

ETSI TR 126 938 V16.0.0 (2020-09)



**Universal Mobile Telecommunications System (UMTS);  
LTE;  
Packet-switched Streaming Service (PSS);  
Improved support for dynamic adaptive streaming  
over HTTP in 3GPP  
(3GPP TR 26.938 version 16.0.0 Release 16)**



---

Reference

RTR/TSGS-0426938vg00

---

Keywords

LTE,UMTS

**ETSI**

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

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

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

---

**Important notice**

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at [www.etsi.org/deliver](http://www.etsi.org/deliver).

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommitteeSupportStaff.aspx>

---

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2020.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.

**3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners.

**GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

---

## Intellectual Property Rights

### Essential patents

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

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

### Trademarks

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

---

## Legal Notice

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

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

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

---

## Modal verbs terminology

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

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

# Contents

Intellectual Property Rights .....	2
Legal Notice .....	2
Modal verbs terminology.....	2
Foreword.....	8
Introduction .....	8
1 Scope .....	9
2 References .....	9
3 Definitions and abbreviations.....	10
3.1 Definitions .....	10
3.2 Abbreviations .....	11
4 Relevant Specifications .....	12
4.1 Overview .....	12
4.2 3GP-DASH as a profile of MPEG-DASH .....	12
4.2.1 General.....	12
4.2.2 Media Codecs .....	13
4.2.3 Media Presentation Description constraints.....	13
4.2.4 Segment format constraints .....	13
4.2.5 Extensions.....	13
4.2.5.1 Media Presentation Description Delta.....	13
5 Deployment Guidelines.....	15
5.1 Introduction .....	15
5.2 Content Authoring Guidelines.....	15
5.3 Client implementation and client operation guidelines .....	15
5.3.1 Guidelines for rate adaptation.....	15
5.3.1.1 Introduction.....	15
5.3.1.2 Rate adaptation in DASH.....	15
5.4 Operational and deployment guidelines .....	16
5.4.1 General.....	16
5.4.2 Proxy/cache switch for DASH service.....	16
5.4.2.1 Assumptions.....	16
5.4.2.2 Description .....	16
5.4.2.3 Working assumption .....	16
5.4.2.4 Recommended Requirements.....	17
6 General Use Cases.....	17
6.1 Introduction .....	17
6.2 Advanced Support for Live Services.....	17
6.2.1 Description.....	17
6.2.1.1 Setup .....	17
6.2.1.2 Use Case A: Start-up latency .....	17
6.2.1.3 Use Case B: Aligned Presentation .....	17
6.2.1.4 Use Case C: End-to-end latency.....	17
6.2.1.5 Use Case D: Time-Shift Buffer.....	17
6.2.1.6 Use Case E: Infrastructure Upgrade.....	17
6.2.1.7 Use Case F: Ad Insertion .....	17
6.2.2 Operation with MPD dynamic Mode.....	18
6.2.2.1 Introduction.....	18
6.2.2.2 Problem Statement .....	18
6.2.2.3 Existing Technologies.....	18
6.2.2.4 Consequences with Existing Technologies .....	18
6.2.2.5 How does DASH solve this?.....	19
6.2.2.5.1 Overview .....	19
6.2.2.5.2 Benefits of this approach .....	19

6.2.2.6	MPD Times .....	20
6.2.2.7	General Derivation .....	20
6.2.2.8	Derivation of MPD Times .....	21
6.2.2.9	Addressing Methods .....	21
6.2.2.9.1	Introduction .....	21
6.2.2.9.2	Playlist-Method .....	21
6.2.2.9.3	Number-Based Template .....	21
6.2.2.10	Scheduling Playout .....	21
6.2.2.11	Validity of MPD .....	22
6.2.3	Mapping Use Cases to Live Operation .....	22
6.2.3.1	Use Case A .....	22
6.2.3.1.1	Description .....	22
6.2.3.1.2	MPD example .....	22
6.2.3.1.3	Client Procedure .....	22
6.2.3.2	Use Case B .....	23
6.2.3.2.1	Description .....	23
6.2.3.2.2	MPD example .....	23
6.2.3.2.3	Client Procedure .....	23
6.2.3.3	Use Case C .....	23
6.2.3.3.1	Description .....	23
6.2.3.3.2	MPD example .....	23
6.2.3.3.3	Client Procedure .....	23
6.2.3.4	Use Case D .....	23
6.2.3.4.1	Description .....	23
6.2.3.4.2	MPD example .....	24
6.2.3.4.3	Client Procedure .....	24
6.2.3.5	Use Case E .....	24
6.2.3.5.1	Description .....	24
6.2.3.5.2	MPD example .....	24
6.2.3.5.3	Client Procedure .....	25
6.2.3.6	Use Case F .....	25
6.2.3.6.1	Description .....	25
6.2.3.6.2	MPD example .....	25
6.2.3.6.3	Client Procedure .....	25
6.2.4	Gap Analysis .....	25
6.2.5	Working Assumptions .....	26
6.3	Use Cases for Content Protection .....	26
6.3.1	Use Case A – Efficient Caching for Multiple DRM Systems .....	26
6.3.1.1	Description .....	26
6.3.1.2	Actors' issues .....	26
6.3.1.3	Analysis in the Context of Rel-10 TS 26.247 .....	26
6.3.2	Use Case B – Signalling of Rights/License Acquisition Information in MPD .....	27
6.3.2.1	Description .....	27
6.3.2.2	Actors' issues: .....	27
6.3.2.3	Analysis in the Context of Rel-10 TS 26.247 .....	27
6.3.3	Use Case C – Time-Varying Decryption Keys .....	28
6.3.3.1	Description .....	28
6.3.3.2	Actors' issues: .....	28
6.3.3.3	Analysis in the Context of Rel-10 TS 26.247 [2] .....	28
6.4	Fast Media Start-up .....	29
6.4.1	Description .....	29
6.4.2	Analysis in the Context of Rel-10 TS 26.247 .....	29
6.5	Advanced Trick Modes .....	29
6.5.1	Description .....	29
6.5.2	Analysis in the Context of MPEG-DASH and Rel-10 TS 26.247 .....	29
6.5.2.1	Overview .....	29
6.5.2.2	Sub-Representations .....	30
6.5.2.3	MPD authoring .....	30
6.5.2.4	Segment generation .....	30
6.5.2.5	Summary .....	32
6.6	Content and Device Interoperability .....	32
6.6.1	Use Case Description .....	32

