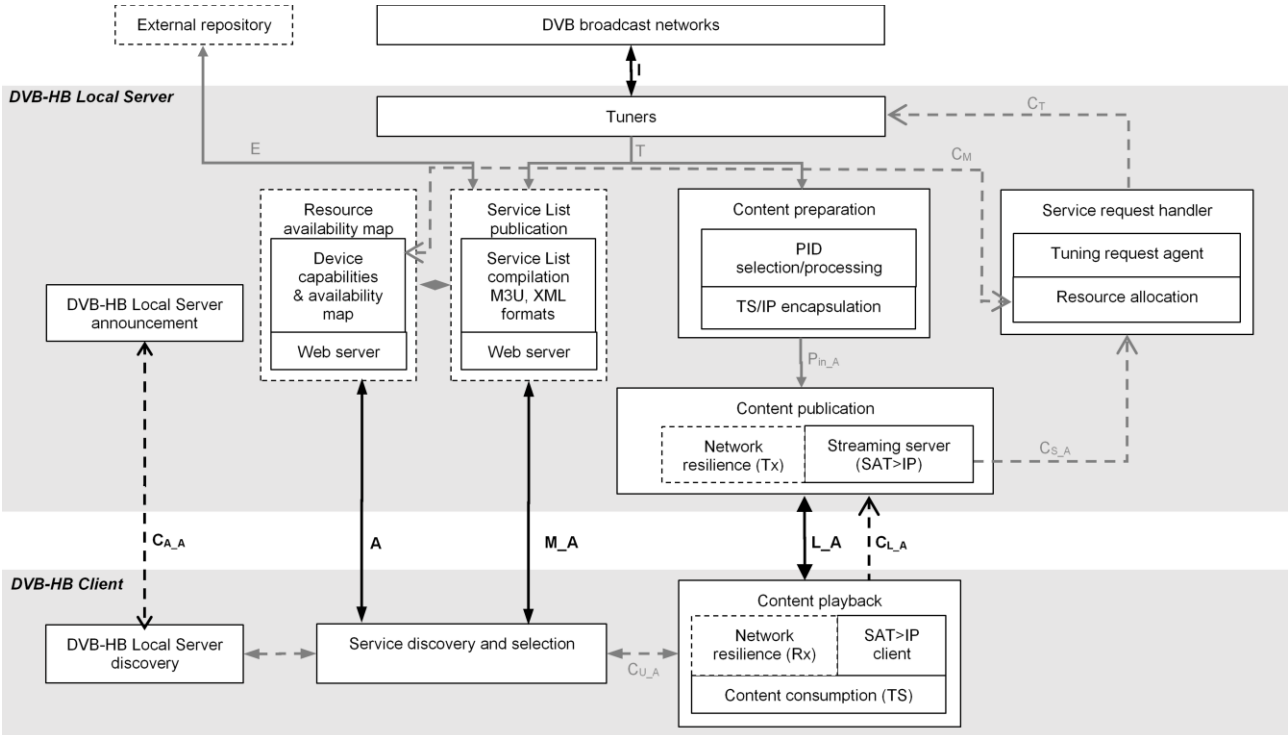

Figure 2: Reference architecture for Profile A

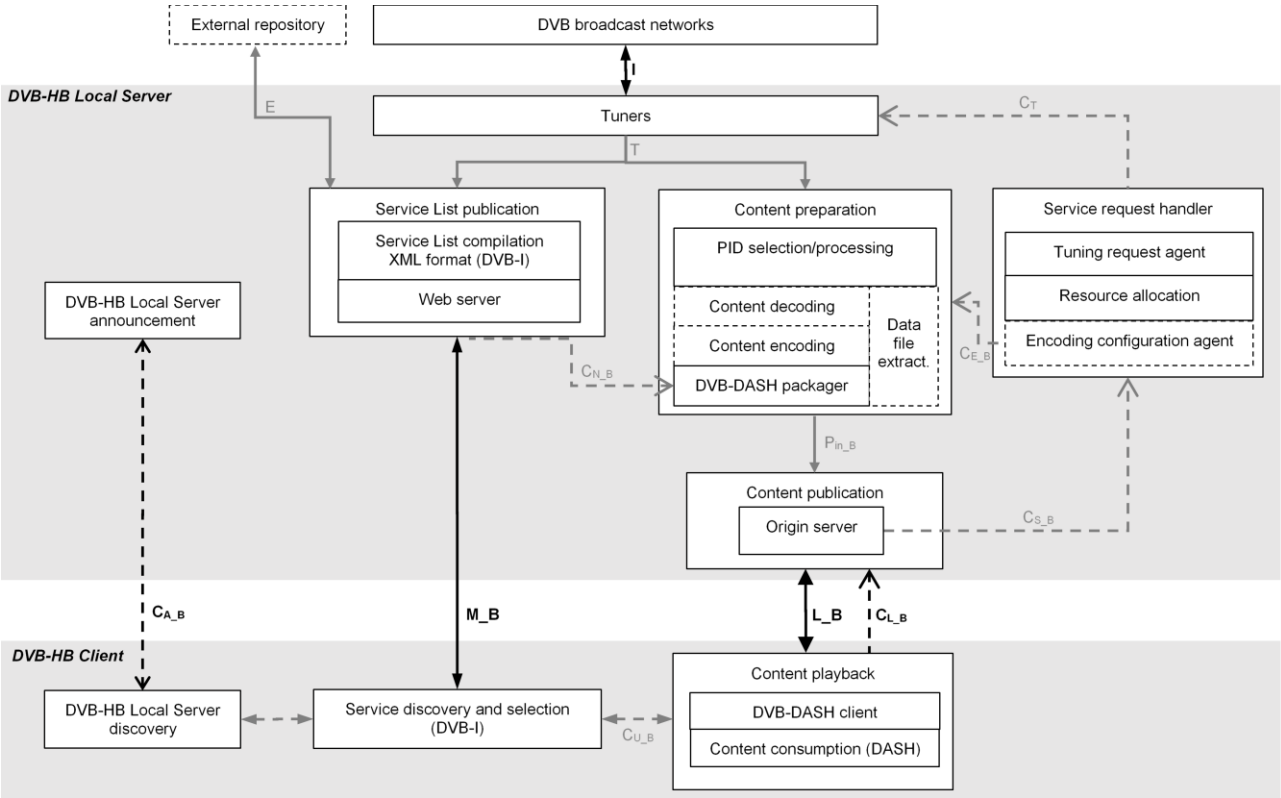

Figure 3: Reference architecture for Profile B

5.3 Reference points

5.3.1 Data plane reference points

5.3.1.1 Introduction

The reference points defined in this clause are used primarily to transport content and service list metadata.

5.3.1.2 Generic data plane reference points (Profiles A and B)

I Input RF interface between Digital Video Broadcasting (DVB) broadcast networks and the Tuners function of a DVB-HB Local Server. Input signals can be any of the following:

- DVB Satellite Framing and Modulation (DVB-S) signals as defined in ETSI EN 300 421 [4],
- DVB Satellite Framing and Modulation, Second Generation (DVB-S2) signals as defined in ETSI EN 302 307-1 [5],
- DVB Satellite Framing and Modulation, Second Generation Extensions (DVB-S2X) signals as defined in ETSI EN 302 307-2 [6],
- DVB Terrestrial Framing and Modulation (DVB-T) signals as defined in ETSI EN 300 744 [7],
- DVB Terrestrial Framing and Modulation, Second Generation (DVB-T2) signals as defined in ETSI EN 302 755 [8],
- DVB Cable Framing and Modulation (DVB-C) signals as defined in ETSI EN 300 429 [9],
- DVB Cable Framing and Modulation, Second Generation (DVB-C2) signals as defined in ETSI EN 302 769 [10].

Carriage of A/V content and related information in a TS is as specified in ETSI TS 101 154 [11], clause 4.

L Interaction between the *Content Playback* function of a DVB-HB Client and the *Content publication* function of a DVB-HB Local Server.

M Interaction between the *Service discovery and selection* function of a DVB-HB Client and the *Service List publication* function of a DVB-HB Local Server. This interface includes the fetching of Service List(s) and Content Guide.

NOTE: The *Service List publication* function may be hosted on a remote server published on the web (how the service list is produced in this case is out of scope of the present document).

A HyperText Transfer Protocol (HTTP) or HyperText Transfer Protocol Secure (HTTPS) interaction between the *Service discovery and selection* function of a DVB-HB Client and the *Resource availability map* function of a DVB-HB Local Server.

T Distribution of the TS, as received by the *Tuners* function, to *Service List publication* and *Content Preparation* functions in a DVB-HB Local Server.

P_{in} Provision of content to a *Content publication* function by a *Content preparation* function within a DVB-HB Local Server. This may be implemented as a push interface, or content may be pulled on demand from a Content packaging function (out of scope of the present document).

E External provision of service list metadata (out of scope of the present document, it may be a proprietary interface).

5.3.1.3 Specific data plane reference points for Profile A

I See clause 5.3.1.2.

L_A Delivery of content via Real Time Streaming Protocol (RTSP)/Real Time Protocol (RTP) or HTTP interaction between the *Content publication* function of a Profile A DVB-HB Local Server and the *Content Playback* function of a Profile A DVB-HB Client, according to the SAT>IP specification [1], and additional optional Application Layer - Forward Error Correction (AL-FEC) packets as defined in clause 8.4.2.

NOTE: **C_{L_A}** and **L_A** reference points physically correspond to requests and responses of the RTSP or HTTP communication between *Streaming Server (SAT>IP)* and *SAT>IP client* subfunctions.

M_A HTTP(S) interaction between the *Service discovery and selection* function of a Profile A DVB-HB Client and the *Service List publication* function of a Profile A DVB-HB Local Server. This interface includes the fetching of the service list(s) in MPEG audio layer 3 Uniform resource locator (M3U) or eXtensible Markup Language (XML) format.

A HTTP(S) interaction between the *Service discovery and selection* function of a DVB-HB Client and the *Resource availability map* function of a DVB-HB Local Server.

T See clause 5.3.1.2.

P_{in_A} Provision of content in the form of TS packets over IP to a *Content publication* function by a *Content preparation* function in a Profile A DVB-HB Local Server (out of scope of the present document).

E See clause 5.3.1.2.

5.3.1.4 Specific data plane reference points for Profile B

I See clause 5.3.1.2.

L_B HTTP(S) interaction between the *Origin server* subfunction of a Profile B DVB-HB Local Server and the *DVB-DASH client* subfunctions of a Profile B DVB-HB Client for content delivery, and eventually the *Data file extraction* subfunction in case of additional data files (e.g., an interactive application).

NOTE 1: The **L_B** reference point aggregates interfaces **D2**, **E1** and **E2** defined in ETSI TS 103 770 [3], i.e., all interaction between *Origin server* and *DVB-DASH client* subfunctions except request for DASH Media Presentation Description (MPD): the latter, while being carried over the same physical interface, is represented by the $\mathbf{C_{L\_B}}$ reference point.

M_B HTTP(S) interaction between the *Service discovery and selection*) function of a Profile B DVB-HB Client and the *Service List publication* function of a Profile B DVB-HB Local Server. This interface includes the fetching of the service list(s) in XML format according to ETSI TS 103 770 [3].

NOTE 2: The **M_B** reference point aggregates interfaces **A1**, **A2**, **B1** and **B2** defined in ETSI TS 103 770 [3].

A HTTP(S) interaction between the *Service discovery and selection* function of a DVB-HB Client and the *Resource availability map* function of a DVB-HB Local Server.

T See clause 5.3.1.2.

$\mathbf{P_{in\_B}}$ Provision of content in the form of files (i.e., DASH segments, eventually data files) to a *Content publication* function by a *Content preparation* function in a Profile B DVB-HB Local Server (out of scope of the present document).

E See clause 5.3.1.2.

5.3.2 Control plane reference points

5.3.2.1 Introduction

The reference points defined in this clause are used for control signalling.

5.3.2.2 Generic control plane reference points (Profiles A and B)

$\mathbf{C_L}$ Control interface for request of a specific service by a DVB-HB Client.

$\mathbf{C_A}$ Control interface for discovery and capability exposure of DVB-HB Local Servers.

$\mathbf{C_S}$ Control interface for forwarding service requests to the *Service Request Handler*) function (out of scope of the present document).

$\mathbf{C_T}$ Control interface for commands to the *Tuners* function (out of scope of the present document).

$\mathbf{C_U}$ Control interface to control content playback in a DVB-HB Client according to the selected service. It is out of scope of the present document, but would generally involve passing a Uniform Resource Locator (URL) to initiate playback.

$\mathbf{C_M}$ Control interface between the *Resource allocation subfunction* of the *Service request handler* function and the *Resource availability map* function, for keeping the availability map up-to-date (out of scope of the present document).

5.3.2.3 Specific control plane reference points for Profile A

$\mathbf{C_{L\_A}}$ Control interface for request of a specific service by a Profile A DVB-HB Client, based on RTSP or HTTP requests according to the SAT>IP specification [1].

NOTE: $\mathbf{C_{L\_A}}$ and **L_A** reference points physically correspond to requests and responses of the RTSP or HTTP communication between *Streaming Server (SAT>IP)* and *SAT>IP client* subfunctions.

$\mathbf{C_{A\_A}}$ Control interface for discovery and capability exposure of Profile A DVB-HB Local Servers according to the SAT>IP specification [1].

$\mathbf{C_{S\_A}}$ Control interface for commands to the *Tuners* function in case of Profile A (out of scope of the present document).

$\mathbf{C_T}$ See clause 5.3.2.2.

C_{U_A} Control interface to control content playback in a Profile A DVB-HB Client according to the selected service. It is out of scope of the present document, but would generally involve passing a URL to initiate playback.

C_M See clause 5.3.2.2.

5.3.2.4 Specific control plane reference points for Profile B

C_{L_B} Control interface for request of a specific service by a Profile B DVB-HB Client, based on HTTP(S) requests of MPD files according to ETSI TS 103 285 [2].

NOTE: The **C_{L_B}** reference point corresponds to interface **D1** defined in ETSI TS 103 770 [3].

C_{A_B} Control interface for discovery and exposure of capabilities of Profile B DVB-HB Local Servers (see also clause 6 and clause 7.2).

C_{S_B} Control interface for commands to the *Tuners* function in case of Profile B (out of scope of the present document).

C_T See clause 5.3.2.2.

C_{N_B} Control interface used by the *Service List publication* function to communicate file names to be adopted by the *DVB-DASH packager* subfunction, as defined in the Service List, for the generation of the requested service according to ETSI TS 103 285 [2].

C_{E_B} Control interface used by the *Encoding configuration agent* subfunction to communicate the encoding parameters, as defined in the Service List for the specific selected service, to the *Content encoding* subfunction.

C_{U_B} Control interface to control content playback in a Profile B DVB-HB Client according to the selected service. It is out of scope of the present document, but would generally involve passing a URL to initiate playback.

C_M See clause 5.3.2.2.

5.4 Functions

5.4.1 DVB broadcast networks

This is an external function, out of scope of the present document. It represents deployed on-the-air DVB satellite and/or terrestrial and/or cable broadcast networks, delivering audio/video services to consumer devices (e.g., TVs/STBs).

5.4.2 Tuners

Hardware modules receiving and demodulating on-the-air DVB-S/DVB-S2/DVB-S2X/DVB-T/DVB-T2/DVB-C/DVB-C2 signals, providing in output the relevant TS. Multiple modules may be available, each of them producing a TS.

5.4.3 Service List publication

5.4.3.1 General description

The *Service List publication* function generates and maintains the Service List and the associated Content Guide describing the services offered by a DVB-HB Local Server, and publishes it on the LAN for DVB-HB Clients requesting it.

The function includes a *Service List compilation* subfunction generating Service List(s) according to the supported formats and a *Web server* subfunction for publication (HTTP(S) server). It may generate the Service List and associated Content Guide locally on the basis of the DVB-Service Information (SI) metadata received over the **T** reference point, or it may rely on metadata retrieved from an *External repository* (how the service list is produced in this second case is out of scope of the present document).

5.4.3.2 Service List publication for Profile A

In case of Profile A, *Service List publication* is an optional function, i.e., DVB-HB Clients can use other means for service discovery, e.g., frequency scan via remote tuning commands.

The *Service List compilation* subfunction should generate service lists according to M3U format and/or according to XML format as defined in ETSI TS 103 770 [3], so that the features associated with metadata of the service list published in this format are made available to the DVB-HB Clients (e.g., Logical Channel Numbering (LCN), regionalisation, multiple instances, etc.).

NOTE: Other service list formats are not prevented by the SAT>IP specification [1].

5.4.3.3 Service List publication for Profile B

In case of Profile B, the *Service List compilation* subfunction generates Service List(s) and associated Content Guide according to the XML formats as defined in ETSI TS 103 770 [3]. Eventually, it may also generate additional signalling files, e.g., an XML-encoded Application Information Table (AIT) associated to an interactive application (see also annex D).

Additionally, it provides file names over the $\mathbf{C_{N\_B}}$ reference point, to be adopted by the *DVB-DASH packager* subfunction, as defined in the Service List, for the generation of the requested service in DVB-DASH format by the DVB-HB Local Server.

5.4.4 Content preparation

5.4.4.1 General description

The *Content preparation* function has the task of producing the selected audio/video content for publication on the LAN, on the basis of the TS demodulated by the *Tuners* function.

It includes a *PID selection/processing* subfunction, with the task of selecting the Packet IDentifier (PID)s associated with the desired service and apply proper manipulations as needed, and additional profile-specific subfunctions, as described in clause 5.4.4.2 and clause 5.4.4.3.

Redistribution of services protected by a Conditional Access System (CAS), including for example decryption and re-protection, is out of scope of the present document, but is not prevented.

5.4.4.2 Content preparation for Profile A

In case of Profile A, in addition to the *PID selection/processing* subfunction described in clause 5.4.4.1, a *TS/IP encapsulation* subfunction has the task of encapsulating TS packets in RTP packets carried in User Datagram Protocol (UDP) packets carried in IP datagrams (i.e., TS/RTP/UDP/IP) as defined in the SAT>IP specification [1].

5.4.4.3 Content preparation for Profile B

In case of Profile B, in addition to the *PID selection/processing* subfunction described in clause 5.4.4.1, the following subfunctions exist:

- *Content decoding* subfunction, with the task of decoding the A/V content (and associated data) of the selected service.
- *Content encoding* subfunction, with the task of transforming the source media streams produced by the *Content decoding* subfunction into encoded media at the desired bit-rate. A single source media stream may be transformed into a number of different encoded representations to match delivery conditions or DVB-HB Client device capabilities. Virtual segment boundary markers may be placed in the encoded media representation to assist an adaptive *Content playback* function in its operation. The output of the *Content encoding* subfunction is a stream formatted so as to be suitable for ingest by the *DVB-DASH packager* subfunction.
- *DVB-DASH packager* subfunction, with the task of ingesting the media streams of one (or more) encoded representations and formatting each one according to the International Organization for Standardization (ISO) Base Media File Format (BMFF) packaging format as defined in ETSI TS 103 285 [2]. In the context of dynamic adaptive streaming, the output of the packager is a sequence of packaged media segments with

representation switching points that are aligned across different representations of the same source media stream.

- *Data file extraction* subfunction, with the task of decoding data files carried over the TS according to a specific format (e.g., an interactive application transported as a Digital Storage Media - Command and Control (DSM-CC) carousel in the incoming TS).

- The *Content preparation* function should also include extraction of subtitles from the incoming TS and their conversion to a DASH-compatible format according to ETSI TS 103 285 [2], clause 7. If the incoming subtitles are encoded as DVB Timed Text Markup Language (TTML) according to ETSI EN 303 560 [i.1], the translation is a simple repackaging. For other forms of subtitles a transcoding is required. DVB bitmap subtitles require Optical Character Recognition (OCR) support for decoding. Teletext subtitles require text remapping and translation of position and colour information.

NOTE: Broadcasters wishing to support Profile B DVB-HB Clients may consider adding a DVB TTML component to services.

5.4.5 Content publication

5.4.5.1 General description

The *Content publication* function has the following tasks:

- Listening to requests from DVB-HB Clients over the $\mathbf{C_L}$ reference point and triggering tuning commands by forwarding such requests in a proper format to the *Service request handler* function over the $\mathbf{C_S}$ reference point;

- Making the content available for delivery to the *Content playback* function over the **L** reference point.

5.4.5.2 Content publication for Profile A

In case of Profile A, the *Streaming server (SAT>IP)* subfunction interacts with the *Content playback* function by receiving requests over the $\mathbf{C_{L\_A}}$ reference point, triggering tuning commands and delivering the requested content over the **L_A** reference point according to the protocols defined in the SAT>IP specification [1].

