Digital Video Broadcasting (DVB); Adaptive media streaming over IP multicast

DVB Document A176

March 2018

THIS IS A PROVISIONAL DVB DOCUMENT. IT MAY BE CHANGED BEFORE FINAL ADOPTION BY DVB.

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), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardisation of radio and television receivers. The EBU is a professional association of broadcasting organisations 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 are 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 standardisation, 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

Video delivery has become a dominant class of traffic on public networks. The wider market has embraced unicast streaming with the ability to adapt to network conditions as a means of delivering media on any type of access network. One of the reasons for its widespread adoption is the reuse of existing network technologies used to deliver other Internet services, in particular HTTP and Content Delivery Networks. Dynamic bit rate adaptation allows the streaming session to degrade gracefully as network conditions worsen, and to recover as they improve.

For consumption of the same linear media stream at the same time by a large audience, the number of simultaneous connections to the edge serving infrastructure carrying the same media payloads results in a high degree of redundancy which can be mitigated by the use of multicast packet replication at Layer 3. Unicast streaming is better suited to unsynchronised media consumption, or consumption of linear streams by smaller audiences.

By combining existing media encoding and packaging formats with the efficiency of point-to-multipoint distribution to the edge of IP-based access networks, it is possible to design a system for linear media distribution that is both efficient and scalable to very large audiences, while remaining technically compatible with the largest possible set of already-deployed end user equipment.

Point-to-multipoint topologies also offer opportunities for efficient pre-positioning of assets to devices at the edge of the network. This supports additional non-linear use cases and can help to alleviate peak demand on the access network at synchronisation points in the linear schedule.

1 Scope

This document specifies a reference functional architecture for an end-to-end system that delivers linear content over Internet Protocol (IP) networks in a scalable and standards-compliant manner. Scalability is achieved by means of IP multicast operating in parallel with and alongside conventional unicast delivery. The individual functions required for such a system are depicted in Figure 2 in clause 4.2, and the interactions between them are shown as named reference points. The functional architecture is intended as an abstract reference: real implementations may, for example, combine multiple functions in a single deployable unit. The architecture is intended to be independent of any particular Internet Protocol address family.

2 References

2.1 Normative references

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

[1] IETF RFC 7230: "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing".

[2] IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

[3] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests”.

[4] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Range Requests".

[5] IETF RFC 7234: "Hypertext Transfer Protocol (HTTP/1.1): Caching".

[6] IETF RFC 7235: "Hypertext Transfer Protocol (HTTP/1.1): Authentication".

[7] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2"

2.2 Informative references

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] IETF RFC 4918: "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)".

3 Definitions and abbreviations

3.1 Definitions

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

content: Any object involved in a streaming session, including the presentation manifest and packaged media. Each such object shall be identifiable by a Uniform Resource Identifier.

service description metadata: The object(s) used to describe the technical parameters of a single linear service. The service description metadata for a particular linear service includes references (such as URLs) to one or more presentation manifests.

NOTE: The format of the service description metadata and the means of its acquisition both lie outside the scope of the present specification.

presentation manifest: The metadata providing information used in the playback of a linear service.

NOTE: Examples include the playlists (.m3u8) for an HLS streaming session, the ISML file for a Microsoft SmoothStreaming session, the Media Presentation Description (MPD) for a DVB DASH session or the SDP file and MPEG-DASH MPD for a 3GPP MBMS session.

service configuration information: The multicast network and transport configuration describing one or more linear services that are available from the network.

NOTE: This information is consumed by the *Service management* subfunction of the *Multicast gateway*.

terminal device: The consumer device that renders the linear service.

3.2 Abbreviations

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

API	Application Programmer’s Interface
BMFF	Base Media File Format
CP	Content Provider
DNS	Domain Name System
DRM	Digital Rights Management
EPG	Electronic Programme Guide
IP	Internet Protocol
ISO	International Organization for Standardization
HLS	HTTP Live Streaming
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
MBMS	Multimedia Broadcast Multicast Services (pertaining to 3GPP)
MPD	Media Presentation Description (pertaining to MPEG-DASH)
MPEG	Moving Pictures Experts Group
MPEG-2 TS	MPEG-2 Transport Stream
MPEG-DASH	MPEG Dynamic Adaptive Streaming over HTTP
PES	Packetized Elementary Stream (pertaining to MPEG-2 Transport Stream)
PID	Packet Identifier (pertaining to MPEG-2 Transport Stream)
RTP	Real-time Transport Protocol
STB	Set-Top Box
URL	Uniform Resource Locator (pertaining to HTTP)

4 Reference architecture

4.1 Reference points

4.1.0 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 particular protocol. The reference points and the logical functions are illustrated by the reference architecture diagram in clause 4.2 and are summarised in the figure below.

