ETSI TS 126 247 V14.5.0 (2018-04)

TECHNICAL SPECIFICATION

Universal Mobile Telecommunications System (UMTS); LTE; Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) (3GPP TS 26.247 version 14.5.0 Release 14)

Reference

RTS/TSGS-0426247ve50

Keywords

LTE,UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or
print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any
existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the
print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

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

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables 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 ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

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.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

The 3GPP transparent end-to-end packet-switched streaming service (PSS) specification consists of seven 3GPP TSs: 3GPP TS 22.233 [1], 3GPP TS 26.233 [2], 3GPP TS 26.234 [3], 3GPP TS 26.244 [4], 3GPP TS 26.245 [5], 3GPP TS 26.246 [6], and the present document.

The TS 22.233 contains the service requirements for the PSS. The TS 26.233 provides an overview of the PSS. The TS 26.234 provides the details of the protocols and codecs used by the PSS. The TS 26.244 defines the 3GPP file format (3GP) used by the PSS and MMS services. The TS 26.245 defines the Timed text format used by the PSS and MMS services. The TS 26.246 defines the 3GPP SMIL language profile. The present document defines Progressive Download and Dynamic Adaptive Streaming over HTTP.

The TS 26.244, TS 26.245 and TS 26.246 start with Release 6. Earlier releases of the 3GPP file format, the Timed text format and the 3GPP SMIL language profile can be found in TS 26.234.

The TS 26.247 starts with Release 10. Earlier releases of Progressive Download and Dynamic Adaptive Streaming over HTTP can be found in TS 26.234.

Introduction

Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) collects a set of technologies how progressive download and adaptive streaming of continuous media may be carried out exclusively over HTTP.

1 Scope

The present document specifies Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH). This specification is part of Packet-switched Streaming Service (PSS). HTTP-based progressive download and dynamic adaptive streaming are separated from TS 26.234 to differentiate from RTP-based streaming that is maintained in TS 26.234. HTTP-based progressive download and dynamic adaptive streaming may be deployed independently from RTP-based PSS, for example by using standard HTTP/1.1 servers for hosting data formatted as defined in the present document.

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1]   3GPP TS 22.233: "Transparent End-to-End Packet-switched Streaming Service; Stage 1".

[2]   3GPP TS 26.233: "Transparent end-to-end Packet-switched Streaming service (PSS); General description".

[3]   3GPP TS 26.234: "Transparent end-to-end packet switched streaming service (PSS); Protocols and codecs".

[4]   3GPP TS 26.244: "Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)".

[5]   3GPP TS 26.245: "Transparent end-to-end packet switched streaming service (PSS); Timed text format".

[6]   3GPP TS 26.246: "Transparent end-to-end packet switched streaming service (PSS); 3GPP SMIL Language Profile".

[7]   3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[8]   IETF STD 0007: "Transmission Control Protocol", Postel J., September 1981.

[9]   IETF RFC 2616: "Hypertext Transfer Protocol – HTTP/1.1", Fielding R. et al., June 1999.

[10]   Open Mobile Alliance, Service and Content Protection for Mobile Broadcast Services, Approved Version 1.0, February 2009.

[11]   ISO/IEC 14496-12:2012 | 15444-12:2012 "Information technology - Coding of audio-visual objects - Part 12: ISO base media file format" | "Information technology - JPEG 2000 image coding system - Part 12: ISO base media file format".

[12]   IETF RFC 2818: "HTTP Over TLS", E. Rescorla, May 2000.

[13]   IETF RFC 5646: "Tags for Identifying Languages", A. Phillips, M. Davis, September 2009.

[14]   (void)

[15]   Open Mobile Alliance: "DRM Content Format V 2.0".

[16]   Open Mobile Alliance: "DRM Content Format V 2.1".

[17]   IETF RFC 3986: "Uniform Resource Identifiers (URI): Generic Syntax", Berners-Lee T., Fielding R. and Masinter L., January 2005.

[18]   IETF RFC 1952: "GZIP file format specification" version 4.3, P. Deutsch, May 1996.

[19]   IETF RFC 1738: "Uniform Resource Locators (URL)", December 1994.

[20]   (void)

[21]   (void)

[22]   OMA-ERELD-DM-V1_2-20070209-A: "Enabler Release Definition for OMA Device +Management, Approved Version 1.2"

[23]   3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".

[24]   IETF RFC 2045: "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".

[25]   IETF RFC 2231: "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations".

[26]   IETF RFC 6381: "The 'Codecs' and 'Profiles' Parameters for "Bucket" Media Types," August 2011.

[27]   Void.

[28]   IEEE 1003.1-2008, IEEE Standard for Information Technology - Portable Operating System Interface (POSIX), Base Specifications, Issue 7

[29]   IETF RFC 4337, "MIME Type Registration for MPEG-4," March 2006

[30]   IETF RFC 3023, "XML Media Types," January 2001.

[31]   3GPP TS 23.203: "Policy and charging control architecture".

[32]   3GPP TS 29.213: "Policy and Charging Control signalling flows and Quality of Service (QoS) parameter mapping".

[33]   3GPP TS 29.214: "Policy and Charging Control over Rx reference point".

[34]   void

[35]   ITU-T Recommendation H.264 (04/2013): "Advanced video coding for generic audiovisual services".

[36]   3GPP TR 26.946: "Multimedia Broadcast/Multicast Service (MBMS); User service guidelines".

[37]   IETF RFC 3629: "UTF-8, a transformation format of ISO 10646," November 2003.

[38]   IETF RFC 4288: "Media Type Specifications and Registration Procedures," December 2005.

[39]   IETF RFC 4648: "The Base16, Base32, and Base64 Data Encodings," October 2006.

[40]   IETF RFC 5234: "Augmented BNF for Syntax Specifications: ABNF", Crocker D. and Overell P., January 2008.

[41]   3GPP TR 26.905: "Mobile stereoscopic 3D video".

[42]   3GPP TS 26.346: " Multimedia Broadcast/Multicast Service (MBMS);Protocols and codecs"

[43]   ISO/IEC 23009-1:2014/Amd. 1:2015/Cor.1:2015" Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats".

[44]   ISO/IEC 23009-3 "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 3: Implementation and Deployment Guidelines".

[45]   ISO/IEC 23009-2 " Information technology -- Dynamic adaptive streaming over HTTP (DASH) --
            Part 2: Conformance and Reference Software".

[46]   3GPP TR 26.938: "Packet-switched Streaming Service (PSS); Improved support for dynamic
            adaptive streaming over HTTP in 3GPP".

[47]   ISO/IEC 23001-7:2015: "Information technology -- MPEG systems technologies -- Part 7:
            Common encryption in ISO base media file format files".

[48]   IETF RFC 7164, "RTP and Leap Seconds", March 2014.

[49]   ITU-T P.1203 (11/2016), "Parametric bitstream-based quality assessment of progressive download
            and adaptive audiovisual streaming services over reliable transport"

[50]   ITU-T P.910 (04/2008), "Subjective video quality assessment methods for multimedia
            applications"

[51]   "Mobile Location Protocol (MLP) ", Open Mobile Alliance, OMA-LIF-MLP-V3_1, Approved
            Version 3.1 – 20 Sep 2011.

[52]   IEEE 1003.1-2008 "IEEE Standard for Information Technology - Portable Operating System
            Interface (POSIX(R))"

[53]   3GPP TS 25.331 "Radio Resource Control (RRC); Protocol specification"

[54]   3GPP TS 26.116: "Television (TV) over 3GPP services; Video profiles".

3   Definitions, abbreviations and conventions

3.1   Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [7], in ISO/IEC 23009-1 [43], section 3.1, and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [7].

frame-packed stereoscopic 3D video: a video consisting of two views in which both views were packed into a single stream before compression.

multiview stereoscopic 3D video: a video consisting of two views packed into a single stream during compression.

3.2   Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [7] and the following apply.
An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [7].

3GP	3GPP file format
3GP-DASH	3GPP Dynamic Adaptive Streaming over HTTP
AHS	Adaptive HTTP Streaming
A/V	Audio/Video
AVC	Advanced Video Coding
DM	Device Management
DRM	Digital Rights Management
HSD	HTTP Streaming and Download
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDR	Instantaneous Decoding Refresh
MPD	Media Presentation Description
MPEG-2 TS	Moving Picture Experts Group Transport Stream

MIME	Multipurpose Internet Mail Extensions
MOS	Mean Opinion Score
NAS	Non-Access Stratum
OMA	Open Mobile Alliance
PDCF	Packetized DRM Content Format
PSS	Packet-switched Streaming Service
QMC	QoE Measurement Collection
QoE	Quality-of-Experience
RFC	Request For Comments
RRC	Radio Resource Control
RTP	Real-time Transport Protocol
SMIL	Synchronised Multimedia Integration Language
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UTF-8	Unicode Transformation Format (the 8-bit form)
XML	eXtensible Markup Language

3.3 Conventions

The naming conventions of ISO/IEC 23009-1 [43], clause 3.3 apply in this specification.

4 Overview

The present document specifies Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) for continuous media. The features are separated from the umbrella specification TS 26.234 [3] to differentiate from RTP-based streaming that is specified and maintained in TS 26.234. Services relying exclusively on these features may be deployed independently from RTP-based PSS servers, for example by using standard HTTP/1.1 servers for hosting the services.

The specification covers the following aspects:

- System Description: describes the relationship to the PSS architecture and refines the architecture, interfaces and protocols that are defined in this specification.

- Progressive Download over HTTP.

- 3GPP Dynamic Adaptive Streaming over HTTP (3GP-DASH) provides an overview of the architecture, the formats and the models that build the basis for 3GP-DASH. Also, 3GP-DASH Profiles provide identifiers and refers to a set of specific restrictions in this or other specifications.

- DASH - Media Presentation describes the data model of a Media Presentation. It also provides an overview on elements and attributes that may be used to describe components and properties of a media presentation in a Media Presentation Description (MPD).

- DASH - Usage of the 3GP file format defines how segments can be formed based on the 3GP file format.

- Quality-of-Experience for Progressive Download and 3GP-DASH.

- Normative annexes for MPD schema (Annex B), Descriptor Scheme Definitions (Annex C), OMA DM QoE Management Object (Annex F), File format extensions for 3GPP DASH support (Annex G) and MIME Type Registration for MPD (Annex H). - Informative annexes for Client Behaviour (Annex A), MPD Examples (Annex D), and Mapping MPD structure and semantics to SMIL (Annex E).

Note: Several of the Annexes refer partially or exclusively to ISO/IEC 23009-1 [43].

5　System Description

5.1　Overview

Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) enables to provide services to deliver continuous media content over Hypertext Transfer Protocol (HTTP) in a sense that all resources that compose the service are accessible through HTTP-URLs and the HTTP/1.1 protocol as specified in RFC 2616 [9] may be used to deliver the metadata and media data composing the service. This enables that standard HTTP servers and standard HTTP caches can be used for hosting and distributing continuous media content. Figure 1 shows the architecture for services using progressive download and Figure 2 shows the architecture for services using 3GP-DASH.

The present document deals with the specification of interfaces between the Client and the Server. Specifically, it defines the formats that may be delivered exclusively over the HTTP interface to enable progressive download and streaming services.
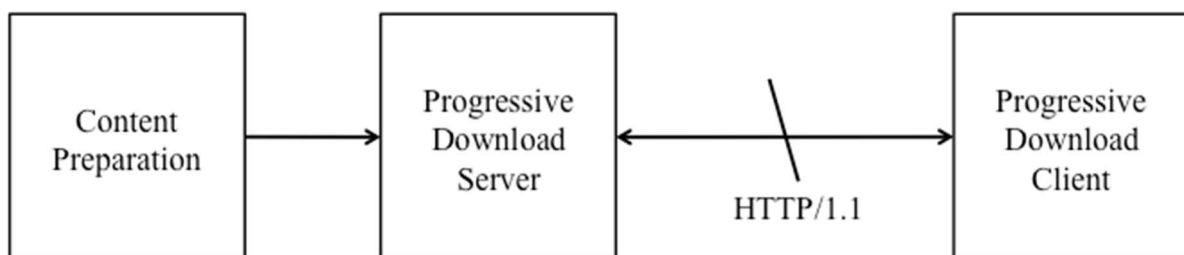
Figure 1: Architecture for Progressive Download over HTTP

Services using the features described in this specification may be deployed within PSS as specified in TS 26.233 [2] and TS 26.234 [3]. In this case the Progressive Download/3GP-DASH Server may be a sub-function of the PSS server and the Progressive Download/3GP-DASH client may be a sub-function of the PSS client.
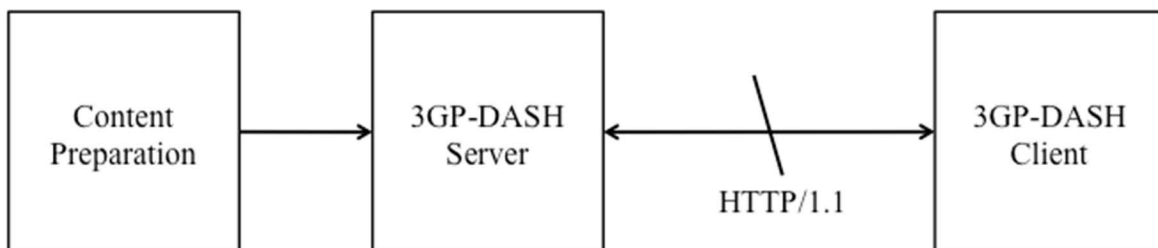
Figure 2: Architecture for 3GP-DASH

Services using the features defined in this specification may also be deployed independent of the PSS servers and clients. In this case the Progressive Download/3GP-DASH client shall support the formats and codecs according to this specification.

Access to services based on the features defined in the present document is introduced in clause 5.2.

The protocol support for services using the features defined in this specification is provided in clause 5.3.

Clients supporting progressive download-based services shall support the features and formats as specified in clause 6 of this specification.

Clients supporting 3GP-DASH shall support the features and formats as specified in clause 7 of this specification.

Clients supporting QoE Metrics and Reporting shall support the features as specified in clause 10 of this specification.

5.2　Service Access

Service access refers to the method by which a Client initially accesses the service. Service access for services based in the specification can be achieved e.g. by a Media Presentation Description or a URL to the media file.

The service access URL can be made available to a client in many different ways. Clients supporting services based on the features in this specification shall be able to access services that are provided through an HTTP-URL. However, it is out of the scope of this specification to mandate any specific mechanism. A preferred way may be to embed URLs for service establishment within HTML pages.

5.3 Protocols

5.3.1 General

Progressive Download and 3GP-DASH clients shall comply with a *client* as specified in RFC 2616 [9]. The resource hosting the 3GP files and DASH Segments shall comply with a *server* as specified in RFC 2616 [9].

Progressive Download and 3GP-DASH clients should use the HTTP GET method or the HTTP partial GET method, as specified in RFC 2616 [9], clause 9.3, to access media offered at HTTP-URLs.

Figure 3 shows a protocol stack for services in the context of this specification. 3GP Files in progressive download as well as Segments based on the 3GPP File Format shall be accessible through HTTP.
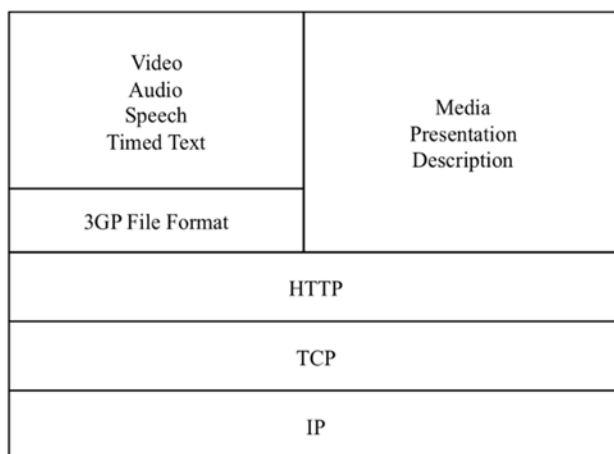
Figure 3: Overview of the protocols stack

Transport security in Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) is achieved using the HTTPS (Hypertext Transfer Protocol Secure) specified in RFC 2818 [12] and TLS as specified in TLS profile of Annex E in TS 33.310 [23]. In case secure delivery is desired, HTTPS should be used to authenticate the server and to ensure secure transport of the content from server to client.

NOTE 1: The use of HTTPS for delivering Media Segments may inhibit caching at proxies and add overhead at the server and the client.

5.3.2 Partial File Handling

In certain cirumstances, for example in the case of MBMS download delivery of 3GP-DASH content, a file or segment may only be partially available on the HTTP server.

If the application, i.e. the 3GP file format client or the 3GP-DASH client supports the handling of partial files, it shall signal its capability using the partial-file-accept request defined in TS26.346 [42], clause 7.9.2.1, i.e. using the Accept header `application/3gpp-partial` in combination with e.g. an HTTP GET request. If the application uses a partial-file-accept request, then the application shall also be able to handle the request response as defined in TS26.346 [42], clause 7.9.2.2.

6 Progressive Download over HTTP

6.1 General

As an alternative to conventional streaming, a client may download, typically through HTTP, a media file that encapsulates continuous media and may play the media from the local storage. A PSS client shall support progressive download and playout of 3GP files [4] as specified in the remainder of this clause.

The media file encapsulating the continuous media is accessed directly by issuing one or more HTTP GET or partial GET requests to the referenced media file. An example of a valid URL is http://example.com/morning_news.3gp.

6.2 Progressive Download

Progressive download uses normal HTTP download using HTTP GET or partial GET requests. The differences between regular download and Progressive Download are that 1) the content may be authored as progressively downloadable, and 2) the terminal recognises that the content is suitable for progressive download.

A client downloading continuous media may decide to start playout of the encapsulated media data before the download of the media file is completed.

6.3 3GPP File Format Profiles

The following profiles of the 3GPP file format in TS 26.244 [4] shall be supported by clients supporting Progressive Download over HTTP:

- Basic profile, and

- Progressive-download profile.

7 3GPP Dynamic Adaptive Streaming over HTTP

7.1 System Description

The 3GPP Dynamic Adaptive Streaming over HTTP (3GP-DASH) specified in this specification provides streaming services over HTTP. 3GP-DASH is a set of profiles of ISO/IEC 23009-1 [43], also known as MPEG-DASH, with some extensions. For this it specifies XML and binary formats that enable delivering content from standard HTTP servers to an HTTP-Streaming client and enables caching content by standard HTTP caches.

The specification for 3GP-DASH primarily defines two formats:

1) The Media Presentation Description (MPD) describes a *Media Presentation*, i.e. a bounded or unbounded presentation of media content. In particular, it defines formats to announce resource identifiers for *Segments* and to provide the context for these identified resources within a Media Presentation. For 3GP-DASH, the resource identifiers are exclusively HTTP-URLs possibly combined with a byte range.

2) The Segment formats specify the formats of the entity body of the HTTP response to an HTTP GET request or an HTTP partial GET request with the indicated byte range through HTTP/1.1 as defined in RFC 2616 [9] to a resource identified in the MPD. Segments typically contain efficiently coded media data and metadata according to or aligned with common media formats.

The MPD provides sufficient information for a client to provide a streaming service to the user by accessing the Segments through the protocol specified in the scheme of the defined resources, in the context of this specification exclusively HTTP/1.1. Such a client is referred to as a 3GP-DASH client in the remainder of the present document. However, this specification does not provide a normative definition for such a client. An informative client model to illustrate the formats defined in this specification is provided in section 7.2. An informative example client behaviour description is provided in Annex A of ISO/IEC 23009-1 [43].

Figure 7-1 shows an architecture in which the formats defined in this specification are typically used. Boxes with solid lines indicate devices that are mentioned in this specification as they host or process the formats defined in this specification whereas dashed boxes are conceptual or transparent. This specification deals with the definition of formats that are accessible on the interface to the 3GP-DASH client, indicated by the solid lines. Any other formats or interfaces are not in scope of this specification. In the considered deployment scenario, it is assumed that the 3GP-DASH client has access to an MPD. The MPD provides sufficient information for the 3GP-DASH client to provide a streaming service to the user by requesting Segments from an HTTP server and demultiplexing, decoding and rendering the included media streams.
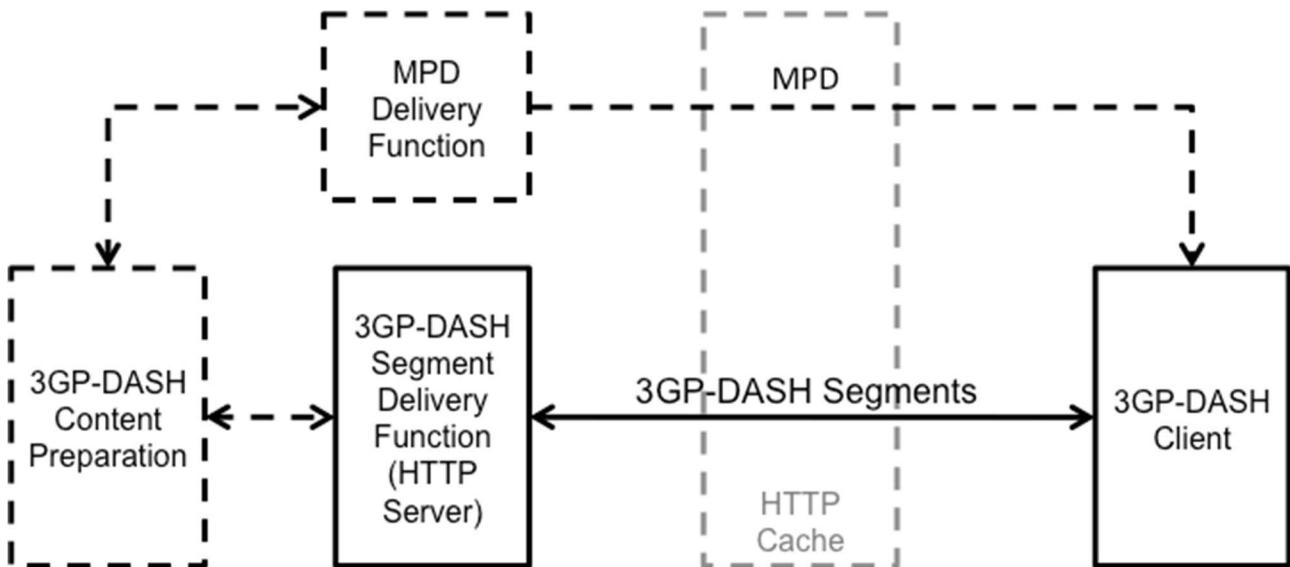
Figure 7-1: System Architecture for 3GP-DASH

The normative aspects of 3GP-DASH formats are defined by

- the profiles defined in clause 7.3.

- the DASH Media Presentation as defined in clause 8.

- the usage of the 3GPP file format for DASH as defined in clause 9.

The clauses mentioned above may refer to normative aspects in clause 10 on Quality-of-Experience, normative Annexes B and H as well as ISO/IEC 23009-1 [43].

7.2   3GP-DASH Client Model

The design of the formats defined in this specification is based on the informative client model as shown in Figure 7-2. The figure illustrates the logical components of a conceptual 3GP-DASH client model. In this figure the 3GP-DASH Access Engine receives the Media Presentation Description (MPD), constructs and issues requests and receives Segments or parts of Segments. In the context of this standard, the output of the DASH Access Engine consists of media in container formats according to the ISO/IEC 14496-12 ISO Base Media File Format [11] and specifically the 3GP file format [4]. In addition, timing information is provided that maps the internal timing of the media to the time line of the Media Presentation.
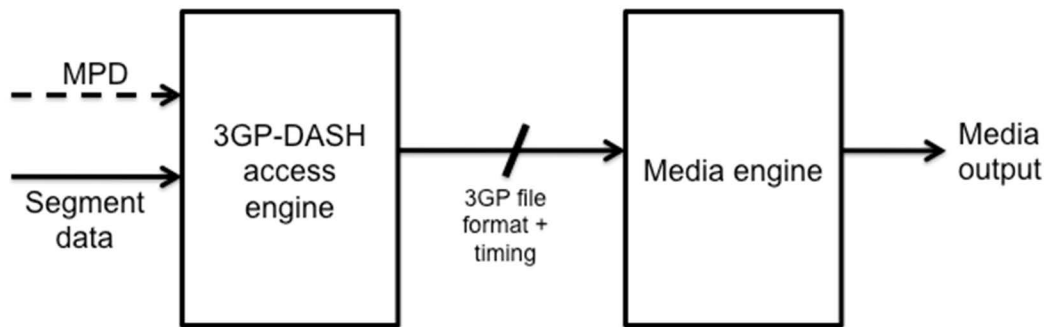
Figure 7-2: 3GP-DASH client Model

7.3 3GP-DASH Profiles

7.3.1 General

Profiles of 3GP-DASH are defined so as to enable interoperability and the signaling of the use of features etc. A profile refers to a set of specific restrictions. Those restrictions might be on features of the MPD as defined in clause 8 of this specification, Segment formats as for example defined in clause 9 of this specification, usage of the network, codec(s) used, content protection formats, or on quantitative measures such as bit-rates, segment lengths, screen size, and so on. Profiles defined in this specification define restrictions on features of this specification, but may additionally impose restrictions on other aspects of media delivery.

NOTE   A profile can also be understood as permission for 3GP-DASH clients that only implement the features required by the profile to process the Media Presentation. However, as 3GP-DASH client operation is not specified normatively, it is also unspecified how a 3GP-DASH client conforms to a particular profile. Hence, profiles merely specify restrictions on MPD and Segments rather than DASH client behaviour.

For details on the use of profiles, refer to ISO/IEC 23009-1 [43], section 8.1.

7.3.2 3GPP Adaptive HTTP Streaming (Release-9 AHS)

Release-9 Adaptive HTTP Streaming as defined in TS 26.234 [3] Release-9, clause 12 is not a profile of this specification. Rel-9 AHS uses a different namespace `"urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:2009"` and a different MIME type signalling `"application/3gpp-ahs+xml"` for the MPD. However, a Media Presentation may be defined such that segments complying with the segment formats in TS 26.234 [3] Release-9, clause 12, also comply with segment formats for this specification.

7.3.3 3GP-DASH Release-10 Profile

7.3.3.1 Introduction

The 3GP-DASH Release-10 profile is identified by the URN "`urn:3GPP:PSS:profile:DASH10`".

This profile includes all features defined in the Release-10 version of this specification in clauses 7.3.6 (media codecs), 7.3.7 (content protection), 8 (Media Presentation Description), 9 (File Format) and 10 (QoE).The `@mimeType` attribute of each Representation shall be provided according to RFC4337. Additional parameters may be added according to RFC6381 [26].

7.3.4 3GP-DASH Release 11 multiview stereoscopic 3D video profile

The 3GP-DASH Release 11 multiview stereoscopic 3D video profile is identified by the URN "`urn:3GPP:PSS:profile:DASH11:MS3D`".

The `@mimeType` attribute of each Representation shall be provided according to RFC4337. Additional parameters may be added according to RFC6381 [26].

This profile includes all features defined in clauses 7.3.7, 8, 9 and 10.

Clients that support 3GP-DASH Release 11 multiview stereoscopic 3D video profile shall support multiview stereoscopic 3D video as specified in clause 7.4 of TS 26.234 [3]. For any other particular continuous media type, the corresponding media decoders are specified in TS 26.234 [3], clause 7.2 for speech, 7.3 for audio, 7.4 for video, 7.9 for timed text and 7.11 for timed graphics. Additionally, the following contraints apply for multiview stereoscopic 3D video bitstreams, if present in a media presentation:

> The DASH multiple views scheme as defined in 5.8.5.6 of ISO/IEC 23009-1 [43] shall be used in the MPD.