Optionally, additional AL-FEC packets associated with the data packets are sent by the *Network resilience (Tx)* subfunction over the **L_A** reference point for error protection, as defined in clause 8.4.2.

5.4.5.3 Content publication for Profile B

In case of Profile B, the *Content Publication* function consists of an *Origin Server* subfunction (HTTP(S) server), which interacts with the *Content playback* function by receiving requests over the $\mathbf{C_{L\_B}}$ reference point, triggering tuning commands and delivering the requested content over the **L_B** reference point according to ETSI TS 103 285 [2].

5.4.6 Service request handler

The *Service request handler* function has the task of converting DVB-HB Client requests, as intercepted by the *Content publication* function and forwarded over the $\mathbf{C_S}$ reference point, to tuning commands sent to the *Tuners* function over the $\mathbf{C_T}$ reference point.

It consists of a *Resource allocation* subfunction, responsible of optimising the available resources of a DVB-HB Local Server (e.g., tuners, Central Processing Unit (CPU), etc.) according to the overall requests simultaneously managed by the device at each point of time, and a *Tuning request agent* subfunction, responsible of sending the actual tuning commands to the *Tuners* function.

Additionally, in case of Profile B, it also consists of an *Encoding configuration agent* subfunction, which may provide the encoding parameters, as defined in the Service List for the specific selected service, to the *Content encoding* subfunction over the $\mathbf{C_{E\_B}}$ reference point.

5.4.7 DVB-HB Local Server announcement

The *DVB-HB Local Server announcement* function has the task of providing advertisements of the presence of a DVB-HB Local Server on the LAN and providing description of the DVB-HB Local Server capabilities (e.g., number and type of tuners, etc.).

5.4.8 DVB-HB Local Server discovery

The *DVB-HB Local Server discovery* function is a function of a DVB-HB Client with the task of searching on the LAN for available DVB-HB Local Servers and retrieving their capabilities. See also clause 6 for protocols used.

5.4.9 Service discovery and selection

The *Service discovery and selection* function is an application of a DVB-HB Client with the task of compiling a local copy of the Service List, collecting Content Guide metadata and controlling the *Content playback* function.

In case of Profile A, Service List(s) and Content Guide data may be retrieved over the **M** reference point or, in alternative, may be compiled by the *Service discovery and selection* function by means of a frequency scan, performed via remote tuning commands generated by the *Content playback* function according to the indications sent over the $\mathbf{C_{U\_A}}$ reference point.

In case of Profile B, Service List(s) and Content Guide data are retrieved over the **M** reference point according to ETSI TS 103 770 [3].

5.4.10 Resource availability map function

The *Resource availability map* function is an optional function with the task of exposing the static capabilities of the device (e.g., number and type of tuners) and the instantaneous resources availability in order to efficiently allow sharing the resources among multiple DVB-HB Clients.

The function includes a *Device capabilities & availability map* subfunction generating the data to be exposed as XML files and a *Web server* subfunction for publication (HTTP(S) server).

To achieve its tasks, the *Resource availability map* function relies on information generated by the *Service List publication* function and its subfunctions.

5.4.11 Content playback

5.4.11.1 General description

The *Content playback* function is the entity in the DVB-HB Client managing request, reception and presentation of content.

5.4.11.2 Content playback function for Profile A

In case of Profile A, request and reception of the selected service are managed by the *SAT>IP client* subfunction. It interacts with the *Content publication* function of the DVB-HB Local Server by sending requests over the $\mathbf{C_{L\_A}}$ reference point and retrieving the requested content over the **L_A** reference point according to the protocols defined in the SAT>IP specification [1].

Optionally, additional AL-FEC packets associated with the data packets are received over the **L_A** reference point by the *Network resilience (Rx)* subfunction for error correction, as defined in clause 8.4.2

The *Content consumption (TS)* subfunction manages presentation of content in TS format for consumption by the user. As browser-based (HTML5+Javascript) players do not support HTTP-based video streaming, a dedicated application is used.

5.4.11.3 Content playback function for Profile B

In case of Profile B, request and reception of the selected service are managed by the *DVB-DASH client* subfunction. It sends HTTP(S) requests to the *Origin server* subfunction of the DVB-HB Local Server over the $\mathbf{C_{L\_B}}$ reference point, retrieving the requested content over the **L_B** reference point as defined in ETSI TS 103 285 [2].

The *Content consumption (DASH)* subfunction manages presentation of content formatted as defined in ETSI TS 103 285 [2] for consumption by the user. The audio/video player may be implemented in a browser (HTML5+Javascript) or as a dedicated application.

5.4.12 External repository

This is an external function, out of scope of the present document. It represents an external repository (e.g., a remote server, a Universal Serial Bus (USB) memory stick, etc.) of service list metadata, accessed by the *Service List publication* function via the proprietary reference point **E** in alternative to generating them locally on the basis of the DVB-SI metadata received over the **T** reference point. How the service list is produced in this case is out of scope of the present document.

6 DVB-HB Local Server announcement and discovery

6.1 Introduction

In Profile A, the *DVB-HB Local Server Announcement* and *DVB-HB Local Server discovery* functions shall be according to the SAT>IP specification [1], i.e., they are implemented using a simplified version of Universal Plug and Play (UPnP) according to ISO/IEC 29341-1-1 [13] to ensure backwards compatibility with existing devices supporting the SAT>IP specification [1].

In Profile B, instead, since it aims to support also browser-based DVB-HB Clients, such method is not applicable in all the use-cases. A discovery mechanism based on UPnP, in fact, relies on the Simple Service Description Protocol (SSDP), which is based on UDP multicast packet transmission, a type of transmission that cannot be supported natively by HTML5/Javascript clients. It is recognized that this issue does not affect all the implementation scenarios that can be foreseen for Profile B DVB-HB Clients. For instance, an exemplary (non-exhaustive) list is given below:

- DVB-HB Client App installed on the user terminal.

  In this case, not being browser-based, there are no specific constraints on the mechanism adopted.

- Browser-based standard DVB-I Client hosted on a well-known web site (e.g., the DVB Internet (DVB-I) reference client).

  In this case automatic discovery is not applicable, and the IP address of the service list exposed by DVB-HB Local Server has to be provided manually.

- Browser-based DVB-HB Client hosted by the DVB-HB Local Server itself.

  In this case, for the first time at least, user has to type the DVB-HB Local Server's address in the browser, so automatic discovery is not applicable. However, the DVB-HB Client may automatically discover additional DVB-HB Local Servers present on the same LAN .

- Browser-based DVB-HB Client hosted on a well-known web site.

  In this case automatic discovery is applicable, however a solution based on UPnP cannot be used.

In order to cover all of the possible alternatives, for Profile B DVB-HB Local Server discovery few optional discovery mechanisms are outlined in this clause, as well as other informative implementation-specific guidelines.

6.2 Profile A DVB-HB Local Server discovery

Profile A DVB-HB Local Server announcement and discovery shall be implemented according to the SAT>IP specification [1]. The mechanism relies on the SSDP as specified in ISO/IEC 29341-1-1 [13].

6.3 Profile B DVB-HB Local Server discovery

6.3.1 Introduction

A Profile B DVB-HB Local Server may implement one or more of the following discovery mechanisms. Different implementations of Profile B DVB-HB Clients may take advantage of one or the other of such mechanisms. Manual input of the DVB-HB Local Server's IP address within the client can always be implemented as a fallback solution.

6.3.2 Optional discovery mechanism based on UPnP

When targeting clients capable of UPnP-based discovery, a Profile B DVB-HB Local Server implementing this method shall announce its presence on the LAN according to the SAT>IP specification [1], with the differences indicated below. The mechanism relies on the SSDP as specified in ISO/IEC 29341-1-1 [13].

With respect to the SAT>IP specification [1], the following differences apply:

- The device type, or Uniform Resource Name (URN), shall be set to: `urn:dvb:metadata:device:HBLocalServer:1`
- The `LOCATION` header shall indicate the URL of the Service List Entry Point exposed by the DVB-HB Local Server (see also clause 7).

6.3.3 Optional discovery mechanism based on HTTP

When targeting clients capable of port-scanning a selected subnet, a Profile B DVB-HB Local Server implementing this method shall announce its presence on the LAN by exposing its Service List Entry Point on Transmission Control Protocol (TCP) port 61277 (HTTP).

During the discovery phase, DVB-HB Clients may scan the local subnet, looking for DVB-HB Local Servers active on that port.

NOTE: The current discovery mechanism requires a-priori knowledge of the local subnet addressing details.

6.3.4 Additional implementation-specific mechanisms

In addition to the optional discovery mechanisms specified above, other implementation-specific mechanisms supporting browser-based clients may be used. In this informative clause, two specific mechanisms are considered since they are closely related to similar solutions adopted in the present document or in other DVB specifications.

User login via remote server

When a remote server is available for client's authentication, this mechanism uses it as a middle node to connect a browser-based DVB-HB Client to a DVB-HB Local Server using the device unique identifier and user credentials associated with it. After logging into the remote server with the associated credentials, the client retrieves the relevant discovery information related to the user's DVB-HB Local Server(s), allowing it to retrieve the DVB-HB Local Server's local IP address.

NOTE 1: This procedure may be done during the first installation, in background or each time the user starts the DVB-HB Client application, depending on the implementation.

Device pairing

When the DVB-HB Local Server has a display (e.g., it is connected to a monitor or TV screen) and the DVB-HB Client runs on a separate personal device (e.g., Personal Computer (PC), laptop, tablet, smartphone, etc.), the two devices can be paired using a selected *"rendezvous"* mechanism (e.g., alphanumeric code, QR code, URL, etc.). Once paired, the DVB-HB Local Server can send its IP address and capabilities to the DVB-HB Client, which can then start sending back the requests.

NOTE 2: This procedure requires a remote server for the *"rendezvous"* mechanism.

NOTE 3: The DVB-HB Local Server and DVB-HB Client may be paired during either the initial setup or at the beginning of a viewing session, depending on the implementation.

7 Exposure of DVB-HB Local Server capabilities

7.1 Introduction

This clause defines how Profile A and Profile B DVB-HB Local Servers can announce their capabilities to DVB-HB Clients, including number and type of available tuners and encoding resources, together with any restrictions because of concurrent usage by other connected DVB-HB Clients.

Clause 7.2 and clause 7.3 provide the detailed set of device capabilities and available resources announced by the DVB-HB Local Server.

A Profile A DVB-HB Local Server shall expose basic information about itself in the form of an XML file (e.g., desc.xml) as defined in the SAT>IP specification [1] and extended as defined in clause 8.3 of the present document. Such file is announced during the DVB-HB Local Server discovery phase (see clause 6).

A Profile B DVB-HB Local Server shall expose basic information about itself in the form of an extended Service List Entry Point XML file as defined in ETSI TS 103 770 [3], clause 5.3.2 and extended as defined in clause 9.2 of the present document. Such file is announced during the DVB-HB Local Server discovery phase (see clause 6).

In both Profiles, the exposed information includes:

- URL of the published DVB-I Service List (optional in Profile A, mandatory in Profile B),
- Device description (i.e., brand, model, device identifier, supported version, etc.),
- Device capabilities and resource availability map (optional).

7.2 Device capabilities

Capabilities of a DVB-HB Local Server are defined according to pseudocode 1, table 1 and table 2.

`HBLocalServerType` replicates the same structure as defined by the SAT>IP specification [1], aligned to UPnP syntax according to ISO/IEC 29341-1-1 [13], also for Profile B DVB-HB Local Servers.

`HBServiceAvailabilityType` is used by both Profile A and Profile B DVB-HB Local Servers to announce their currently available resources. See also clause 8.3 and clause 9.2, respectively, for their use.

Pseudocode 1: DVB-HB Local Server device capabilities schema

```
<complexType name="HBLocalServerType">
  <sequence>
    <element name="DeviceType" type="anyURI"/>
    <element name="UniqueDeviceName">
      <simpleType>
        <restriction base="string">
          <pattern value="uuid:[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}"/>
        </restriction>
      </simpleType>
    </element>
    <element name="UniversalProductCode" minOccurs="0">
      <simpleType>
        <restriction base="string">
          <pattern value="[0-9]{12}"/>
        </restriction>
      </simpleType>
    </element>
    <element name="ModelName" type="mpeg7:TextualType" />
    <element name="FriendlyName" type="mpeg7:TextualType" maxOccurs="unbounded" />
    <element name="Manufacturer" type="mpeg7:TextualType" maxOccurs="unbounded" />
    <element name="ModelNumber" type="mpeg7:TextualType" minOccurs="0" />
    <element name="SerialNumber" type="mpeg7:TextualType" minOccurs="0" />
    <element name="ModelDescription" type="mpeg7:TextualType" minOccurs="0" maxOccurs="unbounded" />
    <element name="ManufacturerURL" type="anyURI" minOccurs="0" />
    <element name="ModelURL" type="anyURI" minOccurs="0" />
    <element name="Icon" type="tva:RelatedMaterialType" minOccurs="0" maxOccurs="unbounded"/>
    <element name="Availability" type="dvbhbx:HBServiceAvailabilityType" minOccurs="0" />
  </sequence>
  <attribute name="specVersion" type="positiveInteger" use="required"/>
</complexType>

<complexType name="HBServiceAvailabilityType">
  <sequence>
    <element name="OrbitalPosition" type="dvbisd:LongitudeType" minOccurs="0" maxOccurs="unbounded"
/>
    <element name="ServiceAvailabilityMapIdleURL" type="dvbisd:ExtendedURIType" />
    <element name="ServiceAvailabilityMapUpdateURL" type="dvbisd:ExtendedURIType" minOccurs="0" />
  </sequence>
  <attribute name="version" type="positiveInteger" use="required"/>
</complexType>
```

Table 1: HBLocalServerType fields

Name	Semantic Definition	Constraints
`DeviceType`	Indicates the category of device, aligned to UPnP syntax. For a Profile B DVB-HB Local Server, it shall be set to `urn:dvb:metadata:device:HBLocalServer:1`.	Mandatory
`UniqueDeviceName`	Universally-unique identifier for the device, whether root or embedded. Must be the same over time for a specific device instance (i.e., must survive reboots). It shall be set to “uuid:” followed by a Universally Unique IDentifier (UUID) suffix specified by the vendor (the UUID string shall have the format specified in ISO/IEC 29341-1-1 [13]).	Mandatory
`UniversalProductCode`	12-digit, all-numeric code that identifies the consumer package. Managed by the Uniform Code Council. Specified by UPnP vendor.	Optional 0 .. 1
`ModelName`	Model name. Should be < 32 characters.	Mandatory
`FriendlyName`	Short description for end user. Multiple values can be specified as long as they have different `@xml:lang` values. Should be < 64 characters.	Mandatory 1 .. ∞
`Manufacturer`	Manufacturer's name. Multiple values can be specified as long as they have different `@xml:lang` values. Should be < 64 characters.	Mandatory 1 .. ∞
`ModelNumber`	Model number. Should be < 32 characters.	Optional
`SerialNumber`	Serial number of the device. Should be < 32 characters.	Optional
`ModelDescription`	Long description for end user. Multiple values can be specified as long as they have different `@xml:lang` values. Should be < 128 characters.	Mandatory 0 .. ∞
`ManufacturerURL`	Web site for manufacturer	Optional
`ModelURL`	Web site for the model	Optional
`Icon`	Icon to depict device in a control point UI	Optional 0 .. ∞
`Availability`	Announces the capabilities of the DVB-HB Local Server in terms of type and number of front-ends (tuners). It also informs about real-time status in terms of used and available resources. See table 2 for semantic definition of `HBServiceAvailabilityType`.	Optional 0 .. 1
`@specVersion`	Defines the version on which the device is implemented	Mandatory

Table 2: HBServiceAvailabilityType fields

Name	Semantic Definition	Constraints
`OrbitalPosition`	Received orbital position(s) expressed in positive or negative degrees representing east and west directions respectively. Multiple occurrences are possible if multiple orbital positions are received. Relevant only if the DVB-HB Local Server also includes satellite tuners.	Optional
`ServiceAvailabilityMapIdleURL`	URL of the ServiceAvailabilityMapIdle.xml file, which provides information on installed tuners and available services. Syntax of the ServiceAvailabilityMap.xml file is given in clause 7.3.	Mandatory
`ServiceAvailabilityMapUpdateURL`	URL of the ServiceAvailabilityMapUpdate.xml file, which is used to update currently used resources in the `ServiceAvailabilityMap` with respect to `ServiceAvailabilityMapIdle` according to IETF RFC 5261 [14] (see also clause 7.3). Present if the DVB-HB Local Server supports resource sharing functionality.	Optional 0 .. 1
`@version`	Version of the `ServiceAvailabilityMap` exposed at URL `ServiceAvailabilityMapIdleURL`. The value shall be identical as inside the XML document and the same as in the DVB-I Service List. The version is incremented each time the files are updated, for example after an installation update or when the exposed DVB-I Service List is updated. DVB-HB Clients supporting resource sharing functionality can check the `@version` attribute and, if different from the previous one, download the new version of ServiceAvailabilityMapIdle.xml and ServiceAvailabilityMapUpdate.xml files.	Mandatory

7.3 Service Availability Map

7.3.1 Introduction

The effective availability of a service for a DVB-HB Client can be impacted by several limitations, including:

- Unavailability of a frontend (tuner),
- Unavailability of a sub-band, or polarization and sub-band, in case of Direct To Home (DTH) reception,
- Processing limitations of a DVB-HB Local Server,
- Bandwidth limitations of a DVB-HB Local Server,
- Number of encoders (if transcoding is used).

The capability of sharing DVB-HB Local Server resources among multiple DVB-HB Clients is an efficient way to increase the effective availability of services.

Obviously, checking service availability is only relevant when there is lack of resources. In other words, if resources are overprovisioned, i.e., the available resources always allow all clients to be served with the requested services, then there is no need to efficiently manage resources or to analyse the actual usage by clients, as all services can simply offered at all times.

The problem of sharing tuning resources is not new and is well known, for example in the context of STBs sharing a common Low Noise Block (LNB) feed (i.e., *daisy chained* LNB) or Personal Video Recorder (PVR) functionalities (i.e., watching one service while recording another one on the same or another multiplex). An additional challenge is the availability of transcoders. Moreover, use of resources by other clients is not known, because they are independent devices. For privacy reasons, a reporting of individual usage is not envisaged, but, fortunately, for assessing available services, only own usage (i.e., which resources would be freed before selecting a service) and a snapshot of what is used globally at server side is relevant – not the individual clients' usage.

Since Profile A DVB-HB Clients request a specific encoding profile linked with the client request, it is assumed that, when a transcoded service is requested, it is on exclusive basis.

Conversely, Profile B DVB-HB Clients can "share" transcoders for the same service.

Although easier to define and handle, simply publishing the number of currently locked tuners would be insufficient, as the number of locked tuners does not permit to a DVB-HB Client to derive available services/resources, unless it assumes that all remaining tuners give access to all services. A simple list of available services provides more usable information, however the DVB-HB Client would not be able to consider what resources are released when switching to other services: the DVB-HB Client cannot assume that all services would be available, as some of the resources currently used may be shared with other clients, i.e., a release of such resource may not always lead to freeing it.

The technical solution described in the following, under responsibility of the *Resource Availability Map* function (see clause 5.4.10), allows instead to cover the above cases.

7.3.2 Logical structure of the Service Availability Map

The comprehensive logical structure is organised in form of an XML document describing the dependencies among resource usage associated to each service. This is described in the file ServiceAvailabilityMap.xml, which is referred to in the device capabilities exposed by the DVB-HB Local Server (see clause 7.2).

Logical structure and potential dependencies are represented in the hierarchy of the XML document. These dependencies are static, as they depend on the DVB-HB Local Server software functionality, hardware features and receiving antennas, while only the instantaneous usage of the resources changes.

A DVB-HB Local Server implementing this feature shall keep an up-to-date Services Availability Map in accordance with its used resources. When accessing the ServiceAvaivabilityMap.xml file, the DVB-HB Client should be able to determine the available services and present them to the user on the UI.

This proposal is applicable to both Profile A and Profile B DVB-HB Local Servers and DVB-HB Clients.

The DVB-HB Local Server may internally use the ServiceAvailabilityMap.xml to manage its resources together with the information on DVB-HB Clients potentially releasing a resource.

7.3.3 ServiceAvailabilityMap.xml document

The ServiceAvailabilityMap.xml document is structured in a hierarchy of `<DependencyResourceGroup>` elements. All child `<DependencyResourceGroup>` elements are dependent to each other, i.e., if one of these child elements is in use by a DVB-HB Client, it limits usage of other child elements belonging to the same parent `<DependencyResourceGroup>` element (if a limit is set). If any of the child elements is used, then the parent `<DependencyResourceGroup>` element becomes used.