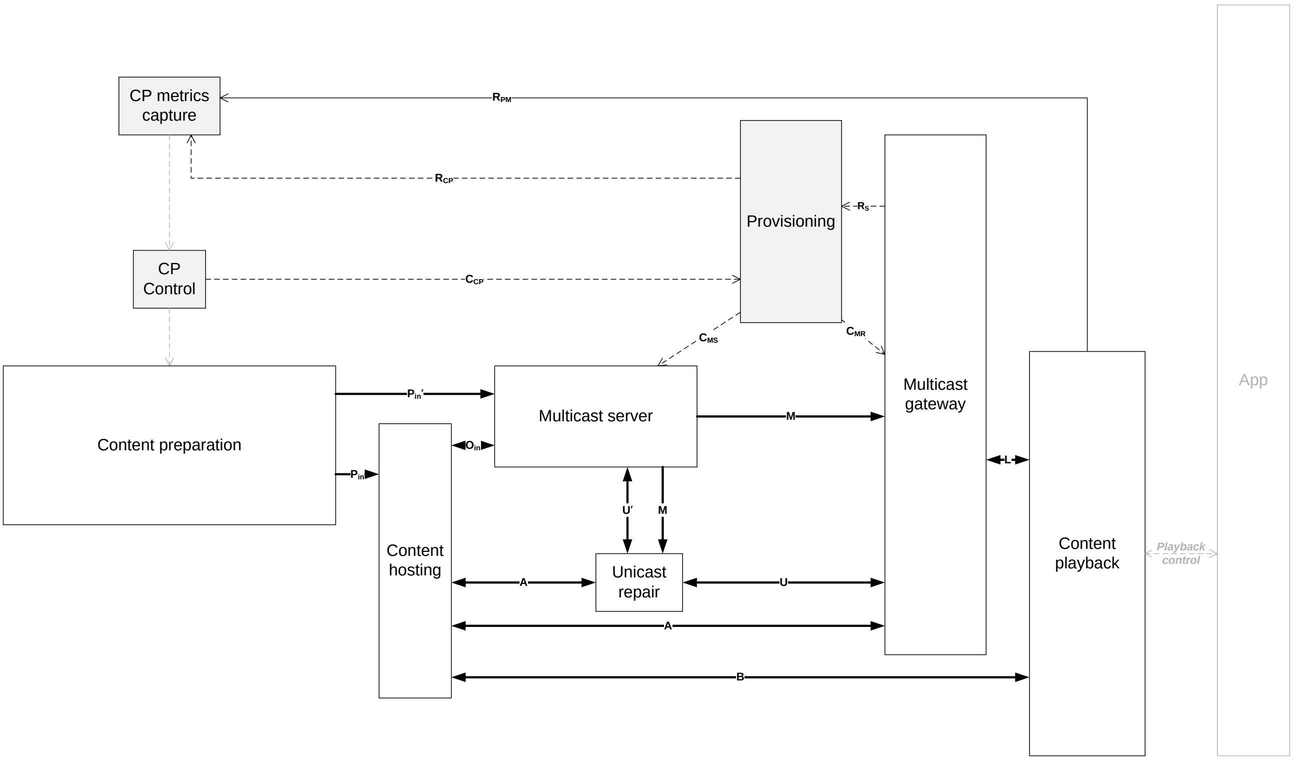

Figure 1: Simplified reference architecture

4.1.1 Data plane reference points

The reference points defined in this clause are used primarily to transport content.

L Unicast HTTP [1, 2, 3, 4, 5, 6] (and, later, HTTPS [7]) interaction between a *Content Playback* function and a *Multicast gateway*. This interface includes the fetching of all specified types of content.

NOTE: When a *Multicast gateway* and a *Content playback* function are co-located on a single end device, such as a set-top box (see clause 5.3), interface **L** may be realised as a local API.

B Bootstrap unicast HTTP(S) interaction directly between a *Content playback* function and a *Content hosting* function. This may be used to fetch presentation manifests and to retrieve content in the unmediated case.

