3GPP TR 26.938 V1.4.0 (2013-09)

Technical Report

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Packet-switched Streaming Service (PSS);
Improved Support for Dynamic Adaptive Streaming over HTTP in 3GPP
(Release 12)

Keywords

UMTS, LTE, IP, MBMS, codec, broadcast, multimedia

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Contents

Foreword

This Technical Report 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.

Introduction

3GPP's Dynamic Adaptive Streaming over HTTP (DASH) specifications were developed in Rel-9 and Rel-10 and are available in TS 26.247 [2]. Despite being integrated into the PSS architecture, the specifications have significant flexibility for deployments also outside the 3GPP services. This has been recognized by other organizations such as MPEG and Open IPTV Forum. In continuous alignment efforts, 3GPP and MPEG have developed a generic format for Dynamic Adaptive Streaming over HTTP (DASH).

These specifications serve an urgent need: With the evolution of radio access technologies towards HSPA & LTE higher data rates are provided allowing more feature rich services with higher quality and access to multimedia services has grown significantly. And the most popular multimedia services today are services delivered over HTTP. Serving content from standard HTTP-servers has many advantages in terms of deployment costs and convergences with regular web services.

With the completion of the specifications, first deployments of services based on DASH and similar technologies are happening. The experiences from initial deployments of massively scalable video streaming delivery over HTTP and advanced radio access technologies result in new use cases, demands and requirements. Improvements for the support of DASH when delivered over 3GPP networks and architectures are expected to be necessary and deployments guidelines are important. Considered improvements are in the area of improved user experience, improved bandwidth efficiency or more efficient delivery over HTTP-caching infrastructures. Furthermore, the combination of DASH with other services and technologies is an ongoing challenge and effort. Not limited to this, but some examples are the delivery of DASH over different 3GPP radio access networks, the combination with presentation technologies such as HTML-5, the support of advanced content protection schemes, or the support for QoS in 3GPP networks.

Service improvements might not require additional TS 26.247 [2] specification work, but do require a detailed analysis of the envisaged use cases, the resulting requirements, the ability to solve these use cases with the existing 3GPP and/or other existing specifications and provide guidelines and deployment examples. The analysis of the use cases may lead to additional specification work, but this should first be identified and justified from the above analysis.

1 Scope

The present document covers:

- deployment guidelines for DASH in 3GPP networks and architectures,
- use cases for the improved support of DASH in 3GPP networks and architectures as well as requirements to support those use cases,
- recommendations for potentially necessary normative specification work in 3GPP,
- recommendations for the documentation of potentially informative guide lining work.

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 TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 26.247: "Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH)".

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

[4] 3GPP TS 26.346: "Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs".

[5] 3GPP TS 26.244: "Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)".

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

[7] ISO/IEC 14496-12 | 15444-12: "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".

[8] IETF RFC 6726 (November 2012): "FLUTE - File Delivery over Unidirectional Transport", T. Paila, M. Luby, R. Lehtonen, V. Roca, R. Walsh.

[9] ISO/IEC 23009-1:2012/Cor1:2013 " Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats".

[10] ISO/IEC 23009-3:2013 " Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 3: Implementation and Deployment Guidelines".

[11] ISO/IEC 23009-2:2013 " Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 2: Conformance and Reference Software".

[12] W3C Working Draft, XMLHttpRequest

[13] ISO/IEC 23009-1:2012/DAmd 1" Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats, Support for Event Messages and extended Audio Channel Configuration".

[14] International Telecommunication Union. *Methodology for the Subjective Assessment of the Quality of Television Pictures*. ITU-R Recommendation BT.500-11. 2002.

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [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 [1].

altitude: number indicating the altitude in meters. The reference altitude, indicated by zero, is set to the sea level.

digital zoom: number indicating the enlargement scale factor of the image due to cropping and interpolating the pixel dimensions back to the original size.

latitude: number indicating the latitude in degrees. Negative values represent southern latitude.

longitude: number indicating the longitude in degrees. Negative values represent western longitude.

optical zoom: number indicating the optical magnification scale factor.

pan: number measured in degrees and corresponding to the compass direction of the component in the plane parallel to the earth's surface of any vector which points in the same direction that the camera is facing. For example, North corresponds to 0 degrees, East corresponds to 90 degrees, etc. If the camera is pointing in a direction perpendicular to the earth's surface (either straight up at the sky or straight down at the ground), then the value of Pan is undefined. For the direction corresponding to Pan, it is useful to have an indication of whether the direction is "true" or "magnetic".

tilt: number measured in degrees corresponding to the rotational position about the axis in the plane of constant amplitude through the camera centre that is perpendicular to the Pan direction. For example, if the camera is pointing parallel to the earth's surface, Tilt is 0. If the camera is pointing straight up towards the sky, the Tilt is 90 degrees and if the camera is pointing straight down towards the earth Tilt is -90 degrees.

rotation: number measured in degrees corresponding to the rotational position about the axis in the direction that the camera is facing. Since Tilt and Rotation are independent parameters, Rotation is defined for a Tilt value of 0, i.e. the camera is first tilted to be pointing parallel to the earth's surface in the direction that would correspond to Pan. Rotation is then the amount of counter-clockwise rotation about the axis that the camera is facing needed to bring a vector initially pointing straight up towards the sky into alignment with the camera "up" direction. In the event that Pan is undefined as the camera is either pointing straight up or straight down, Rotation can be defined as the amount of rotation needed to bring a vector initially pointing North into alignment with the camera "up" direction.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] 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 [1].

3GP	3GPP file format
3GP-DASH	3GPP Dynamic Adaptive Streaming over HTTP
AHS	Adaptive HTTP Streaming
DASH	Dynamic Adaptive Streaming over HTTP (DASH)
FLUTE	File Delivery over Unidirectional Transport
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
MPD	Media Presentation Description
PSS	Packet-switched Streaming Service
SAP	Stream Access Point

URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name

4 Relevant Specifications

4.1 Overview

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 [9] (MPEG-DASH) as well as for 3GPP's Release 10's DASH specification [2] (3GP-DASH).

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

- ISO/IEC 23009-2: Conformance and Reference software [11]
- ISO/IEC 23009-3: Implementation and Deployment Guidelines [10]

Due to the close coordination in the development of 3GP-DASH it is achieved that 3GPP Release 10 DASH can be viewed as a profile of MPEG-DASH with minor extensions. Clause 5.2 provides a description on how 3GP-DASH may be represented as a profile of MPEG-DASH. Beyond the formats defined in both specifications, 3GP-DASH also defines the transport protocol when deployed within PSS as being HTTP [6]. Furthermore, 3GPP also supports the delivery of DASH formats within MBMS [4] using FLUTE [8] as the delivery protocol.

Both specifications rely for the segment formats on the ISO base media file format [7]. In 3GPP, compatibility is achieved with the 3GPP file format [5].

4.2 3GP-DASH as a profile of MPEG-DASH

4.2.1 General

The 3GP-DASH Release-10 profile as defined in TS 26.247 [2], section 7.3.3 and identified by the URN "`urn:3GPP:PSS:profile:DASH10`" may be described as an MPEG-DASH profile as follows:

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

The `@mimeType` attribute of each Representation are expected to be provided according to RFC4337. Additional parameters may be added according to RFC 6381 [26].

4.2.2 Media Codecs

For the 3GP-DASH Release-10 profile clients supporting a particular continuous media type, the corresponding media decoders are specified in TS 26.234 [3], clause 7.2 for speech, clause 7.3 for audio, clause 7.4 for video, clause 7.9 for timed text and clause 11 for timed graphics.

4.2.3 Media Presentation Description constraints

The Media Presentation Description are expected to conform to the following constraints:

- The rules for the MPD and the segments as defined in ISO/IEC 23001-9, section 7.3, apply.
- Representations with value of the `@mimeType` attribute other than `video/mp4`, `video/3gp`, `audio/3gp` or `audio/mp4` may be ignored. Additional profile or codec specific parameters may be added to the value of the MIME type attribute. For details refer to specific parameters below.
- The **`Subset`** element may be ignored.

- Any **SegmentBase, SegmentTemplate** or **SegmentList** element that contain a **SegmentTimeline** element may be ignored.
- Any Representation that contains a **FramePacking** element may be ignored.
- Any Representation that contains an @scanType attribute with value other than "progressive" may be ignored.

4.2.4 Segment format constraints

Representations and Segments referred to by the Representations in the profile-specific MPD for this profile, the following constraints are expected to be met:

- Representations are expected to comply with the formats defined in clause 7.3 in ISO/IEC 23009-1.
- Representations are expected to comply with a 3GP file format profile in a sense that the profile parameter of the @mimeType attribute will contain the '3gh9' brand.

4.2.5 Extensions

4.2.5.1 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 1. The XML-syntax of **x3gpp:DeltaSupport** element is provided in Table 2.

Table 1: 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 will 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 2: 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 includes 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 3.

Table 3: 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 will always be available to clients for regular MPD updates. MPD Delta related procedures are optional at the client.

5 Deployment Guidelines

5.1 Introduction

Deployment guidelines include

- instructions on how content may be offered using the DASH formants,
- instructions on relevant client implementation and operation aspects,
- operational guidelines on how operate a DASH-based service.

5.2 Content Authoring Guidelines

For generic content authoring guidelines please refer to ISO/IEC 23009-3 [10]. 3GPP-specific aspects are TBD.

5.3 Client implementation and client operation guidelines

5.3.1 Guidelines for rate adaptation

5.3.1.1 Introduction

DASH merely specifies the formats for Media Presentation Description (MPD) and media segments, while client operations, such as rate adaptation algorithms, are not specified. The operation of the rate adaptation algorithm might affect the perceived presentation quality as media segments from different representations within one Adaptation Set could be selected to present the same media content component. The perceived quality might be affected by the bitrate of the selected media rate as well as potentially experienced interruptions from buffer underruns due to non-timely arrival of the media segments.

5.3.1.2 Rate adaptation in DASH

When designing rate adaptation algorithm for DASH, one should consider among others:

- that the rate adaptation algorithm is efficiently utilizing the sharable network capacities, which affects playback media quality,
- that the rate adaptation algorithm is capable of detecting network congestion and is able to react promptly to prevent playback interruption,
- that the rate adaptation algorithm can provide stable playback quality even if the network delivery capacities fluctuate widely and frequently,
- that the rate adaptation algorithm is able to tradeoff maximum instantaneous quality and smooth continuous quality, for example by smoothing short-term fluctuation in the network delivery capacities by using buffering, but still switch to better presentation quality/higher bitrates if more long-term bandwidth increase is observed,
- that the rate adaptation algorithm is able to avoid excessive bandwidth consumption due to over-buffering media data.

When implementing rate adaptation in DASH, one could balance between different criteria listed above to improve the overall Quality of Experience (QoE) perceived by the user. The guideline of rate adaptation in DASH is summarized based on the QoE metrics specified in 3GPP TS 26.247 [2].

In absence of other information, e.g. from the radio network status, the measurement for certain QoE metrics may be used in rate adaptation in DASH, e.g.:

- average throughput: average throughput measured by a client in a certain measurement interval;
- Segment Fetch Time (SFT) ratio: the ratio of Media Segment Duration (MSD) divided by SFT. MSD and SFT denote the media playback time contained the media segment and the period of time from the time instant of sending a HTTP GET request for the media segment to the instant of receiving the last bit of the requested media segment, respectively;
- buffer level: buffered media time at a client.

5.4 Operational and deployment guidelines

5.4.1 General

For general operational and deployment guidelines see ISO/IEC 23009-3 [10]. 3GPP-specific aspects are TBD.

5.4.2 Proxy/cache switch for DASH service

5.4.2.1 Assumptions

It is assumed that many HTTP proxy/cache that also support DASH are physically located at or close to different P-GWs or for different RATs, can be deployed within the operator domain.

5.4.2.2 Description

Operator A owns both LTE and WLAN networks and proxy caches are provided for both LTE and WLAN network. Robert accesses a DASH service over an LTE network of operator A. A proxy cache is placed inside the LTE network to reduce latency and scalability in order to optimize HTTP delivery over TCP/IP. Robert enters the office covered by the WLAN network also owned by operator A. The operator A knows LTE traffic surging in the office area and the WLAN network is available for traffic offloading. Robert's UE switches to WLAN for the DASH service. The content is served from another proxy cache within the WLAN network as the proxy cache physically close to the UE and therefore provides higher TCP/IP throughput due to reduced latency.

Note:

It is assumed that mobility events may cause a change in proxy cache.

5.4.2.3 Working assumption

- DASH client may be served by another proxy cache after a mobility event occurs

5.4.2.4 Recommended Requirements

- A change in proxy cache should have minimum impact on the user experience of the DASH service.

6 Use Cases

6.1 Introduction

The following section introduces use cases that are either supported by 3GP-DASH [2], possibly in combination with other 3GPP technologies, or the use cases are in the context of DASH, but may require extensions in 3GP-DASH or other technologies. Therefore, the use cases are analyzed, potential solutions reusing existing technologies are provided and potential gaps are identified.

6.2 Advertisement Insertion - Insertion into On-Demand Content

6.2.1 Description

6.2.1.1 Fixed duration advertisement

In a subway, Paul who is 18 years old boy visits the web site for CoD service with DASH enabled-device. He selects a movie. After the movie is played for 10 min on the screen, the actor drives a luxury sports car with his girl friend. Then the advertisement related to the BMW car company is played for 1 min. After finishing the advertisement, the main movie is continued.

At home, Brian who is 18 years old boy visits the web site for watching the same CoD movie with DASH enabled-device. After the movie has started for 10 min, the same advertisement related to the BMW car company is played for 1min. After finishing the advertisement, the main movie is continued.

6.2.1.2 Variable duration advertisement

Paul who is 18 years old boy visits the web site for CoD service with DASH enabled-device. He selects an action movie. After the movie is played for 10 min on the screen the actor drives a luxury sports car. Then the advertisement related to the car company is played for 1 min. After finishing the advertisement, the main movie is restarted.

Lisa who is 18 years old girl visits the web site for watching the same CoD movie with DASH enabled-device and after the movie have started for 10 min, then the long advertisement for the actress' dresses on is played for 1 min 30 seconds, because she likes shopping. After finishing the advertisement, the main movie is restarted.

6.2.1.3 Trusted Client Playback

Operator Hudu wants to deliver ad supported TV content only to DASH clients that are trusted to play ads according to business rules. Example playback rules could be to not skip ads in fast forward that have not been previously played, and to re-evaluate Remote Periods when they are replayed using rewind/replay.

6.2.2 Analysis in the Context of Rel-10 TS 26.247 [2]

This use can be fulfilled

- by a server that dynamically generates a targeted MPD for each user. Periods are concatenated with appropriate start times.
- If only a common MPD will be used then Periods can be inserted in a remote element and the remote element provides resolution based on user-preferences/location, etc.

Generally, the existing mechanisms in TS 26.247 are sufficient to fulfil the use cases considered in clause 6.2.1.

The benefit of signalling events of MPD updates will be investigated in the context of this use case.

6.2.3 Examples

6.2.3.1 Fixed Duration in On-Demand

The following is an example MPD containing fixed advertisement in the middle of an on demand movie.

```
<?xml version="1.0"?>
<MPD
    xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="urn:mpeg:dash:schema:mpd:2011"
    schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
    type="static"
    mediaPresentationDuration="PT30M"
    availabilityStartTime="2011-12-25T12:30:00"
    minBufferTime="PT4S"
    profiles=" urn:3GPP:PSS:profile:DASH10">

    <BaseURL>http://cdn1.example.com/</BaseURL>
    <BaseURL>http://cdn2.example.com/</BaseURL>

    <Period start="PT0.00S" duration="PT1800S">
        <AdaptationSet mimeType="video/mp4" codecs="avc1.640828" frameRate="30000/1001"
             segmentAlignment="true" startWithSAP="1">
            <BaseURL>video_1/</BaseURL>
            <SegmentTemplate timescale="90000" initialization="$Bandwidth%/init.mp4v"
             media="$Bandwidth%/$Time$.mp4v"/>
            <Representation id="v0" width="320" height="240" bandwidth="250000"/>
            <Representation id="v1" width="640" height="480" bandwidth="500000"/>
            <Representation id="v2" width="960" height="720" bandwidth="1000000"/>
        </AdaptationSet>
    </Period>

    <!-- Advertisement -->
    <Period duration="PT180S">
        <!-- Video -->
        <AdaptationSet mimeType="video/mp4" codecs="avc1.640828" frameRate="30000/1001"
             segmentAlignment="true" startWithSAP="1">
            <BaseURL>ad/</BaseURL>
```

```
            <SegmentTemplate timescale="90000" initialization="$Bandwidth%/init.mp4v"
             media="$Bandwidth%/$Time$.mp4v"/>
            <Representation id="ad1" width="640" height="480" bandwidth="480000"/>
            <Representation id="ad2" width="960" height="720" bandwidth="9600000"/>
        </AdaptationSet>
    </Period>

    <Period duration="PT1800S">
        <AdaptationSet mimeType="video/mp4" codecs="avc1.640828" frameRate="30000/1001"
             segmentAlignment="true" startWithSAP="1">
            <BaseURL>video_2/</BaseURL>
            <SegmentTemplate timescale="90000" initialization="$Bandwidth%/init.mp4v"
              media="$Bandwidth%/$Time$.mp4v"/>
            <Representation id="v0" width="320" height="240" bandwidth="250000"/>
            <Representation id="v1" width="640" height="480" bandwidth="500000"/>
            <Representation id="v2" width="960" height="720" bandwidth="1000000"/>
        </AdaptationSet>
    </Period>
</MPD>
```

6.2.3.2 Targeted Advertisements

The following is an example MPD containing a remote period which is actually the advertisement in the middle of an on demand movie. When resolving the xlink, the advertisement server selects advertisement according to the request URL and feed back the XML element with URLs for the advertisement segment:

```
<?xml version="1.0"?>
<MPD
    xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="urn:mpeg:dash:schema:mpd:2011"
    xlink="http://www.w3.org/1999/xlink"
    schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
    type="static"
    mediaPresentationDuration="PT30M"
    availabilityStartTime="2011-12-25T12:30:00"
    minBufferTime="PT4S"
    profiles=" urn:3GPP:PSS:profile:DASH10">

    <BaseURL>http://cdn1.example.com/</BaseURL>
    <BaseURL>http://cdn2.example.com/</BaseURL>

    <Period start="PT0.00S" duration="PT1800S">
        <AdaptationSet mimeType="video/mp4" codecs="avc1.640828" frameRate="30000/1001"
             segmentAlignment="true" startWithSAP="1">
            <BaseURL>video_1/</BaseURL>
            <SegmentTemplate timescale="90000" initialization="$Bandwidth%/init.mp4v"
             media="$Bandwidth%/$Time$.mp4v"/>
            <Representation id="v0" width="320" height="240" bandwidth="250000"/>
            <Representation id="v1" width="640" height="480" bandwidth="500000"/>
            <Representation id="v2" width="960" height="720" bandwidth="1000000"/>
        </AdaptationSet>
    </Period>

    <!-- Advertisement -->
    <Period xlink:href="http://adserver.com/movie_11/ad_1.period" xlink:actuate="onRequest"
        duration="PT180S"/>
        <Period duration="PT1800S">
        <AdaptationSet mimeType="video/mp4" codecs="avc1.640828" frameRate="30000/1001"
             segmentAlignment="true" startWithSAP="1">
            <BaseURL>video_2/</BaseURL>
            <SegmentTemplate timescale="90000" initialization="$Bandwidth%/init.mp4v"
             media="$Bandwidth%/$Time$.mp4v"/>
            <Representation id="v0" width="320" height="240" bandwidth="250000"/>
            <Representation id="v1" width="640" height="480" bandwidth="500000"/>
            <Representation id="v2" width="960" height="720" bandwidth="1000000"/>
        </AdaptationSet>
    </Period>
</MPD>
```

6.3 Advertisement Insertion - Insertion into Live Content

6.3.1 Description

6.3.1.1 Accessing Live Content

A service provider offers a free live service. When accessing the live service, the service provider wants to insert an advertisement targeted to the user. The duration of the advertisement may be different, depending on the access time, the access location, the user, etc. Only after having completed the advertisement, the URL of the live service is provided.