6.6.2	Working Assumption.....	33
6.6.3	Gap Analysis.....	33
6.7	Advanced Support for Live Services.....	33
6.7.1	Description.....	33
6.7.2	Working Assumption.....	33
6.7.3	Gap Analysis.....	33
6.8	Consistent QoE/QoS for DASH users.....	33
6.8.1	Description.....	33
6.8.2	Proposed Work.....	34
6.8.3	Utilization of QoS Information in DASH.....	34
6.9	DASH as download format.....	35
6.9.1	Description.....	35
6.10	Use Case: Use case description for Efficiency of HTTP-caching infrastructure on DASH.....	37
6.10.1	Description.....	37
6.11	Use Case: Multiple Spectator Views offered with DASH.....	39
6.11.1	Description.....	39
6.11.2	Gap Analysis.....	39
6.11.2.1	File support for timed position/location.....	39
6.11.2.2	MPD indication of position/orientation for a Representation or Segment.....	41
6.11.2.3	Synchronization of media content from different devices.....	42
6.11.2.4	MPD indication of key highlights of the event.....	42
6.11.2.5	MPD indication of capturing parameters.....	42
6.12	Use cases for operator control of video streaming services.....	43
6.12.1	Introduction.....	43
6.12.2	Description.....	43
6.12.2.1	Operator control of video services.....	43
6.12.2.2	Operator control using mobile network subscription.....	43
6.12.2.3	Operator management of video streaming service over the radio network.....	43
6.12.3	Working assumptions.....	43
6.12.4	Conclusions and Gap Analysis.....	44
6.13	Use Cases for DASH Operation with Network Proxy Caches.....	44
6.13.1	Introduction.....	44
6.13.2	Use Case Description.....	44
6.13.2.1	Use Case 1: Fast Startup.....	44
6.13.2.2	Use Case 2: Partial Representation Caching.....	45
6.13.2.3	Use Case 3: Network mobility.....	45
6.13.2.4	Use Case 4: Mobility and Coverage Extension for MBMS-based service.....	45
6.13.3	Solution Analysis: Usage of TS 26.247.....	45
6.13.3.1	MPD Update.....	45
6.13.3.2	Redirections.....	46
6.13.3.3	Bandwidth Throttling.....	47
6.13.3.4	Control Events.....	48
6.13.3	Gap Analysis and Problem Statement.....	49
6.14	Services with caching of DASH content at UE functions.....	50
6.14.1	Use Case Description.....	50
6.14.1.1	Use case 1: HTTP Proxy Cache in UE for transparent reception of DASH content over unicast or broadcast.....	50
6.14.1.2	Use case 2: HTTP Proxy Cache in a home GW (gateway), capable of serving multiple DASH players in the home.....	50
6.14.1.3	Use case 3: Pre-Caching on UE.....	50
6.14.2	Working assumptions.....	51
6.14.3	Gap Analysis.....	51
6.14.4	Recommended Requirements.....	51
6.15	Handling special content.....	52
6.15.1	Description.....	52
6.15.1.1	Enabling users to skip some repetitive/low importance content.....	52
6.15.1.2	Preventing users from skipping special content.....	52
6.15.1.3	Network/Content provider control of special content.....	52
6.15.2	Working assumptions.....	52
6.15.3	Analysis against TS 26.247: Dynamic Services.....	52
6.15.3.1	General.....	52
6.15.3.2	Tools and Evidence.....	53

6.15.3.3	Just-in-Time Content Insertion.....	53
6.15.3.4	Client Behaviour .....	54
6.15.3.5	More complex rules.....	54
6.15.3.6	Addressing the Use Cases .....	54
6.15.3.6.1	Repetitive Content .....	54
6.15.3.6.2	Forced Play-out .....	54
6.15.3.6.3	Flexible Ad Insertion .....	55
6.15.3.4	Gap Analysis and Recommendations .....	57
6.16	Use Cases on DASH Authentication .....	57
6.16.1	Use Case 1 .....	57
6.16.2	Use Case 2 .....	57
6.16.3	Use Case 3 .....	58
6.16.4	Use Case 4 .....	58
6.16.5	Use Case 5 .....	58
6.16.6	Gap Analysis on the Use Cases .....	58
6.16.6.1	Analysis of Use Cases 1-3 on content access authorization.....	58
6.16.6.2	Analysis of Use Cases 4-5 on application-level content/metadata integrity validation: .....	59
6.17	Consistent Quality for DASH users.....	59
6.17.1	Use Case Description.....	59
6.17.2	Gap Analysis.....	59
6.17.3	Overview of Potential Solution Space .....	60
6.17.4	Additional Simulation Results on DASH over GBR and Non-GBR bearers.....	61
7	Advertisement Insertion Use Cases .....	63
7.1	Introduction .....	63
7.2	Use Cases .....	63
7.2.1	Use Case 1: Fixed duration advertisement in On-Demand Content .....	63
7.2.2	Use Case 2: Variable duration advertisement in On-Demand Content.....	64
7.2.3	Use Case 3: Trusted Client Playback.....	64
7.2.4	Use Case 4: Advertisement in Live Content .....	64
7.2.5	Use Case 5: Accessing Live Content .....	64
7.3	Working Assumptions .....	64
7.3.1	Introduction.....	64
7.3.2	Generic assumptions .....	64
7.3.3	Single Media Presentation .....	65
7.3.4	Presentation Layer controlled Ad Insertion .....	65
7.4	Analysis in the Context of TS 26.247.....	65
7.4.1	Single Media Presentation .....	65
7.4.2	Presentation Layer controlled Ad Insertion Analysis .....	70
7.4	Examples .....	72
7.4.1	Fixed Duration in On-Demand .....	72
7.4.2	Targeted Advertisements .....	72
7.4.3	Example call for Ad insertion for DASH.....	74
7.6	Advanced Advertisement insertion in the operator network .....	75
7.6.1	Description.....	75
7.6.1.1	Advertisement insertion based on user subscription .....	75
7.6.1.2	Targeted advertisement insertion .....	75
7.6.1.3	Bandwidth-related advertisement/message .....	75
7.6.2	Working assumptions .....	75
7.6.3	Solution based on TS 26.247 and Gap Analysis .....	75
8	Conclusions and Recommendations.....	76
<b>Annex A:</b>	<b>Detailed Performance Evaluation Results on Quality-Driven Rate Adaptation Algorithms .....</b>	<b>78</b>
A.1	Introduction .....	78
A.2	Simulation Setup .....	78
A.3	Content Preparation.....	79
A.4	DASH Rate Adaptation Algorithms.....	81
A.4.1	Adaptation based on per-segment bitrate information (Non-Quality):.....	81
A.4.2	Adaptation based on per-segment bitrate and quality information (Quality-based).....	82

A.4.3 Bandwidth Model .....83  
A.4.4 Experimental Results.....84  
**Annex B: Evaluation of QoS-Driven DASH Client Adaptation .....87**  
**Annex C: Change history .....97**  
History .....98



---

## Foreword

This Technical Report has been produced by the 3<sup>rd</sup> 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 guiding 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 | 15444-12:2012 "Information technology – Coding of audio-visual objects – Part 12: ISO base media file format" | "Information technology – JPEG 2000 image coding system – Part 12: ISO base media file format".
- [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", January 30, 2014.
- [13] (void)