A HTTP(S) acquisition of content not provided over interface **M**. Used in some deployments by a *Unicast repair service* function to retrieve content from a *Content hosting* function in order to effect content repair. Also used by a *Multicast gateway* for retrieving content directly from a *Content hosting* function via unicast when **U** is unable to perform a content repair.

M Multicast IP content transmission by a *Multicast server* function and reception by a *Multicast gateway* function and, in some deployments, reception by a *Unicast repair service* function.

U Unicast interaction between a *Unicast repair client* in a *Multicast gateway* and a *Unicast repair service*. This interface may be used to carry the payloads used for content repair functions in addition to the requests for such payloads.

U′ The unicast interaction between a *Unicast repair service* and a *Multicast server* as an alternative to fetching repair content over **A**. This interface may be used to carry the payloads used for content repair functions in addition to the requests for such payloads.

P$_{in}$ Publication of content to a *Content hosting* function by a *Content packaging* subfunction. This could be implemented as a push interface, or content could be pulled on demand from a *Content packaging* function.

O$_{in}$ Ingest of content by a *Multicast server* function from a *Content hosting* function. This is typically implemented as a pull interface.

P$_{in}$′ Ingest of content by a *Multicast server* direct from a *Content packaging* function. This is typically implemented as a push interface.

4.1.2 Control plane reference points

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

C$_{MS}$ Control interface for configuration of a *Multicast server* function.

C$_{MR}$ Control interface for configuration of a *Multicast gateway* function.

C$_{CP}$ Control interface for configuration of a *Provisioning* function.

R$_{S}$ Service reporting by a *Multicast gateway* function to a *Service reporting capture* function.

R$_{CP}$ Service reporting by a *Service reporting capture* subfunction to a *Content Provider metrics reporting capture* function.

R$_{PM}$ Reporting of playback metrics by a *Content playback* function to a *Content Provider metrics reporting capture* function.

4.2 Reference architecture diagram

A diagram of the reference architecture is shown on the next page (Figure 2). Logical functions are depicted as named boxes and these may be nested in cases where a high-level function is composed of several subfunctions. Functions that lie within the scope of this specification are shown with black text. Those beyond the scope of the specification (but relevant to the functional architecture) are depicted with grey text. Functions lying primarily in the data plane are shown with unfilled boxes; control plane functions are shaded.

Data plane interactions are depicted using solid lines. Control plane interactions are depicted using dotted lines. Interactions that lie within the scope of the specification are depicted as black lines with a reference point name. Those beyond the scope of the specification (but relevant to the functional architecture) are shown with grey lines.