6.3.1.2 Advertisement in Live Content

When continuing the service, the service provider wants to add a targeted advertisement of a fixed duration at predetermined positions. Depending on certain business rules certain users may also continue to watch the cheerleaders during the break of the live event.

6.3.2 Analysis in the Context of Rel-10 TS 26.247 [2]

The use case in section 6.3.1.1 can be fulfilled "by concatenating" a targeted MPD and the live MPD. However, this is part of a presentation layer outside the scope of DASH.

The use case in section 6.3.1.2 is supported in the same way as discussed in section 6.2.1.1 for fixed duration advertisement. If only a common MPD will be used then targeted advertising is supported by using a remote element for the targeted advertisement.

6.4 Advanced Support for Live Services

6.4.1 Description

6.4.1.1 Setup

A service provider wants to provide a live soccer event using DASH that can potentially be accessed by millions of users. The service provider provides redundant infrastructure in terms of encoders and servers to enable a seamless switch-over in case any of the components fail during the live event or get overloaded.

6.4.1.2 Use Case A

Anna accesses the service in the bus with her mobile DASH-enabled device, and the service is available immediately.

6.4.1.3 Use Case B

Continuing Use Case A, across from her sits Paul, who watches the event on his DASH-enabled laptop. A goal is scored and both, despite watching on different screens, celebrate this event at the same time.

6.4.1.4 Use Case C

Continuing Use Case B, Other people that follow the game on a 3GPP Rel-6 PSS terminal observe the goal within a similar time.

6.4.1.5 Use Case D

Continuing Use Case C, Another goal is scored. Paul tells Anna that the first goal in the game was even more exciting and Anna uses the offering that she can view the event 30 minutes back in time on her DASH-enabled device. After having seen the goal she goes back to the live event.

6.4.1.6 Use Case E

Continuing Use Case D, the football match gets into overtime, the star player of CF Anolacrab, Lenoil Issem, is brought into the game by the coach of the year, Aloidraug, hits twice the post, but can not score. Due to the extraordinary tension in the match, more and more users join such that the service provider requires migrating the service to the redundant infrastructure without interrupting the service to the users.

6.4.1.7 Use Case F

Continuing Use Case E, finally penalty shooting is necessary. The live event is interrupted by a short break during which advertisement is added. The exact timing of the ad breaks is unknown due to the extra time of the extension and the start of the penalty shooting is delayed.

6.4.2 Operation with MPD dynamic Mode

6.4.2.1 Introduction

This section provides an overview on using the MPD dynamic mode and how a client can make use of MPD offerings. The focus is on the client operation here. Details on a possible service offering to fulfil the use cases in section 2.1 is provided below.

6.4.2.2 Problem Statement

Generally, an HTTP streaming client accesses and downloads a manifest, based on which it would like to initiate the live session. Based on this manifest, and for each selected Representation, the client needs to take several decisions:

1) Determine what is the latest segment that is available on server.
2) Determine the segment availability start time of the next segment and possibly future segments.
3) Determine when to start playout the segment and from which presentation timeline in the segment in order to be as close as possible to the live edge.
4) Determine when to check for an updated manifest.

6.4.2.3 Existing Technologies

In existing non-DASH streaming technologies these issues are solved as follows:

- for each segment that is made available, the server publishes a new manifest;
- the client, once joining the service, gets the latest manifest, looks at the playlist and then can access the newest segment;
- the client starts playing out the segment and expects, when playing the segment from the beginning, that it can continue accessing the next segment in time;
- before fetching a new segment (or requiring to fetch one), the client fetches a new manifest providing the location where to get the latest segment.

6.4.2.4 Consequences with Existing Technologies

The following consequences result from this simplified live operation:

- The manifest is updated on the server with each newly available segment:
    - This requires the client to fetch the manifest and use the information in the manifest whenever they join, i.e. joining means manifest fetching and the manifest needs to be the latest.

- This requires that the server needs to update the manifest to accommodate the change whenever a new Segment if produced. The manifest renewal is especially critical in cases where the manifest is distributed through FLUTE or needs to be pushed into caches. In this case along with each new segment, a new manifest needs to be pushed.

- The client does not have any insight at what time the next segment is available/published on the server:
  - It will expect that the next segment is published at the latest after segment duration time. This can be verified by updating the manifest prior to fetching a new segment.

- The client does not have any insight if any presentation time later than the earliest presentation time of the latest available segment can be played out in order to get closer to the live edge without a risk of rebuffering later:
  - As a fact of the loose timing model, and the client not knowing when the next segment becomes available, it can only assume that the earliest presentation time can played.

- The client does not have any insight if playout of other clients that download the same segment is synchronized.

- The client needs to fetch a new manifest when joining the service to obtain the latest information. This "fetching" requires at least one manifest fetch round-trip time and may increase start-up.

In summary, the main reason for all these issues is that existing solutions do not provide a good idea on the exact time schedule of the manifest and media segment creation. As an example, if one operates on 10-second segments, the client has little insight whether the manifest had just been published, or whether it will be published shortly after. So you may still be off by 10-epsilon seconds. In addition, it requires updating the manifest frequently with every segment. No reference clock is available to the client that enables a playout that is closer to the live edge or enables playout synchronized with other clients. At the same time, hiding the publish time from the clients typically provides ensures that the requests for segments from different clients are spread.

6.4.2.5 How does DASH solve this?

6.4.2.5.1 Overview

DASH attempts to address the above-mentioned weaknesses, namely:

- to operate closer to the live edge,
- to synchronize playout of clients that are consuming the same media presentation,
- to avoid regular updates of the MPD on the server and fetches by the client, and
- to avoid fetching the MPD in real-time when joining the service.

DASH uses a wall-clock time documented in the MPD, which sets up the live Media Presentation. DASH assumes that the MPD is generated such that the MPD generation process does have access to an accurate clock. This enables that clients that are synchronized to the wall-clock time by any means can operate closer to the live edge.

6.4.2.5.2 Benefits of this approach

In case the template construction with `@duration` is used, the above approach provides several advantages compared to existing solutions:

1) The MPD does not have to be updated on the server as long as the segment construction can be continued. As long as the client records the fetch time of the MPD, it can download the MPD ahead of time (or keep it in the buffer) for several different services that are anticipated to be accessed, for example different channels.

2) Also, in a multicast environment, the MPD can be distributed only once or at least with a much smaller frequency than for every new segment.

3) The client knows exactly the time when the next segment is available/published on the server. This permits operation closer to the live edge as the client can request the segment as soon as it gets available.

4) In order to accurately tune to the live edge, the client may start presentation of the first segment not from the start, but even somewhere in the middle. The exact timing is obtained by mapping the presentation time to the live edge time.

5) The client can synchronize its playout with other clients.

6) Server operation is simple, i.e. no special server beyond HTTP is required.

DASH uses a wall-clock time documented in the MPD, which sets up the live Media Presentation. DASH assumes that the MPD is generated such that the MPD generation process does have access to an accurate clock. This enables that clients that are synchronized to the wall-clock time by any means can operate closer to the live edge.

Specifically, the following information is available in the MPD when using a number-template-based Representations and using the using the `@duration` attribute:

- **`MPD`**`@availabilityStartTime`: the start time is the anchor for the MPD in wall-clock time. The value is denoted as *AST*.
- **`MPD`**`@minimumUpdatePeriod`: the minimum update period of the MPD. The value is denoted as *MUP*.
- **`MPD`**`@suggestedPresentationDelay`: suggested presentation delay as delta to segment availability start time. The value is denoted as *SPD*.
- **`MPD`**`@minBufferTime`: minimum buffer time, used in conjunction with the `@bandwidth` attribute of each Representation. The value is denoted as *MBT*.
- **`MPD`**`@timeShiftBufferDepth`: time shift buffer depth of the media presentation. The value is denoted as *TSB*.
- **`Period`**`@start`: the start time of the Period relative to the MPD availability start time. The value is denoted as *PS*.
- **`SegmentTemplate`**`@startNumber`: number of the first segment in the Period. The value is denoted as *SSN*.
- **`SegmentTemplate`**`@duration`: the duration of a segment in units of a time. The value divided by the value of `@timescale` is denoted as d.

Also assume that the client did fetch the MPD at fetch time *FT*. Note that a reasonable estimate on the lower value of *FT* is the time when the request for then new MPD is issued and for the higher value *FT* when the MPD is received.

6.4.2.6 MPD Times

For using the same concept with different addressing schemes, the following two values are introduced according to ISO/IEC 23009-1:

- the position of the segment in the Period denoted as k with k=1,2,...
- The MPD start time of the segment at position k, referred to as $MST(k)$.
- The MPD duration of a segment at position k, referred to as $MD(k)$.

Assuming now that the wall-clock time at the client is denoted at *WT*, and then the client can derive the following information:

1. the latest available Period on the server, denoted by its period start time PS^*
2. The segment availability start time of any segment at position k within the Period, denoted as $SAST(k)$.
3. The position of the latest segment that is available on server in the Period, referred to as k^*
4. The address of the latest segment that is available on server
5. The time when to fetch a new MPD based on the current presentation time, or more specifically, the greatest segment position k' within this Period that can be constructed by this MPD.

6. The media presentation time within the Representation that synchronizes closest to the live edge, *MPTL*.

7. The media presentation time within the Representation that synchronizes to other clients, *MPTS*.

6.4.2.7 General Derivation

Using these times, the values from above can be derived as:

1. The latest Period is obtained as the Period for which $AST+PS+MD(1) <= NTP$.

2. The segment availability start time is obtained as:

$$SAST(k) = AST + PS + MST(k) + MD(k)$$

Specifically, For the number-based template with *d* the value for the `@duration` attribute and *SSN* the value of the `@startNumber` attribute this results in:

$$SAST(k) = AST + PS + (k - SSN + 1) * d$$

1. Within this Period the latest segment available on the client is the segment at the position $k*$ which results in the greatest value for $SAST(k*)$ and at the same time is smaller than *NTP*. For the number based template with *d* the value for the `@duration` attribute and *SSN* the value of the `@startNumber` attribute this results in:

$$k* = \text{floor} ((NTP - (AST + PS) - d)/ d) + SSN$$

2. The address of the latest segment is obtained by using the position information $k*$ and then the segment address can be derived. The segment address depends on the addressing method.

3. Within this Period the greatest segment position k' that can be constructed by this MPD is the one that results in the greatest value for $SAST(k')$ and at the same time is smaller than $FT + MUP$.

$$k' = \text{ceil} (FT + MUP - (AST + PS) - d)/ d) + SSN$$

6.4.2.8 Derivation of MPD Times

If the `@duration` attribute is present and the value divided by the value of `@timescale` is denoted as *d* then the MPD times are derived as:

- $MD(k) = d$
- $MST(k) = (k-1)*d$

6.4.2.9 Addressing Methods

6.4.2.9.1 Introduction

The addressing method is independent of the usage of the timeline generation. The interpretation of the `@startNumber` depends on the addressing method.

6.4.2.9.2 Playlist-Method

If the Representation contains or inherits one or more **`SegmentList`** elements, providing a set of explicit URL(s) for Media Segments, then the position of the first segment in the segment list is determined by `@startNumber`. The segment list then provides the explicit URLs.

NOTE: This is not properly documented in ISO/IEC 23009-1 and requires a correction.

6.4.2.9.3 Number-Based Template

If the Representation contains or inherits a **SegmentTemplate** element with *$Number$* then the URL of the media segment at position *k* is obtained by replacing the $*Number*$ identifier by (*k*-1) + @startNumber in the **SegmentTemplate**@media string.

6.4.2.10 Scheduling Playout

The client schedules the playout based on the available information in the MPD.

The media presentation time in a Period is determined for each Representation as presentation time value in the media segments minus the value of the @presentationTimeOffset, if present, for each Representation.

Each segment at position *k* has assigned an earliest media presentation time *EPT*(*k*).

By offering an MPD it is guaranteed that:

1. each segment in this Period is available prior to its earliest presentation time and its duration, i.e. for all *k*,
2. *SAST*(*k*) <= *EPT*(*k*) + (*AST* + *PS*) + *MD*(*k*),
3. If each segment with segment number *k* is delivered starting at *SAST*(*k*) 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 *PT* + (*AST* + *PS*) + *MBT* + *MD*(*k*),
4. A recommended playout-time *MPTS* (*PT*) for a presentation time when operating in sync with other clients is *MPTS*(*PT*) = (*AST* + *PS*) + *PT* + *SPD*,
5. Each segment in this Period is available at least until *SAST*(*k*) + *TSB* + *MD*(*k*).

Using this information, the client can now start scheduling playout taking into account the information in the MPD as well the download speed.

A suitable playout time is *POT*(*PT*) = *MPTS*(*PT*), if the attribute @suggestedPresentationDelay is present. If not, then a suitable playout time takes into account the first, second and fourth constraints, i.e. the segment availability times at the server as well as the bitrate variation of the media stream.

6.4.2.11 Validity of MPD

The MPD can be used to construct and request segments until media time *FT* + *MUP*. The greatest segment position *k'* that can be constructed by this MPD is the one that results in the greatest value for *SAST*(*k'*) and at the same time is smaller than *FT* + *MUP*. Note that the latest segment may be shorter in duration than the other ones.

6.4.3 Mapping Use Cases to Live Operation

6.4.3.1 Use Case A

6.4.3.1.1 Description

Anna accesses the service in the bus with her mobile DASH-enabled device, and the service is available immediately.

6.4.3.1.2 MPD example

Below is a snippet of an MPD example.

```
<MPD availabilityStartTime="2011-12-25T12:30:00" minimumUpdatePeriod="30s"
timeShiftBufferDepth="60s"
minBufferTime="5s"/>
<BaseURL>http://www.example.com/</BaseURL>
<Period start="PT0S"/>
...
</Period>
```

```
<Period start="PT0.10S>
...
   <SegmentTemplate timescale="48000" startNumber="22"
   presentationTimeOffset="2016000" duration="96000"
   initialization="audio/fr/init.mp4a" media="audio/fr/$Number$"/>
...
</Period>
```

6.4.3.1.3 Client Procedure

Assume further that the client has fetched the MPD at fetch time *FT*="2011-12-25T12:30:17" and the wall-clock time is *NTP*="2011-12-25T12:30:27" the DASH service to be accessed. The latest segment number is:

$$k^* = \text{floor}\ (\ (NTP\ -\ (\ AST + PS\ )\ -\ d\ )/\ d\ ) + SSN = \text{floor}\ (15/2) + 22 = 29$$

The URL for the latest segment is `http://www.example.com/audio/fr/29.mp4`. The client access the segment and may start playout with the media time *PT* = (29-22+1)*96000/48000 = 16 at time *MPTS*(*PT*) = (*AST* + *PS*) + *PT* + *MBT* = `"2011-12-25T12:30:31"`, i.e. in 5 seconds. The client may also download earlier segments and may start earlier with the playout process, for example with segment 27.

6.4.3.2 Use Case B

6.4.3.2.1 Description

Continuing Use Case A, across from her sits Paul, who watches the event on his DASH-enabled laptop. A goal is scored and both, despite watching on different screens, celebrate this event at the same time.

6.4.3.2.2 MPD example

Below is a snippet of an MPD example with the suggested presentation delay added.

```
<MPD availabilityStartTime="2011-12-25T12:30:00" minimumUpdatePeriod="30s"
suggestedPresentationDelay="10s"
timeShiftBufferDepth="60s"
minBufferTime="5s"/>
<BaseURL>http://www.example.com/</BaseURL>
<Period start="PT0S"/>
...
</Period>
<Period start="PT0.10S>
...
   <SegmentTemplate timescale="48000" startNumber="22"
   presentationTimeOffset="2016000" duration="96000"
   initialization="audio/fr/init.mp4a" media="audio/fr/$Number$"/>
...
</Period>
```

6.4.3.2.3 Client Procedure

The same procedure as in 2.3.1.3 to extract the MPD information is carried out. For synchronized playout, the client accesses the segment and may start playout with the media time *PT* = (29-22+1)*96000/48000 = 16 at time *MPTS*(*PT*) = (*AST* + *PS*) + *PT* + *SPD* = `"2011-12-25T12:30:36"`, i.e. in 10 seconds. If both clients adhere to the SPD value, synchronized playout can be achieved.

6.4.3.3 Use Case C

6.4.3.3.1 Description

Continuing Use Case B, Other people that follow the game on a 3GPP Rel-6 PSS terminal observe the goal within a similar time.

6.4.3.3.2 MPD example

The same as in 2.3.2.2.

6.4.3.3.3 Client Procedure

The same procedure as in 2.3.1.3 and 2.3.2.3 to extract the MPD information is carried out. However, instead of downloading and playing only segment 29, the client may already download segment 24 or 25 and start playout earlier. While starting playout, the client may gradually fill the buffer with segments up to the segment availability start time.

6.4.3.4 Use Case D

6.4.3.4.1 Description

Continuing Use Case C, Another goal is scored. Paul tells Anna that the first goal in the game was even more exciting and Anna uses the offering that she can view the event 30 minutes back in time on her DASH-enabled device. After having seen the goal she goes back to the live event.

6.4.3.4.2 MPD example

Below is a snippet of an MPD example with the minimum time shift buffer depth of 1 hour is added.

```
<MPD availabilityStartTime="2011-12-25T12:30:00" minimumUpdatePeriod="30s"
suggestedPresentationDelay="10s"
timeShiftBufferDepth="3600s"
minBufferTime="5s"/>
<BaseURL>http://www.example.com/</BaseURL>
<Period start="PT0S"/>
...
</Period>
<Period start="PT0.10S>
...
  <SegmentTemplate timescale="48000" startNumber="22"
  presentationTimeOffset="2016000" duration="96000"
  initialization="audio/fr/init.mp4a" media="audio/fr/$Number$"/>
...
</Period>
```

6.4.3.4.3 Client Procedure

The time has moved forward to at *NTP*="2011-12-25T13:32:57". The operation is based on an MPD that was fetched at time FT="2011-12-25T13:32:32". The client is downloading segment with segment number 1959. The event of the goal happened 30 minutes ago. With the above MPD, the segments are available far into the time-shift buffer of one hour. The client computes the segment with has presentation time roughly 30 minutes back and understands that this 1 059 and starts fetching this to playout the presentation time 30 minutes ago. After watching this for 2 minutes, the user wants to move forward into the future again. Based on an updated MPD (necessary as the live edge is no longer presented in the MPD above, the client can then compute the latest segment at the live edge and perform the same operations as in cases 2.3.1.3, 2.3.2.3 and 2.3.3.3.

6.4.3.5 Use Case E

6.4.3.5.1 Description

Continuing Use Case D, the football match gets into overtime, the star player of CF Anolacrab, Lenoil Issem, is brought into the game by the coach of the year, Aloidraug, hits twice the post, but can not score. Due to the extraordinary tension in the match, more and more users join such that the service provider requires migrating the service to the redundant infrastructure without interrupting the service to the users.

6.4.3.5.2 MPD example

Below is a snippet of an MPD example with a new server location added.

```
<MPD availabilityStartTime="2011-12-25T12:30:00" minimumUpdatePeriod="30s"
suggestedPresentationDelay="10s"
timeShiftBufferDepth="3600s"
minBufferTime="5s"/>
<BaseURL>http://www.example.com/</BaseURL>
<BaseURL>http://www.example-massive-scalable.com/</BaseURL>
<Period start="PT0S"/>
...
</Period>
<Period start="PT0.10S>
...
  <SegmentTemplate timescale="48000" startNumber="22"
  presentationTimeOffset="2016000" duration="96000"
  initialization="audio/fr/init.mp4a" media="audio/fr/$Number$"/>
...
</Period>
```

6.4.3.5.3 Client Procedure

Clients updating the MPD may observe that a new server location is available. Based on poorer download experience with the original server location, the clients are expected to probe the new server location and when observing better download experience, they are expected to use this new server location and move away from the old one.