> One of the following shall be true:

- The base view of the stereoscopic multiview bitstream shall be a complementary representation and the non-base view of the bitstream shall be a dependent representation. The @dependencyId attribute as specified in 5.3.5.2 of ISO/IEC 23009-1 [43] shall be used to indicate the complementary and dependent representations.

- The base view and the non-base view of the stereoscopic multiview bitstream shall reside in the same representation. The SubRepresentation element shall be used for the representation, and the base view and the non-base view shall form separate sub-representations. The @level and @dependencyLevel attributes within the SubRepresentation element shall be used. The Level Assignment box shall be used. For each leaf segment index, that is, each Segment Index box that indexes only subsegments but not other Segment index boxes, there shall be exactly one Subsegment Index box.

7.3.5   3GP-DASH Release 11 frame-packed stereoscopic 3D video profile

The 3GP-DASH Release 11 frame-packed stereoscopic 3D video profile is identified by the URN "urn:3GPP:PSS:profile:DASH11:FPS3D".

The @mimeType attribute of each Representation shall be provided according to RFC4337. Additional parameters may be added according to RFC6381 [26].

This profile includes all features defined in clauses 7.3.7, 8, 9 and 10.

Clients that support 3GP-DASH Release 11 frame-packed stereoscopic 3D video profile shall support frame-packed stereoscopic 3D video as specified in clause 7.4 of TS 26.234 [3]. For any other particular continuous media type, the corresponding media decoders are specified in TS 26.234 [3], clause 7.2 for speech, 7.3 for audio, 7.4 for video, 7.9 for timed text and 7.11 for timed graphics. Additionally, the following contraints apply for frame-packed stereoscopic 3D video bitstreams, if present in a media presentation:

> The FramePacking element as defined in clause 8.4.3.2 shall be used in the MPD.

7.3.6   Media Codecs

For 3GP-DASH clients supporting a particular continuous media type, media decoders are specified in TS 26.234 [3], clause 7.2 for speech, 7.3 for audio, 7.4 for video, 7.9 for timed text and 7.11 for timed graphics.

> NOTE: If 3GP-DASH formatted content is delivered outside the scope of this specification, additional constraints may apply on which media codecs can be used. For example, TS26.346 [42] defines further restrictions when delivered over MBMS.

7.3.7   Content Protection

3GP-DASH clients content protection may support OMA DRM 2.0 [15] or OMA DRM 2.1 [16]. Other content protection schemes may be supported. The ContentProtection element in the MPD should be used to convey content protection information.

When using OMA DRM V2.0 or OMA DRM V2.1 scheme for content protection, the non-streamable Packetized DRM Content Format (PDCF) shall be used. An OMA-DRM encrypted Representation shall include the brand "opf2". OMA-DRM [15] [16] defines the procedures for acquiring the Rights Object from the Rights Issuer to decrypt PDCF protected content.   The   scheme   is   identified   by   a   **ContentProtection**@schemeIdUri   set   to "urn:mpeg:dash:mp4protection" and the **ContentProtection**@value shall include the version number;

it starts with "`odkm`", which is the scheme_type contained in the Scheme Type Box of the PDCF file, followed by a ":" and the scheme_version from the Scheme Type Box of the PDCF file, encoded as up to 8 hexadecimal digits, where the leading '0's may be omitted. For example, for OMA DRM2.0 the value could be "odkm:200".

7.3.8 Partial File Handling

3GP-DASH clients should support partial-file-accept requests and partial file responses as defined in clause 5.3.2. If 3GP-DASH clients support partial file handling they shall use partial-file-accept requests as defined in TS26.346 [42], clause 7.9.2.1.

Without excluding other response options, as a response to a partial-file-accept request using a regular HTTP GET request a 3GP-DASH client may typically receive one of the following responses:

1) 200 OK with Content-Type set to the Media Type of the requested object

2) 200 OK with the Content-Type set to `application/3gpp-partial` and the message format according to the definition in clause 7.9.2.2 of TS26.346 [42].

3) `416 Requested Range Not Satisfiable` with the additional information according to the definition in clause 7.9.2.2 of TS26.346 [42].

4) `404 Not Found`

Case 1 is the regular response.

Guidelines for handling request responses according to case 4 from above are provided in Annex A.7.

Guidelines for handling request responses 2 and 3 from above are provided in Annex A.9.

7.3.9 3GP-DASH Enhanced Interoperability Point

The 3GP-DASH Enhanced interoperability point (IOP) is identified by the URN "`urn:3GPP:PSS:iop:DASH-enhanced`".

This interoperability point includes all features defined in the Release-13 version of this specification in clauses 7.3.6 (media codecs), 7.3.7 (content protection), 8 (Media Presentation Description), 9 (File Format), 10 (QoE), 11 (simple live) and 12 (ad insertion).

A DASH client conforms to the IOP by supporting at least the following features:

- All DASH-related features as defined in clause.8 of this document.

- The file format related aspects defined in clause 9 of this document.

- The QoE related aspects defined in clause 10 of this document.

- The requirements and guidelines in clause 11 for simple live operation.

- The requirements and guidelines in clause 12 for server-based ad insertion.

- The following additional requirements:

    - Segment formats are based on ISO BMFF with fragmented movie files, i.e. (Sub)Segments are encoded as movie fragments containing a track fragment as defined in ISO/IEC 14496-12, plus the following constraints to make each movie fragment independently decodable:

        - Default parameters and flags shall be stored in movie fragments ('`tfhd`' or '`trun`' box) and not track headers ('`trex`' box)

        - The '`moof`' boxes shall not use external data references, the flag '`default-base-is-moof`' shall also be set (aka movie-fragment relative addressing) and `data-offset` shall be used, i.e. `base-data-offset-present` shall not be used (follows ISO/IEC 23009-1).

    - Withing each Adaptation Set the following applies

- Fragmented movie files are used for encapsulation of media data

- (Sub)Segments are aligned to enable seamless switching

- The following additional restrictions are applied.

- IDR-like SAPs (i.e., SAPs type 2 or below) at the start of each (Sub)Segment for simple switching.

- Segments should have almost equal duration.

- only non-multiplexed Representations should be used, i.e. each Representation only contains a single media component.

- Addressing schemes are restricted to

- templates with number-based addressing

- Subsegments with Segment Index. In this case either the `@indexRange` attribute shall be present or the **RepresentationIndex** element shall be present. Only a single `sidx` box shall be present.

- For DRM purposes, suitable DRM may be used together with common encryption as defined in ISO/IEC 23001-7 [47]. If used, only the AES-128 CTR mode shall be used. No specific requirements on a specific DRM system are added.

Content shall only be authored claiming conformance to this IOP if such a client can properly play the content. In addition, the content shall follow the mandatory aspects and should take into account the recommendations and guidelines for content authoring documented in clauses 7.3.6 (media codecs), 7.3.7 (content protection), 8 (Media Presentation Description), 9 (File Format), 10 (QoE), 11 (simple live) and 12 (ad insertion).

The `@mimeType` attribute of each Representation shall be provided according to RFC4337. Additional parameters may be added according to RFC6381 [26].

7.3.10   DASH in TeleVision (TV) over 3GPP Services

If used for TeleVision (TV) over 3GPP Services, 3GP-DASH clients shall support the DASH features defined in clause 5 of TS 26.116 [54].

8   DASH - Media Presentation

8.1   Introduction

MPEG had initiated a standardization process to provide specifications to enable scalable and flexible video distribution that addresses fixed and mobile networks. The work had been in close coordination with a parallel effort in 3GPP such that the two standards are aligned for broad industry support across different access networks. 3GPP's Release-9 specification on Adaptive HTTP Streaming (AHS) [3], section 12 completed in 2010 served as a baseline for MPEG's DASH [43] (MPEG-DASH) as well as for 3GP DASH specification in this document.

In addition to the format specification, MPEG provides additional supporting material as part of MPEG-DASH, namely:

- ISO/IEC 23009-2: Conformance and Reference software [44]

- ISO/IEC 23009-3: Implementation and Deployment Guidelines [45]

Due to the close coordination in the development, 3GP-DASH can be viewed as a set of profiles of MPEG-DASH, MPEG DASH for the segment format uses the ISO base media file format [7]. In 3GPP, compatibility is achieved with the 3GP file format [5].Dynamic Adaptive Streaming over HTTP (DASH) [43] specifies XML and binary formats that enable delivery of media content from standard HTTP servers to HTTP clients and enable caching of content by standard HTTP caches.

ISO/IEC 23009-1 [43] primarily defines two formats:

— The Media Presentation Description (MPD) describes a *Media Presentation*, i.e. a bounded or unbounded presentation of media content. In particular, it defines formats to announce resource identifiers for *Segments* and to provide the context for these identified resources within a Media Presentation. These resource identifiers are HTTP-URLs possibly combined with a byte range.

— The Segment formats specify the formats of the entity body of the HTTP response to an HTTP GET request or a partial HTTP GET with the indicated byte range using HTTP/1.1 as defined in RFC 2616 to a resource identified in the MPD. Segments typically contain efficiently coded media data and metadata conforming to or at least closely aligned with common media formats.

For more details on DASH System description, please refer to ISO/IEC 23009-1 [43], clause 4.1.

For more details on DASH Client model, please refer to ISO/IEC 23009-1 [43], clause 4.2.

For more details on DASH Data model overview, please refer to ISO/IEC 23009-1 [43], clause 4.3.

For more details on DASH Media Stream and Representation properties, please refer to ISO/IEC 23009-1 [43], clause 4.5.

The format of the Media Presentation Description in 3GP-DASH is defined in clause 8.2.

The format of the Segments in 3GP-DASH is defined in section 9.

Figure 8.1: Void

8.2 Media Presentation Description

8.2.1 General

The Media Presentation Description shall conform to a DASH Media Presentation as defined in ISO/IEC 23009-1 [43], clause 5.2.Updates may also be done using MPD delta files as defined in clause 8.5.2. The MIME type of an MPD delta file is defined in Annex H.2.

8.2.2 Schema and 3GPP Extension

The overview of the XML schema of the MPD is provided in below. Specific types, elements and attributes are introduced in the remainder of this clause. The complete MPD schema is provided in Annex B of this specification. In case of any inconsistencies the schema in Annex B takes precedence over the XML-syntax snippets provided in this clause. For the normative schema refer to the schema in Annex B.

The main schema is provided in Table 8-1 with the namespace `"urn:mpeg:dash:schema:mpd:2011"`. The 3GPP extension namespace is provided in Table 8-2 with namespace `"urn:3GPP:ns:DASH:MPD-ext:2011"`. An extension schema for 3GPP in the context of the specification is referred to as "3gpp-2011.xsd". Elements and attributes in the extension namespace are preceded with `"x3gpp:"` throughout this document.

The MPD shall be authored such that, after unrecognized XML attributes or elements are removed, the result is a valid XML document formatted according to the XML schema provided in Annex B and that complies with this specification. Namespaces may be used to extend functionalities. Therefore, all extended elements and attributes added to a **Representation** in particular shall be such that they can be safely ignored by 3GP-DASH clients.

NOTE 1: Based on this if DASH clients remove all XML attributes and elements from the MPD in the DASH namespace (`urn:mpeg:dash:schema:mpd:2011`) and in other namespaces that are not in the XML schema documented in Annex B, the MPD results in a valid XML document which complies with this specification. The DASH client can use such a resulting MPD for presentation of a conforming Media Presentation.

Example for valid MPDs are provided in Annex D.

Table 8-1: Overview of XML schema of the MPD

```xml
<?xml version="1.0"?>
<xs:schema targetNamespace="urn:mpeg:dash:schema:mpd:2011"
    attributeFormDefault="unqualified" elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:x3gpp="urn:3GPP:ns:DASH:MPD-ext:2011"
    xmlns="urn:mpeg:dash:schema:mpd:2011">
    <xs:annotation>
        <xs:appinfo>Media Presentation Description</xs:appinfo>
    </xs:annotation>

    <xs:import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlink.xsd"/>
    <xs:import namespace="urn:3GPP:ns:DASH:MPD-ext:2011" schemaLocation="3gpp-2011.xsd"/>

    <!-- MPD: main element -->
    <xs:element name="MPD" type="MPDtype"/>
    ...

</xs:schema>
```

Table 8-2: Overview of XML schema for 3GPP MPD extensions

```xml
<?xml version="1.0"?>
<xs:schema targetNamespace="urn:3GPP:ns:DASH:MPD-ext:2011"
    attributeFormDefault="unqualified" elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="urn:3GPP:ns:DASH:MPD-ext:2011">
    <xs:annotation>
        <xs:appinfo>Extensions to Media Presentation Description for 3GPP</xs:appinfo>
    </xs:annotation>

    ...

</xs:schema>
```

8.2.3   (void)

8.2.4   (void)

8.3   MPD Assembly

3GP-DASH inherits the ability from ISO/IEC 23009-1 [43] to enable to distribute an MPD not as a fully contained document, but to assemble the document by dereferencing certain links by using subset of W3C XLINK [20].

An MPD in 3GP DASH may reference a *remote element entities* from within a local MPD as defined in ISO/IEC 23009-1 [43], clause 5.5.

8.4   Hierarchical Data Model

8.4.1   General

A Media Presentation is described in the **MPD** element that is contained in an MPD document formatted as defined in clause 8.2.

A Media Presentation consists of:

-    A sequence of one or more Periods described in 8.4.2.

-    Each Period contains one or more Adaptation Sets that itself contains one or more Representations as described in clause 8.4.3. Clause 8.4.3 also defines media content components and Sub-Representations.

-    Each Representation consists of one or more Segments. Segment Information is introduced in clause 8.4.4. Segments contain media data and/or metadata to access, decode and present the included media content.

Beyond the mandatory elements and attributes in ISO/IEC 23009-1 [43], the **MPD** element used in 3GP-DASH may contain the following attributes and elements:

- **MPD**@id

- **MPD**@type

- **MPD**@availabilityStartTime

- **MPD**@availabilityEndTime

- **MPD**@mediaPresentationDuration

- **MPD**@minimumUpdatePeriod

- **MPD**@timeShiftBufferDepth

- **MPD**@suggestedPresentationDelay

- **MPD**@maxSegmentDuration

- **MPD**@maxSubsegmentDuration

- **MPD.ProgramInformation**

- **MPD.UTCTiming**

- **MPD.BaseURL**

- **MPD.x3gpp:DeltaSupport**

- **MPD.Location**

- **MPD.Metrics**

Table 8-5: Void

Table 8-6: Void

8.4.2 Period

8.4.2.1 General

A Media Presentation consists of one or more Periods as defined in ISO/IEC 23009-1 [43], clause 5.3.2. A Period is defined by **Period** element in the **MPD** element.

Beyond the mandatory elements and attributes in ISO/IEC 23009-1 [43], the **Period** element used in 3GP-DASH may contain the following attributes and elements:

- **Period**@xlink:href

- **Period**@xlink:actuate

- **Period**@id

- **Period**@start

- **Period**@duration

- **Period**@bitstreamSwitching

- **Period.BaseURL**

- **Period.SegmentBase**

- **Period.SegmentList**

- **Period.SegmentTemplate**

- **Period.AdaptationSet**

- **Period.EventStream**

- **Period.AssetIdentifier**

Table 8-7: Void

Table 8-8: Void

8.4.2.2   Content Offering with Multiple Periods

Note: This text was adopted for Amd.3 of ISO/IEC 23009-1 [43], which is not yet published. It is expected that the below text will be replaced with a reference to ISO/IEC 23009-1 once Amd.3 is published.

Content with multiple Periods may be created for different reasons, for example:

- to enable splicing of content, for example for ad insertion,

- to provide a synchronization point to avoid drift in segment numbering,

- to remove or add certain Representations in an Adaptation Set,

- to remove or add certain Adaptation Sets,

- to add or remove content offering on certain CDNs,

- to enable signalling of shorter segments, if produced by the encoder

Periods provide opportunities for resync, for ad insertion, for adding and removing Representations, but if they are not used or only used for minor changes, then continuous playout of the client is expected.

In certain circumstances the Media Presentation author offers content in the next Period that is a continuation of the content in the previous Period, possibly in the immediately following Period or in a later Period. The latter case applies for example after an advertisement Period had been inserted. The content provider may provide period-continuous Adaptation Sets as follows:

- The Adaptation Set identifiers are the same across two Periods.

- The sum of the value of the `@presentationTimeOffset` and the presentation duration of all Representations in one Adaptation Set are identical to the value of the `@presentationTimeOffset` of the associated Adaptation Set in the next Period.

- If Representations in both Adaptation Sets have the same value for `@id`, then they sshould have functionally equivalent Initialization Segments, i.e. the Initialization Segment may be used to continue the play-out the Representation.

From a client perspective, at Period boundary typically no continuity in terms of content offering is ensured. The content may be offered with different codecs, language attributes, different protection and so on. The client should play the content continuously across Periods, but there may be implications in terms of implementation to provide fully continuous and seamless playout. It may be the case that at Period boundaries, the presentation engine needs to be reinitialised, for example due to changes in formats, codecs or other properties. This may result in a re-initialisation delay. Such a re-initialisation delay should be minimized.

If the client presents media components of a certain Adaptation Set in one Period, and if the following Period has an identical Adaptation Set identifier, then the client is suggested to continue playing the content in the associated Adaptation Set.

8.4.3 Adaptation Sets and Representations

8.4.3.1 Overview

Periods are further subdivided as follows:

- Each Period contains one or more groups. Groups consist of Adaptation Sets as described in clause 8.4.3.3.

- In case an Adaptation Set contains multiple media content components, then each media content component is described individually as defined in clause 8.4.3.6.

- Each Adaptation Set contains one or more Representations as described in clause 8.4.3.4.

- A Representation may contain one or more Sub-Representations as described in clause 8.4.3.5.

- Adaptation Sets, Representations and Sub-Representations share common attributes and elements that are described in clause 8.4.3.2.

8.4.3.2 Common Attributes and Elements

The elements **AdaptationSet, Representation** and **SubRepresentation** have assigned common attributes and elements.

Beyond the mandatory elements and attributes in ISO/IEC 23009-1 [43], the following common attributes and elements may be used in 3GP-DASH:

- @profiles

- @width

- @height

- @frameRate

- @audioSamplingRate

- @mimeType

- @codecs

- @maximumSAPPeriod

- @startWithSAP

- @maxPlayoutRate

- @codingDependency

- **FramePacking**

- **AudioChannelConfiguration**

- **ContentProtection**

- **EssentialProperty**

- **SupplementalProperty**

- **InbandEventStream**

Table 8-9: Void

Table 8-10: Void

8.4.3.3   Adaptation Set

An Adaptation Set is described by an **AdaptationSet** element. **AdaptationSet** elements are contained in a **Period** element. An Adaptation Set contains alternate Representations, i.e. only one Representation within an Adaptation Set is expected to be presented at a time. All Representations contained in one Adaptation Set represent the same media content components and therefore contain media streams that are considered to be perceptually equivalent.

The Adaptation Set shall conform to the definition in ISO/IEC 23009-1 [43], clause 5.3.3.

Beyond the mandatory elements and attributes in ISO/IEC 23009-1 [43], the **AdaptationSet** element used in 3GP-DASH may contain the following attributes and elements:

- @xlink:href

- @xlink:actuate

- @id

- @group

- *CommonAttributesElements*

- @lang

- @contentType

- @minBandwidth

- @maxBandwidth

- @minWidth

- @maxWidth

- @minHeight

- @maxHeight

- @minFrameRate

- @maxFrameRate

- @segmentAlignment

- @bitStreamSwitching

- @subsegmentAlignment

- @subsegmentStartsWithSAP

- **Accessibility**

- **Role**

- **Rating**

- **Viewpoint**

- **ContentComponent**

- **BaseURL**

- **SegmentBase**

- **SegmentList**

- **SegmentTemplate**

- **Representation**

Table 8-11: Void

Table 8-12: Void

8.4.3.4 Representation

Representations are described by the **Representation** element. **Representation** elements are contained in an **AdaptationSet** element.

A Representation is one of the alternative choices of the complete set or subset of media content components comprising the media content during the defined Period.

A Representation shall conform to the definition in ISO/IEC 23009-1 [43], clause 5.3.4.

Beyond the mandatory elements and attributes in ISO/IEC 23009-1 [43], the **Representation** element used in 3GP-DASH may contain the following attributes and elements:

- Representation@qualityRanking

- Representation@mediaStreamStructureId

- Representation.CommonAttributesElements

- Representation.BaseURL

- Representation.SubRepresentation

- Representation.SegmentBase

- Representation.SegmentList

- Representation.SegmentTemplate

Table 8-13: Void

Table 8-14: Void

8.4.3.5 Sub-Representation

Sub-Representations are embedded in regular Representations and are described by the **SubRepresentation** element. **SubRepresentation** elements are contained in a **Representation** element.

The **SubRepresentation** element describes properties of one or several media content components that are embedded in the Representation. It may for example describe the exact properties of an embedded audio component (language, codec, etc.), an embedded sub-title (language) or it may describe some embedded lower quality video layer (e.g. some lower frame rate, etc.).

A Sub-Representation shall conform to the definition in ISO/IEC 23009-1 [43], clause 5.3.5.

Beyond the mandatory elements and attributes in ISO/IEC 23009-1 [43], the **SubRepresentation** element used in 3GP-DASH may contain the following attributes and elements:

- SubRepresentation@level

- SubRepresentation@dependencyLevel

- SubRepresentation@bandwidth

- SubRepresentation@contentComponent

- *CommonAttributesElements*

Table 8-15: Void

Table 8-16: Void

8.4.3.6   Content Component

Each Adaptation Set contains one or more media content components. The properties of each media content component are described by a **ContentComponent** element or may be described directly on the **AdaptationSet** element if only one media content component is present in the Adaptation Set. **ContentComponent** elements are contained in an **AdaptationSet** element.

A Content Component shall conform to the definition in ISO/IEC 23009-1 [43], clause 5.3.6.

Beyond the mandatory elements and attributes in ISO/IEC 23009-1 [43], the **ContentComponent** element used in 3GP-DASH may contain the following attributes and elements:

- ContentComponent@id

- ContentComponent@lang

- ContentComponent@contentType

- ContentComponent.Accessibility

- ContentComponent.Role

- ContentComponent.Rating

- ContentComponent.Viewpoint

Table 8-17: Void

Table 8-18: Void

8.4.4   Segments and Segment Information

A Segment is the smallest addressable unit described by an MPD and has a defined format. Segment formats are defined in section 9. This clause defines the MPD information for Segments.

Representations are assigned *Segment Information* through the presence of the elements **BaseURL**, **SegmentBase**, **SegmentTemplate** and/or **SegmentList**. The *Segment Information* provides information on the location, availability and properties of all Segments contained in one Representation. Specifically, information on the presence and location of Initialization, Media, Index and Bitstream Switching Segments is provided.

Segment Information shall be provided as defined in ISO/IEC 23009-1 [43], clause 5.3.9 using the elements **BaseURL**, **SegmentBase**, **SegmentTemplate** and/or **SegmentList.**

- The duration of Segments shall be described by the @duration attribute.

- If the **SegmentTemplate** is used, then only the following identifiers as defined in ISO/IEC 23009-1 [43], Table 16 may be used: *$$*, *$RepresentationID$*, *$Number$*, and *$Bandwidth$*. The identifier *$Time$* shall not be used.

Table 8-19: Void

Table 8-20: Void

Table 8-21: Void

Table 8-22: Void

Table 8-23: Void

Table 8-24: Void

Table 8-25: Void

Table 8-26: Void

Table 8-27: Void

8.5 MPD Update

8.5.1 General

If the **MPD**@type is set to 'dynamic', the MPD may be updated during the Media Presentation. Updates typically extend the accessible Segment list for each Representation, introduce a new Period, update Segment locations or terminate the Media Presentation.The MPD update functionality in 3GP-DASH shall use the MPD update functionality in ISO/IEC 23009-1 [43], clause 5.4.

8.5.2 Media Presentation Description Delta

If the **x3gpp:DeltaSupport** element is present in the **MPD** element, the content provider indicates that MPD delta files, as defined in this clause, are supported on the server. The URI of the MPD delta is provided in **x3gpp:DeltaSupport** @sourceURL. The **x3gpp:DeltaSupport** @availabilityDuration element, if present, indicates that the MPD delta file referenced by the URI is available for at least the value of the @availabilityDuration attribute (after this time, the server may redirect the client to the full MPD). If **x3gpp:DeltaSupport** @availabilityDuration is not present, then no information is conveyed about the availability of the MPD delta. If a client request for an MPD delta file results in an error, the client should request a full MPD.

The semantics of the attributes within the **x3gpp:DeltaSupport** element are provided in Table 8-28. The XML-syntax of **x3gpp:DeltaSupport** element is provided in Table 8-29.

Table 8-28: Semantics of `x3gpp:DeltaSupport` element

Element or Attribute Name	Use	Description
x3gpp:DeltaSupport		If present, this element indicates that MPD delta files are supported by the server.
@sourceURL	M	The source string providing the URL of the MPD delta. The URL may be relative to any **BaseURL** on MPD level and reference resolution according to clause 8.2.3 shall be applied.
@availabilityDuration	O	When provided, indicates the duration that the server guarantees the availability of the MPD delta file referenced in @sourceURL after the MPD has been updated. After that the client may be redirected to the full MPD.
Legend: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and preceded with an @.		

Table 8-29: XML-Syntax of `x3gpp:DeltaSupport` element

```
    <!--DeltaSupport for the MPD -->
    <xs:complexType name="DeltaSupportType">
        <xs:sequence>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="sourceURL" type="xs:anyURI" use="required"/>
        <xs:attribute name="availabilityDuration" type="xs:duration"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
```

An MPD delta is a text file that shall include the delta between the MPD that references it and the latest provided MPD. Note that the value of @sourceURL in successive MPDs is necessarily different because it is impossible for the delta between two different MPDs and the most recent MPD to be the same.

The output format consists of one or more structures, each corresponding to a change. The changes are in decreasing line number order. The structure format looks like:

```
    change-command
    to-file-line
    to-file-line...
    .
```

There are three types of change commands change-command. Each consists of a line number or comma-separated range of lines in the first file and a single character indicating the kind of change to make. All line numbers are the original line numbers in the file. The types of change commands and the instructions are provided in Table 8-30.

Table 8-30: Change commands and the instructions for delta MPD files

Change command	Instruction	Example
*l*a	Add text from the second file after line *l* in the first file.	'8a' means to add the following lines after line 8 of file 1
*r*c	Replace the lines in range *r* in the first file with the following lines. Like a combined add and delete, but more compact.	'5,7c' means change lines 5–7 of file 1 to read as the text file 2.
*r*d	Delete the lines in range *r* from the first file.	'5,7d' means delete lines 5–7 of file 1.
NOTE: This is the format supported by the GNU diff utilities, see http://www.gnu.org/software/diffutils/manual/#Detailed-ed		

Regardless of the presence of a **x3gpp:DeltaSupport** element, the full MPD shall always be available to clients for regular MPD updates as defined in clause 8.5.1. MPD Delta related procedures are optional at the client.

8.6 Additional Media Presentation Information

8.6.1 Introduction

The MPD, Periods, Adaptation Sets, Representations and Sub-Representations may have assigned descriptors for describing the content or other elements in the MPD. This clause specifies this descriptive information.