- [14] ITU-R Recommendation BT.500-11: "Methodology for the Subjective Assessment of the Quality of Television Pictures", 2002.
- [15] 3GPP TS 23.203: "Policy and Charging Control Architecture".
- [16] 3GPP TS 23.207: "End-to-End Quality of Service (QoS) Concept and Architecture".
- [17] 3GPP TS 29.213: "Policy and charging control signalling flows and Quality of Service (QoS) parameter mapping"
- [18] 3GPP TS 29.214: "Policy and charging control over Rx reference point"
- [19] 3GPP TS 23.401: "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access"
- [20] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)"
- [21] 3GPP TR 36.814: "Further advancements for E-UTRA physical layer aspects".
- [22] ISO/IEC 23009-1:2014: "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats".
- [23] 3GPP TR 23.705: "System Enhancements for User Plane Congestion Management".
- [24] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".
- [25] IAB VMAP: "Digital Video Multiple Ad Playlist (VMAP) 1.0", July 2012, [www.IAB.net](http://www.IAB.net).
- [26] SCTE 35: "Digital Program Insertion Cueing Message for Cable", 2013, <http://www.scte.org>
- [27] IAB VAST: "Digital Video Ad Serving Template (VAST) 3.0", July,2012, [www.IAB.net](http://www.IAB.net).
- [28] ISO/IEC 23001-10:2014 "Information technology – MPEG systems technologies -- Part 10: Carriage of Timed Metadata Metrics of Media in ISO Base Media File Format".
- [29] IETF RFC 4337: "MIME Type Registration for MPEG-4," March 2006.
- [30] IETF RFC 6381: "The 'Codecs' and 'Profiles' Parameters for "Bucket" Media Types," August 2011.
- [31] IETF RFC 6265: "HTTP State Management Mechanism".
- [32] IETF RFC 2109: "HTTP State Management Mechanism".
- [33] Recommendation ITU-T P.1202.1: "Parametric non-intrusive bitstream assessment of video media streaming quality - Lower resolution application area".
- [34] ISO/IEC 23001-9: " Information technology -- MPEG systems technologies -- Part 9: Common encryption of MPEG-2 transport streams".
- [35] IETF RFC 5681: "TCP Congestion Control".
- [36] Recommendation ITU-T J.181: "Digital program insertion cueing message for cable television systems".

---

## 3 Definitions and abbreviations

### 3.1 Definitions

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

**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.

**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.

## 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
ACDC	Application specific Congestion control for Data Communication
AHS	Adaptive HTTP Streaming
AJAX	Asynchronous JavaScript and XML
AVC	Advanced Video Coding
AVP	Attribute-Value Pairs
BMW	Bayerische Motoren Werke
CAS	Conditional Access System
CBP	Constrained Baseline Profile
CDN	Content Delivery Network
CoD	Content-on-Demand
DASH	Dynamic Adaptive Streaming over HTTP (DASH)
DRM	Digital Rights Management
DVD	Digital Versatile Disc
FLUTE	File Delivery over Unidirectional Transport
GPS	Global Positioning System
HD	High Definition
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
MPD	Media Presentation Description
MOS	Mean Opinion Score
MSS	Maximum Segment Size
MS-SSIM	Multi-Scale Structural SIMilarity
P-GW	PDN Gateway
PDN	Packet Data Network
PSNR	Peak-Signal-to-Noise-Ratio

PSS	Packet-switched Streaming Service
QoE	Quality-of-Experience
QoS	Quality-of-Service
RAT	Radio Access Technology
RTT	Round-Trip Time
SAP	Stream Access Point
SSIM	Structural SIMilarity
TCP/IP	Transmission Control Protocol / Internet Protocol
TV	TeleVision
UE	User Equipment
UPCON	User Plan Congestion control
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UTC	Coordinated Universal Time
VBR	Variable Bit Rate
VGA	Video Graphics Array
VoD	Video-on-Demand
WLAN	Wireless Local Access Network
WQVGA	Wide Quarter Video Graphics Array
WVGA	Wide Video Graphics Array
XML	EXtended Markup Language

---

## 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].

In addition, during 2012 and 2013 MPEG has conducted work on a second edition of DASH [22] that includes additional technologies. These technologies are currently not part of 3GP-DASH, but some the technologies are discussed in the present document as they may serve as a candidate technology to fulfil advanced use cases.

### 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 is expected to be provided according to RFC 4337 [29]. Additional parameters may be added according to RFC 6381 [30].

## 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 [34], 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 4.1. The XML-syntax of **x3gpp:DeltaSupport** element is provided in Table 4.2.

Table 4.1: Semantics of `x3gpp:DeltaSupport` element

Element or Attribute Name	Use	Description
<b>x3gpp:DeltaSupport</b>		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 <code>BaseURL</code> on MPD level and reference resolution according to clause 8.2.3 of TS26.247 [2] 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.
<b>Legend:</b> For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

Table 4.2: 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 4.3.

Table 4.3: Change commands and the instructions for delta MPD files

Change command	Instruction	Example
<code>1a</code>	Add text from the second file after line <code>1</code> in the first file.	'8a' means to add the following lines after line 8 of file 1
<code>rc</code>	Replace the lines in range <code>r</code> 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.
<code>rd</code>	Delete the lines in range <code>r</code> 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 <a href="http://www.gnu.org/software/diffutils/manual/#Detailed-ed">http://www.gnu.org/software/diffutils/manual/#Detailed-ed</a>		

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]. In order to address 3GPP-specific aspects the capabilities of 3GPP PSS codecs as well as bitrates mapped to typical 3GPP networks should be considered.

### 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.

In conjunction with rate adaptation, client implementations can include a request cancellation functionality in order to address stalled or expiring HTTP requests, i.e. if the client identifies that the requested segment/subsegment cannot be received in time., or the case that the data is no longer needed due to user interaction. This is relevant to avoid buffer under-runs and to be reactive to user interactions. However, cancelling an HTTP request does have a disadvantage, as this typically requires tearing down and re-establishing a TCP connection. This should be taken into account before cancelling a TCP connection. Any improvements to support HTTP request cancellation may be suitable.

For improving throughput especially during media startup, HTTP pipelining as defined in RFC 2616 [6] may be used by the client. HTTP servers used for DASH distribution preferably support HTTP pipelining.

Further rate adaptation enhancements based on the availability of quality and QoS information are discussed in clauses 6.8 and 6.17.

## 5.4 Operational and deployment guidelines

### 5.4.1 General

For general operational and deployment guidelines see ISO/IEC 23009-3 [10].

As access bandwidth on 3GPP networks is typically shared among many users and is precious, DASH clients are recommended avoiding excessive download of non-consumed data as well as downloading data that is unsuitable for the consumption. DASH clients are recommended avoiding building large download buffers, for example in the range of several minutes, as user may decide to terminate the viewing or apply fast forwards and so on. In addition a DASH client is recommended avoiding downloading Representations that it is not expected to consume, for example alternate languages etc. Furthermore, DASH clients are recommended not downloading Representations for which no quality gains are obvious under the current consumption model. For example, DASH clients are recommended avoiding downloading full resolution video if video is only played as a thumbnail in an application.

Content authoring is recommended to be such that downloading of unnecessary data can be avoided, for example by not multiplexing multiple languages into a single Representation.

### 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 General Use Cases

### 6.1 Introduction

This clause 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 analysed, potential solutions reusing existing technologies are provided and potential gaps are identified.

### 6.2 Advanced Support for Live Services

#### 6.2.1 Description

##### 6.2.1.1 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.

##### 6.2.1.2 Use Case A: Start-up latency

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

##### 6.2.1.3 Use Case B: Aligned Presentation

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.