EXAMPLE: In a typical case, the parent `<DependencyResourceGroup>` element could be a tuner, while its child `<DependencyResourceGroup>` elements could be the receivable multiplexes, each of them carrying a set of services. When one of the services in a multiplex is selected by a DVB-HB Client, the child `<DependencyResourceGroup>` element becomes 'used', and so also the parent `<DependencyResourceGroup>` element becomes 'used'. This means that other DVB-HB Clients cannot access any service in another multiplex, however the same tuner can still serve any of the services in the same multiplex (see also clause C.1.2).

Figure 4 shows the dependencies within a hierarchy of `<DependencyResourceGroup>` elements.

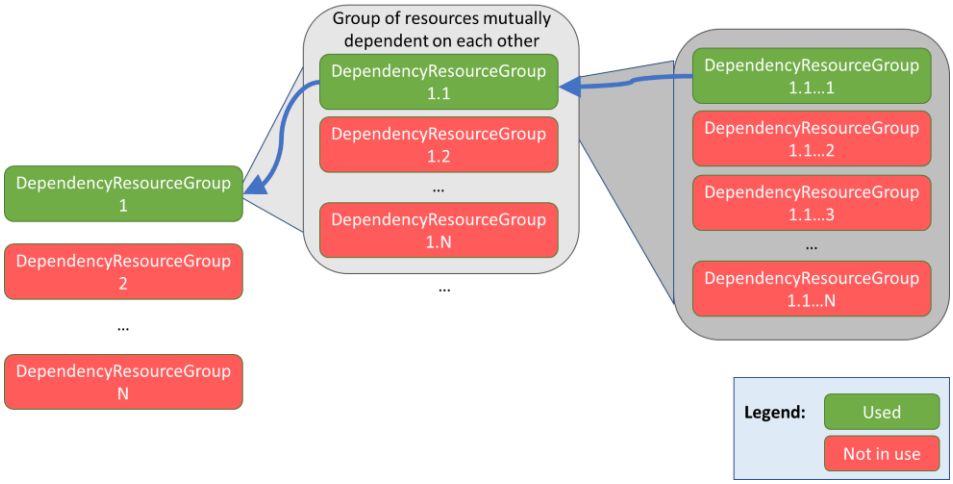

Figure 4: Schematic diagram on hierarchy of DependencyResourceGroup elements

The schema of the `<ServiceAvailabilityMap>` is reported in pseudocode 2 and table 3.

Pseudocode 2: ServiceAvailabilityMap schema

```
<complexType name="ServiceAvailabilityMapType">
  <sequence>
    <element name="HBLocalServerNode" type="dvbhbam:HBLocalServerNodeType"/>
  </sequence>
  <attribute name="version" type="positiveInteger" use="required"/>
</complexType>
<complexType name="ServiceType">
  <attribute name="serviceRef" type="dvbisd:ServiceIdentifierType" use="required"/>
  <attribute name="used" type="nonNegativeInteger" use="required"/>
  <attribute name="transcodedUsed" type="nonNegativeInteger" use="required"/>
</complexType>
<simpleType name="supportedModulationSystemType">
  <restriction base="string">
    <enumeration value="DVB-C"/>
    <enumeration value="DVB-T"/>
    <enumeration value="DVB-T2"/>
    <enumeration value="DVB-S"/>
    <enumeration value="DVB-S2"/>
    <enumeration value="DVB-S2X"/>
  </restriction>
</simpleType>
<complexType name="DependencyResourceGroupType">
  <sequence>
    <element name="supportedModulationSystem" type="dvbhbam:supportedModulationSystemType" minOccu
```

```
rs="0" maxOccurs="unbounded"/>
      <choice>
        <element name="DependencyResourceGroup" type="dvbhbam:DependencyResourceGroupType" maxOccurs
="unbounded"/>
        <element name="Service" type="dvbhbam:ServiceType" maxOccurs="unbounded"/>
      </choice>
    </sequence>
    <attribute name="id" type="string" use="required"/>
    <attribute name="max" type="nonNegativeInteger" use="optional"/>
    <attribute name="used" type="nonNegativeInteger" use="required"/>
  </complexType>
  <complexType name="HBLocalServerNodeType">
    <sequence>
      <element name="supportedModulationSystem" type="dvbhbam:supportedModulationSystemType" minOccu
rs="0" maxOccurs="unbounded"/>
      <element name="DependencyResourceGroup" type="dvbhbam:DependencyResourceGroupType" maxOccurs="
unbounded"/>
    </sequence>
    <attribute name="shared" type="boolean" use="optional" default="true"/>
    <attribute name="totalServedClients" type="nonNegativeInteger" use="required"/>
    <attribute name="totalServedClientsMax" type="nonNegativeInteger" use="required"/>
    <attribute name="totalTranscodedClients" type="nonNegativeInteger" use=" required "/>
    <attribute name="totalTranscodedClientsMax" type="nonNegativeInteger" use=" required "/>
    <attribute name="totalTranscodedServicesMax" type="nonNegativeInteger" use="required"/>
  </complexType>
```

Table 3: ServiceAvailabilityMap fields

Name	Semantic Definition	Constraints
`ServiceAvailabilityMap`	Root node of the XML document.	Mandatory
`@version`	Defines the version of the `ServiceAvailabilityMap`. It shall be identical to the `@version` of the `Availability` element described as part of the device capabilities (see table 2) and of the referred DVB-I Service List.	Mandatory
`HBLocalServerNode`	Node describing the DVB-HB Local Server	Mandatory
`DependencyResourceGroup`	Contains all resources elements which are dependent to each other	Mandatory 0 .. ∞
`Service`	Refers to services as described in a DVB-I Service List XML document. This field should be the lowest child element.	Mandatory 1 .. ∞
`@serviceRef`	Refers to the entry in the DVB-I Service List. See ETSI TS 103 770 [3] for semantic definition.	Mandatory
`@transcodedUsed`	Number of clients connected in transcoded mode for the user. This field is used to determine whether a transcoder would be released if the service is released.	Mandatory
`@id`	Identifier of the `DependencyResourceGroup`	Mandatory
`@supportedModulationSystemType`	Describes the supported modulation system of all subelements. Possible values are: “DVB-T”; "DVB-T2"; “DVB-S”, "DVB-S2", "DVB-S2X"; “DVB-C".	Optional
`@max`	Maximum number of usable elements belonging to the `DependencyResourceGroup`. If there is no limitation, this attribute is omitted.	Optional
`@used`	Number of used elements belonging to the `DependencyResourceGroup`	Mandatory
`@shared`	Describes whether a tuner can share a service resource. Only relevant with Profile A. Default is “true”.	Optional
`@totalServedClients`	Number of currently served clients. For example, it is relevant in case of bandwidth limitation.	Optional
`@totalServedClientsMax`	Number of maximally served clients. For example, it is relevant in case of bandwidth limitation.	Optional
`@totalTranscodedClients`	Number of clients served with a transcoded service on exclusive basis for Profile A	Optional
`@totalTranscodedClientsMax`	Number of clients maximally served with a transcoded service on exclusive basis for Profile A. Not relevant for Profile B.	Optional
`@totalTranscodedServicesMax`	Number of services maximally served with a transcoded service on exclusive basis for Profile B. Not relevant for Profile A.	Optional

Pseudocode 3 shows an example of a simple case with two independent tuners available on exclusive basis which can access all services (for simplicity, only 5 services belonging to different multiplexes are listed). In this example, one service (`tag:ses.com,2019:daserste.ses.com`) is currently served. The described DVB-HB Local Server does not

have transcoding capabilities (i.e., it is a Profile A DVB-HB Local Server, since only Profile A DVB-HB Local Servers are not required to transcode).

Pseudocode 3: Example of a ServiceAvailabilityMap element with 2 independent tuners and 5 services, no transcoding

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceAvailabilityMap version="2005201628"
 xmlns="urn:dvb:metadata:dvbhb-availabilitymap:2020" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
 xsi:schemaLocation="urn:dvb:metadata:dvbhb-availabilitymap:2020 ../schemas/dvbhb-
availabilitymap_v1.0.xsd" >
  <HBLocalServerNode shared="false" totalServedClients="1" totalServedClientsMax="2" totalTranscoded
Clients="0" totalTranscodedClientsMax="0" totalTranscodedServicesMax="0">
    <DependencyResourceGroup id="tuner1" max="1" used="1">
   <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="1" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
    <DependencyResourceGroup id="tuner2" max="1" used="0">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

Pseudocode 4 shows the same example but with transcoding capabilities.

Pseudocode 4: Example of a ServiceAvailabilityMap element with 2 independent tuners and 5 services and transcoding capabilities

```
<ServiceAvailabilityMap version="2005201628">
  <HBLocalServerNode shared="false" totalServedClients="1" totalServedClientsMax="2" totalTranscoded
Clients="1" totalTranscodedClientsMax="2" totalTranscodedServicesMax="2">
    <DependencyResourceGroup id="tuner1" max="1" used="1">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="1" transcodedUsed="1"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
    <DependencyResourceGroup id="tuner2" max="1" used="0">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

In this example, the DVB-HB Local Server has two independent transcoders. A DVB-HB Client (either a Profile A or Profile B DVB-HB Client) is currently using *"Das Erste"* service in transcoded mode.

The complexity of the `ServiceAvailabilityMap` structure depends directly on the dependencies among resources. In the simplest case with a single tuner on exclusive basis there would be just two levels, i.e., one `<HBLocalServerNode>` element containing all services that can be received.

Further examples are provided in annex C.

7.3.4 Management and processing of the Service Availability Map

7.3.4.1 Building the Service Availability Map

The hierarchy of the `<ServiceAvailabilityMap>` element is intended to describe the mutual dependency of several services' usage, whether transcoded or not. Apart from transcoding or limitation on maximum number of clients, only

resources which have a mutual impact on each other's usage need to be grouped in the same `<DependencyResourceGroup>` node.

The structure of the Service Availability Map is highly dependent on the configuration of the receiving installation. The number of levels of `<DependencyResourceGroup>` depends on the configuration. The `@id` attributes shall be different on each level to be uniquely identified.

The ServiceAvailabilityMap.xml file should be available after the installation phase of the DVB-HB Local Server. The actual `@serviceRef` attribute shall be updated correctly in accordance with the DVB-I formatted Service List.

NOTE: To facilitate the installation phase, several beforehand generated ServiceAvailabilityMap.xml versions could be available at product initialization: then only selecting the appropriate ServiceAvailabilityMap.xml version corresponding to the actual configuration would be sufficient.

7.3.4.2 Aligning the Service Availability Map with DVB-I Service List updates

A DVB-HB Local Server implementing this feature shall update the `<Service>` elements in the ServiceAvailabilityMap.xml file to keep it aligned with the DVB-I Service List. Each time the DVB-I Service List is updated and the `@version` attribute has a new value, the DVB-HB Local Server shall also update the ServiceAvailabilityMap.xml file and the `@version` attribute accordingly.

Rebuilding a complete XML hierarchy by the DVB-HB Local Server is not required, if the internal logic for the generation is stored.

For example:

- If, within a `<DependencyResourceGroup>` element, only services of Horizontal polarization and Low band can be received, all services having a Service Instance with these reception parameters are placed here.
- If a DVB-HB Local Server contains several independent tuners (i.e., each of the relevant `<DependencyResourceGroup>` element contains all services), then the updated service list can be directly reflected by updating all `<Service>` entries in each `<DependencyResourceGroup>` element.

7.3.4.3 Updating the Service Availability Map when assigning a service to a client

A DVB-HB Local Server implementing this feature shall update the usage attributes (i.e., `@used`, `@transcodedUsed`, `@totalServedClients`, `@totalTranscodedClients`) in the ServiceAvailabilityMap.xml file each time a stream representing a service is delivered to a DVB-HB Client, according to the procedure shown in pseudocode 5 or any alternative algorithm leading to the same result.

Pseudocode 5: Algorithm to update the Service Availability Map when assigning a service to a DVB-HB Client

```
Identify all entries containing the service with serviceRef corresponding to the service entry in th
e DVB-I formatted service list.
if ("shared" is set to false)
  eliminate serviceRef with used value greater than 0.
if (totalServedClients equals totalServedClientsMax)
  no stream can be provided as the totalClientsServed reached the maximum.
  EXIT
else
  Increase the totalServedClients by 1
  if (a transcoded stream is requested)
    if (the requested transcoded stream is exclusive)
      if the totalTranscodedClients value is equal to totalTranscodedClientsMax,
        no stream can be provided as the totalTranscodedClients reached the maximum.
        Reverse changes.
        EXIT
      else
        increase the @totalTranscodedClient attribute on the HBLocalServerNode element
        for each service entry with serviceRef proceed as follows until a stream can be assigned
          Increase the value of "transcodeUsed" and "@used" attributes of serviceElement by 1
          if (transcodedUsed takes the value 1)
            calculate totalTranscodedServices by counting services with transcodeUsed > 0
            if (totalTranscodedServices equals to totalTranscodedServicesMax)
             no transcoder is available
             reverse the change in the loop of this service entry
             proceed with the next service entry
          else
```

```
          increase the totalTranscodedServices by 1
          foreach DependencyResourceGroup containing the service entry
            If (one of the elements "@used" attributes of the child element changes from 0 to 1)
              if ("@used" attribute of the DependencyResourceGroup is equal to "@max" attribute)
                No DependencyResourceGroup resource is available
                reverse the changes for this service entry loop
                proceed with the next service entry
              else
                increase the "@used" attribute of the DependencyResoureGroup by 1
          A stream corresponding to the requested service could be successfully assigned
    else
      for each service entry with serviceRef proceed as follows until a stream can be assigned
        Increase the value of "transcodeUsed" and "@used" attributes of serviceElement by 1
        if (transcodedUsed takes the value 1)
          calculate totalTranscodedServices by counting services with transcodeUsed > 0
           if (totalTranscodedClients value is equal to totalTranscodedClientsMax)
              no transcoder is available
              reverse the change in the loop of this service entry loop
              proceed with the next service entry
           else
            increase the @totalTranscodedClient attribute on the HBLocalServerNode element
          if (totalTranscodedServices equals to totalTranscodedServicesMax)
            no transcoder is available
            reverse the change in the loop of this service entry loop
            proceed with the next service entry
          else
            foreach DependencyResourceGroup containing the service entry
              if (one of the elements "@used" attributes of the child element changes from 0 to 1)
                if ("@used" attribute of the DependencyResourceGroup is equal to "@max" attribute)
                  No DependencyResourceGroup resource is available
                  reverse the changes for this service entry loop
                  proceed with the next service entry
                else
                  increase the "@used" attribute of the DependencyResoureGroup by 1
            A stream corresponding to the requested service could be successfully assigned
  else
    for each service entry with serviceRef proceed as follows until a stream can be assigned
      Increase the value of "@used" attributes of service Element by 1
      foreach DependencyResourceGroup containing the service entry
        if (one of the elements "@used" attributes of the child element changes from 0 to 1)
          if ("@used" attribute of the DependencyResourceGroup is equal to "@max" attribute)
            No DependencyResourceGroup resource is available
            reverse the changes for this service entry loop
            proceed with the next service entry
          else
            increase the "@used" attribute of the DependencyResoureGroup by 1
      A stream corresponding to the requested service could be successfully assigned
  If (no service resource corresponding to serviceRef can be assigned)
    the service cannot be assigned
```

7.3.4.4 Updating the Service Availability Map when releasing a resource used by a client

Similarly to updating the ServiceAvailabilityMap.xml file when assigning a service to a DVB-HB Client, releasing a resource can be processed by the DVB-HB Local Server as shown in pseudocode 6.

Pseudocode 6: Algorithm to update the Service Availability Map when releasing a resource used by a DVB-HB Client

```
decrement the totalServedClients by 1
if a transcoded stream is released
  if (the requested transcoded stream was exclusive)
    decrement the value of "totalTranscodedClient"
  decrement the value of "transcodeUsed" and "@used" attributes of serviceElement
  if (transcodedUsed takes the value 0)
    decrement "totalTranscodedService"
    if (the requested transcoded stream was non exclusive)
      decrement the value of "totalTranscodedClient"
  foreach DependencyResourceGroup containing the service entry
    if (one of the elements "@used" attributes of the child element changes from 1 to 0)
      decrement the value of the "@used" attribute of the DependencyResoureGroup
else
  decrement the value of "@used" attributes of serviceElement
  foreach DependencyResourceGroup containing the service entry
```

```
    if (one of the elements "@used" attributes of the child element changes from 1 to 0)
      decrement the value of the "@used" attribute of the DependencyResoureGroup
```

7.3.4.5 Verifying the availability of services at a client

When attempting to request one service to the DVB-HB Local Server, it is recommended that the DVB-HB Client checks beforehand the actual availability of the service, so that a potential unavailability could be handled with user friendly experience, e.g.:

- Showing a logo with the currently unavailable service and an information message;
- Not offering the service on the menu or mark it as unavailable (this may be updated just before building the graphics or afterwards).

To assess the actual availability of a resource, a DVB-HB Client implementing this feature shall process the serviceAvailabilityMap.xml file as shown in pseudocode 7, or by any alternative algorithm leading to the same result.

Pseudocode 7: Algorithm to verify the availability of services for a DVB-HB Client

```
Update its local copy of ServiceAvailabilityMap.xml with the "simulated" release of its own resource
 as described in previous clause
if (shared equals false)
  eliminate all services whose attribute "@used" is greater than 0
foreach DepencyResourceGroup of the HBLocalServerNode
  if ("@used" attribute is equal to "@max" attribute)
    eliminate all children whose "@used" attribute is equal to 0
foreach Remaining services element
  if a transcoded stream is requested
    if service entry has an attribute "@transcodedUsed" equal to 0
      if (@totalTranscodedService equals @totalTranscodedServiceMax)
        eliminate the service entry
```

7.3.4.6 Reserving exclusive resources by a DVB-HB Local Server for a client

A DVB-HB Local Server may, with or without request, reserve a resource on exclusive basis. This may be especially the case if the DVB-HB Local Server detects that a DVB-HB Client is zapping quickly through services, or when the DVB-HB Local Server is proceeding with an update scan. A temporary exclusive assignment may be advisable in these circumstances.

In this case the `@used` attribute of the concerned resource is increased by 1. Other DVB-HB Clients may not be able to request a resource as the `<DependencyResourceGroup>` usage would reach its maximum.

With reference to the example shown in pseudocode 3, the DVB-HB Local Server may decide to allocate the second tuner exclusively to one client. In that case, the ServiceAvailabilityMap.xml file may appear as "suboptimal" as two tuners may be used for the same service.

7.3.5 Splitting the ServiceAvailabilityMap.xml file into ServiceAvailabilityIdle.xml and ServiceAvailabilityUpdate.xml

The logical structure and dependencies of the Service Availability Map may be quite complex, as it involves all services in one or several instances (e.g., multiple tuners). Reloading the complete ServiceAvailabilityMap.xml by the DVB-HB Client appears unnecessary, since only the `@used` attributes need to be updated to keep the ServiceAvailabilityMap.xml up-to-date.

For this reason, the ServiceAvailabilityMap.xml shall be split into two files:

1) ServiceAvailabilityMapIdle.xml, whose `@used` attributes are all set to 0 (i.e., idle mode),

2) ServiceAvailabilityMapUpdate.xml, which contains the actual usage by indicating the updated values of the respective `@used` attributes.

At start time, a DVB-HB Client loads the ServiceAvailabilityMapIdle.xml file. Every time it needs an updated resource usage, it retrieves the ServiceAvailabilityMapUpdate.xml and combines it with the ServiceAvailabilityMapIdle.xml file to obtain the actual ServiceAvailabilityMap.xml.

The syntax of ServiceAvailabilityMapUpdate.xml file shall follow IETF RFC 5261 [14].

With reference to the example in pseudocode 3, the ServiceAvailabilityMapIdle.xml file would be as shown in pseudocode 8 and the ServiceAvailabilityMapUpdate.xml file would be as shown in pseudocode 9. When combining them, the result would be according to pseudocode 3.

Pseudocode 8: Example of ServiceAvailabilityMapIdle.xml

```
<ServiceAvailabilityMap version="2005201628">
  <HBLocalServerNode shared="false" totalServedClients="0" totalServedClientsMax="2" totalTranscoded
Clients="0" totalTranscodedClientsMax="2" totalTranscodedServicesMax="2">
    <DependencyResourceGroup id="tuner1" max="1" used="0">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
    <DependencyResourceGroup id="tuner2" max="1" used="0">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