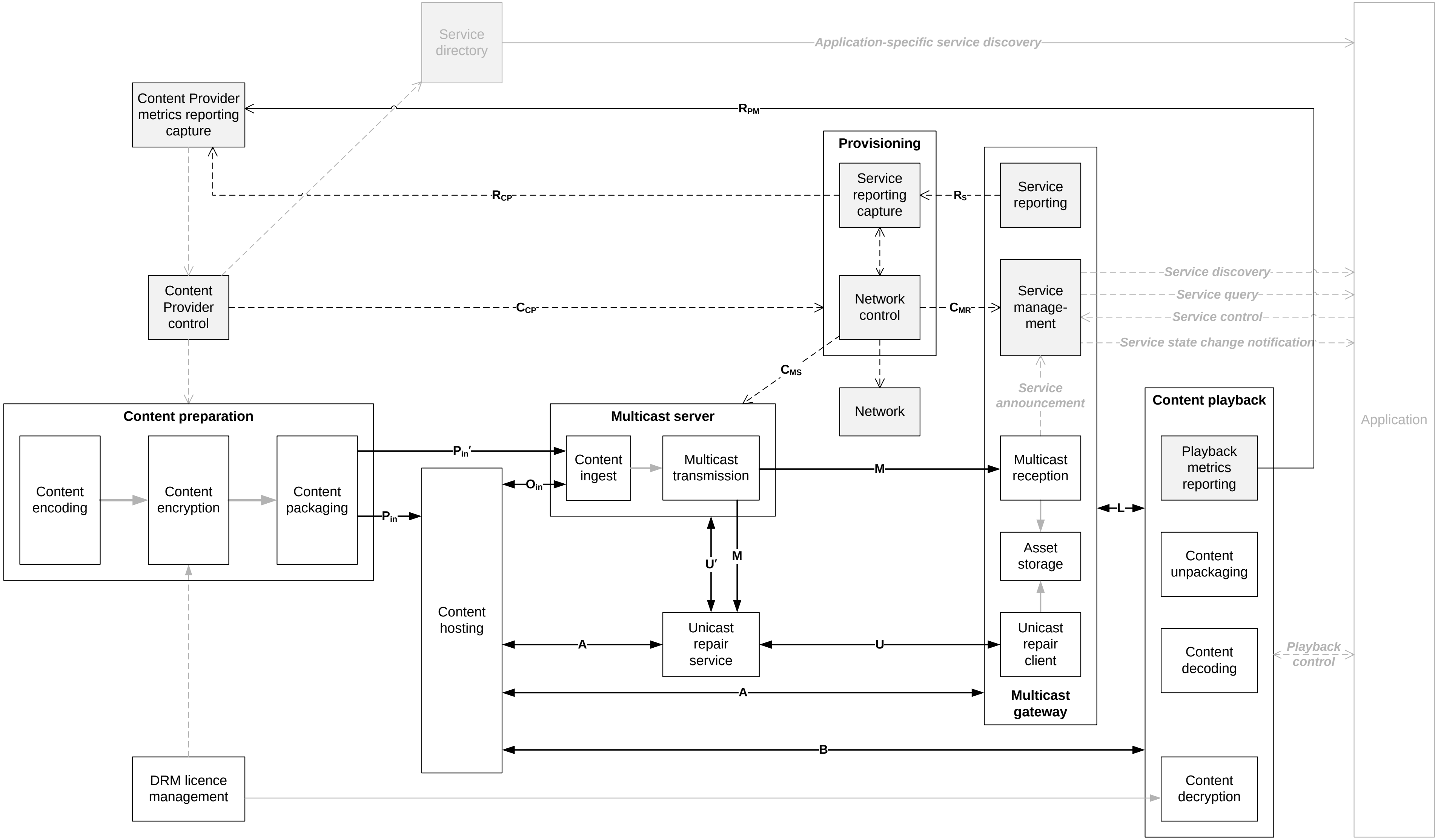

Figure 2: Reference architecture

4.3 Functions

4.3.1 Content preparation

4.3.1.1 Content encoding

The *Content encoding* function transforms source media streams into encoded media with the aim of reducing the bit rate. A single source media stream may be transformed into a number of different encoded representations to match delivery conditions. 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 encoder is a cleartext stream formatted so as to be suitable for ingest by an encrypter or packager. This could, for example, be an MPEG elementary stream, an MPEG-2 Transport Stream, or some other proprietary intermediate format.

4.3.1.2 Content encryption

The *Content encryption* function takes a cleartext stream and encrypts some or all of it to form a ciphertext stream. The encryption keys are obtained from the *DRM licence management* function.

This function is optional in the case where encryption is not a requirement for a particular stream.

4.3.1.3 Content packaging

The *Content packaging* function ingests the media streams of one (or more) encoded representations and formats each one according to a desired packaging format. 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.

Examples of packaging formats are ISO Base Media File Format (also known as fragmented MP4) and fragmented MPEG-2 Transport Stream.

4.3.2 Content hosting

Prepared content is made available by the *Content hosting* function for unicast delivery to the *Multicast server* (for content ingest via interface $\mathbf{O_{in}}$), to the *Multicast gateway* (for cache misses via interface **A)** to the *Unicast repair service* (also via interface **A**), or for instances of the *Content playback* function that are not connecting through a multicast receiver (interface **B**).

The *Content hosting* function may be realised as simple web servers, as part of an origin cluster, or operating as a distributed Content Delivery Network system. As such, load balancing and request distribution techniques (e.g. DNS round-robin, HTTP 302 redirect) may be used to direct clients to the most appropriate content server.

4.3.3 Multicast server

4.3.3.0 Introduction

The *Multicast server* function ingests content from the configured content sources. In the simple case, media streams are retrieved, typically using the same protocols that a media player might employ, via interface $\mathbf{O_{in}}$.

In the *Multicast server*, the payloads of the ingested media stream are encapsulated into the delivery units of the multicast transport protocol and transmitted through the *Network* to subscribed *Multicast gateway* clients using IP multicast via interface **M**.

This entity can be configured by the *Network control* function via interface $\mathbf{C_{MS}}$.

NOTE: This entity uses a Source-Specific Multicast methodology for sending and serving multicast traffic.