##### 6.2.1.4 Use Case C: End-to-end latency

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

##### 6.2.1.5 Use Case D: Time-Shift Buffer

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.

##### 6.2.1.6 Use Case E: Infrastructure Upgrade

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 cannot 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.

##### 6.2.1.7 Use Case F: Ad Insertion

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.

## 6.2.2 Operation with MPD dynamic Mode

### 6.2.2.1 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 clause 6.2.1 is provided in clause 6.2.3.

### 6.2.2.2 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.

### 6.2.2.3 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 latest 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.

### 6.2.2.4 Consequences with Existing Technologies

The following consequences result from this basic live operation as documented in clause 6.2.2.3:

- 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 is produced. The manifest renewal is especially critical in cases where the manifest is distributed through FLUTE [8] 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 be 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 the DASH client may still be off by 10 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.

## 6.2.2.5 How does DASH solve this?

### 6.2.2.5.1 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.

### 6.2.2.5.2 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.

### 6.2.2.6 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  $k=1,2,\dots$
- 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*.

### 6.2.2.7 General Derivation

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

1. The latest Period is obtained as the Period for which  $AST+PS+MD(1) \leq NTP$ .
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  $FT + MUP$ .

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

### 6.2.2.8 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:

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

### 6.2.2.9 Addressing Methods

#### 6.2.2.9.1 Introduction

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

#### 6.2.2.9.2 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.

#### 6.2.2.9.3 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.

### 6.2.2.10 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.  $SAST(k) \leq EPT(k) + (AST + PS) + MD(k)$ ,
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.

### 6.2.2.11 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.

## 6.2.3 Mapping Use Cases to Live Operation

### 6.2.3.1 Use Case A

#### 6.2.3.1.1 Description

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

#### 6.2.3.1.2 MPD example

Below is a snippet of an MPD example. This MPD may have been distributed out-of-band and ahead of time. The DASH client when accessing the service can generally use the MPD to immediately determine the segment information.

```
<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>
```

#### 6.2.3.1.3 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:

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

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.

## 6.2.3.2 Use Case B

### 6.2.3.2.1 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.

### 6.2.3.2.2 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>
```

### 6.2.3.2.3 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.

## 6.2.3.3 Use Case C

### 6.2.3.3.1 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.

### 6.2.3.3.2 MPD example

The same as in clause 6.3.2.2 applies,

### 6.2.3.3.3 Client Procedure

The same procedure as in 6.2.1.3 and 6.2.2.3 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.

## 6.2.3.4 Use Case D

### 6.2.3.4.1 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.



#### 6.2.3.4.2 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>
```

#### 6.2.3.4.3 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 documented in clauses 6.2.1.3, 6.2.2.3 and 6.2.3.3.

#### 6.2.3.5 Use Case E

##### 6.2.3.5.1 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 cannot 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.

##### 6.2.3.5.2 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>
```

### 6.2.3.5.3 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.

### 6.2.3.6 Use Case F

#### 6.2.3.6.1 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.

#### 6.2.3.6.2 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>
```

### 6.2.3.6.3 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.

## 6.2.4 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 too coarse to operate on at the live edge).
- 4) Drift of the video source compared to UTC.
- 5) Leap seconds.

## 6.2.5 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.

## 6.3 Use Cases for Content Protection

### 6.3.1 Use Case A – Efficient Caching for Multiple DRM Systems

#### 6.3.1.1 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.

#### 6.3.1.2 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.

#### 6.3.1.3 Analysis in the Context of Rel-10 TS 26.247

TS 26.247 [2] 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.

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

### 6.3.2.1 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 issue 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.