8.6.2 Program Information

Descriptive information on the program may be provided for each period within the **ProgramInformation** element.

Program Information shall conform to the definition in ISO/IEC 23009-1 [43], clause 5.7.

Table 8-31: Void

Table 8-32: Void

8.6.3 Descriptors

The MPD may contain descriptors that are all in the same format as defined in this clause. The elements of type `DescriptorType` provide a flexible mechanism for DASH content authors to annotate and extend the **MPD**, **Period**, **AdaptationSet** and **Representation** elements.

Descriptors shall conform to the definition in ISO/IEC 23009-1 [43], clause 5.8.1.

The following descriptors may be used in 3GP-DASH:

- Content Protection as defined in ISO/IEC 23009-1 [43], clause 5.8.4.1

- Role as defined in ISO/IEC 23009-1 [43], clause 5.8.4.3. The DASH Role scheme as defined in ISO/IEC 23009-1 [43], clause 5.8.5.4 may be used.

- Rating as defined in ISO/IEC 23009-1 [43], clause 5.8.4.4

- ViewPoint as defined in ISO/IEC 23009-1 [43], clause 5.8.4.5

- Accessibility as defined in ISO/IEC 23009-1 [43], clause 5.8.4.2

- Audio Channel Configuration as defined in ISO/IEC 23009-1 [43], clause 5.8.4.7

- Frame Packing as defined in ISO/IEC 23009-1 [43], clause 5.8.4.6

- Essential Property as defined in ISO/IEC 23009-1 [43], clause 5.8.4.8

- Supplemental Property as defined in ISO/IEC 23009-1 [43], clause 5.8.4.9

- UTC Timing Descriptor as defined in ISO/IEC 23009-1 [43], clause 5.8.4.10. The DASH UTC Timing Schemes as defined in ISO/IEC 23009-1 [43], clause 5.8.5.10 may be used.

Table 8-33: Void

Table 8-34: Void

8.7 Base URL Processing

The **BaseURL** element may be used to specify one or more common locations for Segments and other resources. Base URL Processing shall conform to ISO/IEC 23009-1 [43], clause 5.6.

Table 8-35: Void

Table 8-36: Void

8.8 Event Streams

Events may be signalled in the MPD or within a Representation in order to signal aperiodic information to the DASH client or to an application. Events are timed, i.e. each event starts at a specific media presentation time and typically has a duration. Events may include DASH specific signalling or application-specific events. In the latter case, a proper scheme identifier identifies the application such that the DASH client can forward the event to the proper application.

Events of the same type are clustered in Event Streams. This enables a DASH client to subscribe to an Event Stream of interest and ignore Event Streams that are of no relevance or interest.

Two ways of signalling events are provided, namely

- events signalled in the MPD as defined in 5.10.2 of ISO/IEC 23009-1[34]. Event streams for MPD events are signaled by **Period.EventStream** element.

- events signalled inband in the Representation as defined in 5.10.3 of ISO/IEC 23009-1 [34]. Inband event streams are signaled by the element **InbandEventStream** for a specific Representation or all Representatons in an Adaptation Set.

This specification does not provide any specific information on how to use Event Streams. It is up to the application that employs DASH formats to instantiate the description elements with appropriate scheme information.

If the DASH client is interested in a specific event stream, it is expected to properly extract each event from an event stream and pass it to the application.

9 DASH - Usage of 3GPP File Format

9.1 Introduction

3GPP Dynamic Adaptive Streaming over HTTP uses many elements of fragmented 3GP files to define the Segment formats. This provides Segments according to the requirements defined in clause 8.4.4.1 and enables reuse of existing content, easy encoding and recording, etc. This clause introduces how to use the 3GPP file format as specified in TS 26.244 [4] for DASH Segment formats.

9.2 Segment Types and Formats

9.2.1 Introduction

3GP-DASH defines a Segment format that is used in the delivery of media data over HTTP. A Segment shall contain one or more boxes in accordance with the boxed structure of the ISO-base media file format [11].

For 3GP-DASH the following applies:

- In all cases for which a Representation contains more than one Media Segment, the following applies:

- The Initialization Segment as defined in clause 9.2.2 shall be present. The Initialization Segment shall be available for the 3GP-DASH client before any Media Segment is processed within the Representation.

- Media Segments shall not be self-initializing. The Media Segment format is defined in clause 9.2.3.

- In case a Representation contains only a single Media Segment, then either one of the following two options is used:

1) An Initialization Segment as defined in clause 9.2.2 and one Media Segment as defined in clause 9.2.3.

2) One Self-Initializing Media Segment as defined in clause 9.2.4.

9.2.2   Initialization Segment

The Initialization Segment is conformant with the 3GPP file format, adaptive streaming profile and shall carry "3gh9" as compatibility brand.

The Initialization Segment shall conform to the Initialization Segment in ISO/IEC 23009-1 [43], clause 6.3.3.

9.2.3   Media Segment

9.2.3.1   General

A Media Segment contains and encapsulates media streams that are either described within this Media Segment or described by the Initialization Segment of this Representation or both.

In addition, a Media Segment shall conform to the general format type in ISO/IEC 23009-1 [43], clause 6.2.3.1.

9.2.3.2   Subsegments and Segment Index

Media Segments may contain multiple Subsegments documented by the Segment Index.

Subsegments and Segment Index shall follow the definition in ISO/IEC 23009-1 [43], clause 6.2.3.2.

9.2.3.3   Subsegment Index

Media Subsegments may be indexed further to enable accessing different levels of Subsegments in a Media Subsegment. This Subsegment Index may also be provided in separate Index Segments together with the Segment Index.

Subsegment Index shall follow the definition in ISO/IEC 23009-1 [43], clause 6.2.3.3.

9.2.3.4   3GP-DASH Media Segment Format

A Media Segment conforming to the Media Segment Format for 3GP DASH shall carry '3gmA' as a compatible brand and shall conform to the General format type in ISO/IEC 23009-1 [43], clause 6.3.4.2.

9.2.4   Self-Initializing Media Segment

A Self-Initializing Media Segment shall conform to the Self-Initializing Media Segment in ISO/IEC 23009-1 [43], clause 6.3.5.1.

9.2.5   Media Stream and Segment Properties

9.2.5.1   Media Stream Access Points

To be able to access a Representation, each of the media streams that are contained in the Representation requires Media Stream Access Points (SAPs). For the definition refer to ISO/IEC 23009-1 [43], clause 4.5.1.

9.2.5.2 Non-overlapping Segments and Subsegments

The property of "non-overlapping" Segments (respectively Subsegments) is used to define the terms Segment alignment and Subsegment alignment. The definition is provided in ISO/IEC 23009-1 [43], section 4.5.1.

9.2.5.3 Bitstream concatenation

The definition for bistream concatenation is provided in ISO/IEC 23009-1 [43], section 4.5.2.

9.3 Usage on Server and Client

3GP-DASH uses 3GP files according to the 3GP Adaptive-Streaming profile as specified in TS 26.244 [4]. Content may be prepared as 3GP files according to the 3GP Adaptive-Streaming profile. Initialization Segments and Media Segments may be generated by segmenting such 3GP files. Segment Index 'sidx' boxes may be pre-contained in 3GP files or may be generated during the segmentation process. Clients may store a concatenation of a received Initialization Segment and a sequence of Media Segments from the same Representation to create a compliant 3GP file according to the Adaptive Streaming profile without accessing any media samples.

NOTE: As specified in TS 26.244, the MPD may be linked or embedded in the 'meta' box of the 'moov' box. This enables clients to access the MPD from a 3GP file that was made available from other means than 3GP-DASH (e.g. progressive download).

9.4 Segment Properties with MPD constraints

9.4.1 General

9.4.1.1 Introduction

The content, especially the Segments across Representations at the same media time may have been prepared in a joint or at least coordinated manner. To expose these properties to the client, certain flags in the MPD can be set to true to indicate such coordinated content preparation. Clients consuming 3GP-DASH formatted media presentations may benefit from properly authored content when switching between or presenting Representations.

9.4.1.2 Media Presentation Timeline

One of the key features in DASH is that encoded versions of different media components share a common timeline. The presentation time of access unit within the media content is mapped to the global common presentation timeline for synchronization of different media components and to enable seamless switching of different coded versions of the same media components.

The requirements and recommendations in ISO/IEC 23009-1 [43], clause 7.2.1 and clause 7.3.2 shall apply for 3GP-DASH.

9.4.1.3 Segment Index

If a Segment Index is present in a Media Segment of one Representation within an Adaptation Set, then the constraints in ISO/IEC 23009-1 [43], clause 7.2.2 shall apply for 3GP-DASH.

9.4.2 Segment Alignment

No additional requirements beyond those stated in section 8.4.3.3 are defined.

9.4.3 Bitstream Switching

If the @bitstreamSwitching is set to 'true' for a set of Representations within an Adaptation Set, the conditions in ISO/IEC 23009-1 [43], clause 7.3.3.2 shally apply.

9.4.4   Sub-Representation

If a **SubRepresentation** element is present in a Representation in the MPD and the **SubRepresentation**@level is present, then the conditions in ISO/IEC 23009-1 [43], clause 7.3.4 shally apply.

---

10   QoE for Progressive Download and DASH

10.1   General

A progressive download or 3GP-DASH client supporting Quality of Experience (QoE) shall report QoE metrics according to the QoE configuration. QoE reporting is optional, but if a 3GP-DASH client reports DASH metrics, it shall report all requested metrics.

The quality metrics are defined in subclause 10.2.

The quality metrics applicable for progressive download are specified in section 10.3. In this case the activation and configuration of QoE reporting framework is achieved by a corresponding OMA DM QoE Management Object as specified in Annex F, or by the QMC functionality as specified in Annex L.

The quality metrics for DASH are specified in section 10.4. In this case, QoE reporting may be triggered using the MPD (i.e. when the **Metrics** element is present in the MPD) or using OMA DM QoE Management Object as specified in Annex F, or by the QMC functionality as specified in Annex L. When QoE reporting is triggered via any of the above methods, the 3GP-DASH client is expected to collect quality metrics according to the QoE configuration. When using the MPD or the QMC functionality, the Quality Reporting scheme as defined in section 10.5 may be used.

The QoE configuration shall only be checked by the client when each session starts, and thus all logging and reporting criterias for an ongoing session shall be unaffected by any QoE configuration changes received during that session. This also includes evaluation of any filtering criterias, such as geographical filtering, which shall only be done when the session starts. Thus changes to the QoE configuration will only affect sessions started after these configuration changes have been received.

The quality metric reporting protocol is defined in subclause 10.6. This protocol shall be used when QoE reporting is triggered via the MPD or OMA DM QoE Management Object. When QoE reporting is triggered via the QMC functionality, the reporting is specified in Annex L.

The usage of ITU-T P.1203 [49] Audio/Video Mean Opinion Score (A/V MOS) estimation is defined in Annex K.

10.2   QoE Metric Definitions

10.2.1   Introduction

This section provides the general QoE metric definitions and measurement framework.

The semantics are defined using an abstract syntax. Section 10.6 provides a mapping to an XML schema. Items in this abstract syntax have one of the following primitive types (Integer, Real, Boolean, Enum, String) or one of the following compound types:

- Objects: an unordered sequence of (key, value) pairs, where the key always has string type and is unique within the sequence.

- List: a ordered list of items.

- Set: an unordered set of items.

Additionally, there are two kinds of timestamp defined, i.e. *real time* (wall-clock time) and *media time*.

10.2.2   Void

This clause is void.

Table 25: Void

10.2.3   Representation Switch Events

Annex D.4.4 in ISO/IEC 23009-1 [43] defines the metrics for representation switch events.

Table 26: Void

10.2.4   Average Throughput

This metric in Table 27 indicates the average throughput that is observed by the client during the measurement interval.

Table 27: Average Throughput

Key		Type	Description
AvgThroughput		Object	Average throughput that is observed by the client during the measurement interval
	numbytes	Integer	The total number of the content bytes, i.e. the total number of bytes in the body of the HTTP responses, received during the measurement interval.
	activitytime	Integer	The activity time during the measurement interval in milliseconds. The activity time during the measurement interval is the time during which at least one GET request is still not completed (i.e. excluding inactivity time during the measurement interval).
	t	Real Time	The real time of the start of the measurement interval
	duration	Integer	The time in milliseconds of the measurement interval
	accessbearer	String	Access bearer for the TCP connection for which the average throughput is reported
	inactivitytype	Enum	Type of the inactivity, if known and consistent throughout the reporting period: User request (e.g. pause) Client measure to control the buffer Error case

If the client requests the media Segments from the server separately over multiple non-competing parallel TCP connections established over separate access network bearers named as accessbearer, then the average throughput values should be reported as a list of events with average throughput for each access network and associated access network bearer information reported separately, following the same guidelines as described above.

10.2.5   Initial Playout Delay

This metric in Table 28 signals the initial playout delay at the start of the streaming of the presentation.

Table 28: Initial Playout Delay

Key	Type	Description
InitialPlayoutDelay	Integer	The initial playout delay is measured as the time in milliseconds from the fetch of the first media Segment (or sub-segment) and the time at which media is retrieved from the client buffer.

10.2.6   Buffer Level

Annex D.4.5 in ISO/IEC 23009-1 [43] defines the metrics for buffer level status events.

Table 29: Void

10.2.7   Play List

Decoded samples are generally rendered in presentation time sequence, each at or close to its specified presentation time. A compact representation of the information flow can thus be constructed from a list of time periods during which samples of a single representation were continuously rendered, such that each was presented at its specified presentation time to some specific level of accuracy (e.g. +/-10 ms).

Such a sequence of periods of continuous delivery is started by a user action that requests playout to begin at a specified media time (this could be a "play", "seek" or "resume" action) and continues until playout stops either due to a user action, the end of the content, or a permanent failure.

Table 30 defines the play list event metric.

Table 30: Play List

Key					Type	Description
PlayList					List	A list of playback periods. A playback period is the time interval between a user action and whichever occurs soonest of the next user action, the end of playback or a failure that stops playback.
	Entry				Object	A record of a single playback period.
		start			Real Time	Timestamp of the user action that starts the playback period.
		mstart			Media Time	The presentation time at which playout was requested by the user action.
		starttype			Enum	Type of user action which triggered playout - New playout request (e.g. initial playout or seeking) - Resume from pause - Other user request (e.g. user-requested quality change) - Start of a metrics collection period (hence earlier entries in the play list not collected)
		Trace			List	List of periods of continuous rendering of decoded samples.
			Traceentry		Objects	Single entry in the list.
				representationid	String	The value of Representation@id from which the samples were taken. This is an optional parameter and should not be reported in case of progressive download.
				subreplevel	Integer	If not present, this metric concerns the Representation as a whole. If present, subreplevel indicates the greatest value of any SubRepresentation@level being rendered. This is an optional parameter and should not be reported in case of progressive download.
			start		Real Time	The time at which the first sample was rendered.
			sstart		Media Time	The presentation time of the first sample rendered.
			duration		Integer	The duration of the continuously presented samples (which is the same in real time and media time). "Continuously presented" means that the media clock continued to advance at the playout speed throughout the interval.
			playbackspeed		Real	The playback speed relative to normal playback speed (i.e.normal forward playback speed is 1.0).
			stopreason		Enum	The reason why continuous presentation of this representation was stopped. Either: - representation switch (not relevant in case of progressive download) - rebuffering - user request - end of period - end of content - end of a metrics collection period - failure - other
			stopreasonother		String	The stopreasonother attribute shall be included only if stopreason attribute is included and has the enum value *other*. In this release of the specification, the sender of this string shall set its value to one of the following:: - switch from unicast to broadcast - switch from broadcast to unicast The receiver of this attribute shall ignore this attribute if its string is set to differerent value than the values listed above.

NOTE: The trace may include entries for different representations that overlap in time, because multiple representations are being rendered simultaneously, for example one audio and one video representation.

The playlist includes user actions about start/stop, but also other non-user actions such as adaptation and rebuffering. Thus the playlist may be used to derive many other metrics, and an example calculation of a few stalling-related metrics is shown below.

Assume a user at wall-clock time hh:mm:ss = 09:00:00 clicks to start a 60-second video with the following playout characteristics:
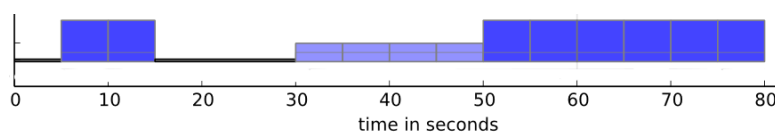
Figure 10-1 Example for Playout

- 5 seconds of initial stalling
- 10 seconds playing with representation 1
- 15 seconds of stalling
- 20 seconds playing with representation 2
- 30 seconds playing with representation 1

This would result in the following (somewhat simplified in this example) playout list being reported by the client:

```
Playlist
  Entry#1
    start = 09:00:00
    mstart = 00:00:00
    starttype = New playout request
    Trace
      Traceentry#1
        representationid = 1
        start = 09:00:05
        sstart = 00:00:00
        duration = 10 seconds
        stopreason = rebuffering
      Traceentry#2
        representationid = 2
        start = 09:00:30
        sstart = 00:00:10
        duration = 20 seconds
        stopreason = representation switch
      Traceentry#3
        representationid = 1
        start = 09:00:50
        sstart = 00:00:30
        duration = 30 seconds
        stopreason = end of content
```

The number of stalling occurances may be calculated by counting how many times a stop reason is specified as "rebuffering".

The time duration for a stalling event may be calculated based on the time difference between the end time of a trace entry with stopreason equal to "rebuffering", and the start time of the next trace entry. In the example above the stalling starts at "Traceentry#1, (start + duration)" = 09:00:05 + 10 secs = 09:00:15, and ends at "Traceentry#2, start" = 09:00:30. Thus the length of the stalling is 15 seconds.

10.2.8 MPD Information

This metric can be used to report Representation information from the MPD, so that reporting servers without direct access to the MPD can understand the used media characteristics.

The metric is reported whenever the client sends any other quality metrics report containing references to a Representation which MPD information has still not been reported.

Table 31 defines the MPD information for quality reporting.

Table 31: MPD Information for Quality Reporting

Key	Type	Description
MPDInformation	Object	
representationid	String	Value of **Representation**@id for the representation addressed by the QoE metrics report.
subreplevel	Integer	If present, value of SubRepresentation@level for the subrepresentation addressed by the QoE metrics report. If not present, the QoE metrics report concerns the representation as a whole.
Mpdinfo	RepresentationType	Provides the MPD information for the representation or subrepresentation identified by representationid and subreplevel, if present. The following attributes and elements shall be present within mpdinfo if they are present for the identified representation or subrepresentation and their values shall be identical to those presented in the MPD: @bandwidth, @qualityRanking, @width, @height, @mimeType, and @codecs.

10.2.9   Playout Delay for Media Start-up

This metric in Table 31a indicates the waiting time that the user experiences for media start-up.

Table 31a: Playout Delay for Media Start-up

Key	Type	Description
PlayoutDelayforMediaStartup	Integer	The playout delay for media start-up is measured as the time in milliseconds from the time instant of DASH player receives play-back-start trigger to the instant of media playout. - If the MPD has been delivered earlier before the user clicks, it may include the process time of MPD, the fetch time of some media segments which are required for media presentation, the process time of segments, and the time for media decode and render to the user. - If no MPD has been fetched earlier, it also needs to add the fetch time of MPD.

10.2.10   Device information

This metric contains information about the displayed video resolution as well as the physical screen characteristics. If the video is rendered in full-screen mode, the video resolution usually coincides with the characteristics of the full physical display. If the video is rendered in a smaller subwindow, the characteristics of the actual video window shown shall be logged.

If known by the DASH client, the physical screen width and the horizontal field-of-view shall also be logged.

The metric is logged at the start of each QoE reporting period, and whenever the characteristics changes during the session (for instance if the UE is rotated from horizontal to vertical orientation, or if the video subwindow size is changed).

Table 31b defines the device information metrics. If an individual metric cannot be logged, its value shall be set to 0 (zero).

Table 31b: Device information

Key	Type	Description
DeviceInformationList	List	A list of device information objects.
Entry	Object	A single object containing new device information.
start	Real-Time	Timestamp when the device information was logged.
mstart	Media-Time	The presentation time at which the device information was logged.
videoWidth	Integer	The width of the displayed video, in screen pixels (not encoded video pixels).
videoHeight	Integer	The height of the displayed video, in screen pixels (not encoded video pixels)
screenWidth	Integer	The total width of the screen, in screen pixels
screenHeight	Integer	The total height of the screen, in screen pixels
pixelWidth	Float	The width of a screen pixel, in mm
pixelHeight	Float	The height of a screen pixel, in mm
fieldOfView	Float	The actual or estimated horizontal angle subtended at the eye by the screen, measured in degrees.

10.3 Quality Metrics for Progressive Download

The following metrics shall be supported by progressive download clients supporting the QoE reporting feature:

- Average Throughput (Clause 10.2.4),

- Initial Playout Delay (Clause 10.2.5),

- Buffer Level (Clause 10.2.6),

- Play List (Clause 10.2.7), and

- Device information (Clause 10.2.10).

10.4 Quality Metrics for DASH

The following metrics shall be supported by 3GP-DASH clients supporting the QoE reporting feature:

- List of of Representation Switch Events (Clause 10.2.3),

- Average Throughput (Clause 10.2.4),

- Initial Playout Delay (Clause 10.2.5),

- Buffer Level (Clause 10.2.6),

- Play List (Clause 10.2.7),

- MPD Information (Clause 10.2.8), and

- Device information (Clause 10.2.10).

The `@metrics` attribute contains a list of quality metric keys listing all metrics that the DASH shall collect and report.

The semantics of the attributes within the **Metrics** element are provided in Table 32. The XML-syntax of a **Metrics** element is provided in Table 33.

Table 32: Semantics of Metrics element

Element or Attribute Name	Use	Description
Metrics		DASH metric element

Element or Attribute Name	Use	Description
@metrics	M	This attribute lists all quality metrics (as a list of quality metric keys as defined in section 10.2, separated by a whitespace) that the client shall report. Certain keys allow specifying a measurement interval or period over which a single value of the metric is derived and potentially also other parameters controlling the collection of the metrics. The parameters, if any, are included in parenthesis after the key and their semantics are specified in clause 10.2 with the metric definition itself.
Range	0..N	When specified, it indicates the time period during which quality metric collection is requested. When not present, quality metric collection is requested for the whole duration of the content.
@starttime	O	When specified, it indicates the start time of the quality metric collection operation. When not present, quality metric collection is requested from the beginning of content consumption. For services with **MPD**@type "Live", the start time of quality metric collection can be obtained in wallclock time by adding the value of this attribute indicated in media time to the value of the **MPD**@availabilityStartTime attribute. For services with **MPD**@type "OnDemand", the start time is indicated in media time and is relative to the *PeriodStart* time of the first period in this MPD.
@duration	O	When specified, it indicates the duration of the quality metric collection interval. The value of this attribute is expressed in media time.
Reporting	1...N	Descriptor that provides information about the requested Quality Reporting method and formats. See clause 10.5 for the 3GP-DASH quality reporting schemes.

Legend:
   For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory.
   For elements: <minOccurs>...<maxOccurs> (N=unbounded)
Elements are **bold**; attributes are non-bold and preceded with an @.

Table 33: XML-Syntax of `Metrics` element

```
<!-- QoE Collection and Reporting -->
<xs:complexType name="MetricsType">
    <xs:sequence>
        <xs:element name="Reporting" type="DescriptorType" maxOccurs="unbounded"/>
        <xs:element name="Range" type="RangeType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="metrics" type="xs:string" use="required"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:complexType name="RangeType">
    <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="startTime" type="xs:duration" use="optional"/>
    <xs:attribute name="duration" type="xs:duration" use="required"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

10.5 Quality Reporting Scheme for DASH

This section specifies a 3GP-DASH quality reporting scheme.

The quality reporting scheme is signaled using in the **Reporting** element in the **Metrics** element. The URN to be used for the **Reporting**@schemeIdUri shall be "urn:3GPP:ns:PSS:DASH:QM10".

The reporting scheme shall use the quality reporting protocol defined in section 10.6.

The semantics and XML syntax of the scheme information for the 3GP-DASH quality reporting scheme are specified in Table 34 and Table 35, respectively.

Table 34: Semantics of Quality Reporting Scheme Information

Element or Attribute Name	Use	Description
@apn	O	This attribute gives the access point that should be used for sending the QoE reports.
@format	O	This field gives the requested format for the reports. Possible formats are: "uncompressed" and "gzip".
@samplepercentage	O	Percentage of the clients that should report QoE. The client uses a random number generator with the given percentage to find out if the client should report or not.
@reportingserver	M	The reporting server URL to which the reports will be sent.
@reportinginterval	O	Indicates the time(s) reports should be sent. If not present, then the client should send a report after the streaming session has ended. If present, @reportingInterval=n indicates that the client should send a report every n-th second provided that new metrics information has become available since the previous report.
LocationFilter	0..1	When present, this element indicates the geographic area(s) or location(s) where quality metric collection is requested. When not present, quality metric collection is requested regardless of the device's location. The LocationFilter element comprises one or more instances of any combination of targeted cell-IDs, polygons and circular areas.Each cell-ID entry in LocationFilter is announced in cellList, and each polygon and circular area entry is announced in the polygonList or and circularAreaList elements, respectively.
cellList	0..N	This element specifies a list of cell identified by E-UTRAN-CGI or CGI.
shape		Geographic area comprising one or more instances of polygonList and/or circularAreaList elements.
polygonList	0..N	This element, when present, comprises a list of 'Polygon' shapes as defined by OMA MLP[51].
@confLevel	O	This attribute indicates the probability in percent that the DASH client is located in the corresponding polygon area. It is defined as 'lev_conf' by OMA MLP. If not present, it has default value of 60.
circularAreaList	0..N	This element, when present, comprises a list of 'CircularArea' shapes as defined by OMA MLP[51].
@confLevel	O	This attribute indicates the probability in percent that the DASH client is located in the corresponding circular area. It is defined as 'lev_conf' by OMA MLP. If not present, it has default value of 60.
Legend: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>…<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and preceded with an @		

Table 35: Syntax of Quality Reporting Scheme Information

```xml
<?xml version="1.0"?>
<xs:schema targetNamespace="urn:3GPP:ns:PSS:AdaptiveHTTPStreaming:2009:qm"
    attributeFormDefault="unqualified"
    elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns="urn:3GPP:ns:PSS:AdaptiveHTTPStreaming:2009:qm">

    <xs:annotation>
        <xs:appinfo>3GPP DASH Quality Reporting</xs:appinfo>
        <xs:documentation xml:lang="en">
            This Schema defines the quality reporting scheme information for 3GPP DASH.
```

```
            </xs:documentation>
        </xs:annotation>

    <xs:element name="ThreeGPQualityReporting" type="SimpleQualityReportingType"/>

    <xs:complexType name="SimpleQualityReportingType">
        <xs:sequence>
            <xs:element name="LocationFilter" type="LocationFilterType" minOccurs="0"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="apn" type="xs:string" use="optional"/>
        <xs:attribute name="format" type="FormatType" use="optional"/>
        <xs:attribute name="samplePercentage" type="xs:double" use="optional"/>
        <xs:attribute name="reportingServer" type="xs:anyURI" use="required"/>
        <xs:attribute name="reportingInterval" type="xs:unsignedInt" use="optional"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:complexType>