4.3.3.1 Content ingest

Both push and pull content ingest methods are possible for the *Multicast server*:

- **HTTP(S) Pull Ingest via interface O_{in}:** The subfunction mimics a conventional adaptive streaming media player and downloads packaged media segments from the *Content hosting* function as described by a presentation manifest. In this case, interface **O_{in}** may be functionally identical to **L** (although its operational behaviour may differ). Segments may, for example, be packaged using MPEG-DASH or HLS. Segments from one or more representations of the presentation may be downloaded simultaneously.

NOTE: DVB-DASH, MPEG-DASH, HLS and other manifest formats may be supported.

- **HTTP(S) Push Ingest via interface $P_{in}′$:** The subfunction offers an HTTP(S) push interface, such as WebDAV [i.1]. The *Content packaging* subfunction uploads media segments to the *Content ingest* function immediately as they are created. Segments may, for example, be packaged using MPEG-DASH or HLS.

- **RTP Push Ingest via interface $P_{in}′$:** The subfunction offers an RTP-based push ingest mechanism to the *Content packaging* subfunction. The packager sends MPEG-2 Transport Stream packets using RTP. Segment boundaries are marked with virtual segment boundary markers.

4.3.3.2 Multicast transmission

This subfunction is responsible for serialising streams received by the *Content ingest* subfunction and for transmitting the serialised streams in the payloads of IP multicast packets via interface **M**.

4.3.4 Unicast repair service

The *Unicast repair service* offers a payload repair function to the *Unicast repair client* in the *Multicast gateway* via reference point **U**.

The following repair modes could be considered for the *Unicast repair service*:

1. The *Unicast repair service* listens to multicast content transmissions over reference point **M** and locally caches a copy of the packet stream which it uses to satisfy repair requests received from the *Unicast repair client*.

2. If the requested packet(s) cannot be satisfied from the *Unicast repair service*’s cache, packet repair requests may be passed to the *Multicast server* via **U′**.

3. Packet repair requests are converted by the *Unicast repair service* to equivalent HTTP(S) requests (e.g. byte ranges) on the *Content hosting* function using an interface identical to reference point **A**.

4. If near-simultaneous requests for the same repair are received by the *Unicast repair service* from multiple *Multicast gateways*, it may be more efficient for the repair packets to be transmitted using multicast via reference point **M**.

4.3.5 Multicast gateway

4.3.5.0 Introduction

The primary purpose of this function is to provide packaged content segments to the *Content playback* function. The *Multicast gateway* may be realised as a forward proxy or as a local origin (including reverse proxy).

The *Multicast gateway* could be instantiated in customer premises equipment like a home gateway device or IP-connected set-top box (STB). It could also be located in an upstream network node as an alternative to the customer’s premises.

Content requests are received, via interface **L**, from one or more instances of the *Content playback* function and these are serviced either directly from content cached in the *Asset storage* subfunction or indirectly via interface **A**, with retrieved content then optionally cached in the *Asset storage* subfunction.

Unicast fill operations are performed via interface **A** until a cache is established in *Asset storage* for a given linear service.

NOTE: Some streams may never be sent using a multicast session, while others may require a short period of time before the cache is established.

4.3.5.1 Service management

The *Service management* subfunction collates service configuration information about multicast content streams available via interface **M** as well as the location(s) of the *Service reporting capture* function. This information may be received from one or more of the following sources:

- Directly from *Network control* via reference point **C$_{\textbf{MR}}$**.
- Indirectly via the *Multicast reception* subfunction, in the case where the information is transmitted over reference point **M**.
- In unicast responses from the *Content hosting* function carried over reference point **A**.

4.3.5.2 Multicast reception

The *Multicast reception* subfunction ingests content streams via interface **M** that have been requested by or configured for an end device. Content that has been received intact may also be cached in *Asset storage* for later use. Content damaged in transit may be repaired using any specified mechanism(s) at the *Multicast gateway*'s disposal (e.g. Forward Error Correction, unicast repair by the *Unicast repair client* via **U** or unicast retrieval via **A**) prior to caching. Irreparable content should not be served via reference point **L**.

4.3.5.3 Unicast repair client

Multicast packet loss detection is performed and recovered from using either Forward Error Correction information received via interface **M**, loss recovery using the *Unicast repair service* via interface **U** (e.g. unicast packet retransmission or multicast segment loss signalling) or, as a last resort, unicast fill via interface **A**.

4.3.5.4 Asset storage

The *Asset storage* subfunction provides temporary storage of assets to be served over interface **L**. Authority over the storage lies with the *Multicast gateway*.

- Managed pre-positioned media content assets. For example, pre-positioning all or part of a popular asset, or advertising material in advance of its availability date to a large population of users.
- Temporary caching of linear media content segments.