Pseudocode 9: Example of ServiceAvailabilityMapUpdate.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xml>
<diff>
  <replace sel="ServiceAvailabilityMap/HBLocalServerNode/@totalServedClients">1</replace>
  <replace sel="ServiceAvailabilityMap/HBLocalServerNode/@totaltranscodedClients">1</replace>
  <replace sel="ServiceAvailabilityMap/HBLocalServerNode/DependencyResourceGroup[@id='tuner1']@used"
>1</replace>
  <replace sel="ServiceAvailabilityMap/HBLocalServerNode/DependencyResourceGroup[@id='tuner1']/Servi
ce[@serviceRef='tag:ses.com,2019:daserste.ses.com'/@used">1</replace>
</diff>
```

7.3.6 Dependencies and backwards compatibility (informative)

Since in the ServiceAvailabilityMap.xml file the services are referred to using the same unique identifier (`@serviceRef` attribute) as in the corresponding DVB-I formatted Service List, only DVB-HB Clients and DVB-HB Local Servers supporting DVB-I formatted Service Lists can take advantage of the service availability feature (i.e., some Profile A devices not supporting DVB-I formatted Service Lists are excluded).

A DVB-HB Client not supporting this feature can anyway continue to offer all services and will still benefit from DVB-HB Local Server resource sharing; the user will eventually get an error message in case of lack of available resources.

7.3.7 Implementation recommendations on client devices (informative)

7.3.7.1 Broadcast-only mode

When switching amongst services (e.g., using P+/P- keys), it is a recommended that the DVB-HB Client does no skip the temporarily unavailable services but shows an info banner. This allows the user to keep orientation on his service list. This is also valid if the user explicitly selects the service by typing the corresponding LCN key.

7.3.7.2 Client mode supporting DVB-HB and DVB-I services

The Service Availability Map feature only addresses availability of broadcast services, received at the DVB-HB Local Server front-ends and redistributed on the LAN.

In case a specific service is flagged as currently unavailable due to lack of resources, a DVB-HB Client with also a broadband connection may use an Internet-delivered DASH alternative Service Instance of the same service, if also declared in the Service List.

8 Extensions to SAT>IP specification

8.1 Introduction

This clause defines the optional extensions to the SAT>IP specification [1], applicable to Profile A DVB-HB Local Servers and Profile A DVB-HB Clients.

8.2 Support to DVB-S2X

The Table in SAT>IP specification [1], clause 3.5.11 is extended as defined in table 4 below.

Table 4: Query attributes for DVB-S, DVB-S2 and DVB-S2X signals

Name	Attribute	Value	Example
Frontend identifier	fe	Numerical value between 1 and 65535. Not required in normal client queries.	`fe=1`
Signal source	src	Numerical value between 1 and 255. Default value is "1".	`src=1`
Frequency	freq	Transponder frequency expressed in MHz as fixed point type or integer value	`freq=11361.75` `freq=11720`
Polarisation	pol	Set to one of the following values: • "h" horizontal linear • "v" vertical linear • "l" circular left • "r" circular right	`pol=h`
Roll-Off	ro	Set to one of the following values: • "0.35", "0.25", "0.20", "0.15", ”0.10", "0.05". For DVB-S this value shall be set to "0.35" in client queries. For DVB-S2 this value shall be set to "0.35", "0.25" or "0.20" in client queries.	`ro=0.35`
Modulation system	msys	Set to one of the following values: • "dvbs", "dvbs2", "dvbs2x".	`msys=dvbs2`
Modulation type	mtype	Set to one of the following values: • "qpsk", "8psk", "8psk-l", "16apsk", "16apsk-l", "32apsk", "32apsk-l", "64apsk", "64apsk-l". For DVB-S this value shall be set to "qpsk" in client queries. For DVB-S2 this value shall be set to "qpsk" or "8psk" in client queries. For DVB-S2X, the possible combinations of mtype and fec shall be according to ETSI EN 302 307-2 [6], Table 1.	`mtype=8psk`
Pilot tones	plts	Set to one of the following values: • "on", "off". For DVB-S this value shall be set to "off" in client queries.	`plts=off`
Symbol rate	sr	Value in kSymb/s	`st=22000`
FEC inner	fec	Set to one of the following values: • "12", "23", "34", "56", "78", "89", "35", "45", "910", "14", "13", 25", "1345", "920", "1120", "2336", "2536", "1318". "59", "2645", "2845", "79", "7790", "815", "3245", "1115". For DVB-S this value shall be set to "12", "23", "34", "56" or "78" in client queries. For DVB-S2 this value shall be set to "12", "23", "34", "56", "78", "89", "35", "45" or "910" in client queries. For DVB-S2X, the possible combinations of mtype and fec shall be according to ETSI EN 302 307-2 [6], Table 1.	`fec=23`
Modcod mode	cm	Set to one of the following values: • "ccm", "vcm". Not used for DVB-S and DVB-S2 (default is "ccm"). If provided for DVB-S or DVB-S2, the parameter shall be ignored.	`cm=ccm`
Input stream identifier	isi	Uniquely identifies an Input Stream Identifier. Numerical value between 0 and 255. Not required for DVB-S and DVB-S2. For DVB-S2X, required only if cm is set to "vcm" and multiple Transport Streams are delivered on the transponder.	`isi=11`
Channel bonding	bond (see NOTE 1)	Set to one of the following values: • "on", "last", "off". Not used for DVB-S and DVB-S2 (default is "off”). If provided for DVB-S or DVB-S2, the parameter shall be ignored.	`bond=off`
List of PIDs	pids	Comma Separated Values (CSV) list of PIDs • Numerical values between 0 and 8191 for Single Programme Transport Stream (SPTS) • "all" for Multi-Programme Transport Stream (MPTS) • "none" for no demux output Not used for DVB-S and DVB-S2 (default is "off”). If provided for DVB-S or DVB-S2, the parameter shall be ignored.	`pids=0,16,201,302`

Name	Attribute	Value	Example
	addpids (see NOTE 2)	Opens new PID filters on the demux for streaming on the network. CSV list of PIDs.	`addpids=307,309`
	delpids (see NOTE 2)	Removes PID filters from the demux. CSV list of PIDs.	`delpids=201,302`
NOTE 1: In case of channel bonding, client shall send multiple SETUP messages, as shown in the example below. The pids parameters can be included in any of those messages. NOTE 2: The addpids or delpids parameters shall not be used in combination with the pids parameter in the same RTSP query. The addpids and delpids parameters may be combined in the same RTSP query.			

Example of tuning command in case of DVB-S2X and channel bonding over three transponders:

```
SETUP rtsp://192.168.178.57:554/?src=1&fe=1&freq=11013&pol=h&ro=0.20&msys=dvbs2x&mtype=8psk&plts=off
&sr=29900&fec=34&cm=ccm&bond=on RTSP/1.0

SETUP rtsp://192.168.178.57:554/?src=1&fe=2&freq=11785&pol=h&ro=0.20&msys=dvbs2x&mtype=16ask&plts=of
f&sr=27500&fec=23&cm=ccm&bond=on RTSP/1.0

SETUP rtsp://192.168.178.57:554/?src=1&fe=3&freq=11766&pol=v&ro=0.20&msys=dvbs2x&mtype=8psk&plts=off
&sr=29900&fec=34&cm=ccm&bond=last&pids=0,1840,1841,1843 RTSP/1.0
```

RTSP DESCRIBE and RTP Control Protocol (RTCP) Announcement syntax of the SAT>IP specification [1] are extended as follows:

- RTSP DESCRIBE

  When supporting DVB-S2X, the attribute syntax in the Session Description Protocol (SDP) implementation is extended as follows:

  - Session level:

    ```
    s=SatIPServer:1 <sat frontends>,<terr frontends>,<cable frontends>
    ```

  - Media level:

    ```
    a=control:stream=<streamID>

    a=fmtp:33 ver=1.3;src=<srcID>;tuner=<feID>,<level>,<lock>,<quality>,<frequency>,<pola
    risation>,<system>,<type>,<pilots>,<roll_off>,<symbol_rate>,<fec_inner>,<modcod_mode>
    ,<isi>,<bond>;pids=<pid0>,…,<pidn>
    ```

NOTE 1: The version number is set to 1.3.

NOTE 2: `<sat frontends>` provides the number of satellite frontends available from the server (DVB-S plus DVB-S2 plus DVB-S2X). `<terr frontends>` provides the number of terrestrial frontends available from the server (DVB-T plus DVB-T2). `<cable frontends>` provides the number of cable frontends available from the server (DVB-C plus DVB-C2).

- RTCP APP Packet String Payload Format for DVB-S2X:

  Similarly to the SDP extension, the APP Packet payload format for RTCP Announcements is extended as follows:

  ```
  ver=1.3;src=<srcID>;tuner=<feID>,<level>,<lock>,<quality>,<frequency>,<polarisation>,<system
  >,<type>,<pilots>,<roll_off>,<symbol_rate>,<fec_inner>,<modcod_mode>,<isi>,<bond>;pids=<pid0
  >,…,<pidn>
  ```

NOTE 3: The version number is set to 1.3.

Example text string:

```
ver=1.3;src=1;tuner=1,240,1,7,12402,v,dvbs2x,8psk,,0.20,29900,34,,,;pids=0,16,56,112,168,179
```

8.3 Description of device capabilities

The "XML Device Description" defined in SAT>IP specification [1], clause 3.4 is extended with a `<dvbhb:X_SATIP_DVBHB>>` element according to pseudocode 10 and table 5.

Pseudocode 10: X_SATIP_DVBHB schema

```
<complexType name="X_SATIP_DVBHB">
  <sequence>
    <element name="ServiceListOffering" type="dvbisld:ServiceListOfferingType" minOccurs="0" />
    <element name="Availability" type="dvbhbx:HBServiceAvailabilityType" minOccurs="0" />
    <element name="AL-FEC" minOccurs="0" >
      <simpleType>
        <restriction base="string">
          <enumeration value="none"/>
          <enumeration value="baselayer"/>
          <enumeration value="base+enhancementlayer"/>
        </restriction>
      </simpleType>
    </element>
  </sequence>
</complexType>
```

Table 5: X_SATIP_DVBHB fields

Name	Semantic Definition	Constraints
`ServiceListOffering`	A list of details and location of the Service List(s) offered by the DVB-HB Local Server, according to ETSI TS 103 770 [3], clause 5.3.5. Present only if the Profile A DVB-HB Local Server exposes the service list according to the DVB-I format.	Optional 0 .. 1
`Availability`	Announces the capabilities of the DVB-HB Local Server in terms of type and number of front-ends (tuners). It also informs about real-time status in terms of used and available resources. See table 2 for semantic definition of `HBServiceAvailabilityType`.	Optional 0 .. 1
`AL-FEC`	Announces the capabilities of the DVB-HB Local Server in terms of support to optional AL-FEC, according to clause 8.4.2.	Optional 0 .. 1

8.4 Network resilience in Profile A

8.4.1 Introduction

Audio/video content redistributed by a Profile A DVB-HB Local Server using UDP-based transport protocol, which does not support packet retransmission, may suffer from network packet losses, especially in case of wireless communication. While error protection mechanisms are provided as part of the relevant network standards, e.g. Wireless Local Area Network (WLAN)/WIreless FIdelity (Wi-Fi) based on the Institute of Electrical and Electronic Engineers (IEEE) 802.11 family, additional network issues might deteriorate the signal to a level that renders the TS packets, which are sent Quasi Error Free (QEF) by the DVB-HB Local Server (assuming QEF broadcast reception), received by the DVB-HB Client device as damaged and unrecoverable.

In the following, an additional option to improve network resilience is defined, based on AL-FEC.

8.4.2 Optional AL-FEC

The optional AL-FEC which can be used in the transmission from a Profile A DVB-HB Local Server to a Profile A DVB-HB Client shall be according to the DVB Internet Protocol TeleVision (DVB-IPTV) specification as defined in ETSI TS 102 034 [12], Annex E.

The presence of an AL-FEC layer shall be signalled by the DVB-HB Local Server in the `<AL-FEC>` element of the `<satip-X_DVBHB>` element, as described in clause 8.3.

If the `<AL-FEC>` is present and its value is different from `none`, the DVB-HB Local Server supports AL-FEC as follows:

- If the value of the `<AL-FEC>` element is `baselayer`, the DVB-HB Local Server supports a Base Layer AL-FEC according to ETSI TS 102 034 [12].
- If the value of the `<AL-FEC>` element is `base+enhancementlayer`, the DVB-HB Local Server supports a Base Layer AL-FEC and an Enhanced Layer AL-FEC according to ETSI TS 102 034 [12].

The UDP port of the AL-FEC Base Layer stream shall be equal to the "`client RTP port`" as per the SAT>IP specification [1] plus 2. The UDP port of the AL-FEC Enhancement Layer stream shall be equal to the "`client RTP port`" as per the SAT>IP specification [1] plus 4.

The RTSP SETUP message defined in the SAT>IP specification [1] is extended with the optional query string parameter `&al-fec`.

EXAMPLE: `rtsp://192.168.128.5/?freq=754&bw=8&msys=dvbt&tmode=8k&mtype=64qam&gi=132&fec=23&pids=0,16,50,201,301&al-fec=baselayer`

8.4.3 Implementation considerations (informative)

Figure 5 describes the functional diagram of the AL-FEC generation included in a Profile A DVB-HB Local Server, showing the layers involved. It is part of the *Network resilience (Tx)* subfunction of the *Content publication* function in the DVB-HB reference architecture (see clause 5).

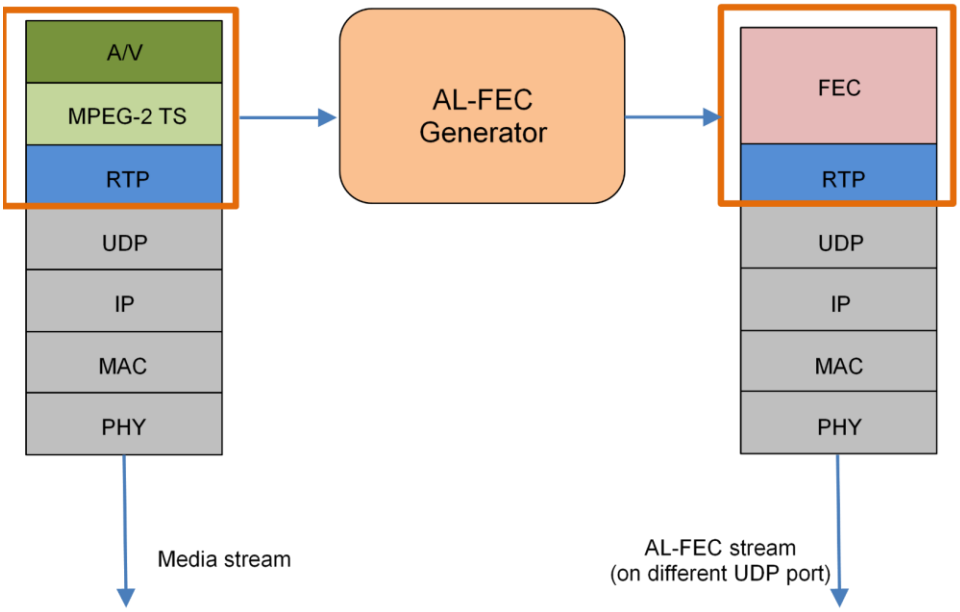

Figure 5: Functional diagram of AL-FEC generation

Lower layer protocols (i.e., UDP encapsulation, Media Access Control (MAC) drivers, etc.) are not involved, and no feedback from the client is required. The solution can be applied on top of any kind of IP network, including any Wi-Fi network of the IEEE 802.11 family.

The solution requires direct control of the RTP layer. This is reasonable as:

- Profile A DVB-HB Local Servers are expected to control the RTP layer anyway, due to the strict requirements on RTP packet generations given by the SAT>IP specification [1];
- Profile A DVB-HB Clients implementing this optional feature need to access the RTP layer to detect IP packet loss (i.e., retrieving the Sequence Number field in RTP packets header [i.2]).

NOTE: In addition to this mechanism based on AL-FEC, Profile A DVB-HB devices may implement additional optimisations at lower layers, leveraging some advanced functions available in specific IEEE 802.11 standards. Some background information is given in annex E.

8.4.4 Backwards compatibility (informative)

The DVB-IPTV AL-FEC consists of a Forward Error Correction (FEC) stream sent in parallel to the (untouched) media stream, on a different UDP port.

Therefore, Profile A DVB-HB Clients not implementing this optional feature can decode the media stream simply ignoring the AL-FEC stream (of course without benefiting from the additional error correction).

9 Extensions to the DVB-I specification

9.1 Introduction

This clause defines the extensions to ETSI TS 103 770 [3], applicable to Profile B DVB-HB Local Servers and Profile B DVB-HB Clients.

These extensions leverage the extensibility defined in ETSI TS 103 770 [3], and are therefore backwards-compatible with DVB-I Clients.

9.2 Extended Service List Entry Points

9.2.1 Introduction

As described in clause 7.2, a Profile B DVB-HB Local Server shall expose basic information about itself in the form of an extended Service List Entry Point XML file.

The `HBxServiceListEntryPointsType` defined in this clause extends the `dvbisld:ServiceListEntryPoints` element of ETSI TS 103 770 [3].

NOTE: Legacy DVB-I Clients can access a Service List Entry Point published by the DVB-HB Local Server to retrieve the URL of the relevant DVB-I Service List, but discarding the additional device description and capabilities carried in the `<HBLocalServerEntity>` element defined in the present document.

9.2.2 Extension to dvbisld:ServiceListEntryPoints

Based on extensibility of `ServiceListEntryPointsType` allowed by ETSI TS 103 770 [3], clause 5.3.2, the extension according to pseudocode 11 and table 6 is defined.

Pseudocode 11: HBxServiceListEntryPointsType schema

```
<complexType name="HBxServiceListEntryPointsType" abstract="false">
  <annotation>
    <documentation xml:lang="en">DVB-
HB extension to "dvbisld:ServiceListEntryPointsType"</documentation>
  </annotation>
  <complexContent>
    <extension base="dvbisd:ExtensionBaseType">
      <sequence>
        <element name="HBLocalServerEntity" type="dvbhbx:HBLocalServerType" minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Table 6: HBxServiceListEntryPointsType fields

Name	Semantic Definition	Constraints
`HBLocalServerEntity`	In case the Service List Entry Point is published by a DVB-HB Local Server, this element provides description and capabilities of the DVB-HB Local Server. See table 1 for semantic definition of `HBLocalServerType`.	Optional 0 .. 1

9.2.3 Example of use of the extended ServiceListEntryPoints

The example in pseudocode 12 describes an extended Service List Entry Point published by a DVB-HB Local Server.