    <xs:simpleType name="FormatType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="uncompressed" />
            <xs:enumeration value="gzip" />
        </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="LocationFilterType">
        <xs:sequence>
            <xs:element name="cellID" type="xs:unsignedLong" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="shape" type="ShapeType" minOccurs="0"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="ShapeType">
        <xs:sequence>
            <xs:element name="PolygonList" type="PolygonListType" minOccurs="0"/>
            <xs:element name="CircularAreaList" type="CircularAreaListType" minOccurs="0"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="PolygonListType">
        <xs:annotation>
            <xs:documentation> see [OMA MLP] </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="Polygon" minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ConfLevel" type="xs:unsignedInt" use="optional"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="CircularAreaListType">
        <xs:annotation>
            <xs:documentation> see [OMA MLP] </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="CircularArea" minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ConfLevel" type="xs:unsignedInt" use="optional"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

</xs:schema>
```

10.6   Quality Reporting Protocol

10.6.1   General

The quality reporting protocol consists of:

- The XML-based report format defined in section 10.6.2

- The reporting protocol defined in section 10.6.3

The MIME type of an XML-formatted QoE report shall be "`application/3gpdash-qoe-report+xml`" as defined in Annex J.

10.6.2   Report Format

The QoE report is formatted as an XML document that complies with the following XML schema:

```xml
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:3gpp:metadata:2011:HSD:receptionreport"
    xmlns:sup="urn:3gpp:metadata:2016:PSS:SupplementQoEMetric"
    xmlns:sv="urn:3gpp:metadata:2016:PSS:schemaVersion"
    xmlns="urn:3gpp:metadata:2011:HSD:receptionreport" elementFormDefault="qualified">


    <xs:element name="ReceptionReport" type="ReceptionReportType"/>

    <xs:complexType name="ReceptionReportType">
        <xs:choice>
            <xs:element name="QoeReport" type="QoeReportType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="skip" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:choice>
        <xs:attribute name="contentURI" type="xs:anyURI" use="required"/>
        <xs:attribute name="clientID" type="xs:string" use="optional"/>
    </xs:complexType>

    <xs:complexType name="QoeReportType">
        <xs:sequence>
            <xs:element name="QoeMetric" type="QoeMetricType" minOccurs="1"
maxOccurs="unbounded"/>
            <xs:element ref="sup:supplementQoEMetric" minOccurs="0" maxOccurs="1"/>
            <xs:element ref="sv:delimiter"/>
            <xs:any namespace="##other" processContents="skip" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="periodID" type="xs:string" use="required"/>
        <xs:attribute name="reportTime" type="xs:dateTime" use="required"/>
        <xs:attribute name="reportPeriod" type="xs:unsignedInt" use="required"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="QoeMetricType">
        <xs:choice>
            <xs:element name="HttpList" type="HttpListType"/>
            <xs:element name="RepSwitchList" type="RepSwitchListType"/>
            <xs:element name="AvgThroughput" type="AvgThroughputType" maxOccurs="unbounded"/>
            <xs:element name="InitialPlayoutDelay" type="xs:unsignedInt"/>
            <xs:element name="BufferLevel" type="BufferLevelType"/>
            <xs:element name="PlayList" type="PlayListType"/>
            <xs:element name="MPDInformation" type="MpdInformationType" maxOccurs="unbounded"/>
            <xs:element name="PlayoutDelayforMediaStartup" type="xs:unsignedInt"/>
        </xs:choice>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>

    <xs:complexType name="HttpListType">
        <xs:choice>
            <xs:element name="HttpListEntry" type="HttpListEntryType" maxOccurs="unbounded"/>
```

```xml
            </xs:choice>
            <xs:anyAttribute processContents="skip"/>
    </xs:complexType>

    <xs:complexType name="HttpListEntryType">
        <xs:choice>
            <xs:element name="Trace" type="HttpThroughputTraceType" maxOccurs="unbounded"/>
        </xs:choice>
        <xs:attribute name="tcpid" type="xs:unsignedInt" use="optional"/>
        <xs:attribute name="type" type="ExtensibleHttpEntryResourceType" use="optional"/>
        <xs:attribute name="url" type="xs:string" use="required"/>
        <xs:attribute name="actualUrl" type="xs:string" use="optional"/>
        <xs:attribute name="range" type="xs:string" use="optional"/>
        <xs:attribute name="trequest" type="xs:dateTime" use="required"/>
        <xs:attribute name="tresponse" type="xs:dateTime" use="required"/>
        <xs:attribute name="responsecode" type="xs:unsignedInt" use="optional"/>
        <xs:attribute name="interval" type="xs:unsignedInt" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>

    <xs:simpleType name="HttpEntryResourceType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="MPD"/>
            <xs:enumeration value="MPDDeltaFile"/>
            <xs:enumeration value="XLinkExpansion"/>
            <xs:enumeration value="InitializationSegment"/>
            <xs:enumeration value="IndexSegment"/>
            <xs:enumeration value="MediaSegment"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="StringPatternType">
        <xs:restriction base="xs:string">
            <xs:pattern value="x:\S.*"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="ExtensibleHttpEntryResourceType">
        <xs:union memberTypes="HttpEntryResourceType StringPatternType"/>
    </xs:simpleType>

    <xs:complexType name="HttpThroughputTraceType">
        <xs:attribute name="s" type="xs:dateTime" use="required"/>
        <xs:attribute name="d" type="xs:unsignedInt" use="required"/>
        <xs:attribute name="b" type="UnsignedIntVectorType" use="required"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>

    <xs:complexType name="RepSwitchListType">
        <xs:choice>
            <xs:element name="RepSwitchEvent" type="RepSwitchEventType" maxOccurs="unbounded"/>
        </xs:choice>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>

    <xs:complexType name="RepSwitchEventType">
        <xs:attribute name="to" type="xs:string" use="required"/>
        <xs:attribute name="mt" type="xs:duration" use="optional"/>
        <xs:attribute name="t" type="xs:dateTime" use="optional"/>
        <xs:attribute name="lto" type="xs:unsignedInt" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>

    <xs:complexType name="AvgThroughputType">
        <xs:attribute name="numBytes" type="xs:unsignedInt" use="required"/>
        <xs:attribute name="activityTime" type="xs:unsignedInt" use="required"/>
        <xs:attribute name="t" type="xs:dateTime" use="required"/>
        <xs:attribute name="duration" type="xs:unsignedInt" use="required"/>
        <xs:attribute name="accessbearer" type="xs:string" use="optional"/>
        <xs:attribute name="inactivityType" type="InactivityType" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>

    <xs:simpleType name="InactivityType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Pause"/>
            <xs:enumeration value="BufferControl"/>
            <xs:enumeration value="Error"/>
```

```xml
                    </xs:restriction>
        </xs:simpleType>
        <xs:complexType name="BufferLevelType">
            <xs:choice>
                <xs:element name="BufferLevelEntry" type="BufferLevelEntryType"
maxOccurs="unbounded"/>
            </xs:choice>

            <xs:anyAttribute processContents="skip"/>
        </xs:complexType>

        <xs:complexType name="BufferLevelEntryType">
            <xs:attribute name="t" type="xs:dateTime" use="required"/>
            <xs:attribute name="level" type="xs:unsignedInt" use="required"/>
            <xs:anyAttribute processContents="skip"/>
        </xs:complexType>

        <xs:complexType name="PlayListType">
            <xs:choice>
                <xs:element name="Trace" type="PlayListEntryType" maxOccurs="unbounded"/>
            </xs:choice>

            <xs:anyAttribute processContents="skip"/>
        </xs:complexType>

        <xs:complexType name="PlayListEntryType">
            <xs:choice>
                <xs:element name="TraceEntry" type="PlayListTraceEntryType" maxOccurs="unbounded"/>
            </xs:choice>
            <xs:attribute name="start" type="xs:dateTime" use="required"/>
            <xs:attribute name="mstart" type="xs:duration" use="required"/>
            <xs:attribute name="startType" type="StartType" use="required"/>
            <xs:anyAttribute processContents="skip"/>
        </xs:complexType>

        <xs:complexType name="PlayListTraceEntryType">
            <xs:attribute name="representationId" type="xs:string" use="optional"/>
            <xs:attribute name="subrepLevel" type="xs:unsignedInt" use="optional"/>
            <xs:attribute name="start" type="xs:dateTime" use="required"/>
            <xs:attribute name="sstart" type="xs:duration" use="required"/>
            <xs:attribute name="duration" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="playbackSpeed" type="xs:double" use="optional"/>
            <xs:attribute name="stopReason" type="StopReasonType" use="optional"/>
            <xs:attribute name="stopReasonOther" type="xs:string" use="optional"/>
            <xs:anyAttribute processContents="skip"/>
        </xs:complexType>

        <xs:simpleType name="StartType">
            <xs:restriction base="xs:string">
                <xs:enumeration value="NewPlayoutRequest"/>
                <xs:enumeration value="Resume"/>
                <xs:enumeration value="OtherUserRequest"/>
                <xs:enumeration value="StartOfMetricsCollectionPeriod"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="StopReasonType">
            <xs:restriction base="xs:string">
                <xs:enumeration value="RepresentationSwitch"/>
                <xs:enumeration value="Rebuffering"/>
                <xs:enumeration value="UserRequest"/>
                <xs:enumeration value="EndOfPeriod"/>
                <xs:enumeration value="EndOfContent"/>
                <xs:enumeration value="EndOfMetricsCollectionPeriod"/>
                <xs:enumeration value="Failure"/>
                <xs:enumeration value="Other"/>
            </xs:restriction>
        </xs:simpleType>

        <xs:complexType name="MpdInformationType">
            <xs:choice>
                <xs:element name="Mpdinfo" type="RepresentationType" maxOccurs="unbounded"/>
            </xs:choice>
            <xs:attribute name="representationId" type="xs:string" use="required"/>
            <xs:attribute name="subrepLevel" type="xs:unsignedInt" use="optional"/>
            <xs:anyAttribute processContents="skip"/>
        </xs:complexType>

        <xs:complexType name="RepresentationType">
```

```
                <xs:attribute name="codecs" type="xs:string" use="required"/>
                <xs:attribute name="bandwidth" type="xs:unsignedInt" use="required"/>
                <xs:attribute name="qualityRanking" type="xs:unsignedInt" use="optional"/>
                <xs:attribute name="frameRate" type="xs:double" use="optional"/>
                <xs:attribute name="width" type="xs:unsignedInt" use="optional"/>
                <xs:attribute name="height" type="xs:unsignedInt" use="optional"/>
                <xs:attribute name="mimeType" type="xs:string" use="required"/>
                <xs:anyAttribute processContents="skip"/>
        </xs:complexType>

        <xs:simpleType name="DoubleVectorType">
                <xs:list itemType="xs:double"/>
        </xs:simpleType>

        <xs:simpleType name="StringVectorType">
                <xs:list itemType="xs:string"/>
        </xs:simpleType>

        <xs:simpleType name="UnsignedIntVectorType">
                <xs:list itemType="xs:unsignedInt"/>
        </xs:simpleType>

 </xs:schema>
```

The following schema is an extension to allow additional QoE metrics.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="urn:3gpp:metadata:2016:PSS:SupplementQoEMetric"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:3gpp:metadata:2016:PSS:SupplementQoEMetric"
    elementFormDefault="qualified">

    <xs:element name="supplementQoEMetric" type="SupplementQoEMetricType"/>

    <xs:complexType name="SupplementQoEMetricType">
        <xs:sequence>
            <xs:element name="deviceinformation" type="DeviceInformationType" minOccurs="0"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="DeviceInformationType">
        <xs:choice>
            <xs:element name="Entry" type="DeviceInformationEntryType" maxOccurs="unbounded"/>
        </xs:choice>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>

    <xs:complexType name="DeviceInformationEntryType">
        <xs:attribute name="start" type="xs:dateTime" use="required"/>
        <xs:attribute name="mstart" type="xs:duration" use="required"/>
        <xs:attribute name="videoWidth" type="xs:unsignedInt" use="required"/>
        <xs:attribute name="videoHeight" type="xs:unsignedInt" use="required"/>
        <xs:attribute name="screenWidth" type="xs:unsignedInt" use="required"/>
        <xs:attribute name="screenHeight" type="xs:unsignedInt" use="required"/>
        <xs:attribute name="pixelWidth" type="xs:double" use="required"/>
        <xs:attribute name="pixelHeight" type="xs:double" use="required"/>
        <xs:attribute name="fieldOfView" type="xs:double" use="required"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
</xs:schema>

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema  xmlns="urn:3gpp:metadata:2016:PSS:schemaVersion"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            targetNamespace="urn:3gpp:metadata:2016:PSS:schemaVersion"
            elementFormDefault="qualified">

    <xs:element name="schemaVersion" type="xs:unsignedInt"/>
    <xs:element name="delimiter" type="xs:byte"/>

</xs:schema>
```

Note: If a supplementQoEMetric needs to be sent when no ordinar QoEMetric are due, a dummy MPDInformation metric shall be sent with codecs="none", bandwidth=0, mimeType="none", representationId="none".

10.6.3   Reporting Protocols

For configuration done via the QMC functionality (see Annex L), the client shall also send QoE reports via the QMC functionality. For MPD or OMA-DM configuration, if a specific metrics server has been configured, the client shall send QoE reports using the HTTP (RFC 2616) [9] POST request carrying XML formatted metadata in its body.
An example QoE reporting based on HTTP POST request signalling is shown below:

```
POST http://www.exampleserver.com HTTP/1.1
Host: 192.68.1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
Content-Type: text/xml; charset=utf-8
Content-Length: 4408
<?xml version="1.0"?>
<ReceptionReport contentURI="http://www.example.com/content/content.mpd" clientID="35848574673"
xmlns="urn:3gpp:metadata:2011:HSD:receptionreport"
    xsi:schemaLocation="urn:3gpp:metadata:2011:HSD:receptionreport DASH-QoE-Report.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <QoeReport periodID="Period1" reportTime="2011-02-16T09:00:00" reportPeriod="500">
        <QoeMetric>
            <HttpList>
                <HttpListEntry type="MPD" url="http://www.example.com/content/content.mpd"
trequest="2011-02-16T08:59:30" tresponse="2011-02-16T08:59:31" interval="50">
                    <Trace s="2011-02-16T08:59:30Z" d="171" b="2367 1990 2463 1254"/>
                </HttpListEntry>
                <HttpListEntry type="InitializationSegment"
url="http://www.example.com/content/initRep1.3gp" trequest="2011-02-16T08:59:40"
tresponse="2011-02-16T08:59:41" interval="200">
                    <Trace s="2011-02-16T08:59:40.5Z" d="159" b="9345"/>
                </HttpListEntry>
                <HttpListEntry type="InitializationSegment"
url="http://www.example.com/content/initRep2.3gp" trequest="2011-02-16T08:59:41"
tresponse="2011-02-16T08:59:42" interval="200">
                    <Trace s="2011-02-16T08:59:41.5Z" d="123" b="6723"/>
                </HttpListEntry>
                <HttpListEntry type="InitializationSegment"
url="http://www.example.com/content/initRep3.3gp" trequest="2011-02-16T08:59:42"
tresponse="2011-02-16T08:59:43" interval="200">
                    <Trace s="2011-02-16T08:59:42.5Z" d="195" b="9786"/>
                </HttpListEntry>
            </HttpList>
        </QoeMetric>
        <QoeMetric>
            <InitialPlayoutDelay>10000</InitialPlayoutDelay>
        </QoeMetric>
    </QoeReport>
    <QoeReport periodID="Period1" reportTime="2011-02-16T09:08:20" reportPeriod="500">
        <QoeMetric>
            <BufferLevel>
                <BufferLevelEntry t="2011-02-16T09:08:19" level="84673"/>
                <BufferLevelEntry t="2011-02-16T09:08:20" level="93874"/>
            </BufferLevel>
        </QoeMetric>
        <QoeMetric>
            <RepSwitchList>
                <RepSwitchEvent to="Rep2"/>
                <RepSwitchEvent to="Rep3"/>
            </RepSwitchList>
        </QoeMetric>
    </QoeReport>
</ReceptionReport>
```

11 Live Services

11.1 Overview Dynamic and Live Media Presentations

DASH Media Presentations with **MPD**@type set to "dynamic" enable that media is made available over time and its availability may also be removed over time. This has two major effects, namely

1) The content creator can announce a DASH Media Presentation for which not all content is yet available, but only gets available over time.

2) Clients are forced into a timed schedule for the playout, such that they follow the schedule as desired by the content author.

Dynamic services may be used for different types of services:

1) **Dynamic Distribution of Available Content**: Services, for which content is made available as dynamic content, but the content is entirely generated prior to distribution. In this case the details of the Media Presentation, especially the Segments (duration, URLs) are known and can be announced in a single MPD without MPD updates.

2) **MPD-controlled Live Service**: Services for which the content is typically generated on the fly, and the MPD needs to be updated occasionally to reflect changes in the service offerings, e.g. announcing a new Period with new media or providing the end of the Media Presentation. The DASH client operates solely on information in the MPD.

Dynamic and Live services are typically controlled by different client transactions and server-side signaling.

For initial access to the service and joining the service, an MPD is required. MPDs may be accessed at join time or may have been provided earlier, for example along with an Electronic Service Guide. The initial MPD or join MPD is accessed and processed by the client and the client is synchronized with the server can analyze the MPD and extract suitable information in order to initiate the service. This includes, but is not limited to:

- identifying the currently active Periods in the service and the Period that expresses the live edge (for more details see below)

- selecting the suitable media components by selecting one or multiple Adaptation Sets. Within each Adaptation Set selecting an appropriate Representation and identifying the live edge segment in each Representations. The client then issues requests for the Segments.

The MPD may be updated on the server based on certain rules and clients consuming the service are expected to update MPDs based on certain triggers.

This clause provides requirements and recommendations for different functions, namely

- for the dynamic segment download in clause 11.2.

- for services with MPD updates in clause 11.3

- for offering live generated services in on-demand mode in clause 11.4

- for client-server timing synchronization in clause 11.5.

- for robust service offerings and corresponding clients in clause 11.6.

11.2 Dynamic Segment Download

11.2.1 Background and Assumptions

The dynamic segment download function is a key component of live services, In addition, the dynamic segment download function may also be used for scheduling playout of on-demand content by the content author. In this subsection, it is assumed that the client has access to a single instance of an MPD and all information of the entire Media Presentation is contained in the MPD.

The timing model of dynamic services, especially the generation of Segment availabilities is a relevant component of a live services. This forms the basis for live services and explains the key concepts and rules for Segment availabilities.

11.2.2 MPD Information and Timing Model

11.2.2.1 MPD Information

If the Media Presentation is of type dynamic, then Segments have different Segment availability times, i.e. the earliest time for which the service provider permits the DASH client to issue a request to the Segment and guarantees, under regular operation modes, that the client gets a 200 OK response for the Segment. The Segment availability times for each Representation can be computed based on the information in an MPD.

For a dynamic service the MPD should at least contain information as available in Table 11-1. Information included there may be used to compute a list of announced Segments, Segment Availability Times and URLs.

Assume that an MPD is available to the DASH client at a specific wall-clock time *NOW*. It is assumed that the client and the DASH server providing the Segments are synchronized to wall-clock, either through external means or through a specific client-server synchronization. Details on synchronization are provided in clause 11.6.

Assuming synchronization, the information in the MPD can then be used by the client at time *NOW* to derive the availability (or non-availability) of Segments on the server.

Table 11-1 -- Information related to Segment Information and Availability Times for a dynamic service

MPD Information	Status	Comment
MPD@type	mandatory, set to "dynamic"	the type of the Media Presentation is dynamic, i.e. Segments get available over time.
MPD@availabilityStartTime	mandatory	the start time is the anchor for the MPD in wall-clock time. The value is denoted as *AST* in the following.
MPD@mediaPresentationDuration	mandatory (for the considered use cases)	provides the duration of the Media Presentation.
MPD@suggestedPresentationDelay	optional, but recommended	suggested presentation delay as delta to segment availability start time. The value is denoted as *SPD*. Details on the setting and usage of the parameter is provided in the following.
MPD@minBufferTime	mandatory	minimum buffer time, used in conjunction with the @bandwidth attribute of each Representation. The value is denoted as *MBT*. Details on the setting and usage of the parameter is provided in the following.
MPD@timeShiftBufferDepth	optional, but recommended	time shift buffer depth of the media presentation. The value is denoted as *TSB*. Details on the setting and usage of the parameter is provided in the following.
Period@start	Mandatory for the first Period in the MPD	the start time of the Period relative to the MPD availability start time.
Representation@availabilityTimeOffset	Optional default	The offset in availability time for this Representation. It may also be available on a Base URL or default NOTE: the value of "INF" implies availability of all segments starts at **MPD**@availabilityStartTime
SegmentTemplate@media	mandatory	The template for the Media Segment assigned to a Representation.

`SegmentTemplate` `@startNumber`	optional default	number of the first segment in the Period assigned to a Representation
`SegmentTemplate` `@timescale`	optional default	timescale for this Representation.
`SegmentTemplate` `@duration`	mandatory	the duration of each Segment in units of a time.

11.2.2.2   Segment Information Derivation

11.2.2.2.1   Introduction

Based on an MPD including information as documented in Table 11-1 and available at time *NOW* on the server, a synchronized DASH client derives the information of the list of Segments for each Representation in each Period. This section only describes the information that is expressed by the values in the MPD. The generation of the information on the server and the usage of the information in the client is provided in clause 11.2.3 and 11.2.4 respectively.

11.2.2.2.2   Definitions

The following definitions are relevant and aligned with ISO/IEC 23009-1:

Based on an MPD including information as documented in Table 11-1 and available at time NOW on the server, a synchronized DASH client derives the information of the list of Segments for each Representation in each Period. This clause only describes the information that is expressed by the values in the MPD. The generation of the information on the server and the usage of the information in the client is provided in clause 11.2.3 and 11.2.4 respectively.

11.2.2.2.2 Definitions

The following definitions are relevant and aligned with ISO/IEC 23009-1:

- available Segment is a Segment that is accessible at its assigned HTTP-URL. This means that a request with an HTTP GET to the URL of the Segment results in a reply of the Segment and 2xx status code.

- valid Segment URL is an HTTP-URL that is promised to reference a Segment during its Segment availability period.

- NOW is a time that is expressing the time on the content server as wall-clock time. All information in the MPD related to wall-clock is expressed as a reference to the time NOW.

11.2.2.2.3   MPD Information

For a dynamic service without MPD updates, the following information shall be present and not present in the MPD (also please refer to Table 11-1):

- The **MPD**@type shall be set to "dynamic".

- The **MPD**@mediaPresentationDuration shall be present, or the **Period**@duration of the last Period shall be present.

- The **MPD**@minimumUpdatePeriod shall not be present.

Furthermore, it is recommended to provide values for **MPD**@timeShiftBufferDepth and **MPD**@suggestedPresentationDelay.

11.2.2.2.4   Period Information

Each Period is documented by a **Period** element in the MPD. An MPD may contain one or more Periods. In order to document the use of multiple Periods, the sequence of Period elements is expressed by an index *i* with *i* increasing by 1 for each new Period element.

- Each regular Period *i* in the MPD is assigned a

  - Period start time *PSwc*[*i*] in wall-clock time,

  - Period end time *PEwc*[*i*], in wall-clock time.

NOTE: An MPD update may extend the Period end time of the last regular Period. For details refer to section 11.3.

The Period start time *PSwc*[*i*] for a regular Period *i* is determined according to section 5.3.2.1 of ISO/IEC 23009-1:

- If the attribute @start is present in the **Period**, then *PSwc*[*i*] is the sum of *AST* and the value of this attribute.

- If the @start attribute is absent, but the previous **Period** element contains a @duration attribute then the start time of the Period is the sum of the start time of the previous Period *PSwc*[*i*] and the value of the attribute @duration of the previous Period. Note that if both are present, then the @start of the new Period takes precedence over the information derived from the @duration attribute.

The Period end time *PEwc*[*i*] for a regular Period *i* is determined as follows:

- If the Period is the last one in the MPD, the time *PEwc*[*i*] is obtained as

  - the sum of *AST* and Media Presentation Duration *MPDur*, with *MPDur* the value of **MPD**@mediaPresentationDuration if present, or the sum of *PSwc*[*i*] of the last Period and the value of **Period**@duration of the last Period.

Else

  - the time *PEwc*[*i*] is obtained as the Period start time of the next Period, i.e. *PEwc*[*i*] = *PSwc*[*i*+1].

11.2.2.2.5   Representation Information

Based on such an MPD at a specific time *NOW*, a list of Segments contained in a Representation in a Period *i* with Period start time *PSwc*[*i*] and Period end time *PEwc*[*i*] can be computed.

Let

- N_s=1,

- *ato* is the value of the @availabilityTimeOffset attribute, if present. Otherwise it is zero.

- *ts* the value of the @timescale attribute

- *t*[*s*] is 0,

- the *d*[*s*] is the value of @duration attribute

- *r*[*s*] is the ceil of (*PEwc*[*i*] - *PSwc*[*i*] - *t*[*s*]/*ts*)*ts*/*d*[*s*])

11.2.2.2.6   Media Time Information of Segment

Each Media Segment at position *k=1,2, ...* for each Representation has assigned an earliest media presentation time *EPT*[*k,r,i*] and an accurate segment duration *SDUR[k,r,j]*, all measured in media presentation time. The earliest presentation time may be estimated from the MPD using the segment availability start time minus the segment duration announced in the MPD. The earliest presentation time may be accurately determined from the Segment itself.

11.2.2.2.7   Segment List Parameters

For each Period *i* with Period start time *PSwc*[*i*] and Period end time *PEwc*[*i*] and each Representation *r* in the Period the following information can be computed:

- the presentation time offset described in the MPD, o[*i,r*]

- the availability time offset of this Representation, ato[*r*]

- the number of the first segment described in the MPD, k1[*i,r*]

- the number of the last segment described in the MPD, k2[*i,r*]

- segment availability start time of the initialization segment SAST[*0,i,r*]

- segment availability end time of the initialization segment SAET[*0,i,r*]

- segment availability start time of each media segment SAST[*k,i,r*], *k*=k1, ..., k2

- segment availability end time of each media segment SAET[*k,i,r*], *k*=k1, ..., k2

- adjusted segment availability start time ASAST[*0,i,r*], *k*=0, k1, ..., k2

- segment duration of each media segment SD[*k,i,r*], k=k1, ..., k2

- the URL of each of the segments, URL[*k,i,r*]

In addition,

- the latest available Period *i*[*NOW*] and the latest segment available at the server k[*NOW*] can be computed. This segment is also referred to as *live edge segment*.

- the earliest available Period *i**[*NOW*] and the earliest segment available at the server k*[*NOW*] can be computed.