### 6.3.2.2 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.

### 6.3.2.3 Analysis in the Context of Rel-10 TS 26.247

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 known 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>

```

### 6.3.3 Use Case C – Time-Varying Decryption Keys

#### 6.3.3.1 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.

#### 6.3.3.2 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.

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

There is no explicit support for key rotation in TS 26.247 [2] and TS 26.244 [5].

## 6.4 Fast Media Start-up

### 6.4.1 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.

### 6.4.2 Analysis in the Context of Rel-10 TS 26.247

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 [2] 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 6.2 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.

## 6.5 Advanced Trick Modes

### 6.5.1 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.

### 6.5.2 Analysis in the Context of MPEG-DASH and Rel-10 TS 26.247

#### 6.5.2.1 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].

### 6.5.2.2 Sub-Representations

### 6.5.2.3 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.

### 6.5.2.4 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.

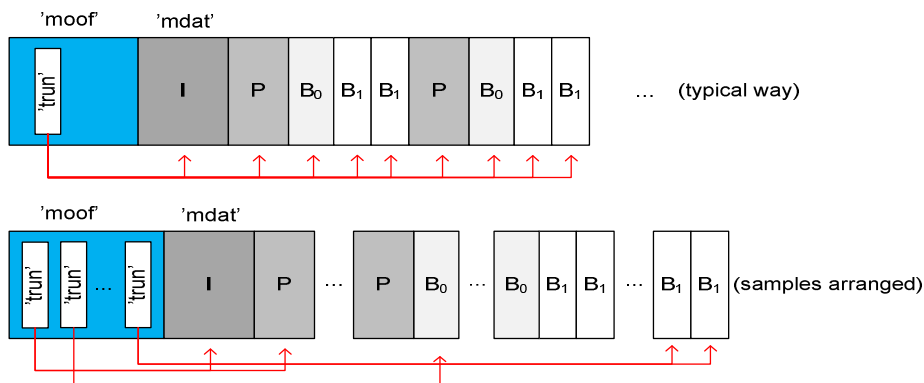
In addition the following conditions as taken from ISO/IEC 23009-3 [10] should be satisfied:

- 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 levels at the

beginning of the Subsegment. As an example Figure 6.1 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<sub>1</sub>B<sub>0</sub>B<sub>1</sub>P... in presentation order.

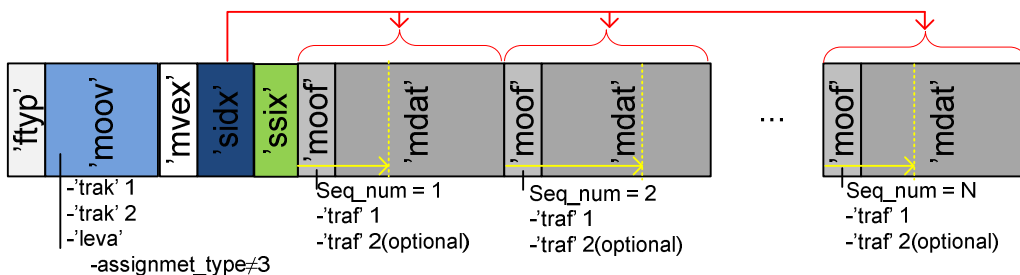


**Figure 6.1: Movie fragment format for arranged samples for easing fast forward with 'ssix' box**

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<sub>0</sub> frames, level 2 B<sub>1</sub> 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.

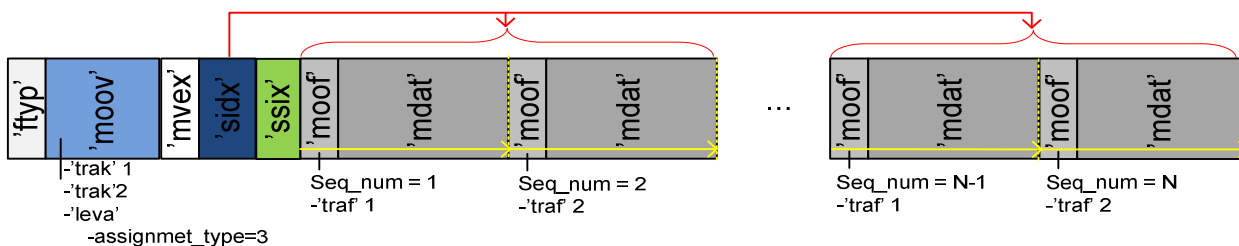
In Figure 6.2 an example of the format segment for supporting **SubRepresentations** for an assignment type other than 3 is shown. In this case the 'moof' box contains all the tracks and a Subsegment should consist of a single movie fragment. The yellow arrows and the dashed lines correspond to the position 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 6.2: Example of usage of 'ssix' box for Sub-Representations for self-initializing segment with assignment type in 'leva' box other than 3**

In Figure 6.3, 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 6.3: Example of usage of 'ssix' box for Sub-Representations for self-initializing 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.

### 6.5.2.5 Summary

Trick modes are well supported in TS 26.247.

## 6.6 Content and Device Interoperability

### 6.6.1 Use Case 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 WQVGA 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 one 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.

## 6.6.2 Working Assumption

Specific profile restrictions are aligned with common industry practice.

## 6.6.3 Gap Analysis

3GP-DASH as defined TS 26.247 lacks a profile that is aligned with existing industry practices.

# 6.7 Advanced Support for Live Services

## 6.7.1 Description

The use extends the live service use cases in clauses 6.2, 6.3 and 6.4.

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 or broadcast as for example defined in TS 26.346 [3].

## 6.7.2 Working Assumption

Tools defined for advanced live services are preferably aligned with technologies defined in other organizations, such as in HTTP/1.1 or MPEG-DASH second edition [22].

## 6.7.3 Gap Analysis

3GP-DASH as defined TS 26.247 lacks tools and guidelines for low-latency live services.

## 6.7.4 Potential Solutions

MPEG-DASH second edition [22] defines certain tools for this purpose that may be considered to improve latency.

For Live services, chunked delivery per RFC 2616 [6] in TS 26.247 may be explicitly enabled so as to support partial delivery of Segments prior to their availability.

# 6.8 Consistent QoE/QoS for DASH users

## 6.8.1 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.

## 6.8.2 Proposed Work

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

- 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.
- 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 optimization on the various components.

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

## 6.8.3 Utilization of QoS Information in DASH

The topic of QoS support for DASH services has been an active area of discussion in 3GPP SA4 since Release 10 and has resulted in specification work on the derivation of QoS mapping guidelines from the DASH MPD in 3GPP TS 26.247 (informative Annex I) [2] to be used by the application function (AF) of 3GPP Policy Charging and Control (PCC) architecture [15], [16], [17], and [18].

A DASH client may take into consideration available QoS information when requesting representations such that the consumed content bandwidth remains within the limits established by the signalled QoS information. Detailed LTE-based evaluation results on the performance benefits of utilizing QoS information in DASH client adaptation logic are provided in Annex B.

It should be noted that in the 3GPP system, PCC-level signalling can already accomplish the communication of network QoS information to the client device (user equipment or UE) and therefore the DASH client can locally (within the UE) obtain the QoS information via internal APIs.

The PCC architecture is defined in TS 23.203 [15] and provides the Rx reference point, which enables the application layer to authorize a specific usage. In this architecture the DASH HTTP streaming server or any other function in the HTTP streaming path (e.g. an HTTP proxy) can act as Application Function and interact with the PCRF via the Rx reference point for QoS control. It is assumed here that the AF has knowledge of the application type and of the MPD. The relevant AVPs are the ones enabling the PCRF to establish bearers with correct characteristics for DASH users. The AVPs are defined in TS 29.214 [18]. The further PCRF mapping from AVP to IP QoS parameter mapping is defined in TS 29.213 [17].

Figure 6.4 depicts an example PCC architecture delivering end-to-end QoS support for DASH services with the capability to interpret the media presentation description (MPD) in order to gain information on the application-layer parameters for DASH content. In the current PCC architecture, the application function (AF) interacts with the applications requiring dynamic policy and charging control. Hence, in order to provide QoS for DASH services, the AF can extract DASH content information from the MPD, map it into the appropriate attribute-value pairs (AVPs), and provide the AVPs to the policy and charging rules function (PCRF) over the Rx reference point. The PCRF combines the DASH-related AVPs received over the Rx reference point and the input received from the Gx and Gxa/Gxc reference points with user-specific policies data from the subscriber profile repository (SPR) to form session-level policy decisions and provides those to the PCEF and BBERF. In other words, the PCRF takes the subscriber information into account when setting QoS. Access-specific QoS parameters are then communicated to the UE from PCEF/BBERF. In particular, [19] describes how the UE acquires QoS information during dedicated bearer activation and bearer modification with bearer QoS update. It is also noted in [19] that "application usage of the EPS bearer QoS information is implementation dependent".

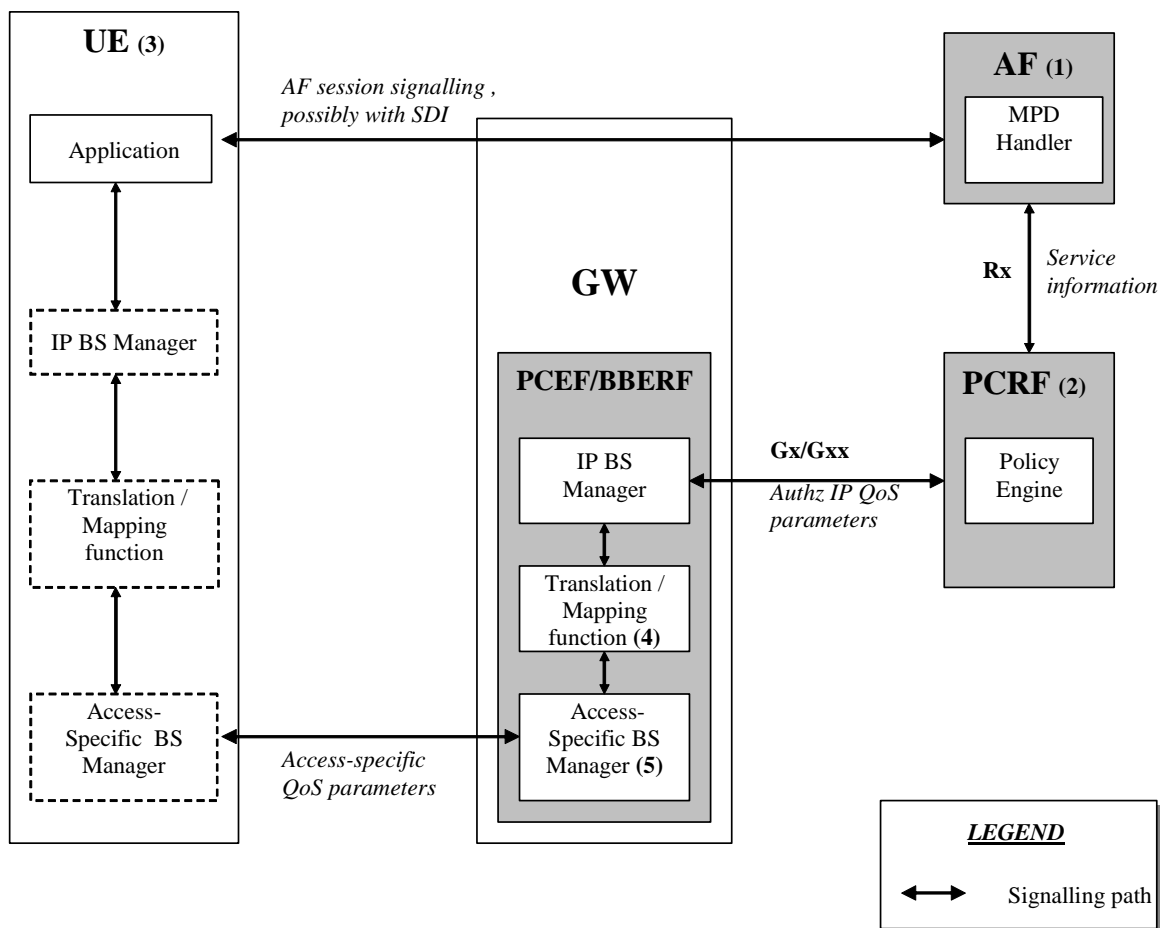


Figure 6.4: An example policy and charging control (PCC) architecture to deliver QoS for DASH services

## 6.9 DASH as download format

### 6.9.1 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 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 that 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 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>