Pseudocode 12: Example of extended Service List Entry Point published by a Profile B DVB-HB Local Server

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xml>
<!-- Example of ServiceListEntryPoint published by a DVB-HB Local Server -->
<ServiceListEntryPoints xmlns="urn:dvb:metadata:servicelistdiscovery:2020" xmlns:xsi="http://www.w3.
org/2001/XMLSchema-instance" xsi:schemaLocation="urn:dvb:metadata:dvbhb-
```

```
extensions:2020 ../schemas/dvbhb-extensions_v1.0.xsd" xmlns:dvbhbx="urn:dvb:metadata:dvbhb-
extensions:2020" xmlns:dvbisd="urn:dvb:metadata:servicediscovery:2020" xmlns:tva="urn:tva:metadata:2
019">
  <ServiceListRegistryEntity>
    <Name>DVB-HB Local Server Registry</Name>
  </ServiceListRegistryEntity>
  <ProviderOffering>
    <Provider>
      <Name>DVB-HB Local Server</Name>
    </Provider>
    <ServiceListOffering>
      <ServiceListName>DVB-HB Local Server #1</ServiceListName>
      <ServiceListURI contentType="application/xml">
        <dvbisd:URI>http://192.168.1.101/dvbhb/servicelist.xml</dvbisd:URI>
      </ServiceListURI>
    </ServiceListOffering>
  </ProviderOffering>
  <Extension xmlns:dvbhbx="urn:dvb:metadata:dvbhb-
extensions:2020" xsi:type="dvbhbx:HBxServiceListEntryPointsType" extensionName="DVB-HB">
    <dvbhbx:HBLocalServerEntity specVersion="1">
      <dvbhbx:DeviceType>urn:dvb:metadata:device:HBLocalServer:1</dvbhbx:DeviceType>
      <dvbhbx:UniqueDeviceName>uuid:67969aea-73f9-11ea-bc55-0242ac130003</dvbhbx:UniqueDeviceName>
      <dvbhbx:ModelName>DVB-HB Local Server</dvbhbx:ModelName>
      <dvbhbx:FriendlyName>Multi-tuner DVB-HB Profile B Local Server</dvbhbx:FriendlyName>
      <dvbhbx:Manufacturer>DVB-HB Local Server Manufacturer</dvbhbx:Manufacturer>
      <dvbhbx:SerialNumber>HB-0242ac130003</dvbhbx:SerialNumber>
      <dvbhbx:ManufacturerURL>http://www.dvbhb-localserver-manufacturer.com</dvbhbx:ManufacturerURL>
      <dvbhbx:Availability version="1">
        <dvbhbx:ServiceAvailabilityMapIdleURL contentType="application/xml">
          <dvbisd:URI>http://192.168.1.101/dvbhb/ServiceAvailabilityMapIdle.xml</dvbisd:URI>
        </dvbhbx:ServiceAvailabilityMapIdleURL>
        <dvbhbx:ServiceAvailabilityMapUpdateURL contentType="application/xml">
          <dvbisd:URI>http://192.168.1.101/dvbhb/ServiceAvailabilityMapUdpate.xml</dvbisd:URI>
        </dvbhbx:ServiceAvailabilityMapUpdateURL>
      </dvbhbx:Availability>
    </dvbhbx:HBLocalServerEntity>
  </Extension>
</ServiceListEntryPoints>
```

9.3 Identification of services originated by the DVB-HB Local Server

9.3.1 Introduction

The Service List published by a DVB-HB Local Server can identify, per service, that the service provided originates from the DVB-HB Local Server tuners and not from an external source.

This identification is achieved by means of `<OriginalDeliverySource>` element as defined in `HBxDASHDeliveryParametersType`, which extends `dvbisd:DASHDeliveryParametersType`.

NOTE: Legacy DVB-I Clients can access an extended `<DASHDeliveryParameters>` element, included in a DVB-I Service List published by the DVB-HB Local Server, to retrieve the URL of the relevant DASH MPD, but discarding the additional identification of the source.

9.3.2 Extension to dvbisd:DASHDeliveryParametersType

Based on extensibility of `DASHDeliveryParametersType` allowed by ETSI TS 103 770 [3], clause 5.5.18.6, the extension according to pseudocode 13 and table 7 is defined.

Pseudocode 13: HBxDASHDeliveryParametersType schema

```
<complexType name="HBxDASHDeliveryParametersType" abstract="false" >
  <annotation>
    <documentation xml:lang="en">DVB-
HB extension to dvbisd:DASHDeliveryParametersType</documentation>
  </annotation>
  <complexContent>
    <extension base="dvbisd:ExtensionBaseType">
      <sequence>
        <element name="OriginalDeliverySource" minOccurs="0">
          <simpleType>
```

```
            <restriction base="anyURI">
              <enumeration value="urn:dvb:metadata:source:dvb-t"/>
              <enumeration value="urn:dvb:metadata:source:dvb-s"/>
              <enumeration value="urn:dvb:metadata:source:dvb-c"/>
            </restriction>
          </simpleType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Table 7: HBxDASHDeliveryParametersType fields

Name	Semantic Definition	Constraints
`OriginalDeliverySource`	If this service instance is generated by a DVB-HB Local Server, identifies the original delivery source for this service instance. Possible values are: `urn:dvb:metadata:source:dvb-t`, `urn:dvb:metadata:source:dvb-s`, `urn:dvb:metadata:source:dvb-c`.	Optional 0 .. 1

9.3.3 Example of use of the extended DASHDeliveryParametersType

The example in pseudocode 14 describes an extended DASH service instance published by a DVB-HB Local Server.

Pseudocode 14: Example of extended DASH service instance published by a DVB-HB Local Server

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xml>
<ServiceList version="1" xmlns="urn:dvb:metadata:servicediscovery:2020" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xsi:schemaLocation="urn:dvb:metadata:dvbhb-
extensions:2020 ../schemas/dvbhb-extensions_v1.0.xsd" xmlns:dvbhbx="urn:dvb:metadata:dvbhb-
extensions:2020" xmlns:tva="urn:tva:metadata:2019">
  <Name>Example of DASH service instance published by a DVB-HB Local Server</Name>
  <ProviderName>HB Local Server</ProviderName>
  <Service version="1">
    <UniqueIdentifier>tag:192.168.1.101,2020:dvb-s/318.5200.3401/service1</UniqueIdentifier>
    <ServiceInstance>
      <DASHDeliveryParameters>
        <UriBasedLocation contentType="application/dash+xml">
          <URI>http://192.168.1.101/dash/11766V_service1.mpd</URI>
        </UriBasedLocation>
        <Extension xmlns:dvbhbx="urn:dvb:metadata:dvbhb-
extensions:2020" xsi:type="dvbhbx:HBxDASHDeliveryParametersType" extensionName="DVB-HB">
          <dvbhbx:OriginalDeliverySource>urn:dvb:metadata:source:dvb-s
          </dvbhbx:OriginalDeliverySource>
        </Extension>
      </DASHDeliveryParameters>
    </ServiceInstance>
    <ServiceName>Service1</ServiceName>
    <ProviderName>Provider1</ProviderName>
  </Service>
</ServiceList>
```

10 Conversion of DVB-SI metadata to DVB-I format (informative)

10.1 Introduction

The service list published by a DVB-HB Local Server in a DVB-I compatible format (be it a Profile A DVB-HB Local Server implementing this optional functionality or a Profile B DVB-HB Local Server) can be generated by the *Service List publication* function based on the DVB SI metadata according to ETSI EN 300 468 [15], carried in the input TS as received at the reference point **T** (see also clause 5).

In the following, description on how to map some of the SI fields to DVB-I metadata is provided.

Alternatively, the generation of the service list in a DVB-I compatible format (or part of it) may rely on an external repository, as described in clause 5.4.12. It is recommended that the DVB-I metadata is always aligned with the actual SI.

NOTE: The acquisition time, which is dependent on repetition rates of the SI within a broadcast signal, can affect the generation or update of local DVB-I metadata.

10.2 Service description

Table 8 lists the association of some of the fields of the SI Service Description Table (SDT) with the elements in the output DVB-I Service List.

Table 8: Generation of DVB-I Service List elements from SI SDT fields

DVB-I service discovery element	SI table	SI descriptor	SI field
`Service.ServiceName` (multiple service names can be specified as long as they have different `@xml:lang` values)	SDT	`service_descriptor` as defined in ETSI EN 300 468 [15], clause 6.2.36	*service_name*
		`multilingual_service_name_descriptor` as defined in ETSI EN 300 468 [15], clause 6.2.25	*service_name*
`Service.ProviderName` (multiple service provider names can be specified as long as they have different `@xml:lang` values)		`service_descriptor` as defined in ETSI EN 300 468 [15], clause 6.2.36	*provider_name*
		`multilingual_service_name_descriptor` as defined in ETSI EN 300 468 [15], clause 6.2.25	*provider_name*
`Service.ServiceType` and `Service.ServiceInstance.ContentAttributes` (see ETSI TS 103 770 [3], Table 67)	See ETSI TS 103 770 [3], Table 107		

10.3 Content Guide

10.3.1 ScheduleInfo endpoint

A DVB-HB Local Server should support a Content Guide server functionality as defined in ETSI TS 103 770 [3], clause 6.

A DVB-HB Local Server should host such Content Guide `ScheduleInfo` endpoint via the *Web server* subfunction of the *Service List publication* function. The endpoint URL should be included in the `ServiceList.ContentGuideSource` element (or in the `Service.ContentGuideSource` elements, if different for the different services) in the published DVB-I Service List.

Alternatively, the Content Guide server functionality may rely on a remote server on the cloud. It is recommended that the DVB-I Content Guide metadata is always aligned with the actual SI Event Information Table (EIT).

NOTE: The acquisition time, which is dependent on repetition rates of the SI within a broadcast signal, can affect the generation or update of local DVB-I metadata.

A DVB-HB Local Server should construct the Content Guide metadata from the SI information. Table 9 lists the correspondence between the main Content Guide metadata items and the SI fields.

Table 9: Association of SI EIT fields with DVB-I Content Guide elements

DVB-I Content Guide element	SI table	SI descriptor	SI field
`ProgramDescription.ProgramLocationTable.Schedule@serviceIDRef` (set to the `UniqueIdentifier` of the corresponding service in the Service List)	EIT	-	*service_id*
Not available. Content Guide is pulled by client on request and not pushed by server.		-	*version_number*
`ProgramDescription.ProgramLocationTable.Schedule.ScheduleEvent.Program@crid` and `ProgramDescription.ProgramInformationTable.ProgramInformation@programId`		-	*event_id*
`ProgramDescription.ProgramLocationTable.Schedule.ScheduleEvent.PublishedStartTime`		-	*start_time*
`ProgramDescription.ProgramLocationTable.Schedule.ScheduleEvent.PublishedDuration`		-	*duration*
`ProgramDescription.ProgramInformationTable.BasicDescription.Synopsis` (with `@length="medium"`)		`short_event_descriptor`	*event_name*
`ProgramDescription.ProgramInformationTable.BasicDescription.CreditsLists` or `ProgramDescription.ProgramInformationTable.BasicDescription.Synopsis` (with `@length="long"`)		`extended_event_descriptor`	*item_description* and *item*
See table 8		`component_descriptor`	
`ProgramDescription.ProgramInformationTable.BasicDescription.Genre`		`content_descriptor`	
`ProgramDescription.ProgramInformationTable.BasicDescription.ParentalGuidance.MinimumAge`		`parental_rating_descriptor`	*rating*

10.3.2 Example of Content Guide (present/following)

Pseudocode 15 shows an example of a query to the `ScheduleInfo` endpoint on present/following schedule, i.e.:
`<ScheduleInfoEndpoint>?sid=<service_id>&now_next=true`

Pseudocode 15: Example of result of a query to the Content Guide server (present/following)

```
<?xml version="1.0" encoding="UTF-8"?>
<TVAMain xmlns="urn:tva:metadata:2019"
 xmlns:mpeg7="urn:tva:mpeg7:2008" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:tva:metadata:2019 ../schemas/tva_metadata_3-1.xsd"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xml:lang="eng">
  <ProgramDescription>
    <ProgramInformationTable xml:lang="eng">
      <ProgramInformation programId="crid://service1/now">
        <BasicDescription>
          <Title type="main">Title of current programme</Title>
          <Synopsis length="medium">Description of current programme</Synopsis>
          <ParentalGuidance>
            <mpeg7:MinimumAge>14</mpeg7:MinimumAge>
          </ParentalGuidance>
        </BasicDescription>
      </ProgramInformation>
      <ProgramInformation programId="crid://service1/next">
        <BasicDescription>
          <Title type="main">Title of next programme</Title>
          <Synopsis length="medium">Description of next programme</Synopsis>
          <ParentalGuidance>
            <mpeg7:MinimumAge>0</mpeg7:MinimumAge>
          </ParentalGuidance>
        </BasicDescription>
      </ProgramInformation>
    </ProgramInformationTable>
    <ProgramLocationTable xml:lang="eng">
      <Schedule serviceIDRef="service1" start="2021-01-17T10:57:00Z" end="2021-01-17T12:12:00Z">
        <ScheduleEvent>
          <Program crid="crid://service1/now"/>
          <PublishedStartTime>2021-01-17T10:57:00Z</PublishedStartTime>
          <PublishedDuration>PT45M</PublishedDuration>
```