Based on the above information, for each Representation *r* in a Period *i*, the segment availability start time *SAST[k,i,r]*, the segment availability end time of each segment *SAET[k,i,r]*, the segment duration of each segment *SD[k,i,r]*, and the URL of each of the segments, *URL[k,i,r]* within one Period *i* be derived as follows using the URL Template function `URLTemplate(ReplacementString, Address)`:

- k=0

- SAST[0,i,r] = PSwc[i]

- ASAST[0,i,r] = PSwc[i] - ato

- for s=1, ... N_s [i,r]

  - $k = k + 1$

  - $SAST[k,i,r] = PSwc[i] + (t[s,i,r] + d[s,i,r] - o[i,r])/ts$

  - $ASAST[k,i,r] = SAST[k,i,r] – ato$

  - $SD[k,i,r] = d[s,i,r]/ts$

  - $SAET[k,i,r] = SAST[k,i,r] + TSB + d[s,i,r]/ts$

  - if **SegmentTemplate**@media contains $Number$

    - *Address=*@startNumber

    - *URL[k,i,r]* = URLTemplate ($Number$, *Address*)

  - for $j = 1, ..., r[s,i,r]$

    - $k = k + 1$

    - $SAST[k,i,r] = SAST[k-1,i,r] + d[s,i,r]/ts$

    - $ASAST[k,i,r] = SAST[k,i,r] – ato$

    - $SAET[k,i,r] = SAST[k,i,r] + TSB + d[s,i,r] /ts$

    - $SD[k,i,r] = d[s,i,r] /ts$

- if **SegmentTemplate**@media contains $Number$

    - Address = Address + 1

    - URL[k,i,r] = URLTemplate ($Number$, Address)

- k2[*i,r*] = *k*

- *SAET*[*0,i,r*] = *SAET*[k2[*i,r*]*,i,r*]

Note that not all segments documented above may necessarily be accessible at time *NOW*, but only those that are within the segment availability time window. Hence, the number of the first media segment described in the MPD for this Period, k1[*i,r*], is the smallest *k*=1, 2, ... for which *SAST*[*k,i,r*] >= *NOW*.

The latest available Period *i*[*NOW*] is the Period *i* with the largest *PEwc*[*i*] and *PEwc*[*i*] is smaller than or equal to *NOW*.

The latest available segment *k*[*NOW*] available for a Representation of Period *i*[*NOW*] (also the live edge segment) is the segment with the largest *k*=0,1,2,... such that *SAST*[*k,i,r*] is smaller than or equal to *NOW*. Note that this contains the Initialization Segment with *k*=0 as not necessarily any media segment may yet be available for Period *i*[*NOW*]. In this case, last media segment k2[*i*[*NOW*]-1*,r*], i.e., the last media segment of the previous Period is the latest accessible media Segment.

However, if the @availabilityTimeOffset is present, then the segments for this Representation are available earlier than the nominal segment availability start time, namely at *ASAST*[*k,i,r*].

11.2.2.2.8   URL Generation with Segment Template

The function URL Template function URLTemplate(ReplacementString, Address) generates a URL. For details refer to ISO/IEC 23009-1, section 5.3.9.4. Once the Segment is generated, processing of the Base URLs that apply on this segment level is done as defined in ISO/IEC 23009-1, section 5.6.

11.2.3   Service Offering Requirements and Guidelines

11.2.3.1   General Service Offering Requirements

For dynamic service offerings, the MPD shall conform to the MPD in clause 8 and shall at least contain the mandatory information as documented in Table 11-1.

If such an MPD is accessible at time *NOW* at the location **MPD.Location**, then

- all Segments for all Representations in all Periods as announced in an MPD shall be available latest at the announced segment availability start time *SAST*[*k,i,r*] at all *URL*[*k,i,r*] as derived in section 11.2.2.2;

- all Segments for all Representations in all Periods as announced in an MPD shall at least be available until the announced segment availability end time *SAET*[*k,i,r*] at all *URL*[*k,i,r*] as derived in section 11.2.2.2;

- for all Media Segments for all Representations in all Periods as announced in an MPD the Segment in this Period is available prior to the sum of Period start, earliest presentation time and segment duration, i.e. *SAST*[*k,i,r*] <= *PSwc*[*i*] + *SD*[*k,r,i*] + *EPT*[*k,r,i*];

- if a Media Segments with segment number *k* is delivered over a constant bitrate channel with bitrate equal to value of the @bandwidth attribute then each presentation time *PT* is available at the client latest at time with a delay of at most *PT* + *MBT*.

11.2.3.2   Dynamic Service Offering Guidelines

11.2.3.2.1   Introduction

In order to offer a simple dynamic service for which the following details are known in advance,

- start at wall-clock time `START`,

- exact duration of media presentation `PDURATION`,

- location of the segments for each Representation at "`http://example.com/$RepresentationID$/$Number$`",

a service provide may offer an MPD as follows:

Table 11-2 – Basic Service Offering

MPD Information	Value
MPD@type	dynamic
MPD@availabilityStartTime	START
MPD@mediaPresentationDuration	PDURATION
MPD@suggestedPresentationDelay	SPD
MPD@minBufferTime	MBT
MPD@timeShiftBufferDepth	TSB
MPD.BaseURL	"http://example.com/"
Period@start	PSTART
Representation@bandwidth	BW
SegmentTemplate@media	"$RepresentationID$/$Number$"
SegmentTemplate@startNumber	1
SegmentTemplate@duration	SDURATION

Note that the setting of capitalized parameters is discussed in clause 11.2.3.2.2.

Based on the details in clause 11.2.2.2, the Segment Information is derived as:

- k1 = 1

- k2 = `ceil(PDURATION/SDURATION)`

- for k = 1, ..., k2

  - SAST[k] = START + PSTART + k*SDURATION

  - SAET[k] = SAST[k] + TSB + SDURATION

  - SD[k] = SDURATION

  - URL[k] = http://example.com/$RepresentationID$/k

- The segment availability times of the Initialization Segment are as follows:

  - SAST[0] = START + PSTART

  - SAET[0] = SAET[k2]

11.2.3.2.2 Basic Parameter Settings

In the following recommendations are provided for the

- Time Shift Buffer Depth (`TSB`):

  - If the content should be consumed at the live edge, then the time shift buffer depth should be set short. However, the `TSB` should not be smaller than the recommended value of 4*`SDURATION` and 6 seconds in media time in order for the client to do some prebuffering in more difficult network conditions.

  - If no restrictions on the accessibility of the content are provided, then the `TSB` may be set to a large value that even exceeds `PDURATION`.

- Suggested Presentation Delay (`SPD`)

- If synchronized play-out with other devices adhering to the same rule is desired and/or the service provider wants to define the typical live edge of the program, then this value should be provided. The service provider should set the value taking into account at least the following:

  - the desired end-to-end latency

  - the typical required buffering in the client, for example based on the network condition

  - the segment duration `SDURATION`

  - the time shift buffer depth `TSB`

- A reasonable value may be 2 to 4 times of the segment duration `SDURATION`, but the time should not be smaller than 4 seconds in order for the client to maintain some buffering.

- Segment Duration (`SDURATION`)

  - The segment duration typically influences the end-to-end latency, but also the switching and random access granularity as in DASH-264/AVC each segment starts with a stream access point which can also be used as s switch point. The service provider should set the value taking into account at least the following:

    - the desired end-to-end latency

    - the desired compression efficiency

    - the start-up latency

    - the desired switching granularity

    - the desired amount of HTTP requests per second

    - the variability of the expected network conditions

  - Reasonable values for segment durations are between 1 second and 10 seconds.

- Minimum Buffer Time (`MBT`) and bandwidth (`BW`)

  - the value of the minimum buffer time **does not provide any instructions to the client on how long to buffer the media**. The value describes how much buffer a client should have under *ideal* network conditions. As such, `MBT` is not describing the burstiness or jitter in the network, it is describing the burstiness or jitter in the **content encoding**. Together with the `BW` value, it is a property of the content. Using the "leaky bucket" model, it is the size of the bucket that makes `BW` true, given the way the content is encoded.

  - The minimum buffer time provides information that for each Stream Access Point (and in the case of DASH-IF therefore each start of the Media Segment), the property of the stream: If the Representation (starting at any segment) is delivered over a constant bitrate channel with bitrate equal to value of the `BW` attribute then each presentation time *PT* is available at the client latest at time with a delay of at most *PT* + *MBT*.

  - In the absence of any other guidance, **the `MBT` should be set** to the maximum GOP size (coded video sequence) of the content, which quite often is identical **to the maximum segment duration**. The *MBT* may be set to a smaller value than maximum segment duration, but should not be set to a higher value.

11.2.3.3   Content Offering with Periods

11.2.3.3.1   General

For content offered within a Period, and especially when offered in multiple Periods, then the content provider should offer the content such that actual media presentation time is as close as possible to the actual Period duration. It is recommended that the Period duration is the maximum of the presentation duration of all Representations contained in the Period.

A typical Multi-Period Offering is shown in Table 11-4. This may for example represent a service offering where main content provided in Period 1 and Period 3 are interrupted by an inserted Period 2.

Table 6 Multi-Period Service Offering

MPD Information	Value
MPD@type	dynamic
MPD@availabilityStartTime	START
MPD@mediaPresentationDuration	PDURATION
MPD@suggestedPresentationDelay	SPD
MPD@minBufferTime	MBT
MPD@timeShiftBufferDepth	TSB
MPD.BaseURL	"http://example.com/"
Period@start	PSTART
SegmentTemplate@media	"1/$RepresentationID$/$Number$"
SegmentTemplate@startNumber	1
SegmentTemplate@duration	SDURATION1
Period@start	PSTART2
Representation@availabilityTimeOffset	ATO2
SegmentTemplate@media	"2/$RepresentationID$/$Number$"
SegmentTemplate@startNumber	1
SegmentTemplate@duration	SDURATION2
Period@start	PSTART3
SegmentTemplate@media	"1/$RepresentationID$/$Number$"
SegmentTemplate@startNumber	STARTNUMBER2
SegmentTemplate@duration	SDURATION1
SegmentTemplate@presentationTimeOffset	PTO

Based on the details in section 11.2.2.2, the Segment Information is derived as:

- Period 1

    - PSwc[1] = START + PSTART

    - PEwc[1] = START + PSTART2

    - k1 = 1

    - k2 = ceil((PSTART2-PSTART1)/SDURATION)

    - for k = 1, ..., k2

        - SAST[k] = PSwc[1] + k*SDURATION

        - SAET[k] = SAST[k] + TSB + SDURATION

        - SD[k] = SDURATION

        - URL[k] = http://example.com/1/$RepresentationID$/k

    - SAST[0] = PSwc[1]

    - SAET[0] = SAET[k2]

- Period 2

    - PSwc[2] = START + PSTART2

    - PEwc[2] = START + PSTART3

- k1 = 1

- k2 = `ceil((PSTART3-PSTART2)/SDURATION2)`

- for k = 1, ..., k2

    - SAST[k] = PSwc[2] + k*`SDURATION2`

    - ASAST[k] = SAST[k] – `ATO2`

    - SAET[k] = SAST[k] + `TSB` + `SDURATION2`

    - SD[k] = `SDURATION2`

    - URL[k] = http://example.com/2/$RepresentationID$/k

- SAST[0] = PSwc[2]

- SAET[0] = SAET[k2]

- Period 3

    - PSwc[3] = `START` + `PSTART3`

    - PEwc[3] = `START` + `PDURATION`

    - k1 = 1

    - k2 = `ceil((PSTART3-PDURATION)/SDURATION1)`

    - for k = 1, ..., k2

        - SAST[k] = PSwc[3] + k*`SDURATION1`

        - SAET[k] = SAST[k] + `TSB` + `SDURATION1`

        - SD[k] = `SDURATION1`

        - URL[k] = " http://example.com/1/$RepresentationID$/(k+STARTNUMBER2-1) "

    - SAST[0] = PSwc[3]

    - SAET[0] = SAET[k2]

Note that the number k describes position in the Period. The actual number used in the segment template increased by the one less than the actual start number.

11.2.3.4   Joining Recommendation

By default, an MPD with **MPD**`@type="dynamic"` suggests that the client would want to join the stream at the live edge, therefore to download the latest available segment (or close to, depending on the buffering model), and then start playing from that segment onwards.

However there are circumstances where a dynamic MPD might be used with content intended for playback from the start, or from another position. For example, when a content provider offers 'start again' functionality for a live program, the intention is to make the content available as an on-demand program, but not all the segments will be available immediately.

This may be signalled to the DASH client by including an MPD Anchor, with either

- the `t` parameter, or

- both the `period` and `t` parameter, in the MPD URL provided to the DASH client, or

The format and behaviour of MPD Anchors is defined in section C.4 of ISO/IEC 23009-1.

For example to start from the beginning of the MPD the following would be added to the end of the MPD URL provided to the DASH client:

- #t=0

Or to start from somewhere other than the start, in this case 50 minutes from the beginning of the period with Period ID "`program_part_2`":

- #period=program_part_2&t=50:00

Where an MPD Anchor is used it should refer to a time for which segments are currently available in the MPD.

11.2.4 Client Operation, Recommendations and Guidelines (informative)

11.2.4.1 Basic Operation

A DASH client is guided by the information provided in the MPD. A simple client model is shown in Figure 11-1.
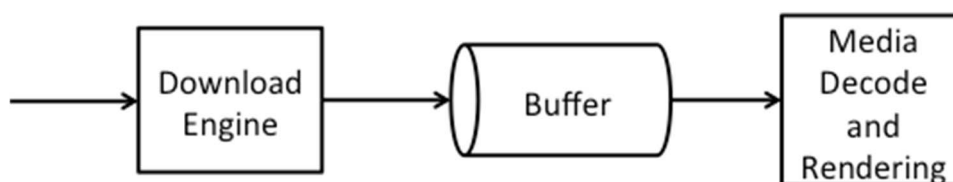
Figure 11-1 Simple Client Model

Assume that the client has access to an MPD and the MPD contains the parameters in Table 11-1 and Table 11-3, i.e. it consumes a dynamic service with fixed media presentation duration, but possibly multiple Periods

The following example client behaviour may provide a continuous streaming experience to the user:

1) The client parses the MPD, selects a collection of Adaptation Sets suitable for its environment based on information provided in each of the **AdaptationSet** elements.

2) Within each Adaptation Set it selects one Representation, typically based on the value of the @bandwidth attribute, but also taking into account client decoding and rendering capabilities.

The client creates a list of accessible Segments at least for each selected Representation taking into account the information in the MPD as documented in Table 11-1 and Table 11-3 and the current time *JOIN* in the client and in particular the segment closest to the live edge referred to the *live edge segment*. For details refer to section 11.2.4.2.
For this it needs to take into account the latest Period *i*[*NOW*]. The latest Period and the latest segment are obtained as follows with i* the index of the last Period:

- if *NOW* <= PSwc[1]

- no segment is yet available

- else if *NOW* > PSwc[i*]

- the last one and the latest segment is available is k2[i*]

- else if *NOW* > PSwc[i*] + TSB

- no segment is available any more

- else if PSwc[1] < *NOW* <= PEwc[i*]

  - i' the such that PSwc[i'] < NOW <= PEwc[i']

  - k[*NOW*] = MIN(floor ((*NOW* - PEwc[*i*'] - PSwc[*i*'])/SDURATION[i']), k2)

Note again that if k[*NOW*] is *0*, then only the Initialization Segment is available. If the Period is not the first one, then the last available Media Segment is the last Media Segment of the previous Period.

3) The client downloads the initialization segment of the selected Representations and then accesses the content by requesting entire Segments or byte ranges of Segments. Typically at any time the client downloads the next segment at the larger of the two: (i) completion of download of current segment or (ii) the Segment Availability Start Time of the next segment. Note that if the `@availabilityTimeOffset` is present, then the segments may be downloaded earlier, namely at the adjusted segment availability start time. Based on the buffer fullness and other criteria, rate adaptation is considered. Typically the first media segment that is downloaded is the *live edge segment*, but other decisions may be taken in order to minimize start-up latency. For details on initial buffering, refer to section 11.2.4.4.

4) According to Figure 11-1 media is fed into buffer and at some point in time, the decoding and rendering of the media is kicked off. The downloading and presentation is done for the selected Representation of each selected Adaptation. The synchronization is done using the presentation time in the Period. For synchronized playout, the exact presentation times in the media is expected to be used.

5) Once presentation has started, the playout process is continuous. The playout process expects media to be present in the buffer continuously. If the **MPD**`@suggestedPresentationDelay` is present, then this value may be used as the presentation delay *PD*. If the **MPD**`@suggestedPresentationDelay` is not present, but the client is expected to consume the service at the live edge, then a suitable presentation delay should be selected, typically between the value of `@minBufferTime` and the value of `@timeShiftBufferDepth`. It is recommended that the client starts rendering the first sample of the downloaded media segment *k* with earliest presentation time EPT(*k*) at *PSwc*[*i*] + (EPT(*k*) - *o*[*r,i*]) + *PD*. For details on selecting and minimizing end-to-end latency as well as the start-up latency, see section 11.2.4.4.

6) The client may request Media Segments of the selected Representations by using the generated Segment list during the availability time window.

7) Once the presentation has started, the client continues consuming the media content by continuously requesting Media Segments or parts of Media Segments and playing content that according to the media presentation timeline. The client may switch Representations taking into updated information from its environment, e.g. change of observed throughput. In a straight-forward implementation, with any request for a Media Segment starting with a stream access point, the client may switch to a different Representation. If switching at a stream access point, the client is expected to switch seamlessly at such a stream access point.

8) With the wall-clock time *NOW* advancing, the client consumes the available Segments. As *NOW* advances the client possibly expands the list of *available* Segments for each Representation in the Period according to the procedures specified in 11.2.4.2.

9) Once the client is consuming media contained in the Segments towards the end of the announced media in the Representation, then either the Media Presentation is terminated, a new Period is started (see subsection or the MPD needs to be refetched. Once the client is consuming media contained in the Segments towards the end of the announced media in the Representation, and the Representation is contained not in the last Period, then the DASH clients generally needs to reselect the Adaptation Sets and a Representation in same manner as described in bullet 1) and 2). Also steps 3), 4), 5) and 6) need to be carried out at the transition of a Period. Generally, audio/video switching across period boundaries may not be seamless. According to ISO/IEC 23009-1, section 7.2.1, at the start of a new Period, the playout procedure of the media content components may need to be adjusted at the end of the preceding Period to match the *PeriodStart* time of the new Period as there may be small overlaps or gaps with the Representation at the end of the preceding Period. Overlaps (respectively gaps) may result from Media Segments with actual presentation duration of the media stream longer (respectively shorter) than indicated by the Period duration. Also in the beginning of a Period, if the earliest presentation time of any access unit of a Representation is not equal to the presentation time offset signalled in the `@presentationTimeOffset`, then the playout procedures need to be adjusted accordingly.

The client should play the content continuously across Periods, but there may be implications in terms of implementation to provide fully continuous and seamless playout. It may be the case that at Period boundaries, the presentation engine needs to be reinitialized, for example due to changes in formats, codecs or other properties. This may result in a re-initialization delay. Such a re-initialization delay should be minimized. If the Media Presentation is of type dynamic, the addition of the re-initialisation delay to the playout may result in drift between the encoder and the presentation engine. Therefore, the playout should be adjusted at the end of each Period to provide a continuous presentation without adding drift between the time documented in the

MPD and the actual playout, i.e. the difference between the actual playout time and the Period start time should remain constant.

If the client presents media components of a certain Adaptation Set in one Period, and if the following Period has assigned an identical Asset Identifier, then the client should identify an associated Period and, in the absence of other information, continue playing the content in the associated Adaptation Set.

If furthermore the Adaptation Sets are period-continuous, i.e. the presentation times are continuous and this is signalled in the MPD, then the client is expected to seamlessly play the content across the Period boundary. Most suitably the client may continue playing the Representation in the Adaptation Set with the same @id, but there is no guarantee that this Representation is available. In this case the client is expected to seamlessly switch to any other Representation in the Adaptation Set.

For details on MPD updates and refetching, please refer to section 11.3.

11.2.4.2   Joining, Initial Buffering and Playout Recommendations

11.2.4.2.1   General

A DASH client is expected to start playout from:

-   The time indicated by the MPD Anchor, if one is present

-   The live edge, if there is no MPD Anchor and **MPD**@type="dynamic".

11.2.4.2.2   Joining at the live edge

For joining at the live edge there are basically two high-level strategies:

-   Every client participating in the service commits to the same presentation delay (PD) relative to the announced segment availability start time at start-up and in continuous presentation, possible using one suggested by the Content Provider and then attempts to minimise start-up latency and maintain the buffer. The content provider may have provided the **MPD**@suggestedPresentationDelay (SPD) or may have provided this value by other means outside the DASH formats. The content author should be aware that the client may ignore the presence of **MPD**@suggestedPresentationDelay and may choose its own suitable playout scheduling.

-   The client individually picks the presentation delay (PD) in order to maximize stable quality and does this dependent on its access, user preferences and other considerations.

In both cases the client needs to decide, which segment to download first and when to schedule the playout of the segment based on the committed PD.

A DASH client would download an available segment and typically render the earliest presentation time EPT(k) of the segment at $PSwc[i] + (EPT(k) - o[r,i]) + PD$. As PD may be quite large, for example in order to provision for downloading in varying bitrate conditions, and if a segment is downloaded that was just made available it may result in larger start up delay.

Therefore, a couple of strategies may be considered as a tradeoff of for start-up delay, presentation delay and sufficient buffer at the beginning of the service, when joining at the live edge:

1)   The client downloads the next available segment and schedules playout with delay PD. This maximizes the initial buffer prior to playout, but typically results in undesired long start-up delay.

2)   The client downloads the latest available segment and schedules playout with delay PD. This provides large initial buffer prior to playout, but typically results in undesired long start-up delay.

3)   The client downloads the earliest available segment that can be downloaded to schedules playout with delay PD. This provides a smaller initial prior to playout, but results in reasonable start-up delay. The buffer may be filled gradually by downloading later segments faster than their media playout rate, i.e. by initially choosing Representations that have lower bitrate than the access bandwidth.

11.2.5   Considerations on live edge)

A DASH client should avoid being too aggressive in requesting segments exactly at the computed segment availability start time, especially if it is uncertain to be fully synchronized with the server. If the DASH client observes issues, such as 404 responses, it should back up slightly in the requests.

In addition, for a content authoring to avoid too aggressive requests and possible 404 responses, the content author may schedule the segment availability start time in the MPD with a small safety delay compared to the actual publish time. This also provides the content author a certain amount of flexibility in the publishing of Segments. However, note that such safety margins may lead to slightly increased end-to-end latencies, so it is a balance to be taken into account.

11.3   Live Services with MPD Updates

11.3.1   Background and Assumptions

If many cases, the service provider cannot predict that an MPD that is once offered, may be used for the entire Media Presentations. Examples for such MPD changes are:-The duration of the Media Presentation is unknown

-   The Media Presentation may be interrupted for advertisements which requires proper splicing of data, for example by adding a Period

-   Operational issues require changes, for example the addition of removal of Representations or Adaptation Sets.

-   Operational problems in the backend.

-   Changes of segment durations, etc.

In this case the MPD typically only can describe a limited time into the future. Once the MPD expires, the service provider expects the client to recheck and get an updated MPD in order to continue the Media Presentation.

The main tool in MPEG-DASH is Media Presentation Description update feature as described in section 5.4 of ISO/IEC 23009-1. The MPD is updated at the server and the client is expected to obtain the new MPD information once the determined Segment List gets to an end.

If the MPD contains the attribute **MPD**@minimumUpdatePeriod, then the MPD in hand will be updated. The DASH client typically frequently polls the MPD update server whether an MPD update is available or the existing MPD can still be used. The update frequency is controlled by MPD based on the attribute **MPD**@minimumUpdatePeriod.

11.3.2   Preliminaries

11.3.2.1   MPD Information

As the MPD is typically updated over time on the server, the MPD that is accessed when joining the service as well as the changes of the MPD are referred to as MPD instances in the following. This expresses that for the same service, different MPDs exist depending on the time when the service is consumed.

Assume that an MPD instance is present on the DASH server at a specific wall-clock time *NOW*. For an MPD-based Live Service Offering, the MPD instance may among others contain information as available in Table 11-5. Information included there may be used to compute a list of announced Segments, Segment Availability Times and URLs.

Table 11-5 – Information related to Live Service Offering with MPD-controlled MPD Updates

MPD Information	Status	Comment
MPD@type	mandatory, set to "dynamic"	the type of the Media Presentation is dynamic, i.e. Segments get available over time.

`MPD@availabilityStartTime`	mandatory	the start time is the anchor for the MPD in wall-clock time. The value is denoted as *AST*.
`MPD@minimumUpdatePeriod`	mandatory	this field is mandatory except for the case where the `MPD@mediaPresentationDuration` is present. However, such an MPD falls then in an instance as documented in section 11.2.
`Period@start`	mandatory	the start time of the Period relative to the MPD availability start time. The value is denoted as *PS*.
`SegmentTemplate@media`	mandatory	the template for the Media Segment
`SegmentTemplate@startNumber`	optional default	the number of the first segment in the Period. The value is denoted as *SSN*.
`SegmentTemplate@duration`	mandatory	the duration of each Segment in units of a time. The value divided by the value of `@timescale` is denoted as *MD*[*k*] with k=1, 2, ... The segment timeline may contain some gaps.

11.3.2.2   Segment Information Derivation

Based on an MPD instance including information as documented in Table 11-5 and available at time *NOW* on the server, a DASH client may derive the information of the list of Segments for each Representation in each Period.

If the Period is the last one in the MPD and the `MPD@minimumUpdatePeriod` is present, then the time *PEwc*[*i*] is obtained as the sum of *NOW* and the value of `MPD@minimumUpdatePeriod`. Note that with the MPD present on the server and *NOW* progressing, the Period end time is extended. This issue is the only change compared to the segment information generation in section 11.2.2.2.

If the `MPD@minimumUpdatePeriod` is set to 0, then the MPD documents all available segments on the server.

11.3.3   Service Offering Requirements and Guidelines

11.3.3.1   General

The same service requirements as in section 11.2.2.1 hold for any time *NOW* the MPD is present on the server with the interpretation that the Period end time *PEwc*[*i*] of the last Period is obtained as the sum of *NOW* and the value of `MPD@minimumUpdatePeriod`.

In order to offer a simple live service with unknown presentation end time, but only a single Period and the following details are known in advance,

-   start at wall-clock time `START`,

-   location of the segments for each Representation at "
    `http://example.com/$RepresentationID$/$Number$`",

a service provider may offer an MPD with values according to Table 11-6.

Table 11-6 – Basic Service Offering with MPD Updates

MPD Information	Value
MPD@type	dynamic
MPD@availabilityStartTime	START
MPD@publishTime	PUBTIME1
MPD@minimumUpdatePeriod	MUP
MPD.BaseURL	"http://example.com/"
Period@start	PSTART
SegmentTemplate@media	"$RepresentationID$/$Number$"
SegmentTemplate@startNumber	1
SegmentTemplate@duration	SDURATION

Based on the details in section 11.2.2.2 and 11.4.2.2, the Segment Information can be derived at each time *NOW* by determining the end time of the Period $PEwc[1] = NOW + $ MUP.

The service provider may leave the MPD unchanged on the server. If this is the case the Media Presentation may be terminated with an updated MPD that

- adds the attribute **MPD**@mediaPresentationDuration with value PDURATION

- removes the attribute **MPD**@minimumUpdatePeriod

- changes the **MPD**@publishTime attribute to PUBTIME2

The MPD must be published latest at the end of the Media Presentation minus the value of MUP, i.e. PUBTIME2 <= START + PSTART + PDURATION - MUP.

The minimum update period may also be changed during an ongoing Media Presentation. Note that as with any other change to the MPD, this will only be effective with a delay in media time of the value of the previous MUP.

The principles in this document also holds for multi-period content, for which an MPD update may add a new Period. In the same way as for signalling the end of the Media Presentation, the publish time of the updated MPD with the new period needs to be done latest at the start of the new Period minus the value of the **MPD**@minimumUpdatePeriod attribute of the previous MPD.