6.4.3.6 Use Case F

6.4.3.6.1 Description

Continuing Use Case E, finally penalty shooting is necessary. The live event is interrupted by a short break during which advertisement is added. The exact timing of the ad breaks is unknown due to the extra time of the extension and the start of the penalty shooting is delayed.

6.4.3.6.2 MPD example

Below is a snippet of an MPD example with a new Period added for ad insertion and then the live program is continued.

```
<MPD availabilityStartTime="2011-12-25T12:30:00" minimumUpdatePeriod="30s"
suggestedPresentationDelay="10s"
timeShiftBufferDepth="3600s"
minBufferTime="5s"/>
<BaseURL>http://www.example.com/</BaseURL>
<BaseURL>http://www.example-massive-scalable.com/</BaseURL>
<Period start="PT0S"/>
...
</Period>
<Period start="PT0.10S>
...
  <SegmentTemplate timescale="48000" startNumber="22"
  presentationTimeOffset="2016000" duration="96000"
  initialization="audio/fr/init.mp4a" media="audio/fr/$Number$"/>
...
</Period>
<Period start="PT1H.45M.15S">
...
  <SegmentTemplate timescale="44100" duration="44100"
  initialization="http://adserver.com/audio/fr/init.mp4a"
  media="http://adserver.com/audio/fr/audio/fr/$Number$"/>
...
</Period>
<Period start="PT1H.46M.10S">
...
  <SegmentTemplate timescale="48000" startNumber="189030"
  presentationTimeOffset="18146784000" duration="96000"
```

```
  initialization="audio/fr/init.mp4a" media="audio/fr/$Number$"/>
...
</Period>
```

6.4.3.6.3 Client Procedure

With another update the client obtains an MPD with a new Period that points to an ad server. The advertisement is scheduled for 60 seconds and after this it returns to the main program.

6.4.4 Gap Analysis

Despite the improved timing control and the advantages of the DASH solution, the following aspects are crucial and may need more considerations, especially when operating on a low-latency live service:

1) The server and the client need to have accurate UTC timing. There is no requirement how to implement this, but it still requires implementation of a globally accurate timing standard on both ends. NTP is considered as one option, but the NTP protocol may not be accessible to clients that rely on the HTTP protocol only. Simpler methods for client-server synchronization may be desired.

2) Server overload as all clients may access the segment at the same time as the segment availability time is exposed explicitly. This problem needs further investigation.

3) A more accurate resolution of time is necessary (seconds may be to coarse to operate on at the live edge).

4) Drift of the video source compared to UTC.

5) Leap seconds.

6.4.5 Working Assumptions

As MPEG has ongoing work and core experiments on improved live services, it is proposed to complete the work in MPEG, but potentially send 3GPP specific requirements to MPEG in order to ensure that these aspects are taken into account.

6.5 Use Cases for Content Protection

6.5.1 Use Case A – Efficient Caching for Multiple DRM Systems

6.5.1.1 Description

Service provider "WebMedia" acquires various video contents from movie studios and TV broadcasters for delivery over DASH to its registered subscribers. Some of these programs, such as newly-released movies and hit-series TV episodes, are premium content for which the content providers assign usage rights or licenses through WebMedia for access by its subscribers. WebMedia employs three popular DRM systems, Playball, Fairgame and Grapewine to provide the requisite content protection. Furthermore, a common encryption mechanism "FooCrypt", featuring the use of a single encryption algorithm (but changeable across programs) and common encryption parameter values for any given content item, is implemented by all three of these DRM systems. This enables distribution and caching of the same segments despite different DRM systems are used. WebMedia specifies the use of FooCrypt for content encryption, and ensures that all WebMedia-capable end user devices support FooCrypt in conjunction with one of Playball, Fairgame or Grapewine DRM Agents. Upon DASH-based consumption of such encrypted content, the DRM Agent grants the security key for content decryption and rendering in accordance to the DRM rights or license associated with that content item. WebMedia passes content usage information and any payments to its content providers as dictated by business agreements.

6.5.1.2 Actors' issues

- Content Provider – Wants to ensure controlled (and possibly paid) access premium content delivered by its designated service provider, in accordance with assigned usage rights.

- Service Provider – Wants the ability to honour business contract with content provider for rights-based content access. Desires the simplicity and cost effectiveness of providing a single encrypted version of protected content that is compatible with multiple DRM systems to be supported.

- User device vendors – want to implement content protection which meet service provider requirements with minimum complexity.

- End user – wants seamless user experience in viewing HTTP streamed content, and be fully agnostic of any underlying DRM and decryption mechanisms.

6.5.1.3 Analysis in the Context of Rel-10 TS 26.247 [2]

TS 26.247 is agnostic to the DRM that is used. However, neither in TS 26.247 nor in 3GPP file format TS 26.244 [5] is there explicit support for common encryption.

6.5.2 Use Case B – Signalling of Rights/License Acquisition Information in MPD

6.5.2.1 Description

The MLF (Major League Football) via its designated DASH service provider, wishes to make available live transmission of the 2012 SuperBall game to its subscribers. The game event is targeted to Snoozzz.com enabled clients, all of which support the OpraDRM content protection standard, on devices equipped with the OpraDRM Agent. The OpraDRM related protection information, i.e. the URL to the rights issuer operated by Snoozzz.com for acquiring the associated rights and keys, is nominally contained in the DASH initialization segment. The service provider expects a large audience turnout for reception of the game transmission, but unfortunately, its rights issuer servers have limited TPS (Transactions Per Second) capability to handle the expected high traffic load. Snoozzz.com is concerned of potentially inferior user experience in excessive start-up delay of playout. Therefore, it desires an alternative means for delivering rights/licenses acquisition information, or the rights/license itself, to user devices prior to the game. This would allow for the rights/licenses and corresponding key material to be already fetched, cached and made ready for use by the device when the SuperBall game begins.

6.5.2.2 Actors' issues:

- Content Provider – Wants to ensure its designated service provider can deliver protected live events with low start-up delay.

- Service Provider – Wants to minimize or better control start-up delay given limited processing capability of its servers to handle rights/licenses acquisition traffic.

- User device vendors – Want to implement content protection which meet service provider requirements with minimum complexity and high performance.

- End user – Wants seamless user experience in viewing live HTTP streamed content, and be fully agnostic of any underlying DRM and decryption mechanisms.

6.5.2.3 Analysis in the Context of Rel-10 TS 26.247 [2]

TS 26.247 is agnostic to the DRM that is used. However, neither in TS 26.247 nor in 3GPP file format TS 26.244 [5] is there explicit support for common encryption.

In addition TS 26.247 allows the use of Early Available Periods as defined in clause 8.4.2 of TS 26.247. An MPD may contain a Period that can be provided without a start time of the period. This means that the Period is expected to occur and any resources indicated in the Period structure are available, but the actual start time of the Period is only determined with updates of the Period. In this case, Content Protection relevant scheme specific information, such as DRM server URL etc., may be added to the content protection element.

In this case the structure of the Period is announced without a start time (see example below in bold).

```
<Period  start="PT00H"  id="1">
  .....
```

```
</Period>
<Period  id="10"
   <Representation  id="QVGA-LQ"  mimeType="video/3gp; codecs='avc1.42E00C, mp4a.40.2'"
   bandwidth="192000"  width="320"  height="240"...>
      ...
      <SegmentTemplate ...>
        <Initialization="http://www.example.com/rep-QVGA-LQ/seg-init.3gp"/>
           ...
      </SegmentTemplate>
      <ContentProtection ... />
   </Representation>
</Period>
```

At the time when it is know that the first Media Segment is available, the start time may be added.

```
<Period  start="PT00H"  id="1">
  .....
</Period>
....
<Period  start="PT06H"  id="10"
   <Representation  id="QVGA-LQ"  mimeType="video/3gp; codecs='avc1.42E00C, mp4a.40.2'"
   bandwidth="192000"  width="320"  height="240"...>
      ...
      <SegmentTemplate...>
        <Initialization="http://www.example.com/rep-QVGA-LQ/seg-init.3gp"/>
           ...
      </SegmentTemplate>
      <ContentProtection ... />
   </Representation>
</Period>
```

6.5.3 Use Case C – Time-Varying Decryption Keys

6.5.3.1 Description

A service provider employs broadcast delivery of live TV services to its users. For protected content, it needs the ability to dynamically change encryption keys, as well as introduce new keys, over the duration of a program transmission. One reason is to ensure greater security for premium content delivery. The service provider also requires the ability to provide overlaid rendering onto the main program, a prerecorded program segment that is protected with a different key; it wants to be able to make such combined presentation decision just before content delivery time.

6.5.3.2 Actors' issues:

- Service Provider – for protected live programs, it wants:

  a) The capability to dynamically vary encryption keys and the associated rights/licenses for greater security;

  b) The ability to carry time-varying encryption keys inband with the live program delivery;

  c) The usability of key rotation with inband key delivery mechanism by both content protection (DRM) and service protection (CAS) technologies;

  d) The capability for key rotation with inband key delivery mechanisms to be useable by multiple concurrent schemes within a given protection technology type (DRM or CAS) in the course of common encryption of the content (see Use Case 3.5.1);

  e) The capability for key rotation with inband key delivery mechanisms to be useable by multiple concurrent schemes within multiple protection technology type (DRM and CAS) in the course of common encryption of the content (see Use Case 3.5.1);

f) Ability for "just in time" decision on combining the presentation of prerecorded content, protected by different keys, with the main program.

- User device vendors – Wants to be able to acquire any changed or new licenses/keys in a timely manner to avoid potential presentation problems.
- End user – Wants seamless user experience in viewing live HTTP streamed content, and be fully agnostic of any underlying DRM and decryption mechanisms.

6.5.3.3 Analysis in the Context of Rel-10 TS 26.247 [2]

There is no explicit support for key rotation in TS26.247 and TS26.244.

6.6 Fast Media Start-up

6.6.1 Description

Alice clicks through in a browser session to start consuming DASH. She is delivered an initial download of data and this allows her user agent to begin fetching media segments. In due time, Alice's user agent downloads all of the MPD data and Alice is enabled to use all features enabled by the MPD.

6.6.2 Analysis in the Context of Rel-10 TS 26.247 [2]

Media startup is influenced among others by:

- the size of the initial MPD
- the amount of requests before the first media is downloaded

Different means exist in TS 26.247 to keep the initial MPD of a service compact and minimize the amount of requests necessary to start playout. Among others, the following tools may be used:

- For On-Demand cases, Self-Initializing Media Segments may be used. In this case the MPD is compact and the download of the Segment Index allows for proper scheduling of subsegment requests. The segment index itself may be hierarchically, so downloading of the initial portion is sufficient to start media download. Typically two or at most three requests are necessary to obtain the first media.
- Segment templates: By using segment templates, the MPD size is small and independent of the segment size. The usage of segments to generate the appropriate HTTP-URLs is discussed in detail in TS 26.247 as well as in clause 6.4 of the present document. Only the MPD request and the request for the Initialization Segment and the first media segment is necessary.
- Xlink: In order to keep the initial MPD small, Segments or Periods with a later media presentation time may be added to remote elements by using xlink. Xlink may be used in combination with Periods, Adaptation Sets and Segment Lists. It also allows that in case of dynamic live services, "older" content is moved to remote elements.

Generally, TS 26.247 provides sufficient means to support fast media start-up.

6.7 Advanced Trick Modes

6.7.1 Description

Lisa consumes the latest series of the show "X" in SD coded at a bitrate of 2 MBit/s that is distributed to an DASH-ready Client set. Her client is equipped with a H.264/AVC video decoder that is capable to handle H.264/AVC High Profile level 3.0. All of a sudden the phone rings and she pauses the service.

After the phone call, she resumes the service, but realizes that she wants to go backward in time, as she cannot remember the start of the scene. She seeks backward to the last scene changes and resumes the service from there.

After a while she needs to leave for her Football practice and she decides to continue to watch the movie from her smart phone with H.264/AVC CBP level 1.3. She enters the service and does a fast-forward 64-times of the original speed to the position where he stopped on the TV set. Once she is close, she reduces the search speed gradually down until she recognizes the position. Once the position found, she resumes the service at normal playback speed.

She meets her friend Max and pauses the service. She remembers the great scene in the show wants to share the scene with her friend. She seeks backward in-time and finally gets to the scene and shares it with her friend.

6.7.2 Analysis in the Context of MPEG-DASH and Rel-10 TS 26.247 [2]

6.7.2.1 Overview

TS 26.247 and MPEG-DASH provide different means to support trick modes. The most relevant ones are:

- dedicated trick mode Representations with frequent IDR frames or IDR-frame only. The latter can be provided by using the @codingDependency flag may be set to false for video representations to indicate that a Representation is IDR-frames only.
- Sub-Representations may be used to signal temporal subsequences in Representations. This is discussed in more detail in the following aligned with ISO/IEC 23009-3 [10].

6.7.2.2 Sub-Representations

6.7.2.3 MPD authoring

The MPD file for this use case should be prepared in accordance to general constraints for ISO Base media file format On Demand profile, specified in clauses 8.1, 8.3.1, and 8.3.2 of ISO/IEC 23009-1.

In addition, the following conditions should be satisfied:

- The **`SubRepresentation`** element should be contained at the Representation.
- The **`SubRepresentation`**`@level` should be present
- The SubRepresentation@dependencyLevel should be provided to indicate the dependencies among SubRepresentations.

6.7.2.4 Segment generation

Media segments for this use case should be prepared in accordance to general constraints for ISO Base media file format On Demand profile, specified in clauses 8.1, 8.3.1, and 8.3.3 of ISO/IEC 23009-1.

In addition the following conditions as taken from ISO/IEC 23009-3 [10] should be satisfied:

1) The Initialization Segment should contain the Level Assignment ('`leva`') box with the same levels as provided in SubRepresentation@level.
2) All Media Segments should conform to Sub-Indexed Media Segments as defined in ISO/IEC 23009-1, clause 6.3.4.4 and therefore should include 'sims' as compatible brand in the 'styp' box.
3) If the SubRepresentations defined by the levels in the 'leva' box have assignment type equal to 0 or 1 for a track, the Media Segments should contain the 'sbgp' (sample to group) box in the corresponding 'traf' and the 'sgpd', in case the corresponding 'sgpd' is not included in the 'stbl' in the Initialization Segment.
4) If the SubRepresentations defined by the levels in the 'leva' box have assignment type equal to 2, a single movie fragment is contained in the Subsegment and each of the level contains data of a single track of the tracks indicated in the 'trak' boxes in the 'moov' box.
5) If the SubRepresentations defined by the levels in the 'leva' box have assignment type equal to 3, more than one movie fragment is contained in the Subsegment and each of the level contains data of a movie fragment.

6) If the SubRepresentations defined by the levels in the 'leva' box have assignment type equal to 4 for a track, the Media Segments will contain the 'sbgp' (sample to group) box in the corresponding 'traf' and the 'sgpd', in case the corresponding 'sgpd' is not included in the 'stbl' in the Initialization Segment. Furthermore, the Media Segment will contain a 'udta' box with a 'strk' box. The 'stsg' box in the 'strd' of the 'strk' contains information to identify the sample grouping information in the 'sbgp' box.

7) Data from lower levels should not depend on data in higher levels.

There are four possibilities of generating the segments in order to allow for trick modes, i.e. 3), 4), 5) and 6).

When (3) is considered and assuming the trick mode is performed only for the video media component, there is a single track with sample groups for describing the different level (e.g. the 'tele' sample group). In this case, as well as if level definition is based on subtracks (6), it is necessary to arrange all the samples belonging to each of the leves at the beginning of the Subsegment. As an example Figure 1 shows how this can be done for fast forwarding using the sample grouping 'tele' for a video stream encoded with AVC with GOP size 4 using bi-predictive hierarchical pictures, i.e. with Structure $IB_1B_0B_1P$… in presentation order.

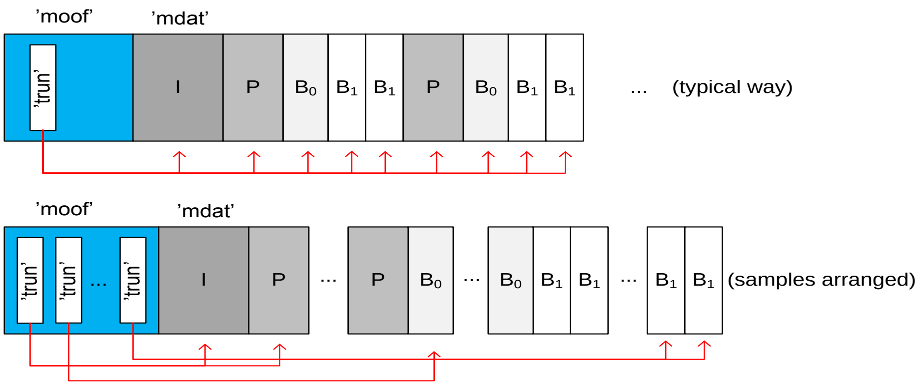


Figure 1: Movie fragment format for arranged samples for easing fast forward with 'ssix' box

Since it is necessary to group the samples in temporal order it is necessary to split the 'trun' in multiple 'trun'-s. For such an arrangement of the samples in an order different from the decoding order, it is necessary to add multiple 'trun' boxes in order to still provide the correct decoding time. Whenever two contiguous samples in the 'mdat' do not have decoding time following each other, a new 'trun' is needed. Then the different levels could be described e.g. as level 0 containing I and P frames, level 1 containing B_0 frames, level 2 B_1 frames and so forth.

If (4) of (5) are considered, i.e. 'leva' box with assignment type 2 or 3, the usage of extractors would be needed for preparing the content for allowing fast forward trick mode.

In Figure 2 an example of the format segment for supporting SubRepresentations for an assignment type other than 3 is shown. In this case the 'moof' box contains all the tracks and a Subsegment should consist of a single movie fragment. The yellow arrows and the dashed lines correspond to the position until which the data belonging to the first level is present, which is indicated in the 'ssix'. In this example only two levels are considered and the second expands until the end of the Subsegment (in this case movie fragment).

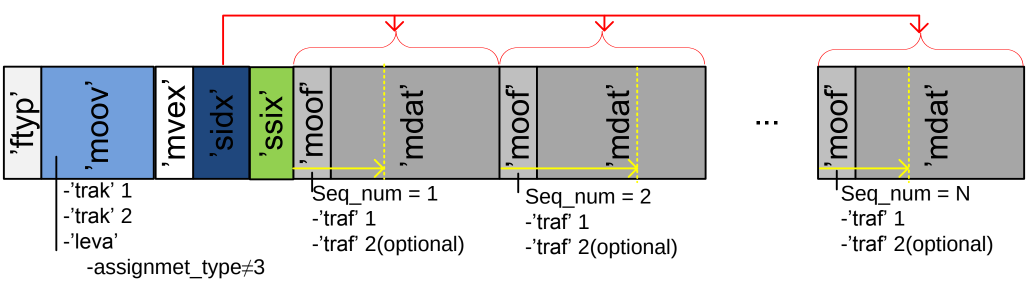


Figure 2: Example of usage of 'ssix' box for Sub-Representations for self-initialising segment with assignment type in 'leva' box other than 3

In Figure 3, an example of a Segment format for assignment type equal to three is shown. As it can be seen in this figure, each of the movie fragments contained within a Subsegment contains data from different tracks. In this case the byte ranges provided by the 'ssix' box should contain whole numbers of movie fragments.