```
        </ScheduleEvent>
        <ScheduleEvent>
          <Program crid="crid://service1/next"/>
          <PublishedStartTime>2021-01-17T11:42:00Z</PublishedStartTime>
          <PublishedDuration>PT30M</PublishedDuration>
        </ScheduleEvent>
      </Schedule>
    </ProgramLocationTable>
  </ProgramDescription>
</TVAMain>
```

10.4 LCN Tables, Subscription Package and regionalisation

The DVB-I specification includes standardised LCN, regionalisation and Subscription Package features [3]. A DVB-HB Local Server should be able to publish a DVB-I Service List in an ordered way by using these features.

A broadcast Integrated Receiver Decoder (IRD) derives a selection of service instances, their order based on the transmitted logical channel descriptors, potentially bouquet selection or subscription level and potentially regionalisation information and associated rules. A DVB-HB Local Server should be able to extract this information from the broadcast transmission and to generate corresponding metadata.

NOTE: It is understood that supporting various rules, syntaxes, etc. may be felt as challenging by a manufacturer, although implemented in conventional IRDs. IRDs may implement subset of these features (e.g., only LCNs without regional descriptors, no simulcast processing, etc.).

With some exceptions, all these elements are defined outside DVB and there is no unique correspondence that can be defined in a specification. However, some elements are provided below on how to construct an ordered service list:

1) **Generate the `<service>` element**

   - Gather all services as a result of a scan of `SDTactual`/`SDTother`, private tables, private descriptors or other transmitted data.

   - Identify duplicate identical instances of the same service. These duplicates have identical audio/video coding and same editorial content at all times. Duplicates should be deleted or inserted as alternate `<ServiceInstance>` elements under the same `<Service>` element.

2) **Identify the Regions and the Subscription Packages**

   - In the input TS, according to ETSI EN 300 468 [15], regions can be defined as target regions, channel lists, bouquet sections, network sections. Hence, name can be extracted from target region descriptor, channel list names, bouquet names, multilingual bouquet names, network names, network identifiers.

   - Subscription Package may be defined in channel lists, bouquet sections or network sections.

3) **Merge Service Instances associated to the same editorial content into the same `<Service>` element**

   - Identify all instances of the same service and merge each instance as a `<ServiceInstance>`. There may be several non-identical instances of the same service, that always carry the same editorial content but differ in their audio/video coding or format (simulcast). Those instances use the same LCN, Simulcast Event Descriptor, a service replacement descriptor, the same or similar instance name. The priority (i.e., `@priority` attribute of the `<ServiceInstance>` element) is defined according to respective rules associated with the LCN syntaxes.

4) **Generate the `<LCNTable>` element**

   - If different LCN values result from the combination of a Subscription Package and region, generate an `LCNTable` for each pair of region and Subscription Package.

   - If different LCN values result from different Subscription Packages, generate an `LCNTable` for each Subscription Package.

   - If different LCN values result from different regions, generate an `LCNTable` for each region.

   - If different bouquets representing Subscription Package and different regions, generate an `LCNTable` for each pair of region and Subscription Package.

5) **Address services without LCN or outside a `service_list_descriptor`**

- For a given `LCNTable`, services without an LCN can be placed at the end of last assigned LCN or in a specific range.
- If a `service_list_descriptor` is present, services not present in the `service_list_descriptor` should be removed from the `LCNTable`.

Table 10 lists the association of some of the fields of the SI tables with the elements in the output DVB-I Service List.

Table 10: Association of Logical Channel Numbers distributed over DVB broadcast signalling with DVB-I Service List elements

DVB-I service discovery element	SI table and field
`RegionList.Region.RegionName` `LCNTable.TargetRegion` corresponding to respective `@regionId`	"Regions" as delivered via broadcast signalling (examples: postcodes, indexes, names, etc.)
`LCNTable.SubscriptionPackage`	Subscription Package as delivered via broadcast signalling
`LCNTable.LCN@channelNumber` with `@serviceRef` set to the `UniqueIdentifier` of the corresponding service in the Service List	Logical Channel Numbers distributed over broadcast signalling

In cases where it can be assumed that all DVB-HB Clients in the LAN are associated with the same region (i.e., the region where the DVB-HB Local Server is installed) and the same Subscription Package, a DVB-HB Local Server may simplify the generation and maintenance of the Service List by applying user selection of a single region and Subscription Package during the configuration phase of the device.

10.5 Mandatory elements of the DVB-I Service List in the absence of corresponding SI metadata or not specified in the received SI metadata

Not all elements in the DVB-I Service List can be directly derived from SI metadata. Table 11 provides suggestions on how a DVB-HB Local Server can complete the DVB-I Service List by including at least all mandatory elements according to ETSI TS 103 770 [3].

Table 11: Generation of DVB-I Service List mandatory elements

DVB-I service discovery element	Description
`ServiceList.Name`	Manufacturer's choice. It is the value shown on the client UI.
`ServiceList.ProviderName`	Manufacturer's choice. It may be shown on the client UI.
`ServiceList@version`	Integer value, defined by the DVB-HB Local Server, incremented at each change in the Service List
`Service.UniqueIdentifier`	Unique ID of the service. Refer to ETSI TS 103 770 [3], clause 5.2.2 for the suitable formats. For instance, the DVB-HB Local Server may construct it on the basis of on-air parameters (e.g., `tag:192.168.1.101,2020:dvb-s/318.5200.3401/service1`, i.e., using DVB triplet + service name). A DVB-HB Local Server relying on an external repository (see clause 5.4.12) may use the same `UniqueIdentifier` as used in public DVB-I Service Lists for the same service.
`Service.ServiceName`	See table 8
`Service.ProviderName`	See table 8
`Service@version`	Integer value, defined by the DVB-HB Local Server, incremented at each change in the Service definition
`ContentGuideSource.ProviderName`	Manufacturer's choice. It may be shown on the client UI.
`ContentGuideSource.ScheduleInfoEndpoint`	See clause 10.3
`LCNTableEntry@channelNumber`	See table 10
`LCNTableEntry@serviceRef`	See table 10
`DASHDeliveryParameters.UriBasedLocation`	The DVB-HB Local Server can autonomously define the URL associated to each service. For instance, the URL may be related to tuning parameters (e.g., similar format to RTSP commands in SATellite over Internet Protocol (SAT>IP®) [1]), or may be assigned according to different criteria.

11 Considerations on the use of HTTP and HTTPS in combination with browser-based DVB-HB Clients (informative)

11.1 Introduction

In case of Profile B, the interaction over the **M_B** reference point between the *Service discovery and selection* function of a DVB-HB Client and the *Service List publication* function of a DVB-HB Local Server (see also clause 5) is based on HTTP or HTTPS.

The present document supports the delivery by the DVB-HB Local Server of content and metadata on the LAN in a way that can be also handled by browser-based DVB-HB Clients using technologies such as HTML5 and JavaScript. Objects fetched via HTTP or HTTPS could be:

- The HTML5/JavaScript DVB-HB Client itself,
- The Service List Entry Point (see also clause 9.2),
- The Service List,
- DASH-delivered audio/video streams.

The use of HTTP and HTTPS in combination with browser-based DVB-HB Clients may have technical implications, addressed in the following.

11.2 Background

The acronym HTTPS is used to reference HTTP over Transport Layer Security (TLS). TLS is a presentation layer protocol (Layer 6 of the ISO/OSI model), designed to provide a secure channel between two communicating peers [i.3]. HTTP data is handled by TLS as agnostic application data payload [i.4], as TLS is designed to be protocol-independent [i.3].

The security of TLS is guaranteed by fulfilling the following communication properties:

- Authentication (of the server side and, optionally, of the client side),
- Confidentiality (i.e., data is visible only at the endpoints),
- Integrity (i.e., data cannot be modified by attackers).

The above properties are dependent on the parameters/messages exchanged during the TLS handshake phase, where the following tasks are performed in an attacker-resistant fashion:

- Authentication of communicating parties by means of Certificate Authority (CA) certificates issued by trusted authorities;
- Cryptographic modes negotiation (to encrypt transmitted data);
- Shared keying material transmission.

NOTE: The current version of TLS is 1.3, which supersedes and obsoletes previous versions by defining new procedures for TLS ticket mechanism, keys derivation and Online Certificate Status Protocol (OSCP) messages exchange. The present document implies the use of TLS 1.3 unless otherwise stated.

Given its level of security, HTTPS is now a requirement for most apps and web pages. Many HTTP Streaming technologies are moving to HTTPS content delivery to be compatible with HTTPS platforms/apps. The reason behind this is that almost the entirety of modern web browsers does not allow mixing of HTTP and HTTPS resources (as known as *Mixed content*, see clause 11.3.2) as this would expose the unencrypted, clear text HTTP content to sniffers and man-in-the middle attacks. Some browsers still allow mixed content only from the *localhost*, but most of them do not allow this practice at all.

From a DVB-HB standpoint, this represents a significant technical issue, because, on one hand, a successful and secure service delivery endorses the use of HTTPS, but, on the other hand, since 2015 CAs can no longer issue publicly-trusted certificates for any host name that cannot be publicly verified (i.e., internal names) and/or which contains a reserved IP address [i.5] (both these cases represent typical LAN and home networks scenarios). This change was deemed necessary to address some security vulnerabilities, including but not limited to:

- Non-uniqueness of internal names and, therefore, of the associated certificates,
- Non-verifiability of internal names from CAs,
- Address Resolution Protocol (ARP) spoofing,
- Dynamic Host Configuration Protocol (DHCP) spoofing,
- Domain Name Server (DNS) cache poisoning,
- Exposure to attacks for hosts with internal Fully Qualified Domain Name (FQDN).

11.3 Enforcing security

11.3.1 Reliability of communication in the LAN

According to ETSI TS 103 770 [3], any communication between DVB-I Service List Providers and DVB-I Clients is required to use HTTPS except for the specific case of a Service List that a client obtains from a server located on the same private subnet. In that specific case, HTTP may be used.

As a consequence, a DVB-HB Local Server, which is located in the same LAN as the DVB-HB Client, can be considered a secure source and therefore it is allowed to expose its metadata either over HTTP or HTTPS.

NOTE: A growing number of features in modern browsers Application Programming Interface (API) are migrating to secure-only origins (HTTPS) as HTTP is gradually being marked as non-secure. One of those features, particularly relevant for video consumption, is Encrypted Media Extensions (EME), which is involved in the playback of encrypted audio and video [i.6].

11.3.2 Combination of HTTP and HTTPS (“mixed content”)

A DVB-HB Client loaded as HTTP (e.g., retrieved from the DVB-HB Local Server, or from a public web server) is allowed to fetch resources (e.g., the Service List, or DASH-delivered audio/video streams) either as HTTP or HTTPS.

Instead, a browser-based DVB-HB Client loaded as HTTPS (e.g., from a public web server) is not allowed to fetch resources (e.g., the Service List, or DASH-delivered audio/video streams) from a DVB-HB Local Server as HTTP, but is forced to use HTTPS. This is known as *mixed content* issue and can be divided in two categories:

- *Mixed passive/display content*. Less harmful for the user. Passive contents include `<img>` and `<video>` elements/requests. User-agents can decide to handle this type of mixed content as *optionally-blockable* when the risk of allowing its usage is outweighed by the risk of breaking significant portions of the web (see also [i.7]).

- *Mixed active content*. It has access to the whole Document Object Model (DOM) of the HTTPS page and can modify its behaviour (potentially including malicious JavaScript code). Active content includes `<script>`, `<iframe>` and `XMLHttpRequest` elements/requests. This type of mixed content is usually blocked by all user-agents.

TLS support by the DVB-HB Local Server is the only way to overcome the mixed content issue.

NOTE: It is worth mentioning that also mobile operating systems are starting to mandate the use of HTTPS connections for their apps, hence this requirement will be relevant in the future also for app-based clients.

To fulfil this, DVB-HB Local Servers and DVB-HB Clients operating in the LAN should support the authentication mechanism of the communicating parties. This mechanism relies on the presence of a certificate issued by a trusted CA that is transmitted from the server to the client during the handshake procedure. However, for a DVB-HB Local Server, issuing a certificate would not be possible, as, according to the CA/Browser Forum BR [i.8]:

"The CA MUST confirm that the Applicant controls the Fully-Qualified Domain Name (FQDN) or IP address or has been granted the right to use it by the Domain Name Registrant or IP address assignee, as appropriate. Wildcard FQDNs are permitted.

As of the Effective Date of these Requirements, prior to the issuance of a Certificate with a subjectAlternativeName extension or Subject commonName field containing a Reserved IP Address or Internal Name, the CA SHALL notify the Applicant that the use of such Certificates has been deprecated by the CA / Browser Forum and that the practice will be eliminated by October 2016. Also as of the Effective Date, the CA SHALL NOT issue a certificate with an Expiry Date later than 1 November 2015 with a subjectAlternativeName extension or Subject commonName field containing a Reserved IP Address or Internal Name. Effective 1 October 2016, CAs SHALL revoke all unexpired Certificates whose subjectAlternativeName extension or Subject commonName field contains a Reserved IP Address or Internal Name."

In other words, the CAs are not allowed to issue publicly-trusted certificates to devices operating in the LAN.

Since obtaining a TLS server certificate for a DVB-HB Local Server has become a non-trivial task in recent years, in the next clauses the possible approaches to overcome this issue are suggested.

11.3.3 TLS certificates issued by the DVB-HB Local Server

This approach is also known as *self-signed certificates* and requires the DVB-HB Local Server to act as a private CA, possibly limiting this function to the home LAN environment and to the DVB-HB services. This approach, in fact, is typically used for closed groups and private services.

The implementation details of this solution are out of scope of the present document, but a plethora of options and open source solutions exist at the time of this writing.

NOTE: Even if the CA/Browser Forum does not prohibit the enterprise and private use of self-signed certificates, this approach does not provide the same level of security that a certificate signed by a publicly trusted CA is able to guarantee. This, therefore, may result in web browser warnings on browser-based clients, while native clients may be configured to allow TLS with self-signed certificates by default.

11.3.4 TLS intermediate certificates with DDNS

This approach is known to be used by some existing client-server media player systems. Despite the higher security level offered (i.e., it uses publicly-trusted CA intermediate certificates), this method is more complex than the previous one and needs some extra prerequisites to be matched in order to work properly:

- A new wildcard-capable and unique intermediate certificate to be issued for every DVB-HB Local Server by signing a commercial agreement with a valid CA;
- A DNS server supporting Distributed Domain Name Server (DDNS) hosted for DVB-HB purposes at a known public domain (e.g., `dvbhbdns.host`);
- A recognized hashing algorithm to provide each DVB-HB Local Server with a unique key (`HASH`).

The main steps for this approach are the following:

1) Set up a DDNS space on the DNS server.
2) Issue a wildcard-capable intermediate certificate for the DVB-HB Local Server during the initial setup. The intermediate certificate should have the address in the form: `*.HASH.DOMAIN` where:
   - `HASH` is the unique key, uniquely identifying the DVB-HB Local Server (e.g., it can be the `UniqueDeviceName` as defined in clause 7.2).
   - `DOMAIN` is the public domain of the DNS server (e.g., `dvbhbdns.host`).
3) At setup time, or every time the DVB-HB Local Server changes its local IP address, prompt a DDNS update request from the DVB-HB Local Server to the DNS server. The update should allow to resolve any `IP.HASH.DOMAIN` to the IP address, where:
   - `IP` is the server's private IP address. This address may change frequently depending on the local network conditions.
   - `HASH` is the unique key, uniquely identifying the DVB-HB Local Server.
   - `DOMAIN` is the public domain of the DNS server (e.g., `dvbhbdns.host`).

NOTE: In this approach `IP` and `HASH` have to be known by the user by following the DVB-HB Local Server discovery phase (see also clause 6).

An example of this implementation is represented in figure 6.

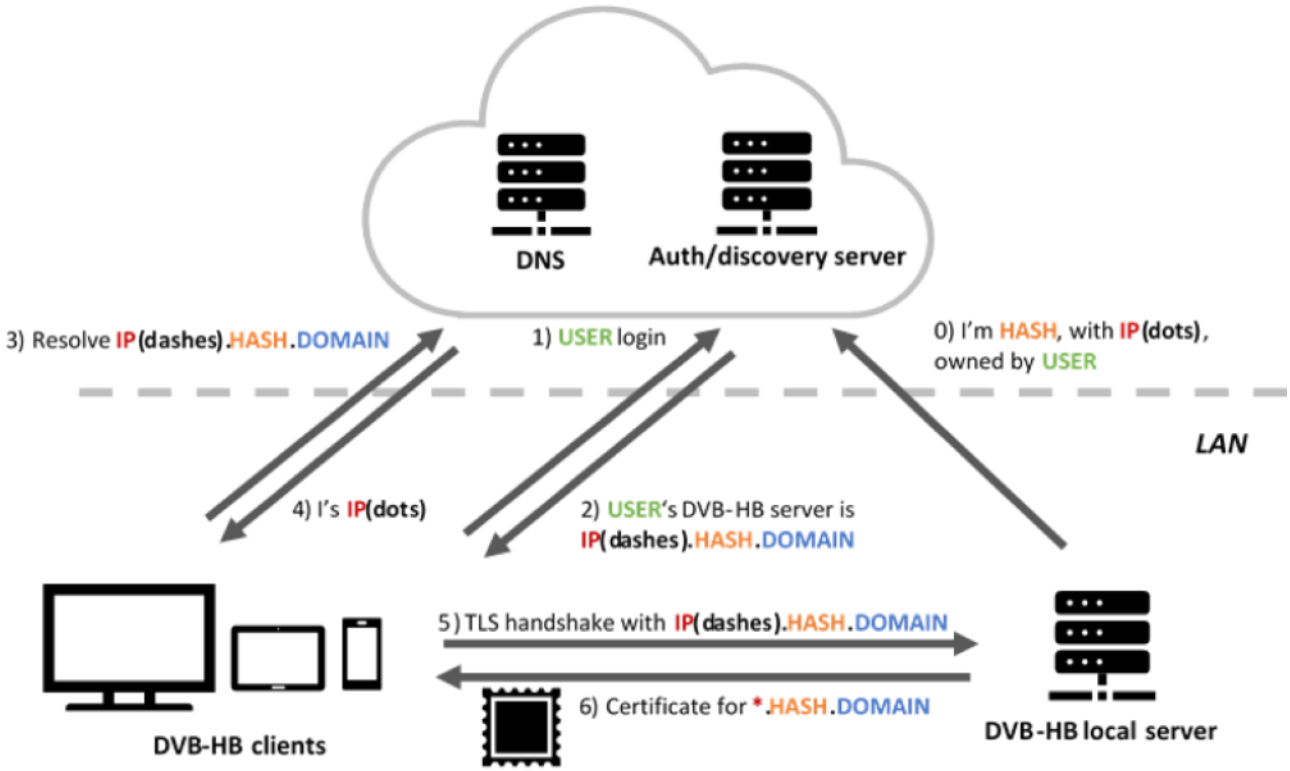

Figure 6: DVB-HB Local Server with TLS intermediate certificate

The advantages of this method are:

- The DNS setup allows to be redirected to any (i.e., public or private) IP address using HTTP over TLS.
- The wildcard certificate is valid independently of the IP address used (i.e., public or private).
- The `HASH` guarantees the uniqueness of the DVB-HB Local Server, which is of pivotal importance for the security of the authentication process.
- The `DOMAIN` can be any, therefore it should not be a problem to obtain the domain validation.

A drawback of this approach is that the DVB-HB Client has to know in advance that the DVB-HB Local Server uses a TLS intermediate certificate, so that it can start the process by querying the DDNS.

This approach is preferably coupled with a DVB-HB Local Server discovery mechanism based on a remote authentication server (see clause 6.3.4).

12 Guidelines on encoding and packaging (informative)

12.1 Introduction

12.1.1 General

The present clause attempts to provide some recommendations on how a DVB-HB Local Server can process the received audio and video bit streams to ensure interoperability with DVB-I Clients. DVB-HB Local Servers have limited resources, typically like any other consumer electronics products, i.e., processing power below PC CPUs, hardware accelerated decoding and, potentially, hardware accelerated encoding with one single video encoding capability. Audio encoding is more lightweight and might be performed as software encoding.

Elementary recommendations for packaging and/or encoding by a DVB-HB Local Server are provided. Basically, two options are possible:

- Simple repackaging as DASH without transcoding,

- Transcoding and packaging as DASH.

12.1.2 A note on support of mobile devices

ETSI TS 101 154 [11], annex L1 and annex L2.4 to L2.17 define *conformance points* for mobile devices. These are very close to conformance points for broadcast services. Whilst these conformance points may be supported in many mobile devices, some other references [i.11] recommend other parameters to ensure interoperability with a larger device population. Choosing these parameters does not preserve the original quality of the broadcast service (beyond the losses inferred by transcoding).

It is a DVB-HB Local Server manufacturer's choice to decide the strategy to choose. Ideally, both profiles should be offered – with and without transcoding to the recommended interoperability points.

12.2 Simple video repackaging without transcoding

12.2.1 Overview and limitations

Broadcast services that are encoded in Advanced Video Coding (AVC) or High Efficiency Video Coding (HEVC) can be repackaged without encoding since they are compliant with ETSI TS 101 154 [11], annex L1 and annex L2.4.

A DVB-HB Local Server can avoid transcoding and preserve the original broadcast quality by simply repackaging the TS video to demultiplexed video into ISO BMFF format segments according to ETSI TS 103 285 [2]. Demultiplexing, segmentation and remultiplexing in ISO BMFF are lightweight processes that can be implemented in software, together with the corresponding MPD, expected to be relatively static.

Benefit is that the original broadcast quality is preserved with low resources. Also, DVB-HB Local Servers not equipped with an encoder could offer at least some services in DVB-I compliant format. Another possibility is to use this method to serve some DVB-HB Clients with some services, saving encoding resources to serve other DVB-HB Clients that request services which cannot be repackaged without encoding, or are not supporting the interoperability points referred to as above.

The drawbacks are that repackaging keeps the same bit-rate as the incoming broadcast signal, which might be challenging for delivery within the home in some cases. Furthermore, the segment length is related to the Random Access Point (RAP) interval, which may lead to larger delays. Finally, received broadcast signals that are encoded in MPEG-2 are assumed not to be supported, as MPEG-2 video is not part of ETSI TS 101 154 [11], annex L. Also, simple repackaging leads to unaligned audio and video segments. The only solution to obtain aligned segments would be to re-encode audio and video with the same audio and video segment length multiple of the duration between RAPs and either a multiple of audio frames (if the audio codec uses audio frames) or a multiple of the duration between two samples (equal to reciprocal of sample frequency).

12.2.2 Parameters

12.2.2.1 Segment length

Broadcast services use a regular Group Of Pictures (GOP) structure with respect to RAP spacing, to enable fast channel switching. A GOP between two or more RAPs is suitable for segmentation. The maximal duration is rarely beyond 1-2 s, however there is no exact prediction or rule on the duration.

Table 12 shows a sample analysis conducted on terrestrial services and satellite services encoded in AVC (720p50 or 1080i50) during a 10 minutes observation time.

Table 12: Analysis of RAP spacing in broadcast services

Service	Video format	Seconds between RAPs
"Hessen Fernsehen HD"	720p50	0,02 - 0,78 (0,02 s steps)
"ARTE"	1080i50	0,04 - 1,28 (0,04 s steps)
"C8"	1080i50	0,12 - 1,44 (0,04 s steps)

Such a characteristic can be used to define segments which are in the order of 1 s. ETSI TS 103 285 [2] allows indeed a variation of 50% in segment length.

In the MPD a duration of 2 s can be used, which would allow a variation of up to 1 s. A DVB-HB Local Server can then group the GOP in a way that the segment stays close to 1 s.

12.3 Transcoding and packaging

12.3.1 General

A pragmatic option may be to systematically transcode and repackage the incoming broadcast signal to ensure interoperability. In this case, [i.11] may be a suitable option for transcoding parameters.

12.3.2 Parameters

12.3.2.1 Single or multiple representations

Optimisations exist for multiple bit-rate encoding [i.9],[i.10]; however, a single representation is considered as enough to ensure interoperability with DVB-I Clients.

Encoding and Signalling are defined in ETSI TS 101 154 [11], annex L and ETSI TS 103 285 [2].

12.3.2.2 Segment length

It is recommended that segment length corresponds to the duration of a GOP between two or more RAPs. This allows to encode separately segments and limit latency.

For example, in case of broadcast MPEG-2 Standard Definition (Video) (SD) services, a GOP is fixed with 12 pictures, which leads to segments of 0,48 s or 0,96 s.

Annex A (normative): Schemas

A.1 DVB-HB extensions

Pseudocode A.1: Schema of DVB-HB extensions

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:dvbisld="urn:dvb:metadata:servicelistdiscover
y:2020" xmlns:dvbhbx="urn:dvb:metadata:dvbhb-
extensions:2020" xmlns:dvbisd="urn:dvb:metadata:servicediscovery:2020" xmlns:mpeg7="urn:tva:mpeg7:20
08" xmlns:tva="urn:tva:metadata:2019" targetNamespace="urn:dvb:metadata:dvbhb-
extensions:2020" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <import namespace="urn:tva:metadata:2019" schemaLocation="tva_metadata_3-1.xsd"/>
  <import namespace="urn:tva:mpeg7:2008" schemaLocation="tva_mpeg7.xsd"/>
  <import namespace="urn:dvb:metadata:servicediscovery:2020" schemaLocation="dvbi_v2.0.xsd"/>
  <import namespace="urn:dvb:metadata:servicelistdiscovery:2020" schemaLocation="dvbi_service_list_d
iscovery_v1.1.xsd"/>
  <!-- Extension to ServiceListEntryPoints (for Profile B) -->
  <complexType name="HBxServiceListEntryPointsType" abstract="false">
    <annotation>
      <documentation xml:lang="en">DVB-
HB extension to "dvbisld:ServiceListEntryPointsType"</documentation>
    </annotation>
    <complexContent>
      <extension base="dvbisd:ExtensionBaseType">
        <sequence>
          <element name="HBLocalServerEntity" type="dvbhbx:HBLocalServerType" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="HBLocalServerType">
    <sequence>
      <element name="DeviceType" type="anyURI"/>
      <element name="UniqueDeviceName">
        <simpleType>
          <restriction base="string">
            <pattern value="uuid:[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}"/>
          </restriction>
        </simpleType>
      </element>
      <element name="UniversalProductCode" minOccurs="0">
        <simpleType>
          <restriction base="string">
            <pattern value="[0-9]{12}"/>
          </restriction>
        </simpleType>
      </element>
      <element name="ModelName" type="mpeg7:TextualType"/>
      <element name="FriendlyName" type="mpeg7:TextualType" maxOccurs="unbounded"/>
      <element name="Manufacturer" type="mpeg7:TextualType" maxOccurs="unbounded"/>
      <element name="ModelNumber" type="mpeg7:TextualType" minOccurs="0"/>
      <element name="SerialNumber" type="mpeg7:TextualType" minOccurs="0"/>
      <element name="ModelDescription" type="mpeg7:TextualType" minOccurs="0" maxOccurs="unbounded"/
>
      <element name="ManufacturerURL" type="anyURI" minOccurs="0"/>
      <element name="ModelURL" type="anyURI" minOccurs="0"/>
      <element name="Icon" type="tva:RelatedMaterialType" minOccurs="0" maxOccurs="unbounded"/>
      <element name="Availability" type="dvbhbx:HBServiceAvailabilityType" minOccurs="0"/>
    </sequence>
    <attribute name="specVersion" type="positiveInteger" use="required"/>
  </complexType>
  <!-- used for both Profiles A and B -->
  <complexType name="HBServiceAvailabilityType">
    <sequence>
      <element name="OrbitalPosition" type="dvbisd:LongitudeType" minOccurs="0" maxOccurs="unbounded
"/>
      <element name="ServiceAvailabilityMapIdleURL" type="dvbisd:ExtendedURIType"/>
      <element name="ServiceAvailabilityMapUpdateURL" type="dvbisd:ExtendedURIType" minOccurs="0"/>
    </sequence>
    <attribute name="version" type="positiveInteger" use="required"/>
  </complexType>
```

```
  <!-- Additions to desc.xml (for Profile A) -->
  <complexType name="X_SATIP_DVBHB">
    <sequence>
      <element name="ServiceListOffering" type="dvbisld:ServiceListOfferingType" minOccurs="0"/>
      <element name="Availability" type="dvbhbx:HBServiceAvailabilityType" minOccurs="0"/>
      <element name="AL-FEC" minOccurs="0">
        <simpleType>
          <restriction base="string">
            <enumeration value="none"/>
            <enumeration value="baselayer"/>
            <enumeration value="base+enhancementlayer"/>
          </restriction>
        </simpleType>
      </element>
    </sequence>
  </complexType>
  <complexType name="HBxDASHDeliveryParametersType" abstract="false">
    <annotation>
      <documentation xml:lang="en">DVB-
HB extension to dvbisd:DASHDeliveryParametersType</documentation>
    </annotation>
    <complexContent>
      <extension base="dvbisd:ExtensionBaseType">
        <sequence>
          <element name="OriginalDeliverySource" minOccurs="0">
            <simpleType>
              <restriction base="anyURI">
                <enumeration value="urn:dvb:metadata:source:dvb-t"/>
                <enumeration value="urn:dvb:metadata:source:dvb-s"/>
                <enumeration value="urn:dvb:metadata:source:dvb-c"/>
              </restriction>
            </simpleType>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

A.2 DVB-HB Service Availability Map

Pseudocode A.2: ServiceAvailabilityMap schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:dvbhbam="urn:dvb:metadata:dvbhb-
availabilitymap:2020" xmlns:dvbisd="urn:dvb:metadata:servicediscovery:2020" targetNamespace="urn:dvb
:metadata:dvbhb-
availabilitymap:2020" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <import namespace="urn:dvb:metadata:servicediscovery:2020" schemaLocation="dvbi_v2.0.xsd"/>
  <element name="ServiceAvailabilityMap" type="dvbhbam:ServiceAvailabilityMapType"/>
  <complexType name="ServiceAvailabilityMapType">
    <sequence>
      <element name="HBLocalServerNode" type="dvbhbam:HBLocalServerNodeType"/>
    </sequence>
    <attribute name="version" type="positiveInteger" use="required"/>
  </complexType>
  <complexType name="ServiceType">
    <attribute name="serviceRef" type="dvbisd:ServiceIdentifierType" use="required"/>
    <attribute name="used" type="nonNegativeInteger" use="required"/>
    <attribute name="transcodedUsed" type="nonNegativeInteger" use="required"/>
  </complexType>
  <simpleType name="supportedModulationSystemType">
    <restriction base="string">
      <enumeration value="DVB-C"/>
      <enumeration value="DVB-T"/>
      <enumeration value="DVB-T2"/>
      <enumeration value="DVB-S"/>
      <enumeration value="DVB-S2"/>
      <enumeration value="DVB-S2X"/>
    </restriction>
  </simpleType>
  <complexType name="DependencyResourceGroupType">
    <sequence>
      <element name="supportedModulationSystem" type="dvbhbam:supportedModulationSystemType" minOccu
rs="0" maxOccurs="unbounded"/>
      <choice>
```

```
        <element name="DependencyResourceGroup" type="dvbhbam:DependencyResourceGroupType" maxOccurs
="unbounded"/>
        <element name="Service" type="dvbhbam:ServiceType" maxOccurs="unbounded"/>
      </choice>
    </sequence>
    <attribute name="id" type="string" use="required"/>
    <attribute name="max" type="nonNegativeInteger" use="optional"/>
    <attribute name="used" type="nonNegativeInteger" use="required"/>
  </complexType>
  <complexType name="HBLocalServerNodeType">
    <sequence>
      <element name="supportedModulationSystem" type="dvbhbam:supportedModulationSystemType" minOccu
rs="0" maxOccurs="unbounded"/>
      <element name="DependencyResourceGroup" type="dvbhbam:DependencyResourceGroupType" maxOccurs="
unbounded"/>
    </sequence>
    <attribute name="shared" type="boolean" use="optional" default="true"/>
    <attribute name="totalServedClients" type="nonNegativeInteger" use="required"/>
    <attribute name="totalServedClientsMax" type="nonNegativeInteger" use="required"/>
    <attribute name="totalTranscodedClients" type="nonNegativeInteger" use=" required "/>
    <attribute name="totalTranscodedClientsMax" type="nonNegativeInteger" use=" required "/>
    <attribute name="totalTranscodedServicesMax" type="nonNegativeInteger" use="required"/>
  </complexType>
</schema>
```