4.3.5.5 Service reporting

Service-related metrics (e.g. telemetry and analytics data) are reported by the *Service reporting* subfunction to the *Service reporting capture* subfunction via interface **R$_{\textbf{S}}$**.

4.3.6 Provisioning

4.3.6.0 Introduction

The purposes of the *Provisioning* function are:

1. To collect service reporting information centrally from the deployed *Multicast gateway* instances.
2. To configure resources in the *Network*.
3. To configure the *Multicast server* to use the configured *Network* resources.
4. To configure the *Multicast gateway* to use the configured *Network* resources.

The *Provisioning* function may be influenced by the *Content Provider control* function via reference point **C$_{\textbf{CP}}$**.

4.3.6.1 Service reporting capture

Service reporting information captured by the *Multicast gateway* is supplied to the *Service reporting capture* function via interface **R$_{\textbf{S}}$**. The reports may include metrics and other key indicators describing the performance of the service (e.g. cache hit ratio, viewership). The metrics depend on which channels are requested, when channels are established and how many segments are in cache. The service reporting information could be used for instance to improve service performance and to configure multicast channels.

The *Service reporting capture* function may also export service reporting information to the *Content Provider metrics reporting capture* function via reference point $\mathbf{R_{CP}}$. This information may include data on multicast content and bit rate.

4.3.6.2 Network control

This function is responsible for controlling, configuring and provisioning *Network* resources. This includes the resources for multicast transmission (over reference point **M**) and well as those for unicast operation (over reference points **U**, **A** and **B**).

In systems with centralised co-ordination, the *Network control* function distributes configuration information about the set of available multicast streams to the *Network* resources and may additionally distribute this configuration information to the *Multicast server* (via $\mathbf{C_{MS}}$) and/or to the *Multicast gateway* (via $\mathbf{C_{MR}}$). The configuration information about the set of available multicast streams can be updated according to *Content Provider control* policy rules and/or the number of client requests.

4.3.7 Content Provider control

This function uses the control interface $\mathbf{C_{CP}}$ provided by the *Network control* function to provision information about services that can be made available over the multicast delivery path **M**. A single *Content Provider control* function may be interacting with multiple *Network control* functions, each one of the latter operated by a different network provider.

4.3.8 Content playback

4.3.8.0 Introduction

This is the entity managing the request, reception, decryption and presentation of content. It only supports unicast delivery via reference point **L**. Playback behaviour is agnostic to the delivery path traversed by the content.

The *Content playback* function may be located separately from the *Multicast gateway* on an end device such as a smartphone (clauses 5.1 and 5.2). It may alternatively be combined with a *Multicast gateway*, for example in a set-top box or connected TV set (clause 5.3).

Additional functions of the *Content playback* function are:

- To retrieve, via interface **B**, a presentation manifest for the linear service.
- To retrieve, via interface **B**, any content that is not intended to be retrieved via the *Multicast gateway*.

4.3.8.1 Content unpackaging

The *Content unpackaging* subfunction is responsible for extracting elementary stream data from retrieved transport objects and presenting it to the *Content decryption* and *Content decoding* subfunctions. For example, with ISO Base Media File Format segments this involves extracting the appropriate media data box(es), while with MPEG-2 Transport Streams the desired PID is filtered and the payloads of reassembled PES packets are extracted.

4.3.8.2 Content decryption

If a Digital Rights Management system is in operation, the *Content decryption* subfunction is responsible for obtaining appropriate decryption keys from the *DRM licence management* function and for decrypting any encrypted elementary streams.

4.3.8.3 Content decoding

The *Content decoding* subfunction is responsible for parsing and interpreting the contents of elementary media streams, allowing them to be rendered for playback on, for example, a screen or loudspeakers.

4.3.8.4 Playback metrics reporting

The *Playback metrics reporting* subfunction reports information relating to the behaviour and quality of experience of content playback to the *Content Provider metrics reporting capture* function via reference point $\mathbf{R_{PM}}$. The metrics may include (but are not limited to) details of HTTP request/response transactions, initial playback delay, buffer level, representation switching events and measured network throughput. The playback metrics reported by this function are directly related to the end user quality of experience and may be used to optimise the experience either at the Content Provider or in the *Network*.

4.3.9 DRM licence management

This optional entity is responsible for core content protection services, providing appropriate encryption keys to be used by the *Content encryption* function, and supplying licences to the *Content decryption* subfunction to enable the *Content playback* function to decrypt encrypted content.