```

## 6.9.2 Analysis against TS26.247

3GP-DASH formats are defined to support download services, especially for advanced use cases for which access to individual components of the media is relevant. However, TS26.247 in section 6.3 only permits two types of file format for progressive download. In order to support this feature, a dedicated profile may be defined:

- single period

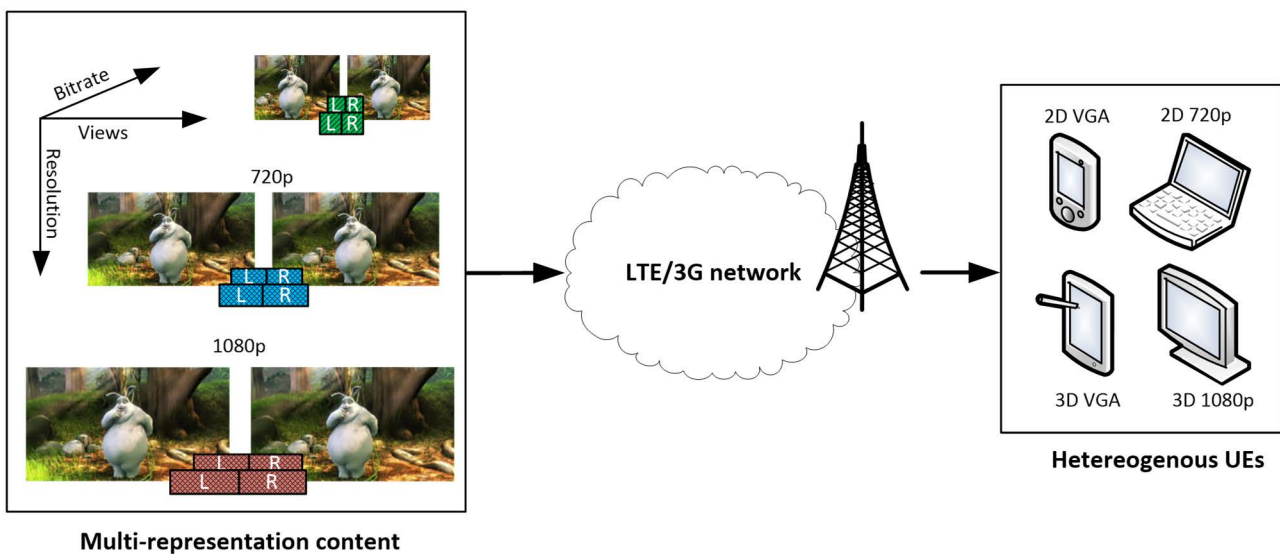
- single segment per Representation
- presence of segment index for download scheduling
- no requirements on segment alignment and IDR frames

This profile may then be added to section 6 of TS 26.247 [2] as a supported profile for progressive download over HTTP. Implementation guidelines may be added to provide details on how to use the format as download and progressive download format.

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

### 6.10.1 Description

As illustrated in Figure 6.5, 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 1080p. 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 6.5: Multi-representation content and UE heterogeneity ((c) copyright 2008, Blender Foundation / [www.bigbuckbunny.org](http://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 6.6. 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.

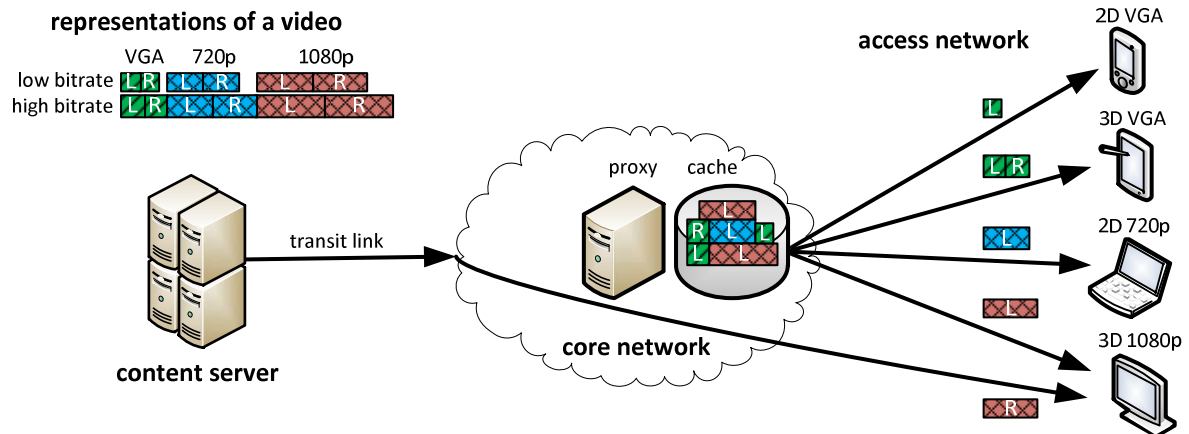


Figure 6.6: Illustration of the infrastructure for a DASH-based service

## 6.10.2 Analysis against TS 26.247

The use is fully supported since Rel-11 of TS 26.247. The example below shows a service offering that enables the use cases as documented in Figure 6.6.

```
<?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="10"/>
      <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="10"/>
      <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>
```

```
</AdaptationSet>  
</Period>  
</MPD>
```

## 6.11 Use Case: Multiple Spectator Views offered with DASH

### 6.11.1 Description

Much of the discussion on 3GP-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 WLAN RF fingerprint, WLAN and cellular fingerprint, Bluetooth beams, etc.) The determination of location and recording orientation is obviously outside the scope of 3GP-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 3GP-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 3GP-DASH.