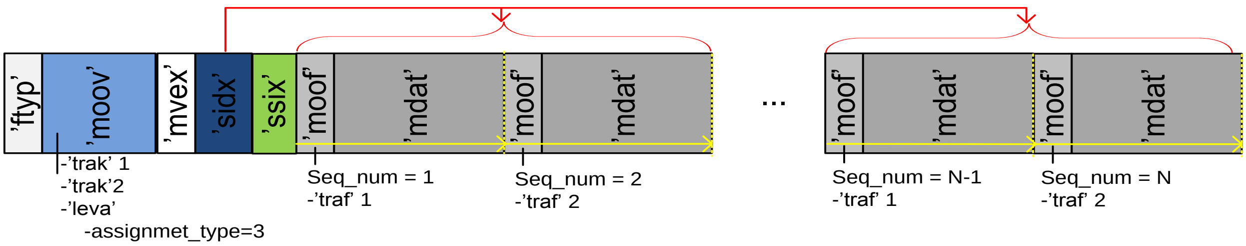


Figure 3: Example of usage of 'ssix' box for Sub-Representations for self-initialising segment with assignments type in 'leva' box equal to 3

In general, when the tracks are used to perform sub-representation extraction (e.g. for trick modes), if several tracks describe one media component, extractors are used and only one track of those is played accessing to the samples in other tracks by reference by the extractors. The usage of extractors should be done very carefully if combined with sub-representations. If extractors are used in higher levels pointing to lower levels there would not be any problem at the client side but if extractors are used in the segments, and these are stored in the lower level pointing to data in higher levels, DASH Clients may try to access non existing data. Therefore, special care should be taken to use extractors in lower levels if assignment type other than 3 is used and the padding_flag in the 'leva' box is set.

6.7.2.5 Summary

Trick modes are well supported in TS 26.247.

6.8 Content and Device Interoperability

6.8.1 Description

Streaming service provider WebMedia has decided to deploy streaming services based on 3GP-DASH to 3GPP UEs. WebMedia does not have any requirements on the codecs, but due to its large library encoding and transcoding of content is expensive. WebMedia wants to distribute the content efficiently through a CDN and also efficiently over 3GPP access networks. At the same time WebMedia wants to avoid to provide specific Representations for 3GPP devices, but wants to distribute the same content to TV sets, Set-Top boxes and other fixed net devices. To fulfil regulatory requirements and other user demands, WebMedia also needs support for subtitles and closed captioning.

For cost and scalability reasons, WebMedia wants to use exactly one Representation to provide a certain quality for audio and video. WebMedia wants to deliver content to three different terminal classes:

- Devices supporting WWVGA video rendering and stereo sound, i.e. 400x240@24fps
- Devices supporting WVGA video rendering and stereo sound, i.e. 800x480@25fps
- Devices supporting HD video rendering and multichannel sound with display capabilities to be defined. For HD video, WebMedia also has requirements on content protection, namely that at least on of the three DRMs are supported: ReadyPlay, WineWide and PlayFair.

WebMedia is also very cautious to provide best user experience when switching Representations within one Adaptation Set are switched for rate adaptation purposes. Therefore, the MPEG DASH functionalities

- segment indexing for On-demand services
- number-based segment templating for their newly introduced live service
- segment alignment and subsegment alignment
- starts with SAP is set to 2

is integrated in the content preparation.

6.9 Advanced Support for Live Services

6.9.1 Description

The use extends the live service use cases in clauses 6.3, 6.4, 6.5 and 6.6.

In certain deployment scenarios, low latency for a live distribution service is essential. One example is the in-venue distribution of an event, such as sports event or a concert. In this case, the delay between the actual live action and the presentation on a mobile device is most appropriate as low as possible in the range of a few seconds at most. Other low latency use cases include betting applications or events with user interaction, etc.

In a deployment scenario, the distribution may happen completely or partially not over unicast, but supported by multicast as for example defined in TS 26.346 [3].

6.10 Consistent QoE/QoS for DASH users

6.10.1 Description

A network operator deploying DASH services or a network operator supporting the delivery of DASH services of a service provider has the ambition to provide consistent quality for users in its network. For this purpose, the content provider wants to provide sufficient QoE to all users that have been granted acquisition to the network and the service. It may also have the ambition to provide certain premium users to maintain a certain service quality when the user plane is congested.

The operator may want to influence its QoS control and radio resource management to actively support such use cases.

The following three cases can happen:

- The operator is able to read the MPD and knows how client will use the MPD in terms of issuing requests for specific Adaptation Sets, Representations, and so on.
- The operator is able to read the MPD but does not know how client will use the MPD.
- The operator does not have access to the MPD.

6.10.2 Proposed Work

To identify the relevant aspects in this area it is proposed to work on the following aspects.

- The details of the following is still FFS:
  Define an end-to-end reference architecture to understand the different components in the application and network and to what extent they influence the rate control. This includes servers, QoS architecture, RRM and schedulers as well as the clients implementation for among others, TCP congestion control, HTTP usage, selection of Adaptation Sets, buffer sizes and rate adaptation.
- The details of the following is still FFS:
  Define a simple reference client architecture that decomposes the different components in the client that influence performance of the work.
- Define some indicative performance metrics for streaming experience to discuss different options and solutions.
- Identify the performance of a DASH client when operating with and without specific treatment in the network, possibly with client support.
- Communicate with SA1 on the specific DASH-specific aspects for UPCON and the progress on UPCON.
- Considering potential optimisation on the various components.

The benefit of signalling events of MPD updates will be investigated in the context of this use case.

6.11 DASH as download format

6.11.1 Description

A content provider wants to offer content with multiple languages as well as with possible different video bitrates such that clients can access according to their capabilities and user preferences. The content provider is looking for a format that is supported in 3GPP and it also considers distribution over HTTP/TCP/IP and multicast MBMS.

The content provider is quite disappointed to not find any such format until he reads TS 26.247 [2] and MPEG DASH. Looking at the example for the basic-on-demand profile copied below, the DASH Media Presentation perfectly describes a format the can also be used for download services. The MPD permits to offer DVD-like content as download content. The content provider decides to use the MPEG DASH formats for download delivery over unicast and multicast, but it is required to provide all the necessary signalling to completely support this.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:DASH:schema:MPD:2011"
  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011"
  type="static"
  mediaPresentationDuration="PT3256S"
  minBufferTime="PT1.2S"
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

  <BaseURL>http://cdn1.example.com/</BaseURL>
  <BaseURL>http://cdn2.example.com/</BaseURL>

  <Period>
    <!-- English Audio -->
    <AdaptationSet mimeType="audio/mp4" codecs="mp4a.0x40" lang="en" subsegmentAlignment="true">
      <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
      <Representation id="1" bandwidth="64000">
        <BaseURL>7657412348.mp4</BaseURL>
      </Representation>
      <Representation id="2" bandwidth="32000">
        <BaseURL>3463646346.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
    <!-- French Audio -->
    <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40.2" lang="fr" subsegmentAlignment="true">
      <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
      <Role schemeIdUri="urn:mpeg:dash:role" value="dub"/>
      <Representation id="3" bandwidth="64000">
        <BaseURL>3463275477.mp4</BaseURL>
      </Representation>
      <Representation id="4" bandwidth="32000">
        <BaseURL>5685763463.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
    <!-- Timed text -->
    <AdaptationSet mimeType="text/mp4" codecs="3gp.text" lang="fr" lang="de">
      <Role schemeIdUri="urn:mpeg:dash:role" value="subtitle"/>
      <Representation id="5" bandwidth="256">
        <BaseURL>796735657.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
    <!-- Video -->
    <AdaptationSet mimeType="video/mp4" codecs="avc1.4d0228" subsegmentAlignment="true">
      <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
      <Representation id="6" bandwidth="256000" width="320" height="240">
        <BaseURL>8563456473.mp4</BaseURL>
      </Representation>
      <Representation id="7" bandwidth="512000" width="320" height="240">
        <BaseURL>56363634.mp4</BaseURL>
      </Representation>
      <Representation id="8" bandwidth="1024000" width="640" height="480">
        <BaseURL>562465736.mp4</BaseURL>
      </Representation>
      <Representation id="9" bandwidth="1384000" width="640" height="480">
        <BaseURL>41325645.mp4</BaseURL>
      </Representation>
      <Representation id="A" bandwidth="1536000" width="1280" height="720">
        <BaseURL>89045625.mp4</BaseURL>
      </Representation>
      <Representation id="B" bandwidth="2048000" width="1280" height="720">
        <BaseURL>23536745734.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

6.12 Use Case: Use case description for Efficiency of HTTP-caching infrastructure on DASH

6.12.1 Description

As illustrated in Figure 4, heterogeneous UEs with varying capabilities in terms of processing power, rendering and display techniques are connected to a 3GPP network. A DASH-based service that offers stored (VoD) and live video content needs to address this device heterogeneity by offering representations that match the capabilities of the UEs, e.g. 2D or 3D video at various resolutions, e.g. as low as VGA up to 1 080p. The nature of wireless connectivity leads to varying channel conditions for UEs that affect available bandwidth and adds further need for adaptivity in terms of multiple representations per content with different bitrates.

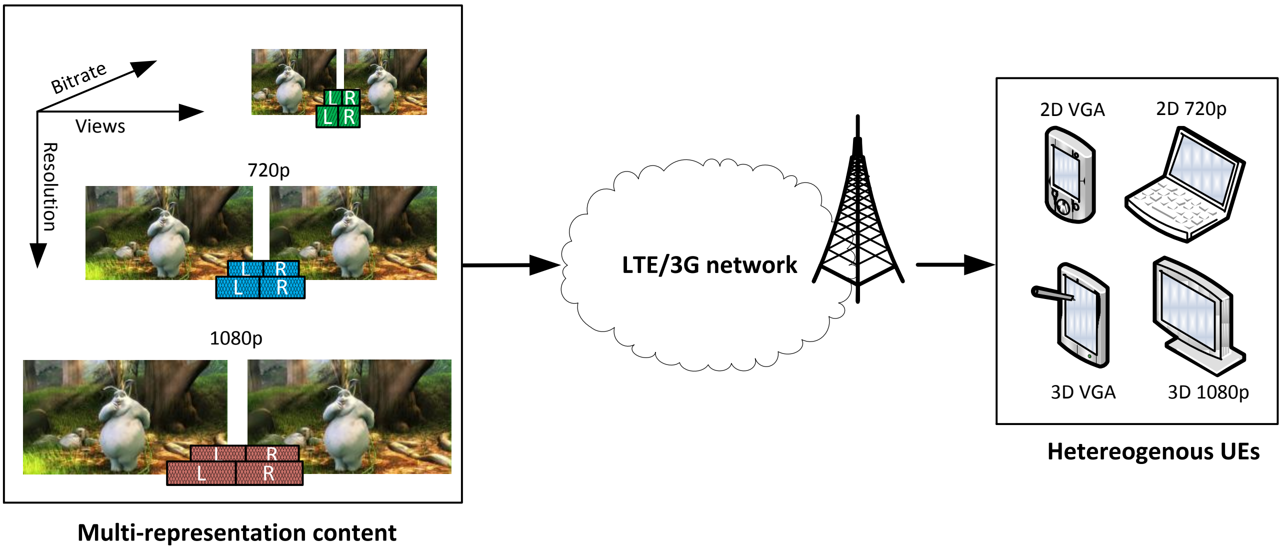


Figure 4: Multi-representation content and UE heterogeneity ((c) copyright 2008, Blender Foundation / www.bigbuckbunny.org)

A DASH-based service is used to deliver multi-representation video content over a 3GPP network to heterogeneous UEs as illustrated in Figure 5. HTTP caching infrastructure is available to be used in the core network. The request characteristics cover different scenarios, e.g. network congestion on the access link or congestion free transmission.

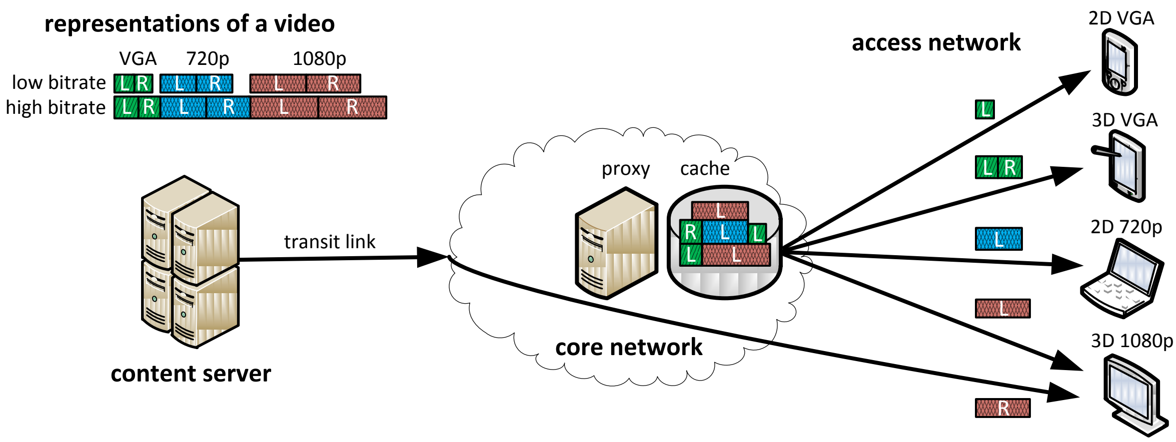


Figure 5: Illustration of the infrastructure for a DASH-based service

6.12.2 Analysis against TS26.247

The use is fully supported since Rel-11 of TS26.247. The example below shows a service offering that enables the use cases as documented in Figure 5.

```xml
<?xml version="1.0"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT3256S"
  minBufferTime="PT10.00S"
  profiles="urn:mpeg:dash:profile:isoff-main:2011">

  <BaseURL>http://www.example.com/</BaseURL>

  <Period duration="PT1256.00S">
    <SegmentList>
      <Initialization sourceURL="seg-m-init-2.3gp"/>
    </SegmentList>
    <AdaptationSet mimeType="video/3gp">
      <Role schemeIdUri="urn:mpeg:dash:stereoid:2011" value="r0"/>
      <!-- 2D-VGA -->
      <Representation id="1" bandwidth="128000" codecs="avc1.64011E">
        <SegmentList duration="10">
          <SegmentURL media="seg-m1-C2view-vga-201.mp4"/>
          <SegmentURL media="seg-m1-C2view-vga-202.mp4"/>
        </SegmentList>
      </Representation>
      <!-- 2D-720p -->
      <Representation id="2" bandwidth="512000" codecs=" avc1.64011F">
        <SegmentList duration="10">
          <SegmentURL media="seg-m1-C2view-720p-201.mp4"/>
          <SegmentURL media="seg-m1-C2view-720p-202.mp4"/>
        </SegmentList>
      </Representation>
    </AdaptationSet>
    <AdaptationSet mimeType="video/3gp">
      <!-- 3D-VGA -->
      <Role schemeIdUri="urn:mpeg:dash:stereoid:2011" value="l0"/>
      <Representation id="3" dependencyId="1" bandwidth="192000" codecs=" mvc1.76001E">
        <SegmentList duration="10">
          <SegmentURL media="seg-m1-C1view-vga-201.mp4"/>
          <SegmentURL media="seg-m1-C1view-vga-202.mp4"/>
        </SegmentList>
      </Representation>
      <!-- 3D-VGA -->
      <Role schemeIdUri="urn:mpeg:dash:stereoid:2011" value="l0"/>
      <Representation id="4" dependencyId="2" bandwidth="768000" codecs=" mvc1.76001F">
        <SegmentList duration="10">
          <SegmentURL media="seg-m1-C1view-1080p-201.mp4"/>
          <SegmentURL media="seg-m1-C1view-1080p-202.mp4"/>
        </SegmentList>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```


6.13 Use Case: Multiple Spectator Views offered with DASH

6.13.1 Description

Much of the discussion on 3GPP-DASH has focused on use cases in which a professional content provider has prepared content such as a movie at different bitrates, resolutions, etc. Another interesting use case is that media is recorded by multiple spectators of an event in different locations and/or recording orientations on handheld devices and uploaded to a server. Location could be determined via GPS for outdoor events or via numerous other means for indoor events (such as WiFi RF fingerprint, WiFi and cellular fingerprint, Bluetooth beams, etc.) The determination of location and recording orientation is obviously outside the scope of 3GPP-DASH and is not proposed to be studied as part of the study item. Users at the same event can record video with cell phones or tablets and upload or stream the content to a server. The transfer of the media content from the recording device to the server is outside the scope of 3GPP-DASH and is not proposed to be studied as part of the study item.

When this content is downloaded via a DASH server, information about the location and orientation of the recording device at the event might be relevant in choosing which Adaptation Set or media presentation time to consume. For example, if users record content at an event such as a hockey game, the DASH client might switch to an Adaptation Set showing a view closer to the net at the time of a goal. On the other hand, if a fight breaks out at centre ice, the client might switch to an Adaptation Set that corresponds to video that was recorded closer to centre ice or had zoomed in on the players. Note that there may be other ways of tagging the content besides location and orientation (for example a particular hockey player that is wearing a helmet cam may be tagged, etc.). Users could go to a website and select relevant times of interest at a particular event and the corresponding times and locations for these instances could be downloaded to the user's client device. For example, a user might select instances of dunks or blocks in a basketball game or instances where a particular player scored, etc. The user might be provided a checklist where they could check multiple types of instances that they are interested in. By downloading the time and position of these relevant instances, the client or user might determine which Segments or Representations to download based on the corresponding view of the event (the position, orientation, amount of zoom, etc. of the camera). The server might also customize the MPD or content for the user based on their selections. Downloading of the relevant times and positions is just an example application and does not need to be included in 3GPP-DASH.

6.13.2 Gap Analysis

6.13.2.1 File support for timed position/location

[Editor's note:this may describe one potential solution, but further work on this section is pending]

The device should be able to record location and orientation information dynamically as media content is recorded. The device location could be described in terms of latitude, longitude, and altitude as is done in the location information box in 3GPP TS 26.244 [5].

The 'Location Information box" [2] (a static box) is as specified in Table 4 below:

Table 4: The Location Information box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'loci'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
Name	String	Text of place name	
Role	Unsigned int(8)	Non-negative value indicating role of location	
Longitude	Unsigned int(32)	Fixed-point value of the longitude	
Latitude	Unsigned int(32)	Fixed-point value of the latitude	
Altitude	Unsigned int(32)	Fixed-point value of the Altitude	
Astronomical_body	String	Text of astronomical body	
Additional_notes	String	Text of additional location-related information	

where Longitude, Latitude, and Altitude have the following semantics:

Longitude: fixed-point 16.16 number indicating the longitude in degrees. Negative values represent western longitude.

Latitude: fixed-point 16.16 number indicating the latitude in degrees. Negative values represent southern latitude.

Altitude: fixed-point 16.16 number indicating the altitude in meters. The reference altitude, indicated by zero, is set to the sea level.

In addition to location, the device orientation can be described according to the direction the camera is facing and how it is tilted and rotated. This is illustrated in Figure 6 below:

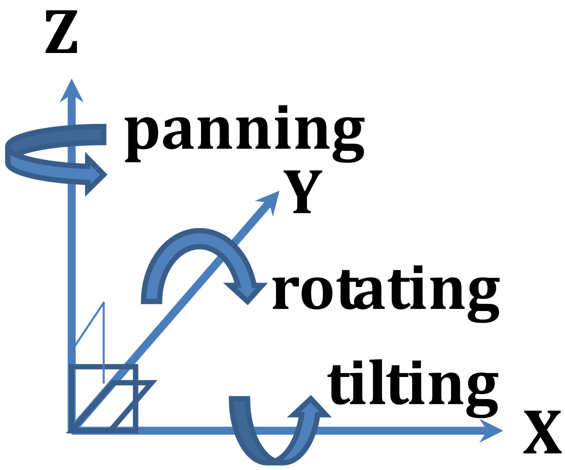


Figure 6: Device orientation

The parameters Pan, Rotation, and Tilt should be defined to describe device orientation just as Longitude, Latitude, and Altitude describe the device's position. In addition to the above parameters, a parameter defining the amount of optical or digital zoom could also be useful as a person farther away with more zoom might have a preferable view to another person who is closer to the event with less zoom.

There is support in the ISO Base Media File Format [3] for a timed metadata track. The SDL code for the sample description box is given as follows:

```
aligned(8) class SampleDescriptionBox (unsigned int(32) handler_type)
   extends FullBox('stsd', 0, 0){
   int i ;
   unsigned int(32) entry_count;
     for (i = 1 ; i <= entry_count ;
                    i++){
              switch
        (handler_type){
         case 'soun': // for audio tracks
            AudioSampleEntry();
            break;
         case 'vide': // for video tracks
            VisualSampleEntry();
            break;
         case 'hint': // Hint track
            HintSampleEntry();
            break;
         case 'meta': // Metadata track
            MetadataSampleEntry();
            break;      }
      }
   }
}
```

MetadataSampleEntry is one of the abstract classes which extends the abstract class SampleEntry. The SDL code for these is given as follows:

```
aligned(8) abstract class SampleEntry (unsigned int(32) format)
extends Box(format){
const unsigned int(8)[6] reserved = 0;
unsigned int(16) data_reference_index;
}

class MetaDataSampleEntry(codingname) extends SampleEntry (codingname) {
}
```

The currently defined classes which extend MetaDataSampleEntry are the following:

```
class XMLMetaDataSampleEntry() extends MetaDataSampleEntry ('metx') {
  string content_encoding; // optional
  string namespace;
  string schema_location; // optional
  BitRateBox (); // optional
}

class TextMetaDataSampleEntry() extends MetaDataSampleEntry ('mett') {
  string content_encoding; // optional
  string mime_format;
  BitRateBox (); // optional
}
```

There is currently no box defined in the ISO base media file format [3] or the 3GPP file format [2] for timed location or orientation information and no XML Schema defined that could be referenced by `XMLMetaDataSampleEntry`. Either a box can be created specifically for device position and orientation that extends MetaDataSampleEntry with a description of the parameters present in a position/orientation sample or an XML schema and namespace can be created with this information and the namespace can be linked to from `XMLMetaDataSampleEntry`.

6.13.2.2 MPD indication of positon/orientation for a Representation or Segment

The Viewpoint Descriptor [4] may be used to express an Adaptation Set that is provided from a specific device..

Using the Adaptation Set with the **`ViewPoint`** descriptor enables the following features:

- a scheme is defined with a `@schemeIdURI` that expresses that the View Point includes the exact position information, e.g ."`urn:3GPP:ns:PSS:DASH:position`".
- the `@value` attribute contains a textual description of the position that is unique and may be offered to the user in order to select the position.
- Multiple Adaptation Sets with the same `@schemeIdURI` and the same `@value` may be offered to express that for example the audio and video are offered by the same position/device.
- an extension namespace is created to express the exact coordinates in the **`ViewPoint`** descriptor. This may be added, but is not essential if the `@value` is sufficiently descriptive. However, the coordinates may be used by the application to position the viewing position.

6.13.2.3 Synchronization of media content from different devices

[Editor's note:this may describe one potential solution, but further work on this section is pending]

The devices should ideally begin recording on Period boundaries. The times to start recording could be published or made known in advance of the event (e.g. on a website). Devices should be synchronized to an accurate wall clock time in some way in order that a change in Segments or Representations with different views can be accomplished without significant gaps in the event timeline. No changes are anticipated to the 3GPP-DASH specification [4] with regards to synchronization.

6.14 Use cases for operator control of video streaming services

6.14.1 Introduction

It is assumed initially that in some cases the video service is treated as coming from an operator directly, i.e. as a PSS service, and in other cases the video service may be treated as regular HTTP traffic.

6.14.2 Description

6.14.2.1 Operator control of video services

A number of subscribers are watching videos in a crowded area (mall on a Saturday afternoon, train station at peak hour while waiting to go to the Alps for the ski season…), saturating the cell capacity.

The operator identifies the congestion and communicate with the UEs to decrease the bitrate for the video to a certain value that would allow the cell to accommodate the load.

6.14.2.2 Operator control using mobile network subscription

As a variant of case described in sub-clause 6.14.2.1, some of the subscribers have registered (in their mobile subscription) to a "multimedia premium" monthly option to be able to enjoy high quality video even in such situations. Other subscribers have elected to only use a "low cost" mobile subscription, allowing them to enjoy good quality in normal time, but conceding that in such situations, their video service would be of a lower quality.

The operator is able to differentiate among these users and communicate different levels of service quality to the different UEs.

6.14.2.3 Operator management of video streaming service over the radio network

As a variant of cases described in sub-clauses 6.14.2.1 and 6.14.2.2, the operator may not want to wait until the cell reaches the congestion level. Instead, it may want to limit the maximum aggregated bandwidth for video in order to leave room for other services, other subscribers, etc.

The operator communicates with the UEs to decrease their service to a lower bitrate (possibly at different levels based on UE subscriptions) even before congestion level is reached.

The operator is able to adjust the preferred bitrate of different UEs based on the power consumption at the base station (e.g. due to their location relative to the radio transmitter).

The operator network interacts with the UEs, based on the loading info of neighbor other access network, and shifts the UEs to other light loaded network (e.g. UMTS, HSPA, etc.) to mitigate the congestion of the current serving network and maintains the video service experience as much as possible.

The operator network offloads the DASH traffic to WLAN network to avoid the congestion of 3GPP network.

6.14.3 Working assumptions

- The network is able to evaluate, where possible, the appropriate bitrate to be used for the DASH users over a 3GPP radio, based on radio conditions and user subscriptions.

- The network is able to, where possible, communicate with the UE to select the appropriate bitrate or range of bitrates for the DASH service currently consumed.

- The network is able to take advantage of some or all general-purpose congestion management and control mechanisms, e.g. as defined in UPCON [reference]. It is expected that SA4 will engage with the working groups developing further congestion control and management mechanisms (e.g. UPCON, ACDC) to ensure that DASH is handled appropriately

6.14.4 Gap Analysis

The benefit of signalling events of MPD updates will be investigated in the context of this use case.

6.15 Use Cases for DASH Operation with Network Proxy Caches

6.15.1 Introduction

It is proposed that the potential benefits of using DASH-aware caches are studied as part of the work done for completing the study item. The improved DASH operation with proxy caches may involve additional signaling between DASH clients and proxy caches. As compatibility with the existing HTTP infrastructure is one of the greatest advantages of DASH, we propose that any additional signaling that is proposed be compatible with HTTP 1.1 and existing HTTP cache implementations in a sense that existing HTTP caches will ignore that additional signaling.

6.15.2 Use Case Decription

6.15.2.1 Use Case 1: Fast Startup

Paul, who lives at Europe, uses DASH-enabled mobile device to view DASH-formatted multimedia clips, whose origin servers are located at outside of Europe. His device accesses to the Internet primarily through mobile (3G and above) networks, and the DASH player in his device uses the HTTP proxy cache provided by his mobile network operator for network connection. In the scenario of on-demand streaming, he frequently seeks to some interesting positions of the playback timeline. During such a seeking to position of playback timeline for an MP and selecting a new MP for presentation, he prefers to fast playback startup by tolerating certain level of low and varying playback quality. In the scenario of live streaming, he is used to perform channel zapping between several live channels. During cannel zapping, he also prefers to fast playback startup by tolerating certain level of low and varying playback quality.

6.15.2.2 Use Case 2: Partial Representation Caching

John uses DASH for media streaming with his DASH-enabled device. Proxy caches are employed to serve DASH clients including John's DASH-enabled device to save bandwidth and reduce delay. John's DASH-enabled device sends HTTP GET segment requests by parsing a specific Media Presentation Description (MPD). Prior to serving John's segment requests, the proxy cache may have served other DASH clients with the same MP where they created HTTP GET segment requests by parsing the same MPD as John's DASH-enabled device. The proxy cache may cache segments which have been sent to other clients for serving future clients requests. As DASH clients request segments, but also switch Representations dynamically, the proxy cache may cache multiple Representations, each of which may be completely or only partially cached. An partially cached Representation is defined as a Representation having segment gaps, i.e. not all segments of the Representation are cached.

John, who lives at Europe, has discovered that his DASH-enabled device suffers from frequent playback quality variation and some playback interruptions, both of which he finds annoying. For streaming quality, John prefers to view streaming in a stable playback quality and fewer or preferably no playback interruptions. In addition, higher presentation media quality is preferable. John has also noticed that such quality variations and playback interruptions typically occur when he views MPs for which origin servers are located at outside of Europe such as Asia.

6.15.2.3 Use Case 3: Network mobility

A service provider deploys the football distribution as a Media Presentation based on DASH. The service provider is collaborating with a mobile operator which deploys CDNs within its distribution network. The mobile operator specifically provides the service through a 3G network as well as to a WiFi network. Each network is supported by a dedicated CDN. In order to avoid overload of the 3G network, only a subset of the Representations are provided in the 3G network. While shopping with his wife, Jari Ragados watches the service in the mall where there is WiFi coverage. After done with shopping, they move to an outdoor cafe without WiFi coverage, but the service is continuously played by the DASH client, just with lower quality.

6.15.2.4 Use Case 4: Mobility and Coverage Extension for MBMS-based service

The son of Jari, Jarison, is watching the game live in the stadium. Jari picks up his son and drives to the stadium and when he gets there, the same service is provided over DASH+MBMS-based broadcast in HD quality. In addition, multiple views are provided close to the stadium, one being close from the seat where Jarison sits. Jari switches to this view which is only provided over unicast while still using the main audio distribution over MBMS. After the game, Jari and Jarison leave the stadium, but continue to watch the interviews from the stadium in the car served through a 3G network.

6.15.3 Solution Analysis: Usage of TS26.247

6.15.3.1 MPD Update

MPD updates are proposed to be used as a mechanism to influence the DASH client's selection of representations. An intermediate entity, e.g. a proxy server running in the network or on the UE itself, generates MPD updates and adds/removes representations that the client should or should not use.

Upon detection of congestion in the core network or on the air interface, the intermediate node, which is keeping track of all ongoing streaming sessions of DASH presentations, will issue a new MPD for each of these presentations. The new MPDs will restrict the set of Representations that are currently available to help ease the congestion situation. However, this measure should be temporary, so that when the congestion situation is alleviated, the old representations should be made available again.

After generating a new MPD, the DASH clients need to be informed about the presence of that new MPD. This may be done in one of the following manners:

In the next segment request by the client, include an MPD update event message to invalidate the current MPD and ask the client to fetch a new MPD.

Reply to all future segment requests from the client with a 404 error message, thus, forcing the clients to fetch a new MPD, after which the requests will be fulfilled again.

For dynamic Presentations only, wait for the next MPD update request and reply with the new MPD.

All of these available options have some (severe) drawbacks.

First of all, in a case of transient congestion, the reaction to the congestion should be as fast as possible, however, all of these approaches require that first a signal is sent to the client and then that a new MPD is fetched before switching takes place.

However, the biggest challenge is to maintain consistency when authoring all the new MPDs, especially, if the original server is also issuing MPD updates (in dynamic Presentations). The complexity will increase with the number of intermediate nodes that are trying to perform the same task on the path. Each node down the path to the UEs will need to integrate and synchronize its MPD updates to all nodes before it. The risk for inconsistencies and DASH client confusion is relatively high.

Thirdly, the MPD describes the property of the content and does not provide any instructions on how the client uses this information for accessing the content. As an example a client may operate on time shift mode

MPD updates are triggered by content changes and are not done for the purpose of operational changes. For example, in case of an On-Demand (type static) or for a templated dynamic session, MPD updates may never happen or may be done infrequently. In addition, a client may rely on an MPD when it for example operates in timeshift buffer. There is some discussion to refetch the MPD in case of HTTP error codes (see Annex A.7 of ISO/IEC 23009-1), but error codes may not be suitable for proper implementation or may result in similar problems as redirection codes discussed in section 6.19.4.2 w/o further specification.

6.15.3.2 Redirections

Redirection is another tool that intermediate nodes may make use of. In this tool, the intermediate node makes use of the HTTP 3xx codes to redirect UE requests to another location.

Redirections may be followed automatically or they may be passed to the user agent for processing. As it is assumed that browser-based DASH implementations will constitute a share of DASH implementations, we need to look at how segments are fetched in such an environment. HTML defines a mechanism for fetching resources in the background. This mechanism is known as AJAX and is defined in [2]. In [2], redirects are to be followed automatically and transparently to the user agent. This is described in section 4.6.7 and is replicated here:

```
4.6.7 Infrastructure for the send() method

The same-origin request event rules are as follows:
If the response has an HTTP status code of 301, 302, 303, 307, or 308
      If the redirect violates infinite loop precautions this is
```