4.3.10 Application

The *Application* is responsible for controlling the *Content playback* function. Examples include an embedded control application on an integrated television set or set-top box ("EPG application") or a third-party application contributed by a Content Provider. The interface the *Application* function uses to control the *Content playback* function is outside the scope of the present specification, but would generally involve passing a reference to a presentation manifest (e.g. the URL of an MPEG-DASH MPD) to initiate playback of a particular linear service.

The *Application* may interact with the *Service management* subfunction of the *Multicast gateway* in order to discover the existence of linear services and control their reception by the *Multicast gateway*. At the present time, these interactions are also out of scope.

The *Application* may alternatively discover the existence of linear services through a private interaction with an application-specific *Service directory* function. This interaction is also outside the scope of the present specification.

4.3.11 Service directory

The *Application* may use a private *Service directory* in order to locate available linear services. The *Service directory* function may be configured by the *Content provider control* function. Because the *Service directory* function is application-specific it is outside the scope of the present specification.

5 Deployment models

5.0 Introduction

The reference architecture described in clause 4 above is intended to support the deployment models described in this clause.

5.1 Multicast gateway deployed in network edge device

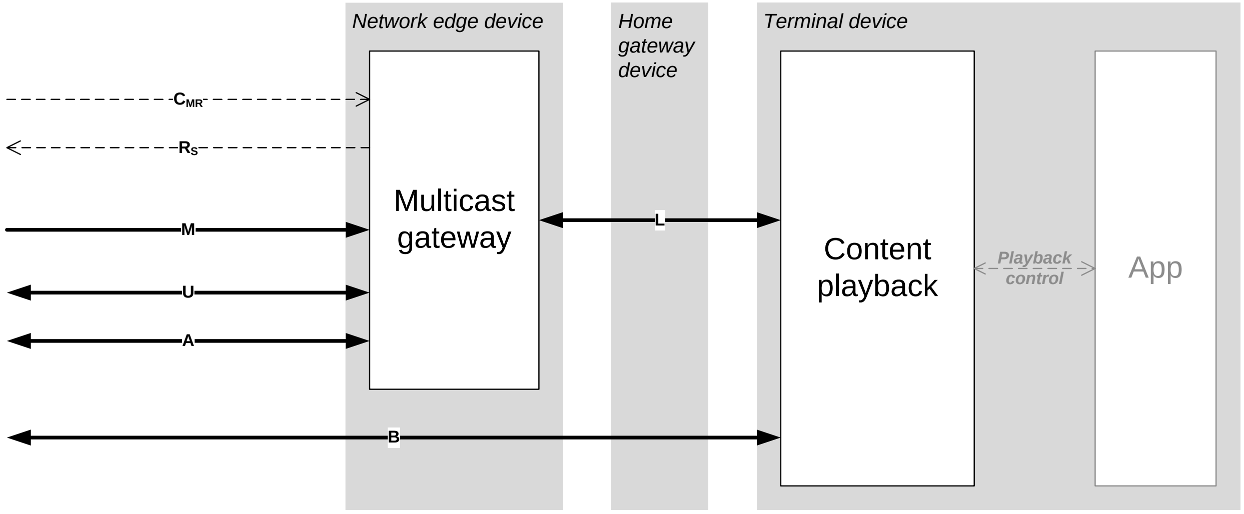

Figure 3

The terminal device does not support reception of IP multicast from the home network. It includes the *Content playback* function, and loads an *Application* to control linear playback.

The *Multicast gateway* is deployed in a network edge device upstream of the terminal device and provides multicast-to-unicast conversion facilities for several homes. All in-scope traffic on the access network between the network edge device and the home gateway device is therefore unicast.