### 6.11.2 Gap Analysis

#### 6.11.2.1 File support for timed position/location

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' [7] (a static box) is as specified in Table 6.2 below:



Table 6.2: The Location Information box

Field	Type	Details	Value
<b>BoxHeader</b> .Size	Unsigned int(32)		
<b>BoxHeader</b> .Type	Unsigned int(32)		'loci'
<b>BoxHeader</b> .Version	Unsigned int(8)		0
<b>BoxHeader</b> .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 16.14 number indicating the longitude in degrees. Negative values represent western longitude.

**Latitude:** fixed-point 16.14 number indicating the latitude in degrees. Negative values represent southern latitude.

**Altitude:** fixed-point 16.14 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 6.7 below:

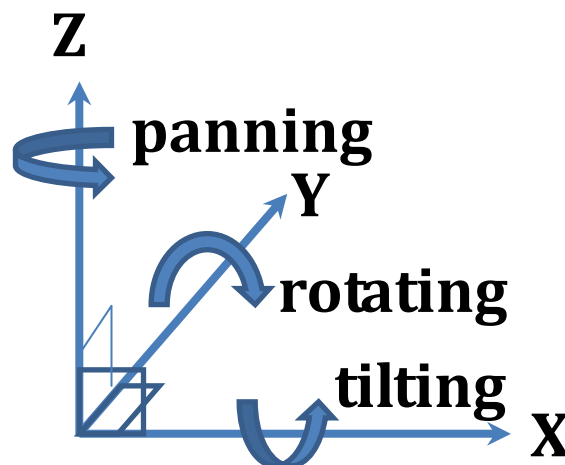


Figure 6.7: Device orientation

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 [7] 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 [7] or the 3GPP file format [5] 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.

### 6.11.2.2 MPD indication of position/orientation for a Representation or Segment

The Viewpoint Descriptor [2] 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.

### 6.11.2.3 Synchronization of media content from different devices

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 3GP-DASH specification [2] with regards to synchronization.

### 6.11.2.4 MPD indication of key highlights of the event

Because spectators are recording different views of the event, some spectators might capture a "highlight" (i.e. a particular occurrence of interest) that other spectators do not capture and vice versa. It would be useful for 3GP-DASH clients to be able to determine which of the available spectator views has captured the highlight. An example of a list of highlights that a user might be interested in for a game is shown in Table 6.3.

**Table 6.3: Example for Highlight Summary**

Highlight ID	Time	Description
0	5:15	Seattle scores first touchdown of the game
1	10:08	San Francisco Player Jones makes a great defensive play at time X in the first quarter
2	35:00	San Francisco scores on 90 yard kickoff return at time Y in the 2 <sup>nd</sup> quarter
3	1:15:00	Seattle scores on 2 <sup>nd</sup> running touchdown just before the half
.		.
.		.
.		.
15	3:15:00	Seattle scores on a Hail Mary pass to win in triple overtime

The following solution requirements are identified:

- A DASH client that has obtained information from the user about which of a particular set of highlights are of interest can use this information to determine if the highlight(s) is/are captured in the Media Presentation, and if so it can select Adaptation Sets that contain the highlight(s) without having to look into the Segments.
- A user or client is able to retrieve a full description of highlights independently from the MPD in order to minimize the size of the MPD. The MPD may contain an identifier.
- A highlight indication may be used at the MPD Level, Period Level, and/or Adaptation Set Level of the MPD to express that a highlight was captured by the offered view(s) of at least one Adaptation Set.

### 6.11.2.5 MPD indication of capturing parameters

The type of the device that recorded the media might be useful to the 3GP-DASH client. Content recorded on a device known to have lower quality hardware specifications, for example, camera quality, audio quality, etc., would be less desirable to download for a 3GP-DASH client consuming the media on a very high end device with high quality hardware specifications. For example, a user might prefer the views recorded by a television camera normally (even at lower bandwidth, and resolution), but when the television camera fails to capture a particular highlight, the client switches Adaptation Sets to a view that was recorded by a handheld device in order to view the highlight.

The following solution requirements are identified:

- A DASH client that has obtained information from the user about preferred capturing parameters, e.g. associated to the device types or the camera parameters, can use this information in selecting Adaptation Sets without having to access Representations.
- Capturing parameters may be used at the Adaptation Set Level of the MPD.

## 6.12 Use cases for operator control of video streaming services

### 6.12.1 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.

### 6.12.2 Description

#### 6.12.2.1 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 communicates with the UEs to decrease the bitrate for the video to a certain value that would allow the cell to accommodate the load.

#### 6.12.2.2 Operator control using mobile network subscription

As a variant of case described in sub-clause 6.12.2.1, 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.

#### 6.12.2.3 Operator management of video streaming service over the radio network

As a variant of cases described in sub-clauses 6.12.2.1 and 6.12.2.2, 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.

### 6.12.3 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 [23]. 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. Based on feedback from SA2 on their discussions during the work on UPCON in Rel-12 and feedback they received from RAN groups, the access to accurate information about the frequently changing radio conditions at the RAN node in a timely manner is seen as almost impossible. Information about user plane congestion at a RAN node should be expected to represent a mean value over a period of time (e.g. a

congestion level for a couple of minutes). However, at the time of the completion of this report the semantics of the congestion information has not been concluded and it was open whether this information can be standardized.

## 6.12.4 Conclusions and Gap Analysis

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

At least a certain amount of use cases may be solved by solutions introduced in clause 6.13.3.

Based on information exchange with 3GPP SA2, the use of the Sh interface from a PSS server is not restricted in TS 23.228 [24] and therefore the use cases above can be fulfilled by using the Sh interface. This is a deployment choice, and therefore is not a gap.

On bandwidth reduction in congestion information, the following aspects should be considered:

- DASH due to the nature of its design reacts to congestion by reducing the requested bitrate without affecting the service continuity based on continuous buffer state observations and bandwidth measurements. The DASH client reacts to reduced TCP throughput by switching to a lower bitrate.
- From analysis in this technical report (see Annex A) it is known that such operation can be supported by the supplying a GBR bearer at a bitrate that enables real-time delivery of at least a minimum set (typically audio and video) of lower quality Representations. Establishing a GBR bearer for DASH is enabled since Rel-10 by using the PCC architecture.
- Based on the second bullet point above if the operator decides to send DASH content over a non-GBR bearer initially, it may be useful to initiate QoS update procedure with GBR QoS parameter once the network gets more loaded. The feasibility of this option requires further clarification with SA2 experts.
- In all cases the reaction to bandwidth provisioning and bandwidth changes is expected to be done transparently by the DASH client without explicit signalling.
- According to 6.12.3 and based on feedback from SA2 on their discussions during the work on UPCON in Rel-12 and feedback they received from RAN groups, the access to accurate information about the frequently changing radio conditions at the RAN node in a timely manner is seen as almost impossible. However, in case SA2 decides to provide UPCON related signalling that is made available to Application Functions/PSS server, we may revisit the details of the signalling and analyse if this information may be useful by the PSS server or the DASH client to improve DASH operation. Considered examples are that the accessible representations under the congestion period are signalled to the DASH client in a scalable manner (see 6.13.3.4 for more discussion) or the DASH client is redirected to an alternative Representation (see 6.13.3.2 for more discussion).