11.3.3.2 Setting the Minimum Update Period Value

Setting the value of the minimum update period primarily affects two main service provider aspects: A short minimum update period results in the ability to change and announce new content in the MPD on shorter notice. However, by offering the MPD with a small minimum update period, the client requests an update of the MPD more frequently, potentially resulting in increased uplink and downlink traffic.

A special value for the minimum update period is 0. In this case, the end time of the period is the current time *NOW*. This implies that all segments that are announced in the MPD are actually available at any point in time. This also allows changing the service provider to offer changes in the MPD that are instantaneous on the media timeline, as the client, prior for asking for a new segment, has to revalidate the MPD.

11.3.3.3 Permitted Updates in an MPD

The permitted updates are provided in section 5.4 of ISO/IEC 23009-1.

In addition, updates in the MPD only extend the timeline. This means that information provided in a previous version of the MPD shall not be invalidated in an updated MPD.

In order to make the MPD joining friendly and to remove data that is available in the past, any segments that have fallen out of the time shift buffer may no longer be announced in the MPD. In this case, the Period start may be moved by changing one or both, **MPD**@availabilityStartTime and **Period**@start. However, this requires that the @startNumber and @presentationTimeOffset need to be updated such that the Segment Information is correct.

If Representations and Adaptations Sets are added or removed or the location of the Segments is changed, it is recommended to update the MPD and provide Adaptation Sets in a period-continuous manner.

11.3.3.4   Last Segment Message

If the @segmentProfiles contains the 'lmsg' brand for a certain Representation, then the 'lmsg' brand for signaling the last segment shall be applied for any content with **MPD**@minimumUpdatePeriod present and the **MPD**@type="dynamic".

DASH clients operating based on such an MPD and consuming the service at the live edge typi-cally need to request a new MPD prior to downloading a new segment. However, in order to minimise MPD requests and resulting traffic load, the client may use one or more of the follow-ing optimisations:

- If the client fetches the MPD using HTTP, the client should use conditional GET methods to reduce unnecessary network usage in the downlink.

- If the @segmentProfiles contains the 'lmsg' brand clients may also rely on the 'lmsg' message and request a new MPD only in case a segment is received with an 'lmsg' brand. Otherwise the client may use template constructions to continue deter-mining the URL and the segment availability start time of segments.

If the attribute **MPD**@minimumUpdatePeriod is set to a value greater than 0 then all Segments with availability start time less than the sum of the request time and the value of the **MPD**@minimumUpdatePeriod will eventually get available at the advertised position at their computed segment availability start time. Note that by providing a **MPD**@minimumUpdatePeriod is set to a value greater than 0, DASH servers reduce the polling frequency of clients, but at the same time cannot expect that clients will request an updated MPD to be informed on changes in the segment URL constructions, e.g. at the start of a new Period.

11.3.4   MPD-based Live Client Operation based on MPD

In an extension to the description in section 11.2.4, the client now has access to an MPD and the MPD contains the **MPD**@minimumUpdatePeriod, for example following the parameters in Table 11-5. The start time of each Period is computed as period start time *PSwc*[*i*] and the MPD-URL does not include any fragment parameters.

The client fetches an MPD with parameters in Table 11-5 at time FetchTime, at its initial location if no **MPD.Location** element is present, or at a location specified in any present **MPD.Location** element. FetchTime is defined as the time at which the server processes the request for the MPD from the client. The client typically should not use the time at which it actually successfully received the MPD, but is expected to take into account delay due to MPD delivery and processing. The fetch is considered successful either if the client obtains an updated MPD or the client verifies that the MPD has not been updated since the previous fetching.

If the client fetches the MPD using HTTP, the client is expected to use conditional GET methods as specified in RFC 2616 to reduce unnecessary network usage in the downlink.

In an extension of bullet 3 in section 11.2.4.1

- the client creates a list of accessible Segments at least for each selected Representation taking into account the information in the MPD as documented in Table 11-5 and the current time *NOW* by using the Period end time of the last Period as FetchTime + MUP.

In an extension of bullet 9 in section 11.2.4.1,

- the client consumes media in last announced Period. Once the client is consuming media contained in the Segments towards the end of the announced Period, i.e. requesting segments with segment availability start time close to the validity time of the MPD defined as FetchTime + MUP, them, then the DASH client needs to fetch an MPD at its initial location if no **MPD.Location** element is present, or at a location specified in any present **MPD.Location** element.

- If the client fetches the updated MPD using HTTP, the client is expected to use conditional GET methods to reduce unnecessary network usage in the downlink.

- The client parses the MPD and generates a new segment list based on the new `FetchTime` and `MUP` of the updated MPD. The client searches for the currently consumed Adaptation Sets and Representations and continues the process of downloading segments based on the updated Segment List.

11.4 Provisioning of Live Content in On-Demand Mode

11.4.1 Scenario

A common scenario for DASH distribution results that a live generated service is also made available for On-Demand offering after the live program is completed. The typical scenario is as follows:

- The Segments as generated for the live service are also used for the On-Demand case. This avoids reformatting and also permits to reuse the Segments that are already cached.

- The MPD is modified to reflect that the content is available as On-Demand now.

- Problems that results from live delivery may be solved, e.g. variable segment durations, or issues of segment unavailability.

- The content may be augmented with ads.

- The content may be trimmed from a longer, e.g. 24/7 stream, at the beginning and/or end.

11.4.2 Content Offering Requirements and Recommendations

In order to provide live content as On-Demand in the above scenario, the following is recommended:

- The same Segments as generated for the live distribution are reused also for static distribution.

- Typically, the Segments also will have the same URL in order to exploit caching advantages.

- An MPD should be generated latest at the end of the live session, but also may be created during an ongoing live session to document a certain window of the program that is offered for On-Demand.

- A new MPD is generated that should contain the following information

  - The **MPD**@type is set to `static`.

  - The **MPD**@availabilityStartTime may be set to any time in the past, for example the time of the original "live" MPD may reused.

  - The attributes @timeShiftBufferDepth and @minimumUpdatePeriod are not present (in contrast to the live MPD) and a @mediaPresentationDuration attribute is added.

  - The window offered by the MPD is expressed by appropriately setting the **Period**@start value (including the presentation time offset and the start number) and the @mediaPresentationDuration attribute. The wall-clock time should be maintained by offsetting the **Period**@start without changing the **MPD**@availabilityStartTime.

  - Content may be offered in the same Period structure as for live or in a different one.

  - If Periods are continuous, it is preferable to remove the Period structure.

  - If new Periods are added for Ad Insertion, the Periods preferably be added in a way that they are at Segment boundaries.

  - The same templating mode as used in the live service should also be used for static distribution.

11.4.3   Client Behavior

For a DASH client, there is basically no difference on whether the content was generated from a live service or the content is provided as On-Demand. However, there are some aspects that may be "left-overs" from a live service distribution that a DASH client is expected to be aware of:

-    The Representations may show gaps in the media offering by using early terminated Periods. Such gaps are expected to be recognized and properly handled.

11.5   Availability Time Synchronization between Client and Server

11.5.1   Background

According to ISO/IEC 23009-1, in order to properly access MPDs and Segments that are available on origin servers or get available over time, DASH servers and clients should synchronize their clocks to a globally accurate time standard.

Specifically Segment Availability Times are expected to be wall-clock accurately announced in the MPD and the client needs to have access to the same time base as the MPD generation in order to enable a proper service. In order to ensure this, this section provides server and client requirements to ensure proper operation of a live service.

11.5.2   Service Provider Requirements and Guidelines

If the Media Presentation is dynamic or if the `MPD`@availabilityStartTime is present then the service shall provide a Media Presentation as follows:

-    The segment availability times announced in the MPD should be generated from a device that is synchronized to a globally accurate timing source, preferably using NTP.

-    The MPD should contain at least one **UTCTiming** element with @schemeIdURI set to one of the following:

     -    urn:mpeg:dash:utc:ntp:2014

     -    urn:mpeg:dash:utc:http-head:2014

     -    urn:mpeg:dash:utc:http-xsdate:2014

     -    urn:mpeg:dash:utc:http-iso:2014

     -    urn:mpeg:dash:utc:http-ntp:2014

-    If the MPD does not contain any element **UTCTiming** then the segments shall be available latest at the announced segment availability time using a globally accurate timing source.

-    If the MPD contains an element **UTCTiming** then

     -    the announced timing information in the **UTCTiming** shall be accessible to the DASH client, and

     -    the segments shall be available latest at the announced segment availability time in the MPD for any device that uses one of announced time synchronization methods at the same time.

If `urn:mpeg:dash:utc:http-head:2014` is used, then the server specified in the `@value` attribute of the **UTCTiming** element should be the server hosting the DASH segments such that with each request the Date general-header field in the HTTP header can be used by the client to maintain synchronization.

Note that in practical deployments segment availability may be an issue due to failures, losses, outages and so on. In this case the Server should use methods as defined in section 11.7 to inform DASH clients about potential issues on making segments available.

A leap second is added to UTC every 18 months on average. A service provider should take into account the considerations in RFC 7164 [Y]. The MPD time does not track leap seconds. If these occur during a live service they may advance or retard the media against the real time.

11.5.3 Client Requirements and Guidelines

If the Media Presentation is dynamic or if the `MPD`@availabilityStartTime is present then client is expected todo the following:

- If the MPD does not contain any element `UTCTiming` it should acquire an accurate wall-clock time from its system. The anticipated inaccuracy of the timing source should be taken into account when requesting segments close to their segment availability time boundaries.

- If the MPD contains one or several elements `UTCTiming` then the client is expected toat least use one of the announced timing information in the `UTCTiming` to synchronize its clock. The client must not request segments prior to the segment availability start time with reference to any of the chosen `UTCTiming` methods.

- The client may take into account the accuracy of the timing source as well as any transmission delays if it makes segment requests.

- Clients is expected to observe any difference between their time zone and the one identified in the MPD, as MPDs may indicate a time which is not in the same timezone as the client.

- If the client observes that segments are not available at their segment availability start time, the client is expected touse the recovery methods defined in section 11.7.

- Clients is expected to not access the `UTCTiming` server more frequently than necessary.

11.6 Robust Operation

11.6.1 General Robustness

General Guidelines are provided ISO/IEC 23009-1 DASH spec in A.7 and in Annex A.9 of this specifications. DASH clients and servers should follow those guidelines.

11.6.2 Synchronization Loss of Segmenter

In order to address synchronization loss issues at the segmenter, the following options from the DASH standard should be considered with preference according to the order below:

1) The server is required to always offer a conforming media stream. In case the input stream or encoder is lost, the content author may always add dummy content. This may be done using a separate Period structure and is possible without any modifications of the standard.

2) Early Terminated Periods as included Cor.1 of the second edition of ISO/IEC 23009-1. Early Terminated Periods may be added that contain both `Period`@start and `Period`@duration. This expresses that for this Period no media is present at least for the time as expressed by the @duration attribute. Such Periods should only be used if Media Presentation author is experiencing issues in generating media, e.g. due to failures of a live feed. The MPD is updated using the @minimumUpdatePeriod, i.e. the timeline is progressing. This permits server to signal that there is an outage of media generation, but that the service is continuing. It is then up to the client to take appropriate actions.

11.6.3 Encoder Clock Drift

In order to support robust offering even under encoder drift circumstances, the segmenter should avoid being synced to the encoder clock. In order to improve robustness, in the case of an MPD-based offering Periods should be added in a period continuous manner. In the case of MPD and segment-based control, the producer reference box should be added

to media streams in order for the media pipeline to be aware of such drifts. In this case the client should parse the segment to obtain this information.

11.6.4   Segment Unavailability

To address signaling of segment unavailability between the client and server and to indicate the reason for this, it is recommended to use regular 404s. In addition, unless a UTC Timing has been defined prior in the MPD, the Date-Header specifying the time of the server should be used. In this case, the DASH client, when receiving a 404, knows that if its time is matching the Date Header, then the loss is due to a segment loss.

11.6.5   Swapping across Redundant Tools

To enable swapping across redundant tools doing hot and warm swaps, the following should be considered

1) the content author is offering the service redundant to the client (for example using multiple BaseURLs) and the client determines the availability of one or the other. This may be possible under certain circumstances

2) Periods may be inserted at a swap instance in order to provide the new information after swap. If possible, the offering may be continuous, but the offering may also be non-continuous from a media time perspective.

3) A completely new MPD is sent that removes all information that was available before any only maintains some time continuity. However, this tool is not fully supported yet in any DASH standard and not even considered.

There is a clear preference for the bullets above in their order 1, 2 and 3 as the service continuity is expected to be smoother with higher up in the bullet list. At the same time, it may be the case that the failure and outages are severe and only the third option may be used.

12   Ad Insertion Enablers

12.1   Background

This section provides recommendations for implementing server-based ad insertion in DASH. In the server-based model, all ad-related information is expressed via MPD and segments, and ad decisions are triggered by client requests for MPDs and for resources described in them (Segments, remote periods).

The server-based model is inherently MPD-centric – all data needed to trigger ad decision is concentrated in the MPD. In case where ad break location (i.e., its start time) is unknown at the MPD generation time, it is necessary to rely on MPD update functionality.

12.2   Enablers

12.2.1   Period elements and Remote Periods

A single ad may be expressed as a single Period element.

Periods with content that is expected to be interrupted as a result of ad insertion should contain explicit start times (**Period**@start), rather than durations. This allows insertion of new periods without modifying the existing periods. If a period has media duration longer then the distance between the start of this period and the start of next period, use of start times implies that a client will start the playout of the next period at the time stated in the MPD, rather than after finishing the playout of the last segment.

An upcoming ad break is expressed as one or more Period elements. Remote periods may have default content that will be played in case dereferencing fails. After dereferencing MPD may contain zero-duration periods or/and remote Periods.

Remote Periods are resolved on demand into one or more than one Period elements. It is possible to embed parameters into the XLink URL of the corresponding remote period, in order to have them passed to the ad decision system via XLink resolver at resolution time.

The remote period may be resolved into a period with zero duration. Such a period element will not contain any Adaptation Sets.

In case of **Period**@xlink:actuate="onRequest", XLink resolution should be done sufficiently early to ensure that there are no artefacts due to insufficient time given to download the inserted content. Care needs to be taken so that the client is given a sufficient amount of time to dereference the upcoming remote period.

In case of Period@xlink:actuate="onRequest", MPD update and XLink resolution should be done sufficiently early to ensure that there are no artefacts due to insufficient time given to download the inserted content.

Period@xlink:actuate="onRequest" shall not be used if MPD@type ="dynamic"

12.2.2   Asset Identifiers

AssetIdentifier descriptors identify the asset to which a Period belongs. This can be used for implementation of client functionality that depends on distinguishing between ads and main content (e.g. progress bar). Periods with same AssetIdentifier should have identical Adaptation Sets, Initialization Segments and same DRM information (i.e., DRM systems, licenses). This allows reuse of at least some initialization data across periods of the same asset, and ensures seamless continuation of playback if inserted periods have zero duration. Period continuity may be signaled.

AssetIdentifier descriptor shall be used for distinguishing parts of the same asset within a multi-period MPD, hence it shall be used for main content.

In order to enable better tracking and reporting, unique IDs should be used for different assets.

In the absence of other asset identifier schemes, a 3GPP-defined scheme may be used with the value of @schemeIdUri set to "urn:org:pss:dash:asset-id:2015". If used, the value of @value attribute descriptor is unspecified. It shall be the same for all parts of an asset.

If a Period has one-off semantics (i.e., an asset is completely contained in a single period, and its continuation is not expected in the future), the author shall not use asset identifier on these assets.

Periods that do not contain non-remote Adaptation Set elements, as well as zero-length periods shall not contain the AssetIdentifier descriptor.

12.2.3   MPD updates

MPD updates are used to implement dynamic behavior. An updated MPD may have additional (possibly – remote) periods. Hence, MPD update should occur as close as possible to the moment the start time of the upcoming ad break is known. Ad breaks can also be canceled prior to their start, and such cancellation will also result in a newer MPD version (i.e., in a need for MPD update).

Frequent regular MPD updates are sufficient for implementing dynamic ad insertion. Unfortunately they create an overhead of unnecessary MPD traffic – ad breaks are rare events, while MPD updates need to be frequent enough if a cue message is expected to arrive only several seconds before the splice point. Use of HTTP conditional GET requests (i.e., allowing the server to respond with "304 Not Modified" if MPD is unchanged) is helpful in reducing this overhead, but asynchronous MPD updates avoid this overhead entirely.

Annex A (informative):
Example DASH Client Behaviour

A.1   Introduction

The information on client behaviour is purely informative and does not imply any normative procedures on DASH client implementations. The Annex primarily refers to Annex A of ISO/IEC 23009-1 [43].

A.2   Overview

See ISO/IEC 23009-1 [43], Annex A.2.

A.3   Segment List Generation

See ISO/IEC 23009-1 [43], Annex A.3.

Table A.1: Void

A.4   Seeking

See ISO/IEC 23009-1 [43], Annex A.4.

A.5   Support for Trick Modes

See ISO/IEC 23009-1 [43], Annex A.5.

A.6   Switching Representations

See ISO/IEC 23009-1 [43], Annex A.6.

A.7   Reaction to Error Codes

See ISO/IEC 23009-1 [43], Annex A.7.

A.8   Encoder Clock Drift Control

See ISO/IEC 23009-1 [43], Annex A.8.

A.9 Handling Partial File Responses

If the DASH client issued a regular HTTP GET request including an accept header with `application/3gpp-partial`, i.e. a partial-file-accept request. it may receive one of the following responses.

- 200 OK with the Content-Type set to `application/3gpp-partial` and the message format according to the definition in clause 7.9.2.2 of TS26.346 [42].

- `416 Requested Range Not Satisfiable` with the additional information according to the definition in clause 7.9.2.2 of TS26.346 [42].

If the **416** code is received, the 3GP-DASH client should conceal the lost media that is contained in this lost segment and should continue its regular operation, by downloading the next Segment in the Representation that contains a random access point, i.e. a SAP type 1 or 2.

If the 200 OK code (or other 2xx success status codes) with the Content-Type set to `application/3gpp-partial` is received, the 3GP-DASH client should continue its regular operation, by downloading the next Segment in the Representation that contains a random access point, i.e. a SAP type 1 or 2. In addition, the 3GP-DASH client may use the information in the body of the response in order to advantageously conceal the lost media as follows:

- If the content range starting from the first byte is included in the response,

    - the 3GP-DASH client may decode and present all complete access units that are included in the initial byte range.

    - The 3GP-DASH client may parse the Segment Index and/or movie fragment header in order to identify SAPs or sync samples that may be used to decode and render additional information.

- If the content range starting from the first position is not included in the response and the extension header `3gpp-access-position` is not present, the 3GP DASH client may discard all information and treat the media segment as lost.

- If the content range starting from the first position is not included in the response and the extension header `3gpp-access-position` is present and has a value that is part of the content range,

    - then the 3GP-DASH client may access and decode from the indicated position using regular box parsing mechanisms in order to identify SAPs or sync samples that may be used to decode and render additional information contained in this range or in other ranges.

A.10 Utilization of QoS Information

A DASH client may take into consideration available QoS information when requesting representations such that the consumed content bandwidth remains within the limits established by the signalled QoS information, see clause 6.8.3 of 3GPP TR 26.938 [46].

A.11 Utilization of Quality Metadata

Quality metadata refers to video quality metrics based on associated measurements of the media data, as described in clause 16 of 3GPP TS 26.244 [4]. A DASH client may take into consideration Quality Metadata when requesting representations, see clause 6.17 of 3GPP TR 26.938 [46].

Annex B (normative): Media Presentation Description Schema

B.1 Introduction

The main schema is provided in Annex B.2 in Table B-1. The main schema refers to the extension schema in Annex B.3 and in section 8.3.

B.2 Main Schema

The XML schema for the MPD is defined in ISO/IEC 23009-1 [43], Annex B. Table B-1 provides the MPD type and the inclusion of the 3GPP extensions in the main schema.

Table B-1: 3GPP Extensions in Main Schema

```xml
<!-- MPD Type -->
<xs:complexType name="MPDtype">
  <xs:sequence>
    <xs:element name="ProgramInformation" type="ProgramInformationType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="UTCTiming" type="DescriptorType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="BaseURL" type="BaseURLType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Location" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Period" type="PeriodType" maxOccurs="unbounded"/>
    <xs:element name="Metrics" type="MetricsType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="x3gpp:DeltaSupport" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string"/>
  <xs:attribute name="profiles" type="xs:string" use="required"/>
  <xs:attribute name="type" type="PresentationType" default="static"/>
  <xs:attribute name="availabilityStartTime" type="xs:dateTime"/>
  <xs:attribute name="availabilityEndTime" type="xs:dateTime"/>
  <xs:attribute name="mediaPresentationDuration" type="xs:duration"/>
  <xs:attribute name="minimumUpdatePeriod" type="xs:duration"/>
  <xs:attribute name="minBufferTime" type="xs:duration" use="required"/>
  <xs:attribute name="timeShiftBufferDepth" type="xs:duration"/>
  <xs:attribute name="suggestedPresentationDelay" type="xs:duration"/>
  <xs:attribute name="maxSegmentDuration" type="xs:duration"/>
  <xs:attribute name="maxSubsegmentDuration" type="xs:duration"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

B.3 3GPP Extension Schema

Table B-2: XML schema of the 3GPP Extensions for MPD

```xml
<?xml version="1.0"?>
<xs:schema targetNamespace="urn:3GPP:ns:DASH:MPD-ext:2011"
    attributeFormDefault="unqualified" elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="urn:3GPP:ns:DASH:MPD-ext:2011">
    <xs:annotation>
        <xs:appinfo>Extensions to Media Presentation Description for 3GPP</xs:appinfo>
    </xs:annotation>

    <xs:element name="DeltaSupport" type="DeltaSupportType"/>

    <!--DeltaSupport for the MPD -->
    <xs:complexType name="DeltaSupportType">
        <xs:sequence>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
```

```
        <xs:attribute name="sourceURL" type="xs:anyURI" use="required"/>
        <xs:attribute name="availabilityDuration" type="xs:duration"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>


</xs:schema>
```

Annex C (normative):
Descriptor Scheme Definitions

C.1   Introduction

This annex defines descriptors that are defined in this specification. In particular the following descriptors are defined

- 3D Video disparity range descriptor scheme in clause C.4.

- 3D Video display information descriptor scheme in clause C.5.

Note: Other descriptors are defined in ISO/IEC 23009-1 [43], section 5.9.

C.2   Void

Table C.1: Void

C.3   Void

C.4   3D Video Disparity Range Descriptor Scheme

The 3D Video Disparity Range description scheme may only be signalled in **SupplementalProperty** element with SupplementalProperty@schemeIdUri set to urn:3GPP:dash:3dVideoDisparityRange:2013. When present the 3D Video Disparity Range description scheme shall be on the Adaptation Set level. The SupplementalProperty@value of 3D Video Disparity Range description scheme shall be present and is defined below using ABNF [40]:

- @value="disparityRange"

- disparityRange = disparityMinValue disparityMaxValue

- disparityMinValue = [%x2D]1*3DIGIT

- disparityMaxValue = [%x2D]1*3DIGIT

disparityMinValue and disparityMaxValue are expressed as the minimum and the maximum disparity, in units of pixels, divided by the width of the 3D video content, in units of pixels, multiplied by one hundred. disparityMinValue and disparityMaxValue shall be the same for all Representations included in an Adaptation Set containing this descriptor scheme.

NOTE 1:   The disparity is the difference between the horizontal positions of a pixel representing the same point in space in the right and left views of a stereoscopic 3D video frame. Positive disparity values move the object away from the viewer whilst negative values move it towards the viewer. A value of zero places the object in the plane of the display screen.

NOTE 2:   The DASH client can calculate the perceived depth of the stereoscopic 3D content based on the provided disparity range and the prevailing contextual information, such as a screen size used to display the content, and the calculated or predefined distance of a viewer from the display. Based on the calculated depth range perceived by a user, the DASH client can determine applicability of the stereoscopic 3D video content provided for a given Adaptation Set. [41]

C.5   3D Video Display Information Scheme

The 3D Video Display Information description scheme may only be signalled in **SupplementalProperty** element with SupplementalProperty@schemeIdUri urn:3GPP:dash:3dVideoDisplayInformation:2013. When present the 3D Video Display Information description scheme shall be present on the Adaptation Set level. The SupplementalProperty@value of 3D Video Display Information description scheme shall be present and is defined below using ABNF [40]:

- @value="displayInformation"

- displayInformation = displayWidth [viewingDistance]

- displayWidth = 1*5DIGIT

- viewingDistance = 1*5DIGIT

displayWidth and viewingDistance provide actual values for which the stereoscopic 3D video content is optimized and are expressed in millimeters. The displayWidth defines the width of the part of the screen actually used for displaying the 3D video content.

NOTE: The DASH client can choose the Adaptation Set which contains the 3D Video Display Information Scheme with displayWidth closest to the actual display size used by the DASH client to display the content.

Annex D (informative): MPD Examples

D.1 On-Demand Service

Table D.1 provides an example MPD for an On-Demand service.

Table D.1: Example MPD for an On-Demand Service

```xml
<?xml version="1.0"?>
<MPD
    profiles="urn:3GPP:PSS:profile:DASH10"
    type="static"
    minBufferTime="PT10S"
    mediaPresentationDuration="PT2H"
    availabilityStartTime="2010-04-01T09:30:47Z"
    availabilityEndTime="2010-04-07T09:30:47Z"
    xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 3GPP-Rel10-MPD.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="urn:mpeg:dash:schema:mpd:2011">
    <ProgramInformation moreInformationURL="http://www.example.com">
        <Title>Example</Title>
    </ProgramInformation>
    <BaseURL>http://www.example.com</BaseURL>
    <Period start="PT0S">
        <AdaptationSet mimeType="video/3gpp">
            <ContentComponent contentType="video"/>
            <ContentComponent contentType="audio" lang="en"/>
            <Representation codecs="s263, samr" bandwidth="256000" id="256">
                <BaseURL>"rep1"</BaseURL>
                <SegmentList duration="1000" timescale="100">
                    <Initialization sourceURL="seg-init.3gp"/>
                    <SegmentURL media="seg-1.3gp"/>
                    <SegmentURL media="seg-2.3gp"/>
                    <SegmentURL media="seg-3.3gp"/>
                </SegmentList>
            </Representation>
            <Representation codecs="mp4v.20.9, mp4a.E1" bandwidth="128000" id="128">
                <BaseURL>"rep2"</BaseURL>
                <SegmentList duration="10">
                    <Initialization sourceURL="seg-init.3gp"/>
                    <SegmentURL media="seg-1.3gp"/>
                    <SegmentURL media="seg-2.3gp"/>
                    <SegmentURL media="seg-3.3gp"/>
                </SegmentList>
            </Representation>
        </AdaptationSet>
    </Period>
    <Period start="PT30S">
        <SegmentTemplate
            duration="10"
            initialization="seg-init-$RepresentationId$.3gp"
            media="http://example.com/$RepresentationId$/$Number$.3gp"/>
        <AdaptationSet mimeType="video/3gpp" codecs="mp4v.20.9, mp4a.E1">
            <ContentComponent contentType="video"/>
            <ContentComponent contentType="audio" lang="en"/>
            <Representation bandwidth="256000" id="1"/>
            <Representation bandwidth="128000" id="2"/>
        </AdaptationSet>
    </Period>
</MPD>
```

D.2 Live Service

Table D.2 provides an example MPD for a live service.

Table D.2: Example MPD for a Live Service