5.2 Multicast gateway deployed in home gateway device

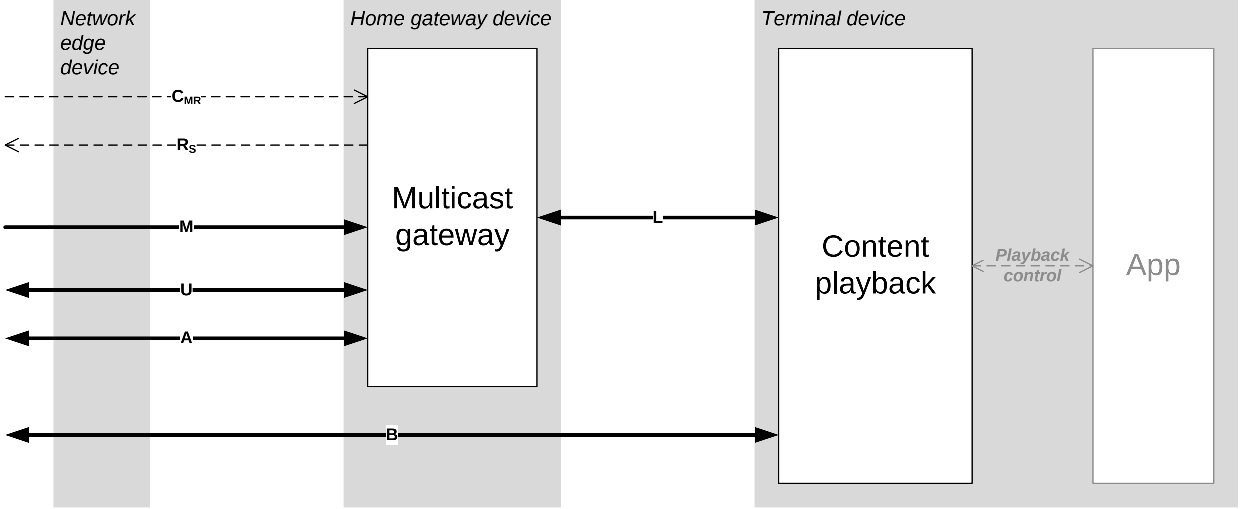

Figure 4

The *Multicast gateway* is deployed in a home gateway device, such as a router (typically supplied by the Internet Service Provider) and provides multicast-to-unicast conversion facilities for multiple terminal devices in the same home network. These terminal devices each have an instance of the *Content playback* function and load the desired *Application*.

5.3 Multicast gateway deployed in terminal device

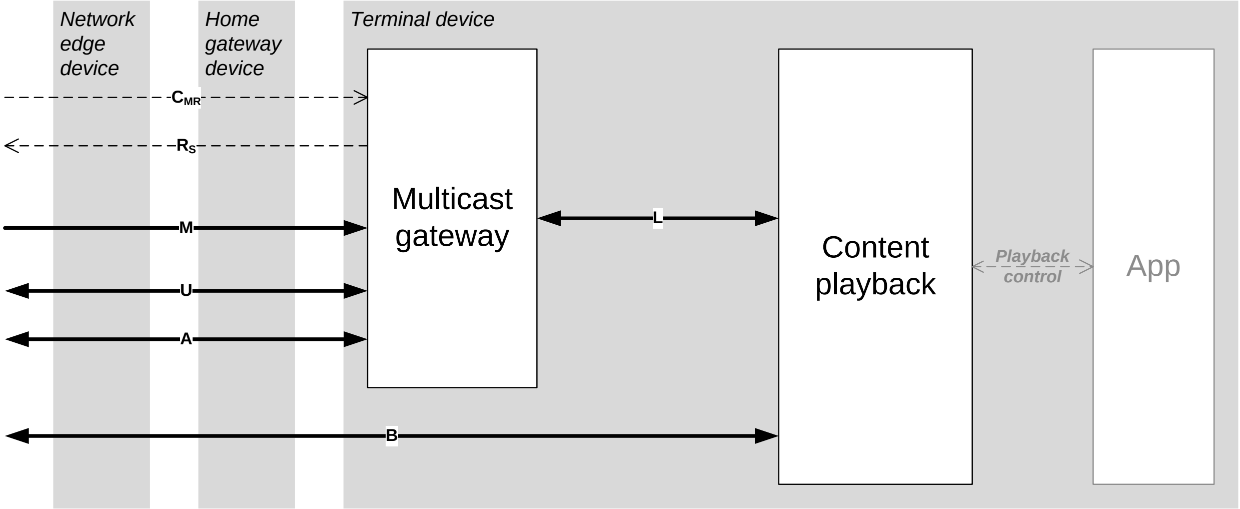

Figure 5

The terminal device supports reception of IP multicast within the home network. Each such terminal device includes both *Multicast gateway* and *Content playback* functions, and loads its own *Application* to control linear playback. In this deployment model, the *Multicast Gateway* function shall provide content services only to its host terminal device.

The home gateway device performs only multicast group subscription operations. (This may result in unpredictable quality behaviour when the home network does not fully support multicast delivery.)

Annex A (informative): TBD

History

Document history		
V0.0.1	February 2018	First pre-publication as a DVB Blue Book.