```
      a network error.
      Otherwise, run these steps:
Set the request URL to the URL conveyed by the Location header.
If the source origin and the origin of request URL are same 
origin transparently follow the redirect while observing the same-
origin request event rules.
Otherwise, follow the cross-origin request steps and terminate the
steps for this algorithm.
```

So assuming that only transparent redirects are supported, the only possible redirects would be to media segments of a different representations. In other words, the client would request a segment from one representation but the response is a redirection to a media segment from another representation. This trick might work only if the used media codecs and configuration information are identical, the request does not contain a byte range, and the segments are time aligned.

Even under those very strong restrictions, the DASH client may still get confused because of the different bandwidth of the segments. The client will be downloading the media segments at a faster pace, if it gets segments from a lower bandwidth representation. This might trigger the client to try to switch to an even higher representation, which will cause even more problems.

Now assuming that redirects do not get followed automatically, in which case, the client will receive a redirection message and the entity body, which might be a textual or HTML fragment. Without knowing the content and format of that response, the DASH client will not be able to interpret the contents of the body and might simply send a GET request to the new resource location.

In other words, the redirection approaches with existing status codes and semantics may not work, unless the client understands the semantics of the redirections and knows how to interpret the body of the redirection message and then extracts the information about the Representations it should consume, this approach will not work.

Additional redirection methods may be considered that take into account:

- redirecting not to a specific object, but to a sequence of object or a new BaseURL
- consistent implementation of such redirection methods in order to ensure that the redirection information is passed to the DASH client.

6.15.3.3 Bandwidth Throttling

The two key issues in video delivery are user experience and delivery efficiency/costs. One of the key performance indicators for video streaming is continuous loss-free playout. Rebuffering and packet losses are considered as the most severe degradation in video delivery. DASH addresses these issues by

relying on a reliable transport protocol, namely HTTP/TCP, and

by providing multiple switchable versions of the same content at different bitrates (aka representations). This enables the client to control its buffer states and choose the requested representations appropriate to the available access bandwidth in order to maintain continuous playout.

Adaptation to changing network conditions such as congestions are naturally handled by TCP congestion control and the end-to-end rate adaptation, driven by the DASH client.

When running TCP in a congested network the TCP throughput is reduced due to increased packet delays and losses. The bandwidth reduction is a reaction to reduced TCP throughput reacting to the above effects packet delay and packet loss. The DASH client will observe the reduced TCP throughout and will therefore use its rate adaptation to adjust the requested bandwidth in order to maintain proper throughout.

The well-known TCP throughput upper bound can be used:

$$\text{rate} < (MSS/RTT)*(C/\text{sqrt}(Loss))$$

Typical examples are: MSS=1460 bytes, RTT between 50ms to 1sec, loss rate 1-e6 to 5%.

As an example, suppose the end-to-end delay of between the HTTP web caching servers and the client was 300 ms, and the packet loss rate was 1%. Based on the TCP equation, the average throughput of a single HTTP connection would be approximately 385 Kbps.

The actual performance of TCP is typically worse than predicted by the TCP equation, and gets harder to model when there are constraints on bandwidth and bandwidth varies. However, there is no counter-proof that the above approach does not work properly.

Users may be differentiated by the applying different packet delays and/or packet losses resulting in more or less bandwidth for the user.

Figure 6.19.4.1 shows the operational principles of DASH-based streaming delivery. The DASH server (in general a plain HTTP server or an intermediate proxy) hosts content at different quality/bitrate levels. Due to congestion, load or other reasons, the e2e network bandwidth may be constrained. The DASH client estimates the available bandwidth and adapts the selected quality/bitrate level to the available bandwidth to ensure continues playback.

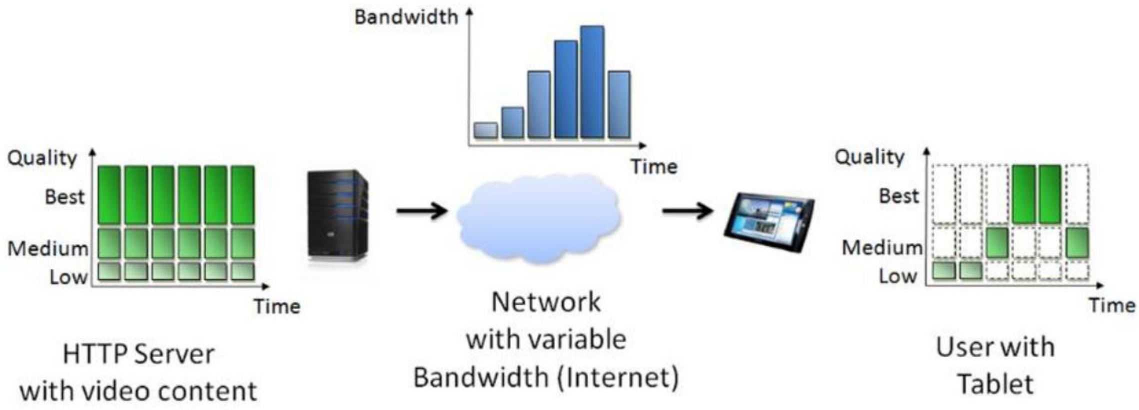


Figure 6.19.4.1 Operational principles of DASH-based streaming

The network "communicates" with the client by applying regular TCP congestion control.

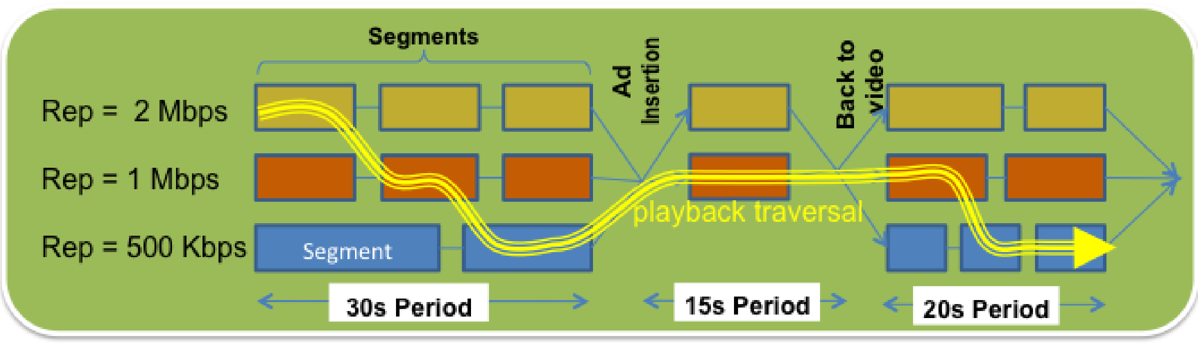


Figure 6.19.4.2 Client Selection process

DASH-based services may be deployed w/o much impact to existing nodes and interfaces in mobile networks. According to Figure 6.19.4.1, the network can modify the bitrate of the DASH session by applying regular TCP congestion control methods (delay and packet-losses) and rely on the DASH client's rate adaptation logic as shown in Figure 3.

Since Release 10, DASH-based services may be QoS-supported. It was agreed on adding in Release 10 the Min-Requested-Bandwidth-UL and Min-Requested-Bandwidth-DL AVPs as part of the media information sent by the Application Function within the Media-Component-Description AVP and allowing the derivation of Authorized Guaranteed Data Rate UL and Authorized Guaranteed Data Rate DL according to the supplied values. TS26.247 provides an informative Annex with an example mapping of DASH/MPD parameters to apply the relevant QoS derivation.

However, for other use cases such as the MBMS use case, this approach fails. In the MBMS use case, the operator or UE might want the DASH client to use a representation that might have higher bandwidth than the one it is currently consuming (e.g. because it is popular, available in the local cache, or because it is delivered over MBMS). Bandwidth throttling does not work in this approach.

6.15.3.4 Control Events

In this approach, the client should be aware about the situation and will cooperate to achieve best user experience by reacting early enough according to the event information.

A set of events may be defined to control the client Representation selection. The following indications may for example be supported:

Preferred Representation: the DASH client is instructed to switch to an alternative representation

Available Bandwidth: the DASH client is informed about the overall bandwidth that is available for the client and requests the client to perform the necessary switching to fit within that bandwidth budget.

Amd.1 of ISO/IEC 23009-1 [13] is currently in preparation in MPEG. In this Amendment the addition of media time related events is considered. Media-time events can be signaled in the MPD or as part of segments. This mechanism may be used to provide information to the DASH client.

If this approach is implemented using the DASH event mechanism, the client may be made aware about the situation and will cooperate to achieve best user experience by reacting early enough.

However, this approach does have the following issues:

The Amendment 1 Events are media-time events. This means that operational real-time instructions to the client are not suitable communicated by these means. For example if the client is operating in the time-shift buffer, the real-time event is missed.

This method is not applicable in the cases where On-Demand profiles are used and short segments are not provided.

The main problem with in-band events is, that the controlling entity needs to have access to the media segments, need to parse and modify them and may even have to provide different segments for each UE or a group of UEs. This may impact the operation including caching and so on.

6.15.3 Gap Analysis and Problem Statement

For the analysis of the use cases the following assumptions are considered

1. The cache operation adheres to some specific rules, e.g. that objects are cached and deleted based on popularity and previous requests.
2. It is assumed that the round trip delay between UE and origin server is significantly longer that the round trip delay between UE and cache server.
3. It is assumed that the content provider produces a large number of representations in this use case.

When the end-to-end system architecture includes a proxy cache, the time for fetching the segment directly from the proxy cache is smaller than the time for the fetching the segment from the origin server. It is especially true in case the network delivery capacity from a remote network element (the origin server or other proxy cache) to the HTTP proxy cache selected by the DASH player is much lower than the network delivery capacity from the HTTP proxy cache selected by the DASH player to the DASH client. Here, the network delivery capacity not only depends on the bandwidths in the delivery network but also depends on the Round Trip Time (RTT) and receiver advertised window as those affect the achievable HTTP/TCP throughput.

The DASH client may suffer from slow playback startup if the requested segment is not cached especially in the case that segment duration is relatively long and either the content has not been prepared for or the client is not capable of subsegment-based rate adaptation.

The proxy cache may cache multiple Representations of an Media Presentation (MP) if other DASH clients requested the same MP through the proxy cache. The content provider may provide a large number of Representations for some popular MP to support hierarchically bandwidth distribution in a large number of DASH clients. The number of Representations and the completeness of Representations of an MP cached by the proxy cache depend on popularity of the MP on the clients connected to the proxy cache and average bandwidth distribution of the clients, dynamic features of the sharable bandwidths of the clients etc.

In case the proxy cache cached large number Representations with fine-scaled bitrate difference, there is a high possibility that the proxy cache does not cache the requested Representation requested by the DASH client but caches the alternative Representations with slight bitrate differences.

Based on this analysis the following gaps are identified:

- The proxy cache has no ability to provide proactive (prior to the request) or reactive (as a response to the request) information to the DASH client about the Representations/Segments that are available on the proxy server and the ones that are available at the origin server.

- The DASH client may typically assume that the bandwidth estimation for one Segment may also be true for all other Segments despite those may be served from different caches, e.g. from the origin server. To prevent this, enhancements may be considered.

- The content provider may have more information than the origin server or the DASH server or the proxy on content that is beneficially cached, e.g. specifically attractive seek points in the content in order to pre-cache such data, etc.

6.16 Services with caching of DASH content at UE functions

6.16.1 Use Case Description

6.16.1.1 Use case 1: HTTP Proxy Cache in UE for transparent reception of DASH content over unicast or broadcast

Mike uses DASH-enabled mobile device with a local HTTP proxy cache to view DASH-formatted multimedia clips. His device accesses to the Internet primarily through 3GPP access networks, and the DASH player in his device uses the HTTP proxy cache in his mobile device to retrieve the DASH content. The HTTP proxy cache, while providing a unique interface to the DASH player, provides 2 interfaces towards the network, one over unicast and one over broadcast MBMS.

6.16.1.2 Use case 2: HTTP Proxy Cache in a home GW (gateway), capable of serving multiple DASH players in the home

In this use case, an HTTP proxy cache in the home GW is capable to serve multiple DASH players in the house, using HTTP interface over possibly over WiFi, or other Ethernet connection in the home.

6.16.1.3 Use case 3: Pre-Caching on UE

An operator offers a TiVo-like video delivery service to its subscribers, enabled by content caching capability in the UEs, which is currently available via significant memory/storage capabilities in mobile devices. This service allows users to select from an operator-specified list of on-demand internet video channels (e.g. popular channels) and/or live TV programs (e.g. evening news, live sports events) with to-be-available future content that they would like to watch from their mobile devices. The content is stored and/or generated at the origin servers based on the DASH file format and an associated media presentation description (MPD) metadata. The users are also provided the option to specify their device capability information and user preferences that help determine the suitable video bitrate, resolution, frame rate etc. parameters to deliver the best possible quality of experience (QoE).

When the relevant videos become available or get posted at the origin servers, the operator network fetches the corresponding MPDs and parts (e.g. first 2 minutes) or all of the DASH media segments and delivers them to the subscribed UE terminals using MBMS and/or unicast-based delivery techniques, preferably during off-traffic (uncongested) hours. The decision on the choice between MBMS/FLUTE or unicast/HTTP methods to efficiently deliver the content to the UEs is made by the operator on a content-by-content basis depending on a various factors including the popularity of the content, number, spatial distribution and subscription profile of the UEs requesting the particular video, bandwidth availability for unicast and broadcast bearers at that time, etc. Upon reception, the UEs cache the videos that were selected by the user, to be presented or played at a later time when requested by the user.

Availability of DASH-formatted video content at the origin servers for such a service delivers a much better QoE to the end user since it allows the operator to fetch and distribute the most suited version of the video based on the device capability profiles, wireless or backhaul link conditions and user preferences, and also it allows the UE to cache the early part of the presentation and adaptively fetch the rest of it with early access to the MPD. The benefit of such a caching functionality in the UEs is mainly two-fold:

1) enhanced QoE, e.g. in the form of reduced startup delay or reduced buffering problems, with accessing DASH-formatted content as due to the use of cache functionality at the UEs and ability to fetch at least the initial part of the content in advance (fetching the whole content may also be possible depending on the nature of the content and operator policy and network conditions),

2) efficient distribution of popular content during off-traffic hours using MBMS/FLUTE-based multicasting and/or HTTP-based unicasting capabilities saving bandwidth for the operator network.

6.16.2 Working assumptions

The following working assumption pertaining to UE HTTP Proxy Cache:

- The UE can logically include an HTTP Proxy Cache and a generic DASH client/player, for use case 1.
- The HTTP Proxy Cache located in a Home GW has the capability to interface towards multiple external DASH players
- The HTTP Proxy Cache has a broadcast receiver (towards the 3GPP network) capable of receiving DASH segments over FLUTE protocol, as currently specified in TS 26.346
- The same HTTP Proxy Cache has a unicast receiver (towards the 3GPP network) capable of receiving DASH segments as HTTP over unicast
- The same HTTP Proxy Cache is used to source the DASH content, with the appropriate representation, towards the requesting DASH player. The DASH content comes from either the unicast or from the broadcast, without knowledge from the DASH player.
- The HTTP Proxy Cache indicates the DASH representations currently cached to the DASH Client in order to increase efficiency of the HTTP Proxy Cache.
- DASH Client in order to increase efficiency of the HTTP Proxy Cache.
- It is possible to provide MPD authoring and client adaptation guidelines for the purpose of caching parts or all of DASH-formatted content at the UEs for later viewing.
- It is possible to provide guidelines for an operator to select (e.g. via access to MPD and/or direct control of the MPD) access method (unicast, MBMS, etc.) for the delivery of DASH-formatted content based on radio conditions, device profiles, user subscription and content popularity.
- It is be possible for an operator to control and enforce policies on the use of available access techniques (unicast, MBMS, etc.) for the delivery of DASH-formatted content in order to optimize utilization of available network resources.

6.16.3 Gap Analysis

There is a need to provide information from the DASH-aware HTTP proxy cache in the operator's network to the DASH client on which representation(s) is or are preferred. This can be achieved for example by having the DASH-aware HTTP proxy cache provide a representation preference indication to the DASH client in an appropriate HTTP header.

NOTE: The actual mechanism is FFS.

For the cases where the DASH-aware HTTP Proxy Cache is not located in the operator's network, there is no necessity to recommend any signalling between the cache and the DASH client.

6.16.4 Recommended Requirements

Derived requirements from the above use cases are listed below.

- The HTTP Proxy Cache can provide on one side an interface towards a DASH client/player, and on the other side, provide 2 interfaces towards the 3GPP access network, one for fetching unicast based DASH content for a DASH server in the network, and another one for fetching DASH content over FLUTE protocol as specified in MBMS TS 26.346.

- A HTTP Proxy Cache can be located in a Home GW, in which case, in addition to the requirement above, it provides or source DASH content towards multiple DASH clients/players.
- DASH clients/players are informed by the HTTP proxy cache, using HTTP signaling, of the representations preferred for playout, depending on
  - whether such representations are already available in the HTTP proxy cache, thus offloading the network;
  - whether such representations are currently obtained by the HTTP proxy cache over broadcast (eMBMS);
  - whether such representations are currently obtained by the HTTP proxy cache over unicast.

6.17 Advertisement insertion in the operator network

6.17.1 Introduction

NOTE: The content to be inserted by the operator is not restricted to advertisements, but can be any regulatory, informative or entertaining content that the operator may wish (or be forced) to present to the subscriber.

6.17.2 Description

6.17.2.1 Advertisement insertion based on user subscription

A subscriber has a "low cost" subscription, allowing a limited bandwidth, restricting his access to high quality multimedia content. As part of his mobile subscription, he has informed his network operator that he is willing to watch some advertisements in exchange of being able to upgrade his multimedia experience (e.g. higher maximum bitrate, or lower degradation of bitrate during congestion situations).