```xml
<?xml version="1.0"?>
<MPD
    profiles="urn:3GPP:PSS:profile:DASH10"
    type="dynamic"
    minBufferTime="PT3S"
    availabilityStartTime="2010-04-26T08:45:00-08:00"
    minimumUpdatePeriod="PT5M0S"
    timeShiftBufferDepth="PT1H30M0S"
    xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 3GPP-Rel10-MPD.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="urn:mpeg:dash:schema:mpd:2011">
    <ProgramInformation moreInformationURL="http://www.example.com">
        <Title>Example 3: 3GPP SA4 Meeting in Vancouver as Live Broadcast</Title>
        <Source>3GPP</Source>
    </ProgramInformation>
    <Period start="PT0S" id="0">
        <AdaptationSet mimeType='video/3gpp' codecs="avc1.42E00B" width="320" height="240"
contentType="video">
            <SegmentTemplate
                duration="60"
                initialization="http://www.ad-server.com/1-day-black/QVGA/0.3gp"
                media="http://www.ad-server.com/1-day-black/QVGA/$Number$.3gp">
            </SegmentTemplate>
            <Representation id="Ad-QVGA" bandwidth="10000">
        </Representation>
        </AdaptationSet>
    </Period>
    <Period start="PT15M0S" id="1">
        <SegmentTemplate
            duration="10"
            initialization="http://www.example.com/Period-2010-04-26T08-45-00/rep-
$RepresentationID$/seg-0.3gp"
            media="http://www.example.com/Period-2010-04-26T08-45-00/rep-
$RepresentationID$/seg-$Number$.3gp"/>
        <AdaptationSet mimeType='video/3gpp'>
            <ContentComponent contentType="video"/>
            <ContentComponent contentType="audio" lang="en"/>
            <Representation
id="QVGA-LQ" codecs="avc1.42E00C, mp4a.40.2" bandwidth="192000" width="320" height="240"/>
            <Representation id="QVGA-HQ" codecs="avc1.42E00C, mp4a.40.2" bandwidth="384000"
width="320" height="240"/>           <Representation id="VGA-LQ" mimeType='video/3gpp'
codecs="avc1.64001E, mp4a.40.2" bandwidth="512000" width="640" height="480"/>
            <Representation id="VGA-HQ" codecs="avc1.64001E, mp4a.40.2" bandwidth="1024000"
width="640" height="480"/>
        </AdaptationSet>
    </Period>
    <Period start="PT2H01M22.12S" id="2">
        <SegmentTemplate duration="10"
            media="http://www.ad-server.com/15min-Ads/$RepresentationID$/$Number$.3gp"
            initialization="http://www.ad-server.com/15min-Ads/$RepresentationID$/0.3gp"/>
        <AdaptationSet mimeType='video/3gpp'>
            <ContentComponent contentType="video"/>
            <ContentComponent contentType="audio" lang="en"/>
            <Representation id="QVGA" codecs="avc1.42E00C, mp4a.40.2" bandwidth="256000"
width="320" height="240"/>
            <Representation id="VGA"  codecs="avc1.64001E, mp4a.40.2" bandwidth="512000"
width="640" height="480"/>
        </AdaptationSet>
    </Period>
    <Period start="PT2H16M22.12S" id="3">
        <SegmentTemplate
            duration="10"
            media="http://www.example.com/Period-2010-04-26T11-01-22/rep-
$RepresentationID$/seg-$Number$.3gp"
            initialization="http://www.example.com/Period-2010-04-26T11-01-22/rep-
$RepresentationID$/seg-0.3gp"
            />
        <AdaptationSet mimeType='video/3gpp' contentType="video">
            <Representation id="QVGA-LQ" codecs="avc1.42E00C" bandwidth="192000" width="320"
height="240"/>
            <Representation id="QVGA-HQ" codecs="avc1.42E00C" bandwidth="384000" width="320"
height="240"/>
            <Representation id="VGA-LQ" codecs="avc1.64001E" bandwidth="512000" width="640"
height="480"/>
            <Representation id="VGA-HQ" codecs="avc1.64001E" bandwidth="1024000" width="640"
height="480"/>
```

```
            </AdaptationSet>
            <AdaptationSet mimeType='audio/3gpp' contentType="audio" lang="en">
                <Representation id="audio" codecs="mp4a.40.2" bandwidth="32000"/>
                <Representation id="audio" codecs="mp4a.40.2" bandwidth="64000"/>
            </AdaptationSet>
        </Period>
</MPD>
```

D.3   MPD Assembly

Table D.3 provides an example MPD with reference to external Period element as provided in Table D.4. An equivalent MPD to the one in Table D.3 after dereferencing with the **Period** element in Table D.4 is shown in Table D.2.

Table D.3: Example MPD with reference to external Period element

```
<?xml version="1.0"?>
<MPD
    profiles="urn:3GPP:PSS:profile:DASH10"
    type="dynamic"
    minBufferTime="PT3S"
    availabilityStartTime="2010-04-26T08:45:00-08:00"
    minimumUpdatePeriod="PT5M0S"
    timeShiftBufferDepth="PT1H30M0S"
    xsi:schemaLocation="urn:mpeg:Ddash:schema:mpd:2011 3GPP-Rel10-MPD.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns="urn:mpeg:dash:schema:mpd:2011">
    <ProgramInformation moreInformationURL="http://www.example.com">
        <Title>Example 3: 3GPP SA4 Meeting in Vancouver as Live Broadcast</Title>
        <Source>3GPP</Source>
    </ProgramInformation>
    <Period start="PT0S" id="0">
        <AdaptationSet mimeType='video/3gpp' codecs="avc1.42E00B" width="320" height="240"
contentType="video">
            <SegmentTemplate
                duration="60"
                initialization="http://www.ad-server.com/1-day-black/QVGA/0.3gp"
                media="http://www.ad-server.com/1-day-black/QVGA/$Number$.3gp">
            </SegmentTemplate>
            <Representation id="Ad-QVGA" bandwidth="10000">
            </Representation>
        </AdaptationSet>
    </Period>
    <Period start="PT15M0S" id="1">
        <SegmentTemplate
            duration="10"
            initialization="http://www.example.com/Period-2010-04-26T08-45-00/rep-
$RepresentationID$/seg-0.3gp"
            media="http://www.example.com/Period-2010-04-26T08-45-00/rep-
$RepresentationID$/seg-$Number$.3gp"/>
        <AdaptationSet mimeType='video/3gpp'>
            <ContentComponent contentType="video"/>
            <ContentComponent contentType="audio" lang="en"/>
            <Representation id="QVGA-LQ" codecs="avc1.42E00C, mp4a.40.2" bandwidth="192000"
width="320" height="240"/>
            <Representation id="QVGA-HQ" codecs="avc1.42E00C, mp4a.40.2" bandwidth="384000"
width="320" height="240"/>      <Representation id="VGA-LQ" mimeType='video/3gpp'
codecs="avc1.64001E, mp4a.40.2" bandwidth="512000" width="640" height="480"/>
            <Representation id="VGA-HQ" codecs="avc1.64001E, mp4a.40.2" bandwidth="1024000"
width="640" height="480"/>
        </AdaptationSet>
    </Period>
    <Period xlink:ref="http://www.example.com/Period.xml" id="2"/>
    <Period start="PT2H16M22.12S" id="3">
        <SegmentTemplate
            duration="10"
            media="http://www.example.com/Period-2010-04-26T11-01-22/rep-
$RepresentationID$/seg-$Number$.3gp"
            initialization="http://www.example.com/Period-2010-04-26T11-01-22/rep-
$RepresentationID$/seg-0.3gp"
        />
        <AdaptationSet mimeType='video/3gpp' contentType="video">
            <Representation id="QVGA-LQ" codecs="avc1.42E00C" bandwidth="192000" width="320"
height="240"/>
```

```
                    <Representation id="QVGA-HQ" codecs="avc1.42E00C" bandwidth="384000" width="320"
height="240"/>
                    <Representation id="VGA-LQ" codecs="avc1.64001E" bandwidth="512000" width="640"
height="480"/>
                    <Representation id="VGA-HQ" codecs="avc1.64001E" bandwidth="1024000" width="640"
height="480"/>
            </AdaptationSet>
            <AdaptationSet mimeType='audio/3gpp' contentType="audio" lang="en">
                    <Representation id="audio" codecs="mp4a.40.2" bandwidth="32000"/>
                    <Representation id="audio" codecs="mp4a.40.2" bandwidth="64000"/>
            </AdaptationSet>
        </Period>
</MPD>
```

Table D.4: External Period

```
<?xml version="1.0"?>
<Period start="PT2H01M22.12S">
    <SegmentTemplate duration="10"
        media="http://www.ad-server.com/15min-Ads/$RepresentationID$/$Number$.3gp"
        initialization="http://www.ad-server.com/15min-Ads/$RepresentationID$/0.3gp"/>
    <AdaptationSet mimeType='video/3gp'>
        <ContentComponent contentType="video"/>
        <ContentComponent contentType="audio" lang="en"/>
        <Representation id="QVGA" codecs="avc1.42E00C, mp4a.40.2" bandwidth="256000"
width="320" height="240"/>
        <Representation id="VGA"  codecs="avc1.64001E, mp4a.40.2" bandwidth="512000"
width="640" height="480"/>
    </AdaptationSet>
</Period>
```

D.4 MPD Deltas

In the following MPD example, the content is 30 minutes in duration. There are 3 Periods, each of 10 minutes duration. Each Period has 3 Representations and each Representation is contained within one 3gp file. Each Representation has audio encoded with Low Complexity-AAC. One Representation of each Period (p1rep1.3gp, p2rep1.3gp, and p3rep1.3gp) has video resolution 320x240 encoded with H.264 baseline profile level 1.1. Another Representation of each Period (p1rep2.3gp, p2rep2.3gp, and p3rep2.3gp) has resolution 320x240 encoded with H.264 baseline profile level 1.3. Finally, a third representation in each period (p1rep3.3gp, p2rep3.3gp, and p3rep3.3gp) has resolution 480x240 encoded with H.264 baseline profile level 2.1. One Representation of each Period has bandwidth of 239 kbps, a second representation has bandwidth of 478 kbps, and a third representation has bandwidth of 892 kbps.

Since each represention is contained in one file, the Initialization Segments and the Media Segments for a representation are accessed with byte ranges. Each **SegmentURL** element in the MPD contains a **mediaRange** attribute and the corresponding byte range for the Initialization Segment or Media Segment. For the example each Segment of all representations is 10 seconds in duration.

Line numbers of the MPD in the example are shown for clarity, although these would not be present in the MPD.

EXAMPLE 1 (add)

The change of adding the SegmentURL element for the next Segment to the Representation of the third Period of the MPD in the example with 239K bandwidth can be described as follows.

```
492a
            <SegmentURL mediaRange="17339554-17642841"/>
.
```

The SegmentURL element is added on a new line after line number 492.

EXAMPLE 2 (replace)

Replacing the line containing the **DeltaSupport** element to correspond to the MPD after the next update can be described as follows.

```
625c
    <DeltaSupport sourceURL="delta2.mpdd" availabilityDuration="PT120S"/>
```

.

EXAMPLE 3(delete)

If lines 8 through 10 of the original MPD are deleted and not present in the updated MPD, the delta to express this is:

```
8,10d
```
.

Below is what the MPD looks like after approximately 29 minutes and 40 seconds. In this case, the MPD is updated approximately every 10 seconds.

```
1<?xml version="1.0"?>
2<MPD    profiles="urn:3GPP:PSS:profile:DASH10"
3        type="dynamic"
4        availabilityStartTime="2010-07-01T05:00:00Z"
5        availabilityEndTime="2010-07-08T05:00:00Z"
6        mediaPresentationDuration="PT2H"
7        minimumUpdatePeriod="PT10S"
8        minBufferTime="PT10S"
9        timeShiftBufferDepth="PT30M"
10       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
11       xmlns:x3gpp="urn:3GPP:ns:DASH:MPD-ext:2011"
12       xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 3GPP-Rel10-MPD.xsd"
13       xmlns="urn:mpeg:dash:schema:mpd:2011">
14
15 <ProgramInformation moreInformationURL="http://www.example.com">
16  <Title>Example</Title>
17  <Source>Example</Source>
18  <Copyright>Example</Copyright>
19 </ProgramInformation>
20 <BaseURL>http://www.example.com</BaseURL>
21 <Period start="PT0S" bitstreamSwitching="true" id="0">
22  <AdaptationSet mimeType="video/3gpp">
23    <ContentComponent contentType="video"/>
24    <ContentComponent contentType="audio" lang="en"/>
25    <Representation id="0" bandwidth="239000" width="320" height="240" codecs="avc1.42E00b,
mp4a.40.2">
26        <BaseURL>"p1rep1.3gp"</BaseURL>
27        <SegmentList duration="10" >
28          <Initialization range="0-985" />
29                        <SegmentURL  mediaRange="986-293761" />
30                        <SegmentURL  mediaRange="293762-592501" />
                                    .
                                    .
                                    .
88                        <SegmentURL  mediaRange="17600065-17894640" />
89        </SegmentList>
90    </Representation>
91    <Representation id="1" bandwidth="478000" width="320" height="240" codecs="avc1.42E00d,
mp4a.40.2" >
92        <BaseURL>"p1rep2.3gp"</BaseURL>
93        <SegmentList duration="10">
94          <Initialization range="0-985" />
95                        <SegmentURL  mediaRange="986-586538" />
96                        <SegmentURL  mediaRange="586539-1184019" />
                                    .
                                    .
                                    .
154                       <SegmentURL  mediaRange="35199171-35788323" />
155       </SegmentList>
156    </Representation>
157    <Representation id="2" bandwidth="892000" width="480" height="240"
codecs="avc1.42E015, mp4a.40.2">
158        <BaseURL>"p1rep3.3gp"</BaseURL>
159        <SegmentList duration="10">
160          <Initialization range="0-985" />
161                       <SegmentURL  mediaRange="986-1093691" />
162                       <SegmentURL  mediaRange="1093692-2208656" />
                                    .
                                    .
                                    .
220                       <SegmentURL  mediaRange="65684646-66784068" />
221       </SegmentList>
```

```
222    </Representation>
223  </AdaptationSet>
224 </Period>
225 <Period start="PT10M0S" bitstreamSwitching="true" id="1">
226  <AdaptationSet mimeType="video/3gpp">
227    <ContentComponent contentType="video"/>
228    <ContentComponent contentType="audio" lang="en"/>
229    <Representation  id="0" bandwidth="239000" width="320" height="240"
codecs="avc1.42E00b, mp4a.40.2">
230      <BaseURL>"p2rep0.3gp"</BaseURL>
231      <SegmentList duration="10">
232            <Initialization range="0-985" />
233                            <SegmentURL  mediaRange="986-296011" />
234                            <SegmentURL  mediaRange="296012-595787" />
                                                .
                                                .
                                                .
292                            <SegmentURL  mediaRange="17647666-17946154" />
293        </SegmentList>
294     </Representation>
295     <Representation id="1" bandwidth="478000" width="320" height="240"
codecs="avc1.42E00d, mp4a.40.2">
296        <BaseURL>"p2rep1.3gp"</BaseURL>
297        <SegmentList duration="10">
298            <Initialization range="0-985" />
299                            <SegmentURL  mediaRange="986-591037" />
300                            <SegmentURL  mediaRange="591038-1190590" />
                                                .
                                                .
                                                .
358                            <SegmentURL  mediaRange="35294377-35891354" />
359        </SegmentList>
360     </Representation>
361     <Representation id="2" bandwidth="892000" width="480" height="240" codecs="avc1.42E015,
mp4a.40.2">
362        <BaseURL>"p2rep2.3gp"</BaseURL>
363        <SegmentList duration="10">
364            <Initialization range="0-985" />
365                            <SegmentURL mediaRange="986-1102088" />
366                            <SegmentURL mediaRange="1102089-2220920" />
                                                .
                                                .
                                                .
424                            <SegmentURL mediaRange="65862331-66976355" />
425        </SegmentList>
426     </Representation>
427  </AdaptationSet>
428 </Period>
429 <Period start="PT20M0S" bitstreamSwitching="true" id="2">
430  <AdaptationSet mimeType="video/3gpp">
431    <ContentComponent contentType="video"/>
432    <ContentComponent contentType="audio" lang="en"/>
433    <Representation  id="0" bandwidth="239000" width="320" height="240"
codecs="avc1.42E00b, mp4a.40.2">
432      <BaseURL>"p3rep0.3gp"</BaseURL>
433      <SegmentList duration="10">
434            <Initialization range="0-985" />
435                            <SegmentURL  mediaRange="986-302469" />
436                            <SegmentURL  mediaRange="302470-597839" />
                                                .
                                                .
                                                .
492                            <SegmentURL  mediaRange="17040002-17339553" />
493      </SegmentList>
494    </Representation>
495    <Representation id="1" bandwidth="478000" width="320" height="240" codecs="avc1.42E00d,
mp4a.40.2" >
496      <BaseURL>"p3rep1.3gp"</BaseURL>
497      <SegmentList duration="10">
498            <Initialization range="0-985" />
499                            <SegmentURL  mediaRange="986-603953" />
500                            <SegmentURL  mediaRange="603954-1194693" />
                                                .
                                                .
                                                .
556                            <SegmentURL  mediaRange="34079046-34678149" />
557      </SegmentList>
```

```
558    </Representation>
559    <Representation id="2" bandwidth="892000" width="480" height="240" codecs="avc1.42E015,
mp4a.40.2" >
560      <BaseURL>"p3rep2.3gp"</BaseURL>
561      <SegmentList duration="10">
562          <Initialization range="0-985" />
563                            <SegmentURL  mediaRange="986-1126190" />
564                            <SegmentURL  mediaRange="1126191-2228575" />
                                                  .
                                                  .
                                                  .
620                            <SegmentURL  mediaRange="63594383-64712374" />
621      </SegmentList>
622    </Representation>
623   </AdaptationSet>
624  </Period>
625  <x3gpp:DeltaSupport sourceURL="delta1.mpdd" availabilityDuration="PT120S"/>
626</MPD>
```

Since the value of @sourceURL in the above MPD is "delta1.mpdd", delta1.mpdd is an empty file at the time of publication of the above MPD.

The following file is delta1.mpdd after the next MPD update. Notice that clients have access to the new value of @sourceURL referenced by the latest MPD via the delta.

```
625c
      <DeltaSupport sourceURL="delta2.mpdd" availabilityDuration="PT120S"/>
.
620a
                                  <SegmentURL mediaRange="64712375-65844316"/>
.
556a
                                  <SegmentURL mediaRange="34678150-35284727"/>
.
492a
                                  <SegmentURL mediaRange="17339554-17642841"/>
.
```

At the next MPD update, "delta1.mpdd" would contain the cumulative update for 2 MPD updates.