Annex B (normative): Electronic attachments

The present document includes an electronic attachment with the following contents:

- dvbhb-extensions_v1.0.xsd - Extensions to SAT>IP® [1] and DVB-I [3] schemas for DVB-HB, as shown in clause A.1.
- dvbhb-availabilitymap_v1.0.xsd - Schema of Service Availability Map (see clause 7.3.3), as shown in clause A.2.
- dvbi_v2.0.xsd - DVB-I service list schema (imported into dvbhb-extensions_v1.0.xsd and dvbhb-availabilitymap_v1.0.xsd).
- dvbi_service_list_discovery_v1.1.xsd - DVB-I service list discovery schema (imported into dvbhb-extensions_v1.0.xsd).
- tva_metadata_3-1.xsd - TV-Anytime schema (imported into dvbhb-extensions_v1.0.xsd, dvbi_service_list_discovery_v1.0.xsd and dvbi_v2.0.xsd).
- tva_mpeg7.xsd - TV-Anytime profile of MPEG-7 schema (imported into dvbhb-extensions_v1.0.xsd, dvbi_service_list_discovery_v1.0.xsd and dvbi_v2.0.xsd).
- avail_map_first_example_update.xml - Example of a `ServiceAvailabilityMapUpdate` XML file (see clause 7.3.3).
- extended_ServiceListEntryPoint.xml - Example of `ServiceListEntryPoint` published by a DVB-HB Local Server (see clause 9.2).
- extended_DASH_service_instance.xml - Example of DASH service instance published by a DVB-HB Local Server (see clause 9.3).
- nownext.xml - Example of result of a query to the Content Guide server for present/following (see clause 10.3).

Annex C (informative): Examples of ServiceAvailabilityMap.xml for different use cases

C.1 One shared tuner

C.1.1 Exclusive assignment of the tuner resources

Pseudocode C.1 shows an example of a Service Availability Map describing one single tuner with exclusive transcoding, no sharing.

Pseudocode C.1: One single tuner with exclusive transcoding, no sharing

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="false" totalServedClients="1" totalServedClientsMax="1" totalTranscoded
Clients="1" totalTranscodedClientsMax="1" totalTranscodedServicesMax="1">
    <DependencyResourceGroup id="tuner" max="1" used="1">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="1" transcodedUsed="1"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

This is the simplest case where a DVB-HB Local Server assigns its single tuner to one single DVB-HB Client. Assuming here that client A is receiving *"Das Erste"* service, in `<HBLocalServerNode>` element, the value of `@totalServedClients` has reached the maximum value, so no further services can be assigned to other clients.

Since client A is using *"Das Erste"*, releasing this resource would set the `@used` attribute on the service to 0. As the `@used` attribute is set to 0, also the `@used` attribute of `<DependencyResourceGroup>` element above would be set to 0 and the `@totalServedClients` attribute would be set to 0. This means that a new client could now offer any of the services again.

Table C.1 shows the result in terms of available services in the situation of pseudocode C.1.

Table C.1: Available services (one single tuner with exclusive transcoding)

Client	Current service	*"Das Erste"*	*"ZDF"*	*"3SAT"*	*"RTL"*	*"SAT.1"*
A	*"Das Erste"*	**Y**	**Y**	**Y**	**Y**	**Y**
B	-	N	N	N	N	N
C	-	N	N	N	N	N
D	-	N	N	N	N	N
E	-	N	N	N	N	N
F	-	N	N	N	N	N
G	-	N	N	N	N	N
H	-	N	N	N	N	N

Pseudocode C.2 shows the same example in a case where transcoded is disabled.

Pseudocode C.2: One single tuner, no transcoding, no sharing

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="false" totalServedClients="1" totalServedClientsMax="1" totalTranscoded
Clients="0" totalTranscodedClientsMax="0" totalTranscodedServicesMax="0">
    <DependencyResourceGroup id="tuner" max="1" used="1">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="1" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
```

```
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

C.1.2 One tuner in shared mode with up to 3 clients served

C.1.2.1 Case 1: All clients on one service - single service per multiplex

Pseudocode C.3 shows an example of a Service Availability Map describing one shared single tuner used by three clients, with no transcoding.

Pseudocode C.3: One shared single tuner used by three clients with no transcoding

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="true" totalServedClients="3" totalServedClientsMax="50" totalTranscoded
Clients="0" totalTranscodedClientsMax="0" totalTranscodedServicesMax="0">
    <DependencyResourceGroup id="tuner" max="1" used="1">
      <DependencyResourceGroup id="mux1" max="1" used="1">
        <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="3" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux2" max="2" used="0">
        <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
        <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux3" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux4" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

Compared with the example in clause C.1.1, the tuner can be shared amongst the DVB-HB Clients on the LAN. Here it is relevant to create `<DependencyResourceGroup>` elements to describe multiplexes.

Assuming that clients A, B and C are on *"Das Erste"* belonging to *Mux1*, i.e., a multiplex carrying a single service, in this case any additional client can only access *"Das Erste"* service belonging to *Mux1*, so there is no benefit with respect to the example in clause C.1.1.

Table C.2 shows the result in terms of available services in the situation of pseudocode C.3.

Table C.2: Available services (one shared single tuner used by three clients, no transcoding)

Client	Current service	*"Das Erste"*	*"ZDF"*	*"3SAT"*	*"RTL"*	*"SAT.1"*
A	*"Das Erste"*	**Y**	N	N	N	N
B	*"Das Erste"*	**Y**	N	N	N	N
C	*"Das Erste"*	**Y**	N	N	N	N
D	-	**Y**	N	N	N	N
E	-	**Y**	N	N	N	N
F	-	**Y**	N	N	N	N
G	-	**Y**	N	N	N	N
H	-	**Y**	N	N	N	N

As only one tuner is available, other services can be accessed only after all clients A, B and C release the resource.

A variant with exclusive transcoding is shown in pseudocode C.4. Here there are three independent transcoders used by three clients; additional clients could access *"Das Erste"* but only in non-transcoded mode.

Pseudocode C.4: One shared single tuner used by three clients with exclusive transcoding

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="true" totalServedClients="3" totalServedClientsMax="50" totalTranscoded
Clients="3" totalTranscodedClientsMax="3" totalTranscodedServicesMax="3">
    <DependencyResourceGroup id="tuner" max="1" used="1">
      <DependencyResourceGroup id="mux1" max="1" used="1">
        <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="3" transcodedUsed="3"/>
      </DependencyResourceGroup>
```

```
      <DependencyResourceGroup id="mux2" max="2" used="0">
        <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
        <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux3" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux4" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

Finally, pseudocode C.5 shows the same example where two clients are using exclusive transcoding, while a third one is using in a shared mode. One additional client could access *"Das Erste"* in shared transcoded mode (Profile B).

Pseudocode C.5: One shared single tuner used by three clients with partially exclusive transcoding

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="true" totalServedClients="3" totalServedClientsMax="50" totalTranscoded
Clients="2" totalTranscodedClientsMax="3" totalTranscodedServicesMax="3">
    <DependencyResourceGroup id="tuner" max="1" used="1">
      <DependencyResourceGroup id="mux1" max="1" used="1">
        <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="3" transcodedUsed="2"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux2" max="2" used="0">
        <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
        <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux3" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux4" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

C.1.2.2 Case 2: All clients on one service - multiple services per multiplex

Pseudocode C.6 also shows an example of a Service Availability Map describing one shared tuner used by three clients, but, with respect to the example in clause C.1.2.1, the selected service is not the only one in the multiplex.

Pseudocode C.6: One shared tuner used by three clients, multiple services per multiplex

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="true" totalServedClients="3" totalServedClientsMax="4" totaltranscodedC
lients="0" totalTranscodedClientsMax="3" totalTranscodedServicesMax="3">
    <DependencyResourceGroup id="tuner" max="1" used="0">
      <DependencyResourceGroup id="mux1" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux2" max="2" used="1">
        <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="3" transcodedUsed="0"/>
        <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux3" max="2" used="0">
        <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux4" max="2" used="0">
        <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

Here, assuming that clients A, B and C are on *"ZDF"* belonging to *Mux2*, additional clients can access services also belonging to *Mux2*, i.e., *"ZDF"* or *"3SAT"* in this case.

In this example, transcoding is available but not used.

Table C.3 shows the result in terms of available services in the situation of pseudocode C.6.

Table C.3: Available services (one shared tuner used by three clients, multiple services per multiplex)

Client	Current service	*"Das Erste"*	*"ZDF"*	*"3SAT"*	*"RTL"*	*"SAT.1"*
A	*"ZDF"*	N	**Y**	**Y**	N	N
B	*"ZDF"*	N	**Y**	**Y**	N	N
C	*"ZDF"*	N	**Y**	**Y**	N	N
D	-	N	**Y**	**Y**	N	N
E	-	N	**Y**	**Y**	N	N
F	-	N	**Y**	**Y**	N	N
G	-	N	**Y**	**Y**	N	N
H	-	N	**Y**	**Y**	N	N

C.2 Two independent tuners

C.2.1 Exclusive assignment of the tuner resources

Pseudocode C.7 shows an example of a Service Availability Map describing two tuners with service exclusive assignment of the tuner, no transcoding enabled. In this representation, two `<DependencyResourceGroup>` elements are defined, one per tuner. Pseudocode C.8 shows the same example with an alternative equivalent representation, with only one `<DependencyResourceGroup>` element defined, merging the two tuners (i.e., `@max` attribute is set to 2).

Pseudocode C.7: Two tuners with exclusive assignment, no transcoding (first representation)

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="false" totalServedClients="2" totalServedClientsMax="2" totalTranscoded
Clients="0" totalTranscodedClientsMax="0" totalTranscodedServicesMax="0">
    <DependencyResourceGroup id="tuner1" max="1" used="1">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="1" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
    <DependencyResourceGroup id="tuner2" max="1" used="0">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

Pseudocode C.8: Two tuners with exclusive assignment, no transcoding (alternative representation)

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="false" totalServedClients="2" totalServedClientsMax="2" totalTranscoded
Clients="0" totalTranscodedClientsMax="0" totalTranscodedServicesMax="0">
    <DependencyResourceGroup id="tuner" max="2" used="1">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="1" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

In this example, the DVB-HB Local Server has two independent tuners which can receive all the services, e.g.:

- Two DVB-T/DVB-T2 tuners,
- Two DVB-S/DVB-S2/DVB-S2X tuners, each with a dedicated LNB feed,

- One DVB-T/DVB-T2 tuner and one DVB-S/DVB-S2/DVB-S2X tuner (assuming that the same services are delivered on terrestrial and satellite networks),
- Etc.

Each tuner can only select one service, irrespective of whether on the same or different multiplex. Transcoding is disabled.

Here client A is served with *"Das Erste"* on *tuner1* on an exclusive basis. There is still one tuner completely available, so other clients may use *tuner2* to access any service. If client A releases *"Das Erste"*, *tuner1* would be completely available again (i.e., `@used` attribute set to 0).

Table C.4 shows the result in terms of available services in the situation of pseudocode C.7.

Table C.4: Available services (two tuners with exclusive assignment)

Client	Current service	*"Das Erste"*	*"ZDF"*	*"3SAT"*	*"RTL"*	*"SAT.1"*
A	*"ZDF"*	**Y**	**Y**	**Y**	**Y**	**Y**
B	-	**Y**	**Y**	**Y**	**Y**	**Y**
C	-	**Y**	**Y**	**Y**	**Y**	**Y**
D	-	**Y**	**Y**	**Y**	**Y**	**Y**
E	-	**Y**	**Y**	**Y**	**Y**	**Y**
F	-	**Y**	**Y**	**Y**	**Y**	**Y**
G	-	**Y**	**Y**	**Y**	**Y**	**Y**
H	-	**Y**	**Y**	**Y**	**Y**	**Y**

C.2.2 Two independent tuners in shared mode receiving all services - one service at a time

C.2.2.1 Case 1: Spare resources on one service

Pseudocode C.9 shows an example of a Service Availability Map describing two tuners with sharing capabilities, in a case where spare resources on a service are still available.

Pseudocode C.9: Two tuners with sharing capabilities, spare resources on a service still available

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="true" totalServedClients="3" totalServedClientsMax="50" totalTranscoded
Clients="0" totalTranscodedClientsMax="0" totalTranscodedServicesMax="0">
    <DependencyResourceGroup id="tuner1" max="1" used="1">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="2" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
    <DependencyResourceGroup id="tuner2" max="1" used="1">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="1" transcodedUsed="0"/>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

The DVB-HB Local Server has two independent tuners which can receive all the services. Transcoding is disabled.

Each tuner can serve several clients (i.e., `@shared` attribute is set to `true`), but only one service per tuner at a time. One service may be served to several clients at the same time.

Here clients A and B are receiving *"Das Erste"* on *tuner1*, while client C is receiving *"SAT.1"* on *tuner2*. Any client can be served with *"Das Erste"* or *"SAT.1"* in shared mode. Moreover, client C can be served with any service, as *tuner2* is currently not shared with other clients.

Table C.5 shows the result in terms of available services in the situation of pseudocode C.9.

Table C.5: Available services (two tuners with sharing capabilities, spare resources still available)

Client	Current service	*"Das Erste"*	*"ZDF"*	*"3SAT"*	*"RTL"*	*"SAT.1"*
A	*"Das Erste"*	**Y**	N	N	N	**Y**
B	*"Das Erste"*	**Y**	N	N	N	**Y**
C	*"SAT.1"*	**Y**	**Y**	**Y**	**Y**	**Y**
D	-	**Y**	N	N	N	**Y**
E	-	**Y**	N	N	N	**Y**
F	-	**Y**	N	N	N	**Y**
G	-	**Y**	N	N	N	**Y**
H	-	**Y**	N	N	N	**Y**

C.2.2.2 Case 2: No spare exclusive transcoding resources on one service

Pseudocode C.10 shows an example of a Service Availability Map describing two tuners with sharing capabilities, in a case where no spare exclusive transcoding resources are still available.

Pseudocode C.10: Two tuners with sharing capabilities, no spare exclusive transcoding resources available

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="true" totalServedClients="3" totalServedClientsMax="50" totalTranscoded
Clients="3" totalTranscodedClientsMax="3" totalTranscodedServicesMax="3">
    <DependencyResourceGroup id="tuner1" max="1" used="1">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="2" transcodedUsed="2"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
    </DependencyResourceGroup>
    <DependencyResourceGroup id="tuner2" max="1" used="1">
      <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="1" transcodedUsed="1"/>
    </DependencyResourceGroup>
  </HBLocalServerNode>
</ServiceAvailabilityMap>
```

Compared with the example in clause C.2.2.1, three clients use exclusive transcoding, client A and client B for *"Das Erste"* service on *tuner1*, client C for *"SAT.1"* service on *tuner2*, so additional clients could offer *"SAT.1"* or *"Das Erste"* only in non-transcoded mode.

Table C.6 shows the result in the situation of pseudocode C.10 for non-transcoded available services, while table C.7 shows the result for transcoded available services.

Table C.6: Non-transcoded available services (two tuners with sharing capabilities, no exclusive transcoding spare resources available)

Client	Current service	*"Das Erste"*	*"ZDF"*	*"3SAT"*	*"RTL"*	*"SAT.1"*
A	*"Das Erste"*	**Y**	N	N	N	**Y**
B	*"Das Erste"*	**Y**	N	N	N	**Y**
C	*"SAT.1"*	**Y**	**Y**	**Y**	**Y**	**Y**
D	-	**Y**	N	N	N	**Y**
E	-	**Y**	N	N	N	**Y**
F	-	**Y**	N	N	N	**Y**
G	-	**Y**	N	N	N	**Y**
H	-	**Y**	N	N	N	**Y**

Table C.7: Transcoded available services (two tuners with sharing capabilities, no exclusive transcoding spare resources available)

Client	Current service	*"Das Erste"*	*"ZDF"*	*"3SAT"*	*"RTL"*	*"SAT.1"*
A	*"Das Erste"*	**Y**	N	N	N	**Y**
B	*"Das Erste"*	**Y**	N	N	N	**Y**
C	*"SAT.1"*	**Y**	**Y**	**Y**	**Y**	**Y**
D	-	N	N	N	N	N
E	-	N	N	N	N	N
F	-	N	N	N	N	N
G	-	N	N	N	N	N
H	-	N	N	N	N	N

C.2.3 Two independent tuners in shared mode receiving all services - multiple services per multiplex

Pseudocode C.11 shows an example of a Service Availability Map describing two tuners with sharing capabilities, which can serve multiple services if they are carried in the same multiplex. Transcoding is not available.

Pseudocode C.11: Two tuners with sharing capabilities, multiple services per multiplex

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="true" totalServedClients="2" totalServedClientsMax="50" totalTranscoded
Clients="0" totalTranscodedClientsMax="0" totalTranscodedServicesMax="0">
    <DependencyResourceGroup id="tuner1" max="1" used="1">
      <DependencyResourceGroup id="mux1" max="1" used="1">
        <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="1" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux2" max="2" used="0">
        <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
        <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
        <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
      <DependencyResourceGroup id="mux3" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
    <DependencyResourceGroup id="tuner2" max="1" used="1">
      <DependencyResourceGroup id="mux1" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:daserste.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux2" max="2" used="1">
        <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="1" transcodedUsed="0"/>
        <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
        <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
      <DependencyResourceGroup id="mux3" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
    </HBLocalServerNode>
</ServiceAvailabilityMap>
```

Table C.8 shows the result in terms of available services in the situation of pseudocode C.11.