The network operator inserts an advertisement before the DASH videos that this subscriber is playing.

The network operator could also insert an advertisement in the middle of the DASH video (especially for longer content) before the user can resume watching the video.

An other subscriber with the same "low cost" subscription has elected not to have advertisements displayed before watching movies. The network operator in this case will not play an advertisement ahead of (or during) the DASH videos (and will not upgrade the subscriber's experience either).

6.17.2.2 Targeted advertisement insertion

The mobile subscription information of a subscriber indicates that an advertisement will be inserted before (or during) DASH video content is played to the UE.

Based on the location of the UE, a specific advertisement can be played (e.g. related to sales of a nearby store).

Based on the subscriber information, a different advertisement can be played (e.g. based on gender, age, etc.), taking privacy in account.

Based on the duration of the video, a different advertisement can be played (e.g. a longer advertisement could be played for a 2-hour movie compared to a 20-minute TV series).

Based on the known available bandwidth, a different advertisement can be played (e.g. simple content authoring (e.g. timed graphics) or high quality video depending on the available bandwidth).

6.17.2.3 Bandwidth-related advertisement/message

A subscriber watches a DASH video over a 3GPP radio network. The subscriber enters an area where the bandwidth capacity is too low for the video content to be played satisfactorily, possibly leading to an empty buffer for an unknown duration.

The network operator inserts a low-bandwidth DASH content (not necessarily an advertisement) until the radio network conditions are sufficient for the original video to resume.

6.17.3 Working assumptions

- The operator's network is deployed in a way that allows the operator to insert DASH content (advertisement or other) before or during a service using DASH.
- The decision for the network operator to insert additional content and the selection of the content to insert can be based on subscription information, bandwidth information, UE location, and video information.

6.18 Handling special content

6.18.1 Description

6.18.1.1 Enabling users to skip some repetitive/low importance content

Lucy wants to watch a number of episodes of a sitcom on her UE using DASH. A TV series usually contains the same introduction and/or conclusion for all the episodes. As Lucy wants to see a number of episodes in a row, she would like her UE to automatically skip over the repetitive content.

The content provider marks the repetitive content in such a way that the UE can skip over it automatically.

6.18.1.2 Preventing users from skipping special content

Herbert is watching a recent movie. As required by law, the content provider inserts some announcements that have to be seen by the viewer before playing the movie.

Additionally, the content provider wants the viewer to be aware of the illegality of some common practices, such as piracy, and wants the user to watch (and not skip over) a short clip explaining what can be done with the video.

Finally, the content provider is providing the video at a promotional rate in exchange of the viewer watching some advertisement. The content provider wants to make sure the viewer does not skip over the advertisement.

For these three video clips, the content provider indicates that the UE is not allowed to skip over this content.

6.18.1.3 Network/Content provider control of special content

In addition to the use case described in clause 6.18.1.2, the network operator and/or the content provider want to make sure that the content is not skipped over by the UE (e.g. if the application does not respect the information).

While it is not possible to ensure that the user is actually watching the content, the network and/or content provider could limit the ability of the UE to fetch certain segments (e.g. movie segments) before a certain time after the special content has been fetched (i.e. not absolute time), and after all the special content segments have been fetched in their entirety.

6.18.2 Working assumptions

- The MPD file format can indicate that some on-demand content can be automatically skipped under certain conditions, including user preferences.
- The MPD file format can indicate that some on-demand content cannot be manually skipped in any circumstance.

- The content provider and/or the network operator are able to control to a certain extent that the UE does not skip over the content marked as "non-skippable", by limiting the ability of the UE to fetch regular content for a certain period after the special content has started to be fetched, and only if the special content has been fetched in its entirety.

6.19 Use Cases on DASH Authentication

6.19.1 Use Case 1

Alice has a DASH-capable client application that allows her watch DASH-formatted content. She is subscribed to Operator BestCoverage Telecom's PSS-based mobile streaming service. She is interested in watching movie "A Dash through the Clouds", which is available in DASH format. The operator is restricting access to the movie only to authorized users and employs 3GPP-based authentication mechanisms for this purpose. Since Alice is already subscribed to the mobile streaming service, her client is authenticated and she enjoys the movie.

6.19.2 Use Case 2

Alice and Bob both have DASH-capable client applications that allow them watch DASH-formatted content. They are both subscribed to Operator BestCoverage Telecom's PSS-based mobile streaming service. Bob has paid for the 'premium streaming' plan while Alice preferred the cheaper 'basic streaming' plan. They are both interested in watching movie "A Dash through the Clouds". The movie is available in DASH format at different bitrates / resolutions. Due to Bob's premium plan subscription, his client application is able to access and receive streams at all bitrates / resolutions offered by the service (by choosing at a given time the best one given the link bandwidth and device capabilities). Alice's client application is restricted from accessing the highest bitrates / resolutions due to her basic subscription, and it can only receive streams from a limited set of the available bitrates / resolutions.

6.19.3 Use Case 3

Operator BestCoverage Telecom has recently invested significantly into its infrastructure and is looking for new business opportunities to increase its service revenues by focusing on the third-party content distribution value chain. Particularly, it wishes to leverage its information systems and network equipment (e.g. Home Subscriber Subsystem or HSS) that contain valuable user information including authentication keys, user identities and user service profiles. Such information enables performing a number of control functions including user authentication, authorization of user access to services and billing on behalf of third-party content providers. It has recently signed a security/authentication related service level agreement (SLA) with DASH content provider MyDASH to distribute its DASH-formatted content by fulfilling user authentication and authorization on behalf of MyDASH over its 3GPP Generic Authentication Architecture (GAA). MyDASH hosts a tiered subscription service and requires enforcement of content-specific access restrictions for client authentication.

6.19.4 Use Case 4

Operator BestCoverage Telecom has recently signed a service level agreement (SLA) with DASH content provider MyDASH to distribute/resell its DASH-formatted content. The operator plans to use the DASH-formatted content from MyDASH to offer various new services to its clients and wants to ensure integrity of the content and associated metadata toward a consistent user experience. Even though the operator believes that it has made the right investments to its infrastructure to ensure security, it also wishes to provision against the potential intrusions to during DASH content delivery from MyDASH. It has therefore made sure to include a provision that commits MyDASH to a specified level of content delivery accuracy, as well as penalty provisions if the specified level of accuracy is not achieved. In response, MyDASH has decided to enable authentication mechanisms for the operator to validate the integrity of the delivered content and MPD.

6.19.5 Use Case 5

Operator BestCoverage Telecom has recently signed service level agreements (SLA) with several DASH content providers to distribute/resell their DASH-formatted content. The operator plans to use these DASH-formatted contents to offer various new PSS-based streaming services to its clients and wants to ensure integrity of the content and associated metadata toward a consistent user experience. In particular, the operator plans to employ service via stream splicing to create media mashups, combining content from multiple sources. A common example here is advertisement insertion, for both VoD and live streams, among other possible media mashups. Such schemes which employ dynamic MPD generation or rewriting must be cognizant of segment URLs and other metadata which should not be modified or removed. Improper modification of these URLs or other metadata may cause playback interruptions, and in the case of unplayed advertisements, may result in loss of revenue for content providers and operator.

6.20 Consistent Quality for DASH users

6.20.1 Use Case Description

The content provider MyDASH aims to offer DASH-formatted content for the 3GPP PSS-based streaming services delivered by the operator BestCoverage Telecom. The operator wishes to ensure that the users in its network experience consistent quality during streaming. MyDASH recognizes that the quality may significantly fluctuate in some of its content due to encoding limitations and decides to enable signalling of quality information to facilitate suitable adaptations on the client end for achieving consistent quality.

6.20.2 Gap Analysis

In DASH MPD, content bitrate characteristics are specified based on `@bandwidth` attribute of the **`Representation`** element in the MPD, allowing for differentiating across DASH representations from a bitrate perspective (averaging time scale defined with respect to **`MPD`**`@minBufferTime`).

If the Segment Index is provided, then this information may be downloaded for different Representations (signalled in-band or out-of-band) and provides a full bitrate over time map. This provides the client more detailed information about the actual bitrate/size of each (sub)segment. However, it may still be the case that the quality fluctuates despite (capped) VBR encoding.

On this front, Figures 6.20.1 and 6.20.2 show the per-segment bitrate and quality for the 5-min test video clip "A_Glimpse_of_China" encoded using unconstrained VBR and capped VBR, respectively. The original clip comes in 720p, 25fps format, encoded at very high rate (81.5Mbps) using H.264 high profile and the content is encoded with x264 with unconstrained VBR and capped VBR with parameters described in Table 6.20.1. The content is encoded into eight representations and the segment length is fixed to 2 seconds (only two representations are depicted in Figures 6.20.1 and 6.20.2, one in the blue curve and other in the green curve). The per-segment MS-SSIM scores of encoded content are calculated and then mapped to MOS scores at the receiving terminal based on an estimation method empirically validated via comprehensive subjective testing (further details are in clause Annex A). As shown in Figures 6.20.1 and 6.20.2, unconstrained VBR encoding yields to a more constant quality than capped VBR encoding. The key observation is that in both cases quality in terms of MOS scores fluctuates significantly and the quality variation becomes larger at a lower bitrate (blue curve).

Table 6.20.1 - Encoding Parameters in x264

Encoding parameter	Value	Note
--profile	High	common parameters
--preset	Veryslow	
--fps	25	
--keyint	50	
--min-keyint	50	
--no-scenecut		
--log-level	Debug	
--bitrate	500, 800, 1000, 1250, 1500, 2000, 2500, 4000	unconstrained VBR
--ratetol	Inf	
--tune	Ssim	
--bitrate	500, 800, 1000, 1250, 1500, 2000, 2500, 4000	constrained VBR
--vbv-maxrate	1000, 1600, 2000, 2500, 3000, 4000, 5000, 8000	
--vbv-bufsize	2000, 3200, 4000, 5000, 6000, 8000, 10000, 16000	

Consequently, DASH-formatted content for a given representation or sub-representation may vary in quality, which is not revealed by the existing @bandwidth attribute or 'sidx' information available at the client. Some potential implications of this may be that it may not be possible to maintain a consistent video quality during playback at the client side and some of the DASH-formatted segments (e.g., those for slow-moving scenes) may be requested and received at quality levels much higher than necessary wasting bandwidth resources.

Signalling of quality-related information about the content (via the DASH MPD or other means) to the DASH client can therefore be desirable. It is envisioned that by receiving information about quality of encoded segments along with their lengths, the client may be able to make more intelligent decisions about which Representation to select to maintain an overall good quality potentially with lower bandwidth consumption and faster buffering.

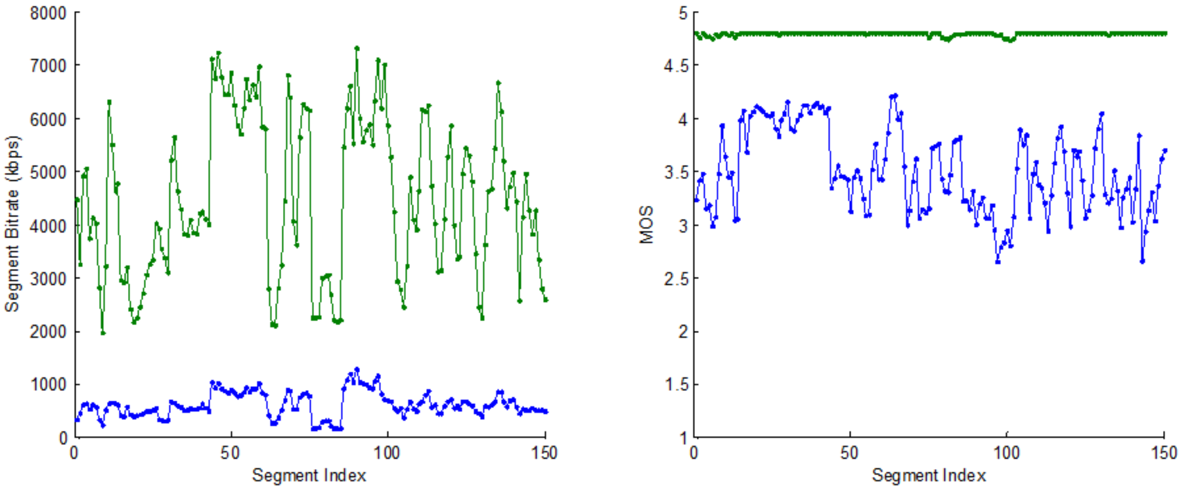


Figure 6.20.1 - Per-segment bitrate and quality for unconstrained VBR

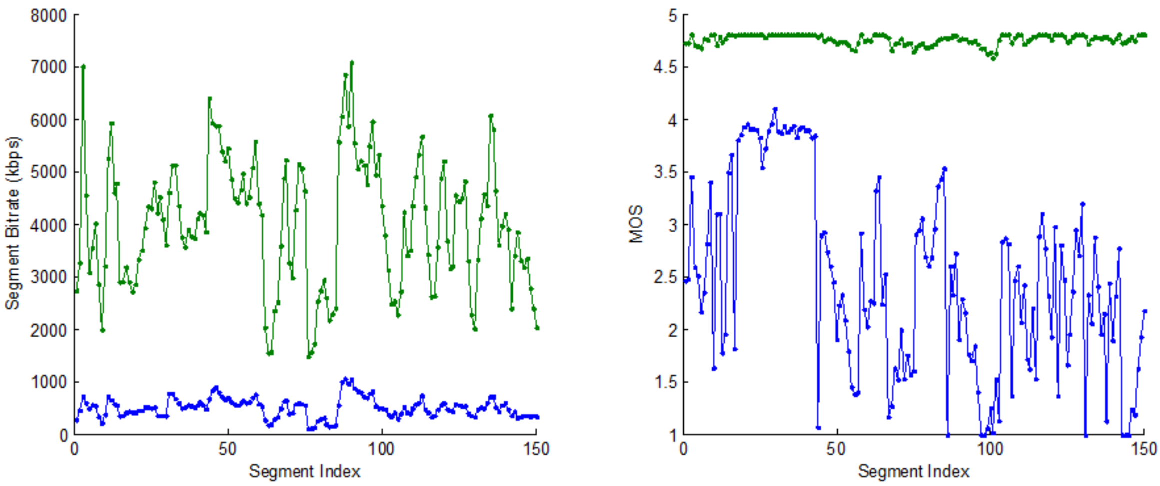


Figure 60.20.2 - Per-segment bitrate and quality for constrained VBR

6.20.3 Overview of Potential Solution Space

Creating suitable 3GP file format based profiles of the metadata track for carriage of quality information can be a potential solution for signalling quality to the DASH client and would meet several design goals in terms of compactness, general applicability, extensibility and backwards compatibility. This is the approach that MPEG has recently agreed (at MPEG#104 in July 2013) toward addressing this use case and a working draft specification has been initiated for this purpose. Such an approach is clearly agnostic to the DASH format and also enables compact and extensible signalling of quality information. Furthermore, it is believed to be possible to reuse a lot of the existing tools from DASH and 3GP file format specifications if this solution is adopted. The metadata track carrying quality information would be linked to the track it describes (e.g., video track) by means of a ‘cdsc’ (content describes) track reference. Different metric types and corresponding storage formats would be identified by their unique sample entry names with suitable 4cc ‘xxxx; definitions for the metrics and can be signalled via the existing ‘codecs’ parameter of the DASH MPD, e.g., ‘psnr’, ‘ssim’, etc.

There are several metrics for signalling of quality that have been proposed in the literature, but there is generally no universal agreement on which one is best in practice. Some of these metrics use dB scale, such as the objective metrics of PSNR, SSIM and MS-SSIM, while others are subjective metrics mapped to the interval associated with 5-level MOS scale. The field of quality measurements is evolving, and hence, it should be possible to consider new metrics to be defined in the future.

Quality information should be accessible for all segments / representations in the adaptation set, so that DASH client can independently retrieve it ahead of time before loading actual data segments. Finally, it should be noted that such addition of quality information should not break existing deployments. Existing clients should be operational by simply ignoring it, while new and more advanced client designs should be able to read and take advantage of it.

Detailed performance evaluation results on quality-driven rate adaptation algorithms are provided in Annex A.

7 Recommendations

<tbd>

Annex A: Detailed Performance Evaluation Results on Quality-Driven Rate Adaptation Algorithms

A.1 Introduction

This annex provides detailed performance evaluation results on quality-driven rate adaptation algorithms. It supports the use case in section 6.20.

A.2 Simulation Setup

Figure A.1 describes our simulation setup. In our simulation, the following tools were used

A. DASH Content Generation

Encoder: x264
Quality generation: MSU tool
(http://compression.ru/video/quality_measure/video_measurement_tool_en.html)
to calculate MS-SSIM, and a MOS Estimator to map MS-SSIM to MOS on different devices (further details on the latter provided below)

DASH encoder (from ITEC web site): http://www-itec.uni-klu.ac.at/dash/?page_id=282
This encoder produces the required MPD files according to DASH specifications. Additional codes are written to add quality information in the MPD file.

B. DASH Server

Microsoft IIS HTTP streaming server in Windows platform. The IIS streaming server supports HTTP streaming which is compatible with VLC DASH plugin.

C. Network Emulator

Apposite Netropy N60 is a hardware emulation engine that enables high-precision emulation of up to 15 separate WAN links to model complex network topologies or run multiple concurrent tests.

D. DASH Client

VLC-2.1.x Player with DASH plug-in (from VideoLan http://nightlies.videolan.org/). VLC player includes a DASH plugin implementation. This implementation was modified to support adding quality and segment size information to the MPD file. In addition, rate adaptation cases were implemented to evaluate the use of quality in rate adaptation.

Summary of details include

1. Parsing quality and size information from the modified MPD file.
2. Adding a sliding window to measure download rates at the client over a defined time interval. This sliding window contains the download rate of previous duration and will be used to estimate the available download rate for next segment.
3. Implementing advanced rate-adaptation algorithms for non-quality and quality-based MPD. More details will be discussed in clause A.4.

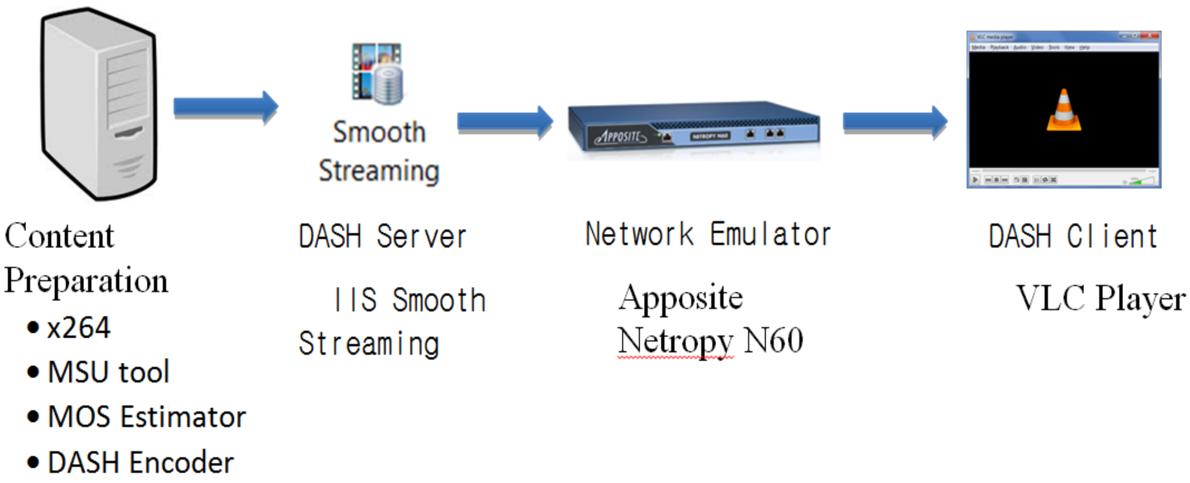


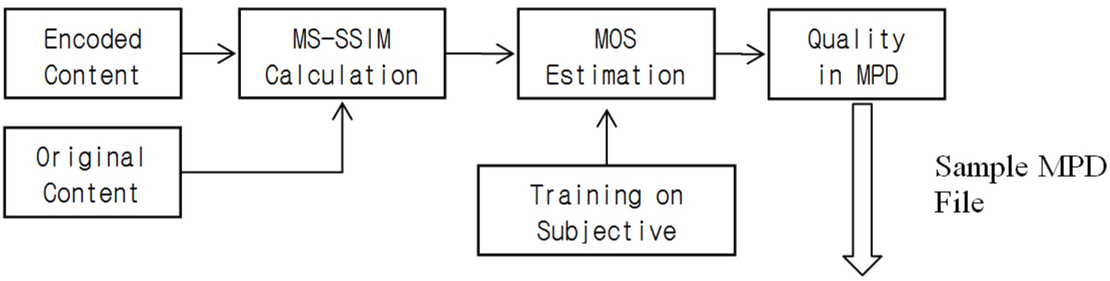
Figure A.1 - Simulation Setup

A.3 Content Preparation

The test clip used in the simulation is a 5-min video “A_Glimpse_of_China_short” provided by Huawei. The clip shows live footage of Shanghai, and offer variety of content: architecture, people, trees, cars, etc. The original clip comes in 720p, 25fps format, encoded at very high rate (81.5Mbps) using H.264 high profile. The content is encoded with x264 with unconstrained VBR and capped VBR and parameters described in Table A.1. The content is encoded into eight representations and the segment length is fixed to 2 seconds.

Note: Used --ratetol inf and --tune ssim for the unconstrained VBR setting since VLC DASH has problems supporting --crf encoding videos.

The quality generation process is shown in Figure A.2. The per-segment MS-SSIM scores of encoded content are calculated using the MSU tool and then mapped to MOS scores on different devices based on an estimation method empirically validated based on comprehensive subjective testing. The 5-point MOS scale was used as shown in Table t2 for the subjective quality evaluation.