```
625c
      <DeltaSupport sourceURL="delta3.mpdd" availabilityDuration="PT120S"/>
.
620a
                                  <SegmentURL mediaRange="64712375-65844316"/>
                                  <SegmentURL mediaRange="65844317-66966044"/>
.
556a
                                  <SegmentURL mediaRange="34678150-35284727"/>
                                  <SegmentURL mediaRange="35284728-35885833"/>
.
492a
                                  <SegmentURL mediaRange="17339554-17642841"/>
                                  <SegmentURL mediaRange="17642842-17943394"/>
.
```

Annex E (normative):
Void

Annex F (normative):
OMA DM QoE Management Object

As an alternative to configuring the QoE reporting for each session via MPD, OMA-DM can be used to specify the QoE configuration. If such an OMA-DM QoE configuration has been specified, it shall be evaluated by the client for all subsequent sessions.

For the OMA-DM QoE configuration the parameters are specified according to the following Managed Object (MO), and represents the same information as specified in section 10.4 and 10.5. Version numbering is included for possible extension of the MO.

The Management Object Identifier shall be: `urn:oma:mo:ext-3gpp-pss-dash-qoe:1.0`.

Protocol compatibility: The MO is compatible with OMA Device Management protocol specifications, version 1.2 and upwards, and is defined using the OMA DM Device Description Framework as described in the Enabler Release Definition OMA-ERELD _DM-V1_2 [22].

The nodes and leaf objects as provided in Figure F.1 shall be contained under the 3GPP_PSS_DASH_QOE node if a client supports the feature described in this clause.

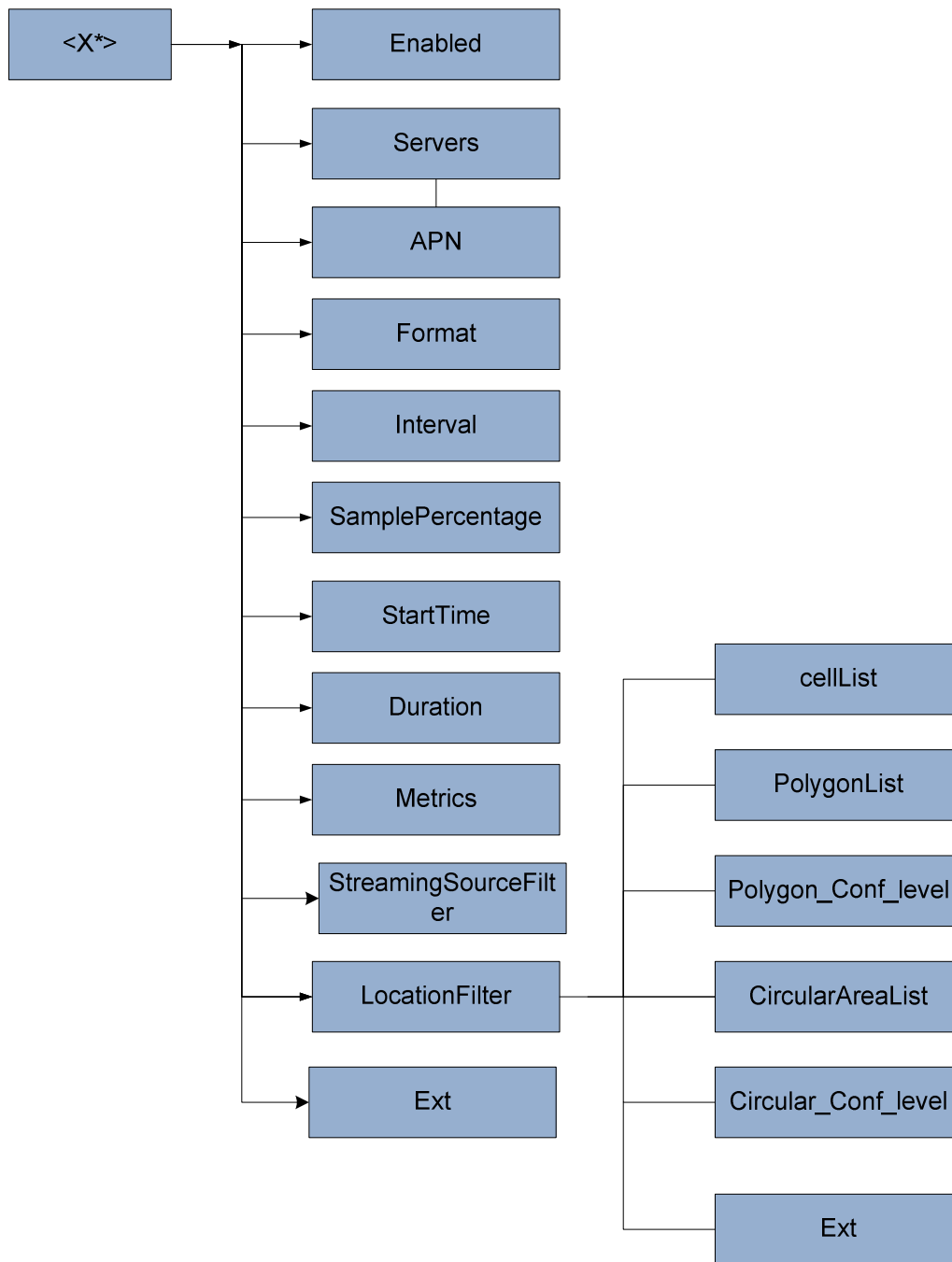**Figure F.1: Nodes and leaf objects**

Node: /<X>

This interior node specifies the unique object id of a QoE metrics management object. The purpose of this interior node is to group together the parameters of a single object.

- Occurrence: ZeroOrOne

- Format: node

- Minimum Access Types: Get

The following interior nodes shall be contained if the client supports the QoE Management Object.

/<X>/Enabled

This leaf indicates if QoE reporting is requested by the provider.

- Occurrence: One

- Format: bool

- Minimum Access Types: Get

/<X>/Servers

This leaf contains a space-separated list of servers to which the QoE reports are transmitted. It is URI addresses, e.g. http://qoeserver.operator.com. In case of multiple servers, the client randomly selects one of the servers from the list, with uniform distribution.

- Occurrence: One

- Format: chr

- Minimum Access Types: Get

- Values: URI of the servers to receive the QoE report.

/<X>/APN

This leaf contains the Access Point Name that should be used for establishing the PDP context on which the QoE metric reports will be transmitted. This may be used to ensure that no costs are charged for QoE metrics reporting. If this leaf is not defined then any QoE reporting is done over the default access point.

- Occurrence: ZeroOrOne

- Format: chr

- Minimum Access Types: Get

- Values: The Access Point Name

/<X>/Format

This leaf specifies the format of the report.If this leaf is not defined the QoE reports shall be sent uncompressed.

- Occurrence: ZeroOrOne

- Format: chr

- Minimum Access Types: Get

- Values: "uncompressed", "gzip"

/<X>/Interval

This leaf specifies how often QoE reports shall be sent. If this leaf is not defined only one QoE report shall be sent after the complete session.

- Occurrence: ZeroOrOne

- Format: int

- Minimum Access Types: Get

- Values: seconds

/<X>/SamplePercentage

This leaf specifies the percentage of sessions for which QoE metrics shall be reported. The client evaluates a random number at start of each session to determine if reporting shall be done for the specific session. If this leaf is not defined QoE reports are sent for every session.

- Occurrence: ZeroOrOne

- Format: float

- Minimum Access Types: Get

- Values: 0.0-100.0.

/<X>/StartTime

This leaf specifies when collection of QoE metrics shall start. It is specified in seconds and is relative to the start of the session. If this leaf is not defined, the QoE collection shall be done from the start of the session.

- Occurrence: ZeroOrOne

- Format: int

- Minimum Access Types: Get

- Values: seconds

/<X>/Duration

This leaf specifies for how long QoE collection shall be done. It is specified in seconds and is relative to the start time of QoE collection. If this leaf is not defined QoE collection shall be done until the end of the session.

- Occurrence: ZeroOrOne

- Format: int

- Minimum Access Types: Get

- Values: seconds.

/<X>/Metrics

This leaf specifies a list of white-space separated metrics which shall be reported, and follows the same syntax as specified for the "@metrics" attribute in Table 32. If this leaf is not defined no QoE reporting shall be done.

- Occurrence: ZeroOrOne

- Format: chr

- Minimum Access Types: Get

- Values: Metrics as specified in section 10.4.

/<X>/StreamingSourceFilter

This leaf controls the Media Presentations for which QoE metrics are collected and reported. The filter is defined as a URL pattern. Each URL pattern is represented by a POSIX regular expression as specified by IEEE Std 1003.1-2008 [52].

If the MPD URL of a Media Presentation does not match any of the regular expressions as provided by the URL patterns in the StreamingSourceFilter values, then reporting is disabled and the DASH client shall not report the requested QoE metrics for that Media Presentation. If no StreamingSourceFilter leaf is present, then reporting is enabled for any Media Presentation.

- Occurrence: ZeroOrMore

- Format: chr

- Minimum Access Types: Get

- Values: a POSIX regular expression providing the URL pattern.

/<X> /<LocationFilter>

This interior node specifies any location filters.

- Occurrence: ZeroOrOne

- Format: node

- Minimum Access Types: Get

/<X>/<LocationFilter>/CellList

This leaf specifies a list of cellIDs.

- Occurrence: ZeroOrOne

- Format: chr

- Minimum Access Types: Get

- Values: a list of ECGI or CGI.

/<X>/<LocationFilter>/PolygonList

This leaf specifies a list of shapes defined as 'Polygon' by OMA MLP[51].

- Occurrence: ZeroOrOne

- Format: chr

- Minimum Access Types: Get

- Values: a list of 'Polygon' defined by OMA MLP[51].

/<X>/<LocationFilter>/Polygon_Conf_Level

This leaf indicates the probability in percent that the DASH client is located in the corresponding polygon area specified by leaf 'PolygonList'. It is defined as 'lev_conf' by OMA MLP. If not present, it has default value of 60.

- Occurrence: ZeroOrOne

- Format: int

- Minimum Access Types: Get

- Values: 'lev_conf' defined by OMA MLP[51].

/<X>/<LocationFilter>/CircularAreaList

This leaf specifies a list of shapes defined as 'CircularArea' by OMA MLP[51].

- Occurrence: ZeroOrOne

- Format: chr

- Minimum Access Types: Get

- Values: a list of 'CircularArea' defined by OMA MLP[51].

/<X>/<LocationFilter>/Circular_Conf_Level

This leaf indicates the probability in percent that the DASH client is located in the corresponding circular area specified by leaf 'CircularAreaList'. It is defined as 'lev_conf' by OMA MLP. If not present, it has default value of 60.

- Occurrence: ZeroOrOne

- Format: int

- Minimum Access Types: Get

- Values: 'lev_conf' defined by OMA MLP[51].

/<X>/Ext

The Ext node is an interior node where the vendor specific information can be placed (vendor includes application vendor, device vendor etc.). Usually the vendor extension is identified by vendor specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include one or more un-standardized sub-trees.

- Occurrence: ZeroOrOne

- Format: node

- Minimum Access Types: Get

Annex G (normative):
File format extensions for 3GP DASH support

G.1 Introduction

This clause documents extensions to the ISO base media file format [11] for the support of 3GPP DASH.

All extensions are documented in ISO base media file format [11].

G.2 Void

G.3 Void

G.4 Void

G.5 Void

G.6 Void

Annex H (normative):
MIME Type Registration for MPD

H.1 MPD MIME Type

H.1.1 Introduction

The MIME type of the MPD is registered and available at the registry at http://www.iana.org/assignments/media-types/application. For formal registration, refer to ISO/IEC 23009-1 [43], Annex C.

H.1.2 Void

H.1.3 Void

H.2 MPD Delta MIME Type

H.2.1 Introduction

This Annex provides the formal MIME type registration for the MPD Delta. The MIME type is registered with IANA and available at http://www.iana.org/assignments/media-types/application/dashdelta.

H.2.2 MIME Type and Subtype

The MIME Type and Subtype are defined as follows:

Media Type Name: application

Subtype name: Standards Tree - dashdelta

Required parameters: none

Optional parameters: none

Encoding considerations: 8-bit text

Security considerations:

A Media Presentation Description (MPD) Delta contains text changes to an MPD. An MPD Delta is used together with a first MPD to construct a second MPD. As such, any security considerations for an MPD (media type application/dash+xml) may also be applicable to an MPD Delta. A MIME type handler would not launch a service with only an MPD Delta.

Further to this, as an MPD Delta performs editing operations on an MPD there are risks that deliberately malformed editing operations could cause security issues.

Interoperability considerations:

Published specification: 3GPP TS 26.247

Applications which use this media type:

various including but not limited to On-Demand Streaming over the Internet, Live Streaming over the Internet, Internet Video, Internet Radio

Additional information:

1. Magic number(s) : none
2. File extension(s) : mpdd
3. Macintosh file type code : none
4. Object Identifiers: none

Person to contact for further information:

1. Name : David Furbeck
2. Email : dfurbeck@blackberry.com

Intended usage : Common

Author/Change controller : 3GPP TSG SA WG4

Annex I (informative):
Signalling of DASH AVP values for QoS handling in the PCC

The PCC architecture is defined in TS 23.203 [31] and provides the Rx reference point, which enables the application layer to authorize a specific usage. In this architecture the DASH HTTP streaming server or any other function in the HTTP streaming path (e.g. an HTTP proxy) can act as Application Function and interact with the PCRF via the Rx reference point for QoS control. It is assumed here that the AF has knowledge of the application type and of the MPD.

The relevant AVPs are the ones enabling the PCRF to establish bearers with correct characteristics for DASH users. The AVPs are defined in TS 29.214 [33]. The further PCRF mapping from AVP to IP QoS parameter mapping is defined in TS 29.213 [32]

Table I.1: Example mapping of MPD parameters to Rx AVPs for 3GP-DASH (PSS)

AVP	Value	Comment
AF-Application-Identifier	"DASH"	Allows to signal the DASH based application hence giving the opportunity to enforce application specific policies
Max-Requested-Bandwidth-DL (NOTE 1)	B1	`B1 = sum of all` **MPD**`@maxBandwidth` (see clause 8.4.3.3) of all media components simultaneously (not mutually exclusive) selectable by the DASH client plus HTTP/TCP/IP overhead and TCP messages for flow control. If this attribute is not present then `B1 = sum of` **MPD**`@bandwidth` attributes of all media components of the available media presentation corresponding to representations or subrepresentations with highest bandwidth simultaneously selectable (not mutually exclusive) by the DASH client plus HTTP/TCP/IP overhead and TCP messages for flow control. Note: the mapping rules to derive the TCP message flow control bandwidth are FFS.
Max-Requested-Bandwidth-UL (NOTE 1)	FFS	For Further Study. If included, should be greater than or equal to Min-Requested-Bandwidth-UL
Min-Requested-Bandwidth-DL (NOTE 1)	B2	`B2 = sum of all` **MPD**`@minBandwidth` (see clause 8.4.3.3) of all media components simultaneously (not mutually exclusive) selectable by the DASH client plus HTTP/TCP/IP overhead and TCP messages for flow control. If this attribute is not present then `B2 = sum of` **MPD**`@bandwidth` attributes of all media components of the available media presentation corresponding to representations or subrepresentations with lowest bandwidth simultaneously (not mutually exclusive) selectable by the DASH client plus HTTP/TCP/IP overhead and TCP messages for flow control. Note: the mapping rules to derive the TCP message flow control bandwidth are FFS.
Min-Requested-Bandwidth-UL (NOTE 1)	FFS	For Further Study. Enough bitrate to cover TCP and HTTP GET requests.
Flow-Description AVP (NOTE 1)	IP addresses and ports	

NOTE 1: AVPs provided within the Media-Component-Description AVP, except Flow-Description AVP that is included within the Media-Sub-Component AVP. Omitted AVPs are not relevant for this functionality.

Annex J (normative):
MIME Type Registration for QoE Reports

J.1 Introduction

This Annex provides the formal MIME-type registration for "application/3gpdash-qoe-report+xml" that identifies XML documents following the schema "urn:3gpp:metadata:2011:HSD:receptionreport" defined in clause 10.6.2. It is referenced from the registry at http://www.iana.org/.

J.2 MIME Type and Subtype

The MIME Type and Subtype are defined as follows:

Media Type name: application

Subtype name: Standards Tree - 3gpdash-qoe-report+xml

Required parameters:

None

Optional parameters:

charset: If a charset parameter is provided its value must be "utf-8".Encoding considerations: 8bit

The utf-8 charset is always used for this type.

Security considerations:

The general XML security issues are addressed in clause 10 of RFC 3023 [83].

The QoE report XML document instances contain information about what content, services and files a particular UE has received. The information also has privacy relevance as it reveals what a specific receiver, which usually can be connected to a specific user, has received.

This media type only contains quality of experience data; there is no defined executable content.

XML from other vocabularies is allowed in this media type. Such material will have its own security considerations.

If operating in an insecure environment and required by the content/service provider, elements and attributes of the report may be encrypted to protect their confidentiality.

If operating in an insecure environment and required by the content/service provider, digital signing and verification procedures may be used to protect data origin authenticity and integrity of the report.

Thus to prevent manipulation of that information it would need to be integrity protected.

Interoperability considerations:

None

Published specification:

3GPP TS 26.247

Applications which use this media type:

3GPP DASH based applications

3GPP progressive download applications

Additional information:

 None

Person & email address to contact for further information:

 Ozgur Oyman (ozgur.oyman@intel.com)
  3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

 None

Author:

 3GPP TSG SA WG4

Change controller:

 3GPP TSG SA WG4

Annex K (normative):
ITU-T P.1203 Audio/Video MOS Estimation

K.1 Introduction

ITU-T P.1203 [49] defines an objective assessment model for Progressive Download and DASH streaming. This model predicts audio only, video only and audio-video MOS on a 5-point Absolute Category Rating scale (see ITU-T P.910 [50]) for monitoring of quality for operation and maintenance purposes. The model can execute in several modes, where Mode 0 is suitable for large-scale network supervision. For more details on this model, please refer to ITU-T P.1203 [49].

K.2 Calculation

The PSS server should support MOS estimation according to ITU-T P.1203 [49] Mode 0. To enable the MOS estimation, the QoE reporting should include the following metrics in either MPD@*metrics* or OMA DM QoE Management Object '*metrics*':

- Initial Playout Delay (section 10.2.5)

- Play List (section 10.2.7)

- MPD Information, unless already available by other means (section 10.2.8)

- Device Information (section 10.2.10)

The attribute **MPD.Metrics.Range**@duration or the OMA DM QoE Management Object '*duration*' shall be set to a value between 1 and 5 minutes.

The value of **MPD.Metrics.Reporting**@reportinginterval or OMA DM QoE Management Object '*Interval*' shall be larger or equal than the value of **MPD.Metrics.Range**@duration or OMA DM QoE Management Object '*duration*'.

The DASH client reports collected metrics using quality metric reporting protocol defined in clause 10.6.

The mapping between QoE metrics and input to ITU-T P.1203 [49] Mode 0 is provided in table K-1 to K-4 below.

Table K-1 mapping between QoE metrics and input in ITU-T P.1203 mode 0 model for video stream

Video Metrics needed for ITU-T P.1203[49] model		QoE metrics defined in clause 10.3 and 10.4		Remark
Description	Value	Metric	Description	
Target Video bit-rate	Bit-rate in kbps.	Mpdinfo	Provides the MPD information for the representation or subrepresentation identified by representationid and subreplevel, if present. Related attributes: @bandwidth, @width, @height, and @codecs. Note: codec attribute includes video codec profile information and video frame rate information	Target Video bit-rate has been supported by @bandwidth attribute. Video frame-rate and Video codec and profile have been supported by @ codec attribute. Video encoding resolution has been supported by @width and @height attributes.
Video frame-rate	Frame rate in frames per second.			
Segment duration	Duration in seconds			
Video encoding resolution	Number of pixels (WxH) in transmitted video			
Video codec and profile	One of: H264-baseline, H264-high, H264-main			

		Play List	Provides information on when and how long different representations or subrepresentations have been used during the session.	Note that the actual timing of the required input for the P.1203 quality model may be influenced by actions such as play/pause.

Table K-2 mapping between QoE metrics and input in ITU-T P.1203 mode 0 model for audio stream

Audio Metrics needed for ITU-T P.1203 model		QoE metrics defined in clause 10.3 and 10.4		Remark
Description	Value	Metric	Description	
Target Audio bit-rate	Bit-rate in kbps.	Mpdinfo	Provides the MPD information for the representation or subrepresentation identified by representationid and subreplevel, if present. Related attributes: @bandwidth, and @codecs. Note: codec attribute includes audio codec profile information, audio sampling frequency and Number of audio channels information	Target Audio bit-rate has been supported by @bandwidth attribute. Audio codec, Audio sampling frequency and Number of audio channels have been supported by @ codecs attribute.
Segment duration	Duration in seconds			
Audio codec	One of: AAC-LC, AAC-HEv1, AAC-HEv2, AC3			
Audio sampling frequency	In Hz			
Number of audio channels	2			
		Play List	Provides information on when and how long different representations or subrepresentations have been used during the session.	Note that the actual timing of the required input for the P.1203 quality model may be influenced by actions such as play/pause.

Table K-3 mapping between QoE metrics and input in ITU-T P.1203 mode 0 model for stalling

Metrics needed for ITU-T P.1203 model		QoE metrics defined in clause 10.3 and 10.4		Remark
Description	Value	Metric	Description	
Buffering event start	The start time of the buffering/stalling event in seconds relative to the start of the original video clip, expressed in media time (not wall clock time) Note: This is 0 for initial buffering.	InitialPlayoutDelay	The playout delay for media start-up is measured as the time in milliseconds from the time instant of DASH player receives play-back-start trigger to the instant of media playout. If the MPD has been delivered earlier before the user clicks, it may include the process time of MPD, the fetch time of some media segments	It is only for initial buffering delay event

			which are required for media presentation, the process time of segments, and the time for media decode and render to the user. If no MPD has been fetched earlier, it also needs to add the fetch time of MPD.	
Event duration	The duration of the buffering/stalling event in seconds.	Play List	A list of playback periods. A playback period is the time interval between a user action and whichever occurs soonest of the next user action, the end of playback or a failure that stops playback.	For the buffering event afterwards

Table K-4 mapping between QoE metrics and input in ITU-T P.1203 mode 0 model for device information

Metrics needed for ITU-T P.1203 model		QoE metrics defined in clause 10.3 and 10.4		Remark
Description	Value	Metric	Description	
The resolution of the image displayed to the user	Number of pixels (WxH) in displayed video	Device Information	videoWidth and videoHeight	.
The device type on which the media is played	pc or mobile	Device Information	screenWidth x pixelWidth, screenHeight x pixelHeight	P.1203 classifies devices up to 12" as mobiles
The size of the video shown	WxH in mm	Device Information	videoWidth x pixelWidth, videoHeight x pixelHeight	

Annex L (normative): QoE Measurement Collection Functionalities

L.1 Configuration and reporting

As an alternative to configuration via MPD or OMA-DM, the QoE configuration can optionally be specified by the QoE Measurement Collection (QMC) functionality. In this case the QoE configuration is received via specific RRC [53] messages over the control plane, and the QoE reporting is also sent back via RRC messages over the control plane.

If QMC is supported, the UE shall support the following QMC functionalities:

- QoE Configuration: The QoE configuration will be delivered via RRC to the UE as a container according to "Application Layer Measurement Configuration" (see [53]). The container is an octet string with a maximum length of 1000 bytes, with gzip-encoded data (see [18]) stored in network byte order. The container shall be uncompressed, and is then expected to conform to XML-formatted QoE configuration data according to clause L.2 in the current specification. This QoE Configuration shall be forwarded to the DASH client.

- QoE Metrics: QoE Metrics from the DASH client shall be XML-formatted according to clause 10.6 in the current specification. The XML data shall be compressed with gzip (see [18]) and stored in network byte order into an octet string container with a maximum length of 8000 bytes. The container shall be delivered via RRC to the eNB according to "Application Layer Measurement Reporting" (see [53]). The behaviour if the compressed data is larger than 8000 bytes is unspecified in this version of the specification.

- The UE shall also set the QMC capability "QoE Measurement Collection for streaming services" (see [53]) to TRUE.

The exact implementation is not specified here, but an example signalling diagram below shows the QMC functionality with a hypothetical "QMC Handler" entity.
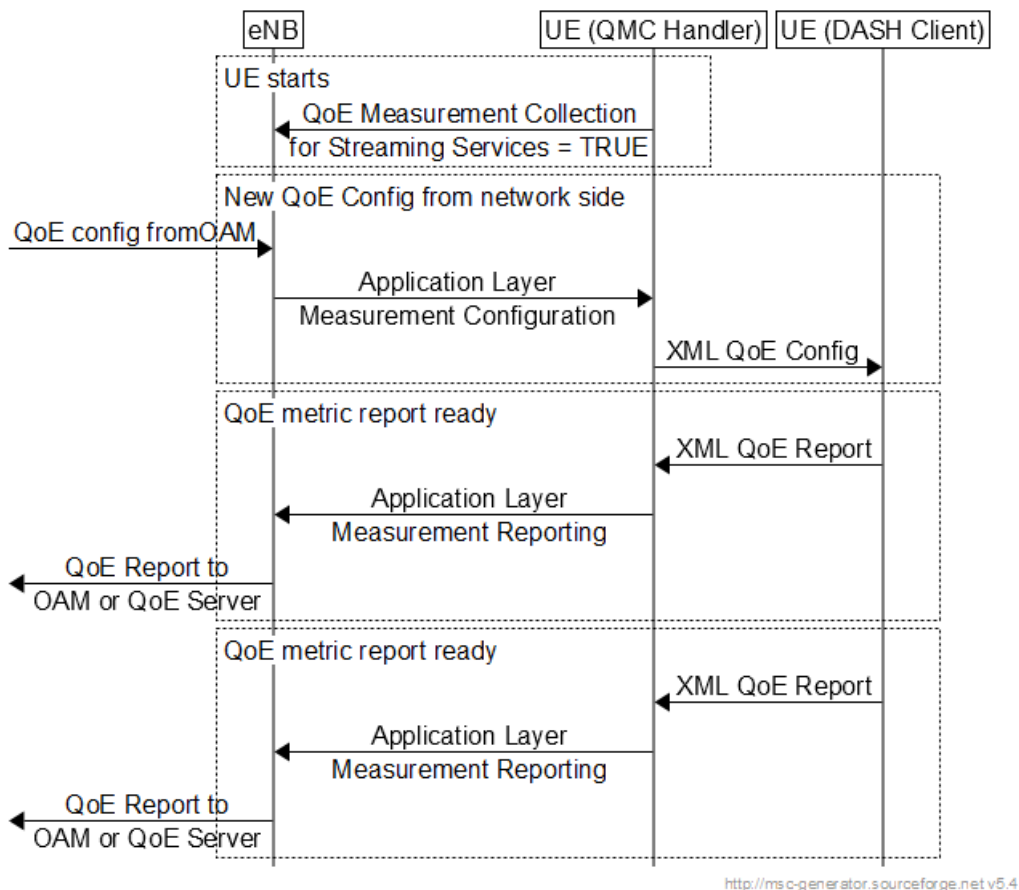
Figure L-1: example signalling diagram

Note that the QMC Handler is only shown here as one possible implementation, and it need not be implemented as such. The corresponding QMC functionality could be built into the DASH client or into other UE entities. In this version of the specification the detailed implementation of the above functionalities is left to the UE vendor.

L.2   XML configuration

When QoE reporting is configured via the QMC functionality, the configuration is done according to the XML schema below. The interpretation of the different elements and attributes are the same as described in clause 10.4 and 10.5 in the current specification.

Note that if geographical filtering is handled on the network side (i.e. QoE reporting is turned on/off by the network depending on the UE location), no LocationFilter should be specified in the QoE Configuration, as this would mean two consecutive filterings.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:3GPP:ns:PSS:DASH:QMC14"
    elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:sv="urn:3gpp:metadata:2016:PSS:schemaVersion"
    xmlns="urn:3GPP:ns:PSS:DASH:QMC14">

    <xs:element name="QmcConfiguration" type="QmcConfigurationType"/>

    <xs:complexType name="QmcConfigurationType">
        <xs:sequence>
            <xs:element name="Range" type="RangeType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="LocationFilter" type="LocationFilterType" minOccurs="0"/>
            <xs:element name="StreamingSourceFilter" type="StreamingSourceFilterType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="metrics" type="xs:string" use="required"/>
```

```xml
            <xs:attribute name="samplePercentage" type="xs:double" use="optional"/>
            <xs:attribute name="reportingInterval" type="xs:unsignedInt" use="optional"/>
            <xs:anyAttribute namespace="##other" processContents="lax"/>
        </xs:complexType>

        <xs:complexType name="RangeType">
            <xs:sequence>
                <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="startTime" type="xs:unsignedInt" use="optional"/>
            <xs:attribute name="duration" type="xs:duration" use="required"/>
            <xs:anyAttribute namespace="##other" processContents="lax"/>
        </xs:complexType>

<xs:complexType name="LocationFilterType">
        <xs:sequence>
            <xs:element name="cellID" type="xs:unsignedLong" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="shape" type="ShapeType" minOccurs="0"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="ShapeType">
        <xs:sequence>
            <xs:element name="PolygonList" type="PolygonListType" minOccurs="0"/>
            <xs:element name="CircularAreaList" type="CircularAreaListType" minOccurs="0"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="PolygonListType">
        <xs:annotation>
            <xs:documentation> see [OMA MLP] </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="Polygon" minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ConfLevel" type="xs:unsignedInt" use="optional"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="CircularAreaListType">
        <xs:annotation>
            <xs:documentation> see [OMA MLP] </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="CircularArea" minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ConfLevel" type="xs:unsignedInt" use="optional"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="StreamingSourceFilterType">
        <xs:sequence>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="streamingSource" type="xs:string" use="required"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
</xs:schema>
```

Annex M (informative):
Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2011-06	52	SP-110305			Version 10.0.0 approved at TSG SA#52		10.0.0
2011-11	54	SP-110794	0003		Alignment with MPEG DASH	10.0.0	10.1.0
2011-11	54	SP-110794	0005	3	QoE Updates for Correction, Clarification and MPEG DASH Alignment	10.0.0	10.1.0
2011-11	54	SP-110794	0006	2	QoS Support for 3GP-DASH Services	10.0.0	10.1.0
2012-06	56	SP-120221	0007	3	Alignment with MPEG DASH	10.1.0	10.2.0
2012-06	56	SP-120221	0010	1	Correction of Table Reference for Change Commands in MPD Deltas	10.1.0	10.2.0
2012-06	56	SP-120221	0012	2	ContentProtection element update to signal version of DRM system	10.1.0	10.2.0
2012-09	57	SP-120504	0009	3	QoE Reporting for DASH over Combined MBMS Download and HTTP-based Delivery	10.2.0	11.0.0
2012-09	57	SP-120509	0014	6	Inclusion of MVC support for DASH	10.2.0	11.0.0
2012-09	57	SP-120509	0015	4	Inclusion of 3D Video Format Information in DASH MPD	10.2.0	11.0.0
2012-12	58	SP-120761	0016	7	Supporting HTTP Partial Response	11.0.0	11.1.0
2013-03	59	SP-130019	0022	1	DASH QoE Reporting Schema bug fix	11.1.0	11.2.0
2013-03	59	SP-130015	0024	1	DASH QoE Reporting Example bug fix	11.1.0	11.2.0
2013-06	60	SP-130184	0026	2	DASH Profiles correction	11.2.0	11.3.0
2013-06	60	SP-130184	0028	1	Correction of Cardinality of DeltaSupport element in Table 8-5	11.2.0	11.3.0
2013-06	60	SP-130184	0030	2	Correction of MPD Delta MIME type information	11.2.0	11.3.0
2013-06	60	SP-130184	0031		General Corrections to DASH	11.2.0	11.3.0
2013-09	61	SP-130350	0037		Corrections to DASH	11.3.0	11.4.0
2013-09	61	SP-130354	0035		3D Video Disparity Range and 3D Video Display Information Descriptor Schemes	11.4.0	12.0.0
2013-12	62	SP-130565	0041	1	Registration of MIME Type for QoE Reports	12.0.0	12.1.0
2013-12	62	SP-130566	0043		Correction on Playlist Metric Attribute	12.0.0	12.1.0
2013-12	62	SP-130578	0044		Update of MPD Delta MIME type information	12.0.0	12.1.0
2013-12	62	SP-130564	0047	1	3GP-DASH MPD Schema Fix	12.0.0	12.1.0
2013-12	62	SP-130565	0050		Corrections to MPD Delta example	12.0.0	12.1.0
2014-03	63	SP-140006	0053	1	Corrections to 3GP-DASH	12.1.0	12.2.0
2014-03	63	SP-140013	0054	1	Missing media decoders specification for 3GP-DASH in MBMS	12.1.0	12.2.0
2014-03	63	SP-140009	0055	1	HEVC support	12.1.0	12.2.0
2014-06	64	SP-140205	0058	2	Registration of MIME Type for QoE Reports	12.2.0	12.3.0
2014-06	64	SP-140205	0061	2	Correction of Cardinality of the SegmentList Element	12.2.0	12.3.0
2014-12	66	SP-140720	0067	1	Correction on Registration of MIME Type for QoE Reports	12.3.0	12.4.0

2014-12	66	SP-140732	0075	1	MI-EMO DASH Robustness	12.3.0	12.4.0
2014-12	66	SP-140737	0074	2	A new QoE metric of playout delay for media start-up	12.4.0	13.0.0
2015-09	69	SP-150449	0082	3	Alignment with ISO/IEC 23009-1:2014 (MPEG-DASH)	13.0.0	13.1.0
2015-09	69	SP-150449	0083	3	Partial File Handling	13.0.0	13.1.0
2015-12	70	SP-150656	0084	3	Event mechanism support for eDASH	13.1.0	13.2.0
2015-12	70	SP-150656	0085	1	Proposed Guidelines on DASH	13.1.0	13.2.0
2015-12	70	SP-150656	0087		DASH enhancements	13.1.0	13.2.0

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2016-06	SA#72	SP-160262	0088	2	F	eDASH: $Time$ needs to go	13.3.0
2016-12	SA#74	SP-160773	0090	4	B	DASH Audio-video MOS support	14.0.0
2016-12	SA#74	SP-160945	0091	3	B	Location based filter and certain streaming program filter support for DASH	14.0.0
2017-03	SA#75	SP-170027	0093	1	C	Remove the HTTP metrics from the QoE reporting	14.1.0
2017-03	SA#75	SP-170027	0094	1	C	QoE Measurement Collection functionalities for QoE reporting	14.1.0
2017-06	SA#76	SP-170323	0095	-	F	Correction for QoE Session Configuration	14.2.0
2017-06	SA#76	SP-170323	0096	-	F	Correction for QoE Device Information	14.2.0
2017-06	SA#76	SP-170323	0097	-	F	Correction for QoE Geographical and Source Filtering	14.2.0
2017-06	SA#76	SP-170323	0098	1	F	Correction for QoE Measurement Collection	14.2.0
2017-09	SA#77	SP-170601	0100	2	F	Correction for Clause Quotation of the Quality Reporting Schemes	14.3.0
2017-09	SA#77	SP-170601	0106	-	F	QoE XML and OMA-DM corrections	14.3.0
2017-09	SA#77	SP-170601	0107	-	F	Clarification for the PlayList Metric	14.3.0
2017-12	SA#78	SP-170822	0118	-	F	Add missing QoE metric to Annex K	14.4.0
2017-12	SA#78	SP-170821	0121	1	A	TV Service Support in DASH	14.4.0
2017-12	SA#78	SP-170821	0126	-	F	Correction to QoE Reporting	14.4.0
2018-03	SA#79	SP-180022	0130	1	F	Correction of QMC Configuration	14.5.0
2018-03	SA#79	SP-180022	0131	3	F	Correction on segment duration information	14.5.0

History

Document history		
V14.1.0	April 2017	Publication
V14.2.0	July 2017	Publication
V14.3.0	October 2017	Publication
V14.4.0	January 2018	Publication
V14.5.0	April 2018	Publication