Table C.8: Available services (two tuners with sharing capabilities, multiple services per multiplex)

Client	Current service	*"Das Erste"*	*"ZDF"*	*"3SAT"*	*"RTL"*	*"SAT.1"*
A	*"Das Erste"*	**Y**	**Y**	**Y**	**Y**	**Y**
B	*"ZDF"*	**Y**	**Y**	**Y**	**Y**	**Y**
C	-	**Y**	**Y**	**Y**	N	N
D	-	**Y**	**Y**	**Y**	N	N
E	-	**Y**	**Y**	**Y**	N	N
F	-	**Y**	**Y**	**Y**	N	N
G	-	**Y**	**Y**	**Y**	N	N
H	-	**Y**	**Y**	**Y**	N	N

C.2.4 Two tuners in shared mode with daisy-chained LNB

Pseudocode C.12 shows an example of a Service Availability Map describing a special case with multiple tuners connected to a daisy-chained LNB.

In this case, though several multiplexes can be received at the same time through the daisy chain, only one polarization and band can be received at a time, which means that only one quarter of the services are available at the same time (in the example, only multiplexes belonging to *polbamd1* can be served).

Pseudocode C.12: Two tuners in shared mode with daisy-chained LNB

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="true" totalServedClients="2" totalServedClientsMax="50" totalTranscoded
Clients="0" totalTranscodedClientsMax="0" totalTranscodedServicesMax="0">
    <DependencyResourceGroup id="LNB" max="1" used="1">
      <DependencyResourceGroup id="polband1" max="2" used="2">
      <DependencyResourceGroup id="mux1" max="1" used="1">
          <Service serviceRef="tag:ses2019;daserste.ses.com" used="3" transcodedUsed="0"/>
       </DependencyResourceGroup>
       <DependencyResourceGroup id="mux2" max="2" used="2">
          <Service serviceRef="tag:ses2019;zdf.ses.com" used="1" transcodedUsed="0"/>
          <Service serviceRef="tag:ses2019;3sat.ses.com" used="1" transcodedUsed="0"/>
       </DependencyResourceGroup>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="polband2" max="2" used="0">
        <DependencyResourceGroup id="mux3" max="1" used="0">
          <Service serviceRef="tag:ses2019;DMAX.ses.com" used="0" transcodedUsed="0"/>
        </DependencyResourceGroup>
        <DependencyResourceGroup id="mux4" max="1" used="0">
          <Service serviceRef="tag:ses2019;Tele5.ses.com" used="0" transcodedUsed="0"/>
        </DependencyResourceGroup>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="polband3" max="1" used="0">
        <DependencyResourceGroup id="mux5" max="1" used="0">
          <Service serviceRef="tag:ses2019;MainFrankenHD.ses.com" used="0" transcodedUsed="0"/>
        </DependencyResourceGroup>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="polband4" max="2" used="0">
        <DependencyResourceGroup id="mux6" max="1" used="0">
          <Service serviceRef="tag:ses2019;RTL.ses.com" max="3" used="0" transcodedUsed="0"/>
        </DependencyResourceGroup>
        <DependencyResourceGroup id="mux7" max="1" used="0">
          <Service serviceRef="tag:ses2019;SAT.1.ses.com" max="3" used="0" transcodedUsed="0"/>
        </DependencyResourceGroup>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
    </HBLocalServerNode>
</ServiceAvailabilityMap>
```

Table C.9 shows the result in terms of available services in the situation of pseudocode C.12.

Table C.9: Available services (two tuners in shared mode with daisy-chained LNB)

Client	Current service	*"Das Erste"*	*"ZDF"*	*"3SAT"*	*"DMAX"*	*"Tele5"*	*"MainFr."*	*"RTL"*	*"SAT.1"*
A	*"Das Erste"*	**Y**	**Y**	**Y**	N	N	N	N	N
B	*"Das Erste"*	**Y**	**Y**	**Y**	N	N	N	N	N
C	*"Das Erste"*	**Y**	**Y**	**Y**	N	N	N	N	N
D	*"ZDF"*	**Y**	**Y**	**Y**	N	N	N	N	N
E	*"3SAT"*	**Y**	**Y**	**Y**	N	N	N	N	N
F	-	**Y**	**Y**	**Y**	N	N	N	N	N
G	-	**Y**	**Y**	**Y**	N	N	N	N	N
H	-	**Y**	**Y**	**Y**	N	N	N	N	N

C.3 Four independent tuners

Pseudocode C.13 shows an example of a Service Availability Map describing four independent tuners in shared mode receiving all services, with multiple services per multiplex at a time.

Pseudocode C.13: Four independent tuners

```
<ServiceAvailabilityMap version="2005201628" >
  <HBLocalServerNode shared="true" totalServedClients="6" totalServedClientsMax="50" totalTranscoded
Clients="0" totalTranscodedClientsMax="0" totalTranscodedServicesMax="0">
    <DependencyResourceGroup id="tuner1" max="1" used="1">
      <DependencyResourceGroup id="mux1" max="3" used="1">
        <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="1" transcodedUsed="0"/>
        <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux2" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux3" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux4" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:hse24.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux5" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:tv5monde.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
    <DependencyResourceGroup id="tuner2" max="1" used="1">
      <DependencyResourceGroup id="mux1" max="3" used="0">
        <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
        <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux2" max="1" used="1">
        <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="2" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux3" max="1" used="1">
        <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux4" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:hse24.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux5" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:tv5monde.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
    <DependencyResourceGroup id="tuner3" max="1" used="1">
      <DependencyResourceGroup id="mux1" max="3" used="0">
        <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
        <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux2" max="1" used="1">
        <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux3" max="1" used="1">
        <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="2" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux4" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:hse24.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux5" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:tv5monde.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
    <DependencyResourceGroup id="tuner4" max="1" used="1">
      <DependencyResourceGroup id="mux1" max="3" used="0">
        <Service serviceRef="tag:ses.com,2019:zdf.ses.com" used="0" transcodedUsed="0"/>
        <Service serviceRef="tag:ses.com,2019:3sat.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux2" max="1" used="1">
        <Service serviceRef="tag:ses.com,2019:RTL.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux3" max="1" used="0">
         <Service serviceRef="tag:ses.com,2019:SAT.1.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux4" max="1" used="1">
        <Service serviceRef="tag:ses.com,2019:hse24.ses.com" used="1" transcodedUsed="0"/>
      </DependencyResourceGroup>
      <DependencyResourceGroup id="mux5" max="1" used="0">
        <Service serviceRef="tag:ses.com,2019:tv5monde.ses.com" used="0" transcodedUsed="0"/>
      </DependencyResourceGroup>
    </DependencyResourceGroup>
```

```
    </HBLocalServerNode>
</ServiceAvailabilityMap>
```

Table C.10 shows the result in terms of available services in the situation of pseudocode C.13.

Table C.10: Available services (four independent tuners)

Client	Current service	*"ZDF"*	*"3SAT"*	*"RTL"*	*"SAT.1"*	*"HSE24"*	*"TV5Monde"*
A	*"ZDF"*	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**
B	*"RTL"*	**Y**	**Y**	**Y**	**Y**	**Y**	N
C	*"RTL"*	**Y**	**Y**	**Y**	**Y**	**Y**	N
D	*"SAT.1""*	**Y**	**Y**	**Y**	**Y**	**Y**	N
E	*"SAT.1""*	**Y**	**Y**	**Y**	**Y**	**Y**	N
F	*"HSE24"*	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**
G	-	**Y**	**Y**	**Y**	**Y**	**Y**	N
H	-	**Y**	**Y**	**Y**	**Y**	**Y**	N

Annex D (informative): Provision of HbbTV applications to DVB-HB Clients

D.1 Use case

Hybrid Broadcast Broadband TeleVision (HbbTV®) applications can be delivered to clients either via IP over the Internet, or within the broadcast TS as a DSM-CC carousel.

The DSM-CC option may be preferred in some cases, e.g., for privacy reasons (i.e., the application asks permissions to establish communication with the broadcaster's server before accessing the Internet URL), or when targeting devices not connected to the Internet (i.e., the application provides some Teletext-like information for non-connected devices).

In the context of DVB-HB, the target device could be a TV/STB with HbbTV® and DVB-I capabilities, or a PC/Tablet/STB running an application implementing HbbTV® browsing and DVB-I.

D.2 Implementation in Profile A

In case of Profile A this is natively covered, i.e., the DVB-HB Client can request all necessary TS PIDs in its RTSP request, and will then take care of decoding the DSM-CC packets.

D.3 Implementation in Profile B

D.3.1 General

In case of Profile B, provision of HbbTV® applications to DVB-HB Clients can be achieved by leveraging specifications ETSI TS 103 770 [3] and ETSI TS 102 809 [16].

If the HbbTV® application is delivered via IP over the Internet, the DVB-HB Local Server has the task of signalling its public URL.

If the HbbTV® application is transported as a DSM-CC carousel, it is a task of the DVB-HB Local Server (namely, the *Content preparation* function in the DVB-HB reference architecture, see clause 5) to extract the original HbbTV® application (i.e., files with path) from the DSM-CC carousel carried in the incoming TS (this task is natively supported if the DVB-HB Local Server is implemented on an HbbTV®-enabled TV or STB), and to keep it up-to-date in case of variations in the data carousel, and to deliver it on the LAN to the DVB-HB Clients as an HTTP session.

D.3.2 Signalling

The original AIT, present in the incoming TS and associated in the Program Map Table (PMT) with the selected service, can be converted into an XML-based AIT by the DVB-HB Local Server, as defined in ETSI TS 102 809 [16], clause 5.4:

- If the HbbTV® application is delivered via IP over the Internet, the XML-based AIT references the public URL of the application files;
- If the HbbTV® application is transported as a DSM-CC carousel, the XML-based AIT references the local URL of the extracted application files, hosted on the DVB-HB Local Server (see also clause D.3.3).

In both cases, the generated XML-based AIT can be signalled in the DVB-I Service List published by the DVB-HB Local Server as a linked application as described in ETSI TS 103 770 [3]. An example is shown in pseudocode D.1.

Pseudocode D.1: Example of a service associated with a linked HbbTV application

```
<Service version="1">
  <UniqueIdentifier>tag:192.168.1.101,2020:Service1+HbbTV</UniqueIdentifier>
  <ServiceInstance>
    <DASHDeliveryParameters>
      <UriBasedLocation contentType="application/dash+xml">
        <URI>http://192.168.1.101/dvbhb/dash/Service1+HbbTV.mpd</URI>
      </UriBasedLocation>
    </DASHDeliveryParameters>
  </ServiceInstance>
  <ServiceName>Service1+HbbTV</ServiceName>
```

```
  <ProviderName>Broadcaster 1</ProviderName>
  <RelatedMaterial>
    <tva:HowRelated href="urn:dvb:metadata:cs:LinkedApplicationCS:2019:1.1"/>
    <tva:MediaLocator>
      <tva:MediaUri contentType="application/vnd.dvb.ait+xml">
        http://192.168.1.101/dvbhb/hbbtv/Service1+HbbTV.aitx
      </tva:MediaUri>
    </tva:MediaLocator>
  </RelatedMaterial>
</Service>
```

NOTE 1: This solution also supports dynamic AITs, provided that the DVB-HB Local Server either updates the DVB-I Service List coherently, or uses the signalling carried by a DVB-DASH `EventStream` as described in ETSI TS 103 770 [3], clause 5.2.3.3.

NOTE 2: This solution is applicable to applications that use relative URLs (i.e., relative to the first page of the application as signalled), while it is not applicable to applications that use dvb:URLs explicitly.

NOTE 3: If the DVB-HB Local Server has knowledge that the DVB-HB Clients do not have access to the Internet, it may include in the XML-based AIT only HbbTV® applications transported as a DSM-CC carousel, for a better Quality of Service (QoS).

D.3.3 Data delivery

If the HbbTV® application is transported as a DSM-CC carousel, the HbbTV® files, after being extracted by the DVB-HB Local Server, can be exposed by its HTTP server (i.e., the *Content publication* function in the DVB-HB reference architecture, see clause 5).

Such files can be retrieved on request (i.e., pulled) and decoded by a DVB-HB Client implementing the HbbTV® stack.

Annex E (informative): Background information on resilience in Wi-Fi networks

E.1 Introduction

TV services redistributed on the LAN by a Profile A DVB-HB Local Server using UDP-based transport protocol, which does not support packet retransmission, may suffer from network packet losses, especially in case of wireless communication.

Some studies exist about measuring the reliability of Wi-Fi networks based on the IEEE 802.11 [i.12] family. Some of them also consider video distribution in broadcast over Wi-Fi, where MAC level retransmissions are not applicable [i.13],[i.14].

The present document, in clause 8.4.2, specifies an optional AL-FEC mechanism, implemented on top of IEEE 802.11 [i.12] technologies, improving network resilience.

Here, an introduction of packet loss models and packet loss reduction techniques available in modern Wi-Fi implementations based on IEEE 802.11 [i.12] standards is provided for information.

NOTE: It is acknowledged that, at the time of writing, some amendments that significantly improve the IEEE 802.11 [i.12] physical layer and radio chains have already been defined (e.g., 802.11ax) or are under development by 802.11 task groups. Therefore, some of the possible issues presented in this document may not be applicable, or be applicable with a lesser degree, to newer Wi-Fi network implementations. Nevertheless, in order to be backwards compatible with older devices and current off-the-shelf devices, the content of this clause is still fully relevant in the DVB-HB scenario.

E.2 Packet loss models

While the first packet loss models that can be found in literature focused only on errors at the physical layer, in reality packets can be lost for many reasons, including issues at the data link layer (MAC).

Packet loss causes can be classified as follows [i.15]:

- Physical layer
  - Interference (hidden terminal, hidden node)
  - Co-existence (Wi-Fi, other wireless services)
  - Fading (multipath effect, weak signal, hand-off / hand-over)
- MAC layer
  - Collisions (competition medium, hidden terminals)
  - Buffer loss (insufficient memory, bufferbloat)

Collisions and buffer losses are, most likely, the main packet loss reasons in the DVB-HB scenario [i.13],[i.15].

Regarding the distribution of the losses, it is generally acknowledged that, for Wi-Fi networks, burst losses are the most common kind [i.15],[i.16]. For video streaming purposes, burst losses generally produce a larger total distortion than an equal number of isolated losses and, in [i.16], the authors proposed a model for estimating video distortion related to the specific loss pattern. The model is proved to be valid for general and complex loss patterns, including burst losses and separate, non-consecutive losses.

NOTE: Burst losses are less pronounced for high bit-rate video streams rather than for low bit-rates. The reason for this is because in high bit-rate video streams each frame is encoded in multiple packets, requiring therefore much longer bursts to have similar effects [i.17].

Generally speaking, Markov Chains with multiple states and variable length burst losses are the dominant strategy for modelling Wi-Fi packet loss [i.15],[i.18]. Two-states Markov/Gilbert-Elliot models are also very popular due to their simplicity [i.19].

E.3 Packet loss reduction techniques

E.3.1 Introduction

As discussed in clause E.2, packet losses can be quite frequent in wireless networks. In this clause some solutions that can improve network robustness and resilience in order to deliver video streams with the appropriate Quality of Experience (QoE)/QoS, specifically tailored to IEEE 802.11 [i.12] networks, are presented.

NOTE: Some of the solutions proposed in this clause may not be implemented in a standardised fashion or in Commercial Off-The-Shelf (COTS) wireless devices.

E.3.2 FEC/ARQ vs. AL-FEC

The Wi-Fi standards of the IEEE 802.11 [i.12] family inherently use FEC at the physical layer and Automatic Repeat reQuest (ARQ) at the link layer to reduce residual packet losses to a minimum [i.19] and prevent packet losses from being recovered by TCP, when used (i.e., losses that are not recovered by FEC/ARQ at link layer are interpreted by TCP as congestion) [i.13].

The FEC schemes used by IEEE 802.11 [i.12] amendments up to date are typically convolutional codes with short constraint lengths, Viterbi maximum-likelihood decoders and variable coding rates, depending on the Modulation and Coding Scheme (MCS) value used at the physical layer for the transmission [i.20].

While the use of FEC is essential for throughput and time-critical applications such as video streaming, the use of simple ARQ schemes might not always be feasible due to the strict time requirements.

In [i.21] the authors developed a model based on the assumption that the main packet loss causes are congestion and wireless channel losses and evaluated the performance gain in terms of Playable Frame Rate (PFR) when using different FEC overhead values (i.e., for standard IEEE 802.11 [i.12] FEC convolutional codes and ARQ with a given maximum number of retransmissions). Results show that the use of FEC with the right amount of overhead might be enough to guarantee a proper playback rate.

NOTE: Higher values of FEC overhead can stress the network and cause congestion.

In real-world IEEE 802.11 [i.12] networks, FEC overhead values have been found to vary between 5% and 45%. In applications where connectionless RTP/UDP streams are employed and retransmissions are not allowed, AL-FEC is a solution which is often recommended in video streaming applications over Wi-Fi [i.14].

E.3.3 Adaptive Cross-layer FEC

In traditional implementations, the FEC function is either implemented at bit-level in the physical layer (e.g., standard IEEE 802.11 [i.12] behaviour) or at the packet-level in the application layer (i.e., AL-FEC). The redundancy introduced by the FEC can be either static or dynamic (i.e., determined by the radio MCS parameters or controlled by the application-layer programs). When dynamic/adaptive AL-FEC is used, the redundancy rate is determined using the information coming from the lower Open Systems Interconnection (OSI) layers. This information, though, may not capture real-time network conditions with sufficient accuracy due to transmission delay and network congestion, resulting in an inefficient packets recovery for time-sensitive applications like video streaming.

In [i.19] the authors proposed a mechanism based on Adaptive Cross-layer Forward Error Correction (ACFEC) to be implemented in the Wi-Fi Access Point (AP). In this way, using the information provided by the AP, it is possible to monitor traffic flows and capture varying channel conditions in real-time, so that the source node can adapt the FEC rate accordingly. The proposed solution targets specifically blocks of video data packets and monitors their transmission status and any eventual failure using ARQ at the MAC layer, in cooperation with UDP protocol at the transport layer.

The solution has been tested using an encoded video sequence, encapsulated in RTP/UDP/IP packets and sent over the WLAN. Simulation results in a network with normal traffic conditions are provided, comparing the performance of ACFEC with static FEC, in terms of number of FEC packets and video Peak Signal to Noise Ratio (PSNR) as a function of the packet loss rate [i.19].

E.3.4 More efficient ARQ techniques

ARQ is a mechanism used to trigger retransmissions of lost packets and conduct data transfer in a reliable way. A retransmission can be triggered in two ways: upon request from the receiver and upon expired timeout of the timer from the sender [i.19].

As mentioned earlier, a traditional ARQ mechanism might not always be applicable for video streaming due to delay and timing constraints imposed by the decoding operations [i.19]. As a result, a consistent number of alternative ARQ schemes have been proposed during the years to target this use-case. The following is a non-exhaustive list of alternative and video related ARQ schemes:

Cross-Layer Perceptual ARQ
ARQ is performed at the application layer, with retransmission opportunities evaluated on a GOP-by-GOP basis, to exploit the information about the perceptual and temporal importance of each packet [i.22].

Soft ARQ
Mechanism based on legacy and delay-constrained ARQ scheme, leveraging layered encoding/transmissions and avoiding retransmitting late data that would not be useful at the decoder, with bandwidth saving. Tests of this solution in high packet loss conditions are reported in [i.23].

Conditional ARQ Retransmission and Low-Delay Interleaving
Hybrid FEC/ARQ scheme, leveraging selective retransmission and video packets interleaving scheme. Selective retransmission is performed using concealment error, a prediction of the distortion introduced by the missing packet and the channel conditions. The interleaving scheme, instead, is used to convert burst errors into correctable bit-errors [i.17].

E.3.5 Other solutions

Also the following solutions, among others, allow to improve resilience in Wi-Fi networks:

Accurate Packet dimensioning
In [i.24] the authors demonstrated the correlation between hidden traffic and packet losses by developing a new model. During their work, they proved that the packet airtime occupation is an important factor in the trade-off between protocol efficiency (i.e., payload time over total time) and the losses related to *hidden nodes* in the network. Therefore, they derived the optimal frame length to be used in case of *hidden nodes* in a closed form and backed up the results with network simulations.

Multipath Video Streaming
An emerging trend in high-quality video streaming is the use of multipath transmissions to tackle the limitations imposed by bandwidth, timing and packet loss constraints. It has been proved by multiple sources that multipath transmissions can lead to substantial performance gains in wireless networks with severe burst-loss rates [i.25]. Moreover, this approach inherently increases bandwidth, network resilience and load balancing of WLAN networks.

NOTE: Smartphones and other hand-held devices may have a single IEEE 802.11 [i.12] radio chain (i.e., 1x1) and, therefore, cannot benefit from a multipath approach.

Annex F (informative): Change History

Date	Revision	Information about changes
02/2021	A179	Initial version