```
<SegmentBase>
   <Initialisation sourceURL="glimpsesh_500kbit/A_Glimpse_of_China_short_500kbit_dashNonSeg.mp4" range="0-818" mos="3.5" />
</SegmentBase>
<SegmentList duration="2">
   <SegmentURL media="glimpsesh_500kbit/A_Glimpse_of_China_short_500kbit_dashNonSeg.mp4" mediaRange="819-81883" mos="3.2" />
   <SegmentURL media="glimpsesh_500kbit/A_Glimpse_of_China_short_500kbit_dashNonSeg.mp4" mediaRange="81884-194588" mos="3.4" />
   <SegmentURL media="glimpsesh_500kbit/A_Glimpse_of_China_short_500kbit_dashNonSeg.mp4" mediaRange="194589-348897" mos="3.5" />
   <SegmentURL media="glimpsesh_500kbit/A_Glimpse_of_China_short_500kbit_dashNonSeg.mp4" mediaRange="348898-510015" mos="3.2" />
   <SegmentURL media="glimpsesh_500kbit/A_Glimpse_of_China_short_500kbit_dashNonSeg.mp4" mediaRange="510016-645088" mos="3.2" />
```


Figure A.2 – Quality generation process

Table A.1 - Definition of 5-point MOS for Subjective Video Quality Evaluation

Scale	Description
5	Excellent: there are no artifacts
4	Good: artifacts are slightly noticeable, but they do not bother me
3	Fair: artifacts are noticeable, and they bother me a little
2	Poor: artifacts are very noticeable, and they bother me a lot
1	Bad: artifacts are severely noticeable, and I would not continue to watch

For the calculation of MOS scores, the subjective quality was modeled as a linear function of an objective quality metric, i.e., MS-SSIM in this case, as shown in Equation 1, where the prediction coefficients α and β are functions of video content characteristics (videos classified based on spatial details, motion levels, bitrate, resolution) and device characteristics (display size and resolution).

$$MOS = \alpha MSSSIM + \beta \qquad (1)$$

As such, this subjective quality estimator uses MS-SSIM, video content characteristics, and device characteristics to determine subjective video quality. The device knows about the device characteristics and can analyze the compressed video to get the content characteristics. However, MS-SSIM is a reference-based objective quality metric and cannot be known by the device unless this information is explicitly signaled to the client. So in order to determine the subjective quality at the client, signaling the quality information is necessary.

Comprehensive subjective testing experiments were conducted based on the standardized procedures established in [14] toward validating the above MOS estimation method for calculating MOS scores from MS-SSIM. All evaluations took place in a usability lab located at the Intel facilities in Hillsboro, Oregon, USA. Five different devices were considered in the experiment - HDTV, Android® Tablet, Android® Phone, iPad® and iPhone®. For each device, about 30 participants are asked to rate the video quality on a 5-point MOS scale as shown in Table A.1 and the subjective results are collected for evaluation.

NOTE: Android® is the trade name of a product supplied by Google Inc. iPad® is the trade name of a product supplied by Apple Computer Inc. iPhone® is the trade name of a product supplied by Apple Computer Inc. This information is given for the convenience of users of the present document and does not constitute an endorsement by 3GPP of the product named. Equivalent products may be used if they can be shown to lead to the same results.

In Figure A.3, the subjective quality MOS values estimated based on MS-SSIM is plotted against the empirically obtained subjective quality scores from experimentation. The results show that subjective video quality can be well estimated if the objective quality score (i.e., MS-SSIM), video content information, and client device characteristics are known.

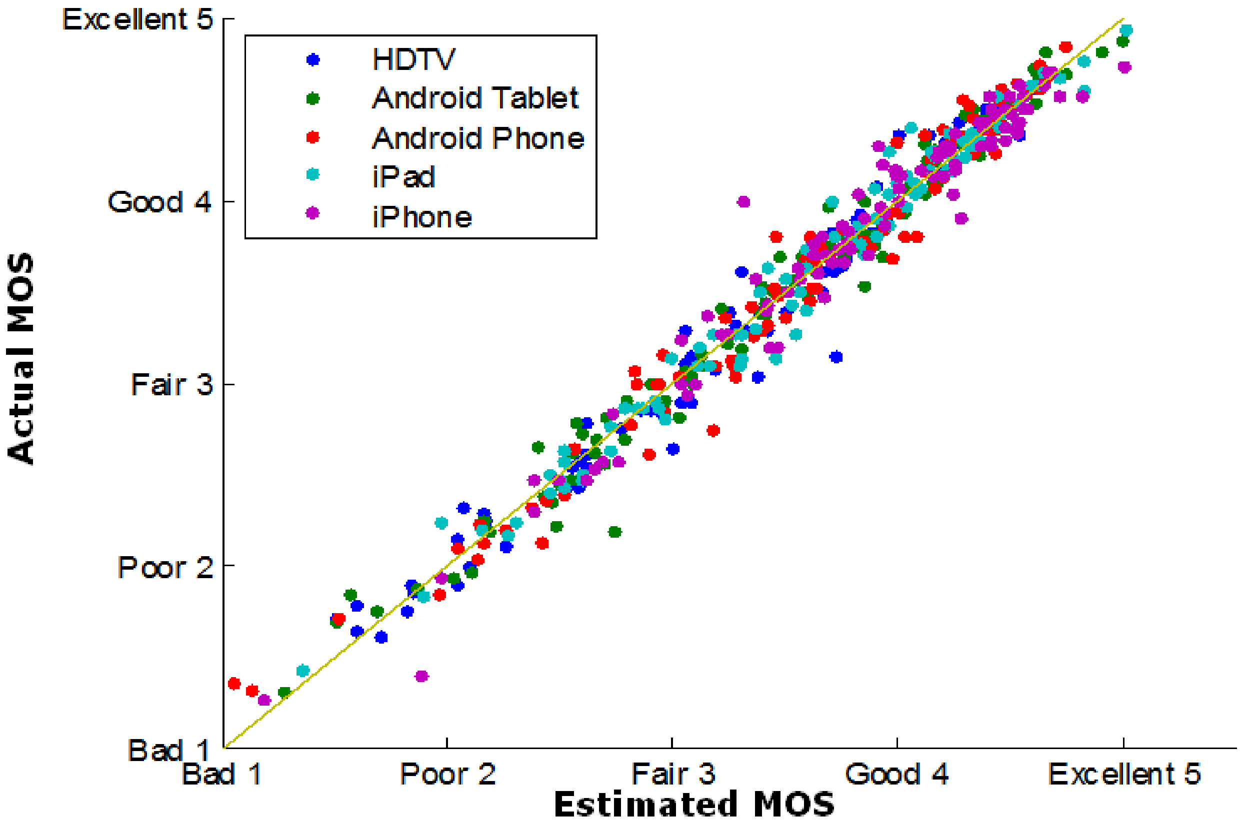


Figure A.3 MOS Estimation based on MS-SSIM, Video Content and Device Information

A.4 DASH Rate Adaptation Algorithms

A.4.1 Adaptation based on per-segment bitrate information (Non-Quality):

The basic idea of the non-quality algorithm is to adapt the segment bitrate to the available network bandwidth with different rate factors, which are determined by the buffer level. When the buffer level is high, the client could perform more aggressively by selecting a representation bitrate bounded by available BW*rate_factor. The flowchart of the algorithm is shown in Figure A.4.. The following parameters are defined in the algorithm:

Buffer percentage parameters: buf_low, buf_med, buf_high
Bitrate threshold factor: rate_factor1, rate_factor2

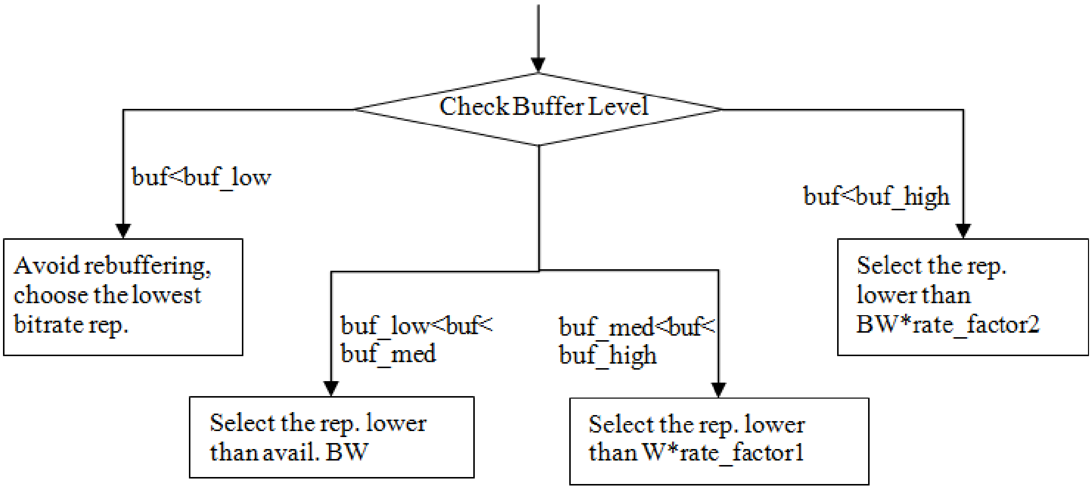


Figure A.4 - Flowchart of Non-quality-based Rate Adaptation Strategy

Algorithm Details:

Condition 0: buf < buf_low

- Select lowest bitrate representation

Condition 1: buf_low < buf < buf_med

- Select highest bitrate representation lower than the available BW

Condition 2: buf_med < buf < buf_high

- Select highest bitrate representation lower than the available BW*rate_factor1

Condition 3: buf > buf_high

- Select highest bitrate representation lower than the available BW*rate_factor2

A.4.2 Adaptation based on per-segment bitrate and quality information (Quality-based)

The main idea of the quality-based algorithm is to maintain a balance among buffer level, selected bitrate and quality based on the available bandwidth and per-segment bitrate/quality information. The flowchart of the quality-based algorithm is shown in Figure A.5 and the following parameters are defined:

Buffer percentage parameters: buf_low, buf_med, buf_high
Quality threshold: quality_min, quality_max
Bitrate threshold factor: rate_factor1, rate_factor2

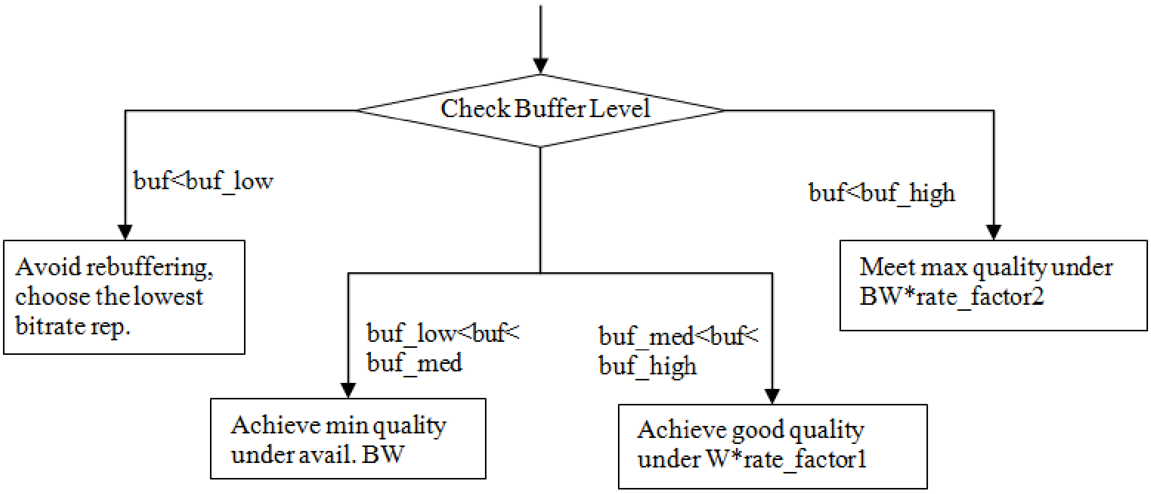


Figure A.5 - Flowchart of Quality-based Rate Adaptation Strategy

Algorithm Details:

Condition 0: buf < buf_low

- Choose lowest bitrate representation

Condition 1: buf_low < buf < buf_med

- Select representations lower than the available BW
    - If quality is higher than quality_min, choose the lowest bitrate representation that satisfies quality_min
    - Else: choose highest quality representation

Condition 2: buf_med < buf < buf_high

- Select representations lower than the available BW*rate_factor1
    - If quality between quality_min and quality_max
        - If lower than avail. BW: select highest quality
        - Else choose highest bitrate representation lower than avail. BW
    - Elseif: quality < quality_min, select the highest bitrate representation
    - Elseif: quality > quality_max, select the lowest bitrate representation

Condition 3: buf > buf_high

- Select representations lower than the available BW*rate_factor2
    - If quality is higher than quality_max, choose the lowest bitrate representation that satisfies quality_max
    - Else: choose highest quality representation

A.4.3 Bandwidth Model

Apposite Netropy N60 Network Emulator was used to emulate various channel conditions. Figure A.6 shows the bandwidth models evaluated in the simulation. In model A, the available bandwidth stays constant at a level for 10s and then switches to a different level following the pattern shown in the figure. In model B, the available bandwidth alternatively switches between 2Mbps and 200kbps every 5s.

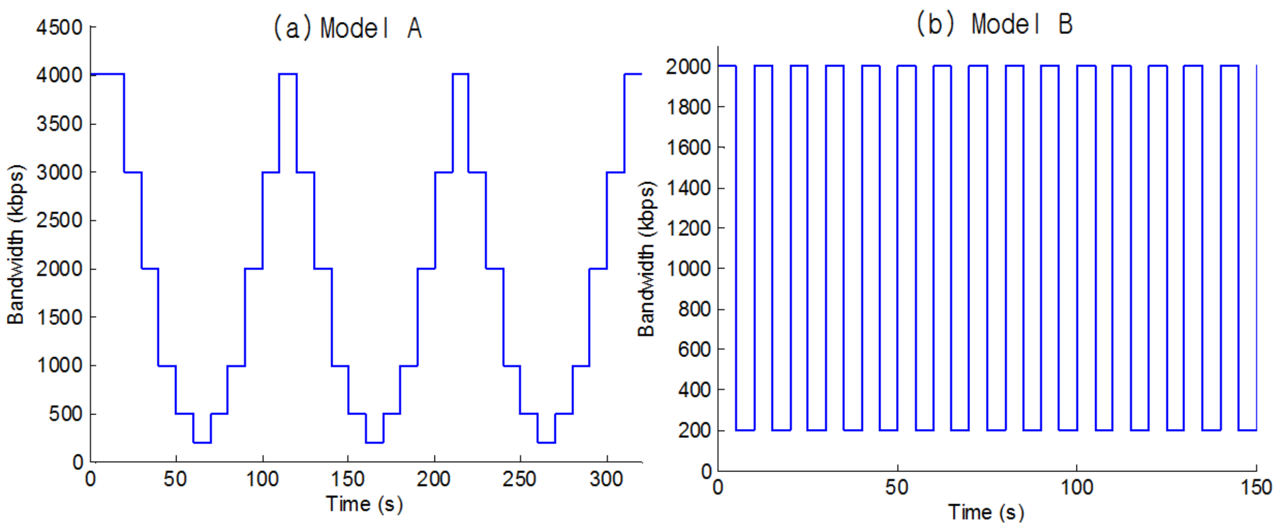


Figure A.6 – Bandwidth Models A and B

A.4.4 Experimental Results

In this section, evaluation results are presented on the performance of non-quality and quality-based DASH rate adaptation algorithms with unconstrained VBR and constrained VBR DASH content in different network scenarios. The VLC buffer size is fixed at 30s. For both algorithms, the pre-defined parameters in Table A.2 are used unless otherwise noted.

Table A.2 - Pre-defined Parameters for Rate Adaptation

	buf_low	buf_med	buf_high	rate_factor1	rate_factor2	Q_{min}	Q_{max}
Non-quality	30%	50%	70%	1.0	1.0	--	--

Quality-based	30%	40%	70%	3.0	3.0	3.0	4.5

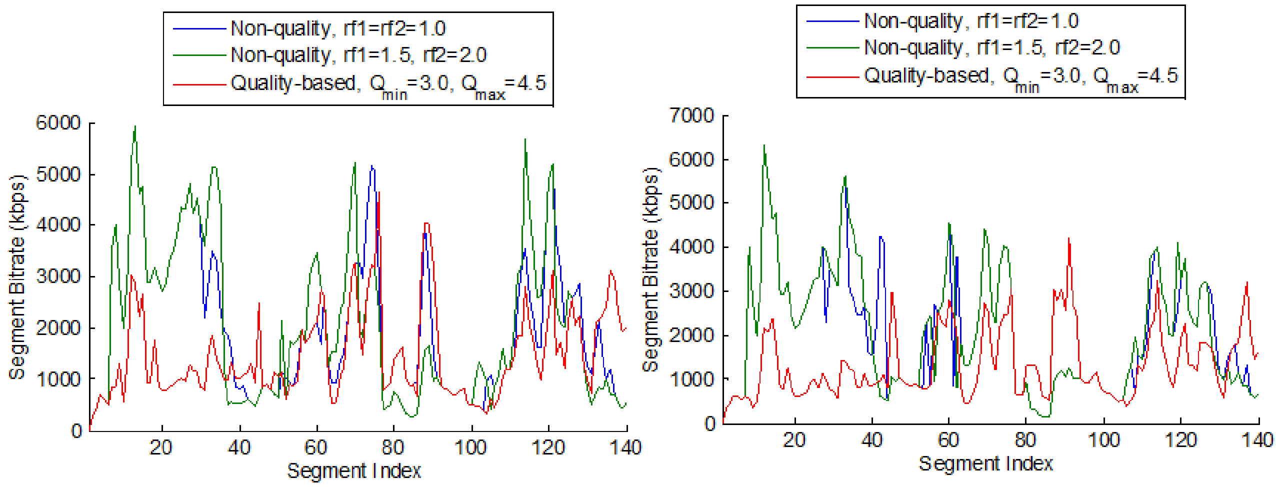
Table A.3 compares the statistics of non-quality and quality-based rate adaptation algorithms under network model A. It is seen that for the constrained VBR case, the quality-based algorithm achieves the same average quality as the non-quality algorithms with less percentage of low-quality periods. At the same time, the quality-based algorithm consumes 27% less bandwidth and maintains a higher buffer level while the non-quality algorithms suffer from one rebuffering event. For the unconstrained VBR case, similar observations can be made that the quality-based algorithm achieves the same average quality with shorter bad-quality periods and consumes 34% less bandwidth with a higher average buffer level. The results show that with the quality information, the DASH client can maintain a better balance among bitrate, quality and buffer level. The results also show that the constrained VBR has worse average quality and larger quality fluctuations under the same network assumption, which indicates that when stricter bitrate variation is applied to the compressed content, it is very important to signal quality information to the DASH client in order to efficiently stream the videos with good user experience.

Table A.3 - Statistics of Non-quality and Quality-based Rate Adaptation Algorithms (Bandwidth Model A)

Metric / Algorithm	Avg. Bitrate (Mbps)	Avg. MOS	MOS<3	Avg. Buffer %	Rebuffering Time (s)	No. Rebuffer
Constrained VBR						
Non-Quality (rf1=rf2=1.0)	2.02	3.98	24.3%	56.0%	4.9	1
Non-Quality (rf1=1.5, rf2=2.0)	2.03	3.92	26.4%	45.5%	14.3	1
Quality-based (Q_{min}=3.0, Q_{max}=4.5)	1.47 (27%)	3.99	15.7%	64.7%	0	0
Unconstrained VBR						
Non-Quality (rf1=rf2=1.0)	2.04	4.21	5.0%	45.0%	0	0
Non-Quality (rf1=1.5, rf2=2.0)	2.05	4.21	5.0%	43.6%	0	0
Quality-based (Q_{min}=3.0, Q_{max}=4.5)	1.34 (34%)	4.22	3.6%	70.5%	0	0

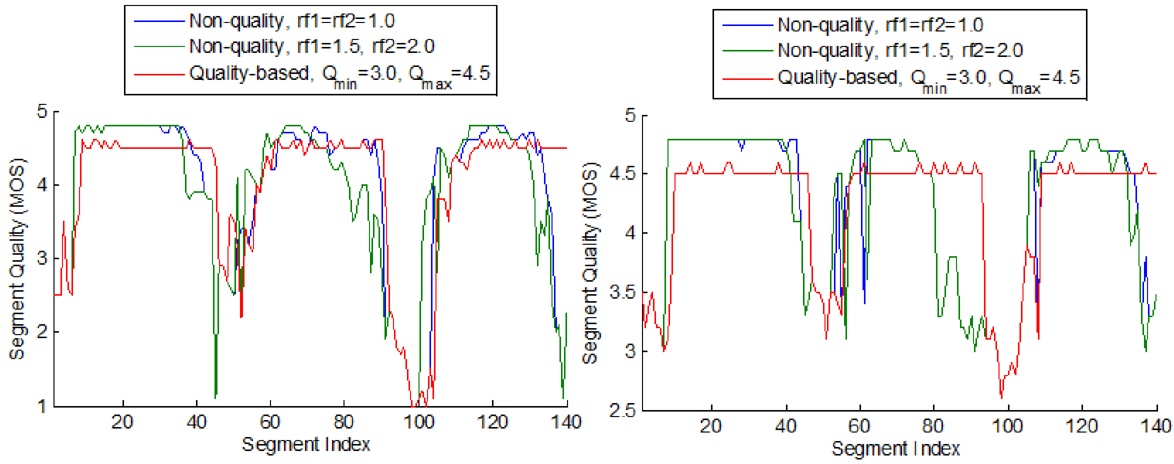
Figures A.7-A.9 present the corresponding streamed segment bitrate, quality and the buffer status. These findings show that the quality-based algorithm can make smarter adaptation decision based on current buffer level, segment bitrates and quality information. In some situations, the algorithm chooses to stream at a lower bitrate when the quality requirement is met, which helps to save bandwidth consumption and fill up the buffer faster. In other situations, the algorithm may request a higher bitrate than the available bandwidth to trade off the quality gains if the buffer level allows. On the contrary, for the non-quality case, requesting a bitrate higher than the available bandwidth without knowing the quality tradeoff could unnecessarily drain the buffer and worsen the overall performance.

Furthermore, another case can be considered where the representations provided to the devices are limited based on the quality (i.e., do not offer representations higher than a MOS=4.5) and the performance can be evaluated under bandwidth model B. The statistics are shown in Table A.4. It is seen that capping the representation bitrate based on the quality information helps to improve the efficiency of the non-quality based algorithm. It limits the unnecessary bandwidth consumption when max quality requirement is met. Compared to the non-quality algorithm without representation capping, it achieves better quality performance and higher buffer level with the same amount of bandwidth consumption. However, quality-based algorithms still yields a better overall performance since capping high bitrate representation would not help the case where the client needs to adapt to several moderate bitrate representations based on the quality information. The results also show that choosing the proper quality parameters for different network conditions helps to improve the performance. This indicates that the quality-based algorithm can be further improved by designing a cost function that takes into account buffer level, segment bitrate, segment quality, and available bandwidth (short-term and long-term) to find the optimal representation.



(a) Constrained VBR (b) Unconstrained VBR

Figure A.7 - Segment Index vs. Segment Bitrate (Bandwidth Model A)



(a) Constrained VBR (b) Unconstrained VBR

Figure A.8 - Segment Index vs. Segment Quality (Bandwidth Model A)

Table A.4 - Statistics of Non-quality and Quality-based Rate Adaptation Algorithms (Bandwidth Model B)

Constrained VBR*				
Algorithm \ Metric	Avg. Bitrate (Mbps)	Avg. MOS	MOS<3	Avg. Buffer %
Non-Quality (rf1=rf2=1.0)	1.36	3.62	32.9%	39.2%
Non-Quality (rf1=1.5, rf2=2.0)	1.33	3.55	35.7%	35.2%
Non-Quality (rf1=rf2=1.0), Capped MOS=4.5	1.27	3.80	25.0%	55.9%
Non-Quality (rf1=1.5, rf2=2.0), Capped MOS=4.5	1.34	3.83	21.4%	51.8%
Quality-based (Q_{min}=3.0, Q_{max}=4.5)	1.14	3.74	16.4%	51.7%
Quality-based (Q_{min}=3.0, Q_{max}=4.0)	1.01	3.75	12.1%	58.0%

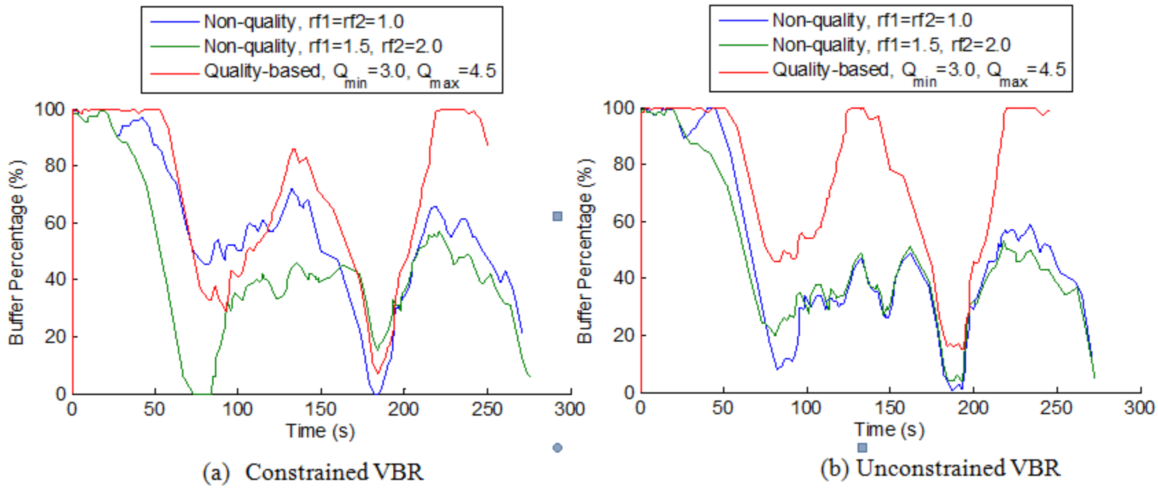


Figure A.9 - Buffer Percentage (Bandwidth Model A)

Annex X: Change history

Change history							
Date	**TSG SA#**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
06-2013	60	SP-130192			Presented for information at TSG SA#60		1.0.0