## 6.13 Use Cases for DASH Operation with Network Proxy Caches

### 6.13.1 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 signalling 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 signalling 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 signalling.

### 6.13.2 Use Case Description

#### 6.13.2.1 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 channel zapping, he also prefers to fast playback startup by tolerating certain level of low and varying playback quality.

### 6.13.2.2 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. A 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.

### 6.13.2.3 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 WLAN 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 WLAN coverage. After done with shopping, they move to an outdoor cafe without WLAN coverage, but the service is continuously played by the DASH client, just with lower quality.

### 6.13.2.4 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.

## 6.13.3 Solution Analysis: Usage of TS 26.247

### 6.13.3.1 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 reflect 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 clause 6.16.4.2 w/o further specification.

### 6.13.3.2 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 currently defined in W3C as XMLHttpRequest in [12] and is also known as AJAX. In [12], redirects are to be followed automatically and transparently to the user agent. This is described in clause 4.6.5 and is replicated here:

#### 4.6.5 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.

### 6.13.3.3 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 throughput and will therefore use its rate adaptation to adjust the requested bandwidth in order to maintain proper throughput.

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

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

Typical examples are: MSS=1 460 bytes, RTT between 50 ms to 1sec, loss rate 1-e6 to 5 %.

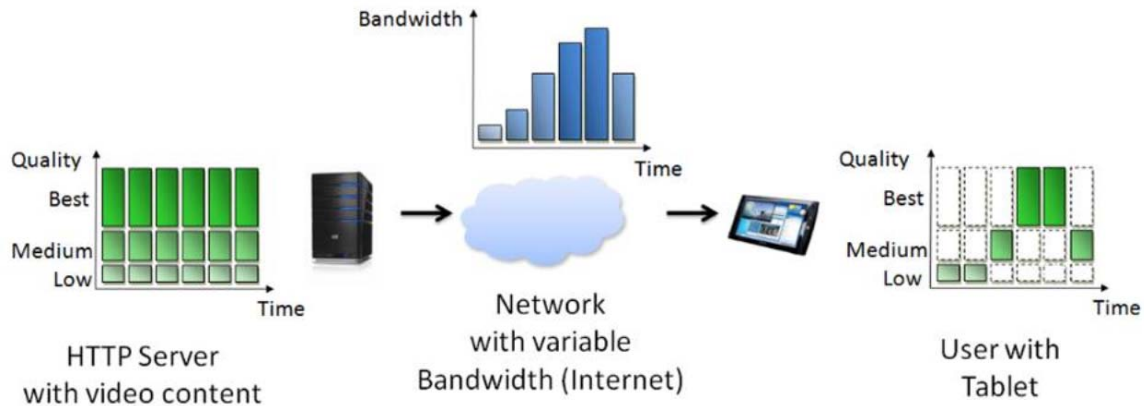
As an example, suppose the end-to-end delay of between the HTTP web caching servers and the client was 300 ms, and the packet loss rate was 1 %. Based on the TCP equation, the average throughput of a single HTTP connection would be approximately 385 Kbps.

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.

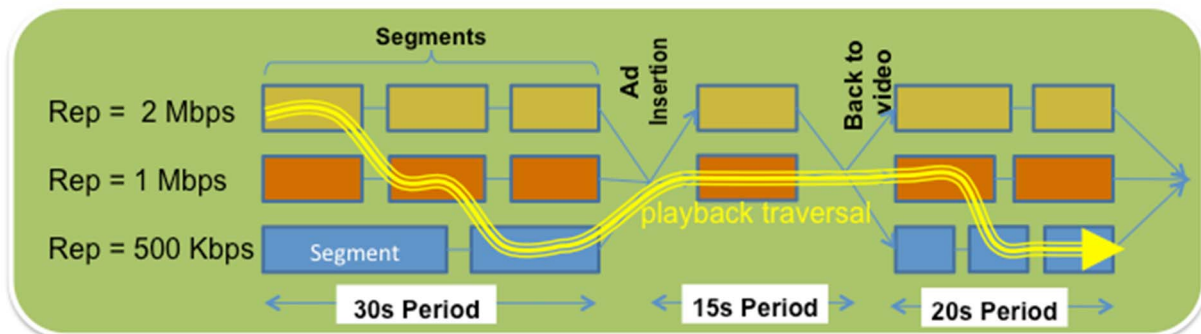
Figure 6.8 shows the operational principles of DASH-based streaming delivery. The DASH server (in general a plain HTTP server or an intermediate proxy) hosts content at different quality/bitrate levels. Due to congestion, load or other reasons, the e2e network bandwidth may be constrained. The DASH client estimates the available bandwidth and adapts the selected quality/bitrate level to the available bandwidth to ensure continues playback.





**Figure 6.8: Operational principles of DASH-based streaming**

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



**Figure 6.9: Client Selection process**

DASH-based services may be deployed w/o much impact to existing nodes and interfaces in mobile networks. According to Figure 6.8, 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 6.9.

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.

#### 6.13.3.4 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.

The second edition of ISO/IEC 23009-1 [22] was recently completed in MPEG. In this second edition the addition of media time related events is enabled. Media-time events can be signalled 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.

### 6.13.3 Gap Analysis and Problem Statement

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

- The cache operation adheres to some specific rules, e.g. that objects are cached and deleted based on popularity and previous requests.
- It is assumed that the round trip delay between UE and origin server is significantly longer than the round trip delay between UE and cache server.
- 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.

## 6.14 Services with caching of DASH content at UE functions

### 6.14.1 Use Case Description

#### 6.14.1.1 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.

#### 6.14.1.2 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 WLAN, or other Ethernet connection in the home.

#### 6.14.1.3 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.

## 6.14.2 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 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.

## 6.14.3 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.

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.

## 6.14.4 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.

## 6.15 Handling special content

### 6.15.1 Description

#### 6.15.1.1 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.

#### 6.15.1.2 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.

#### 6.15.1.3 Network/Content provider control of special content

In addition to the use case described in clause 6.15.1.2, 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.

### 6.15.2 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.

### 6.15.3 Analysis against TS 26.247: Dynamic Services

#### 6.15.3.1 General

The content provider can determine the schedule for playout for an individual client generating a dynamic service for each client. This is shown in Figure 6.10. The DASH/PSS server or some aware proxy creates a dynamic MPD with regular MPD updates for a client. By this means, the content server can at least provide certain restrictions on the playout behaviour of the client. As the content itself is unmodified, this provides a scalable way to do per client or per client group adaptation.

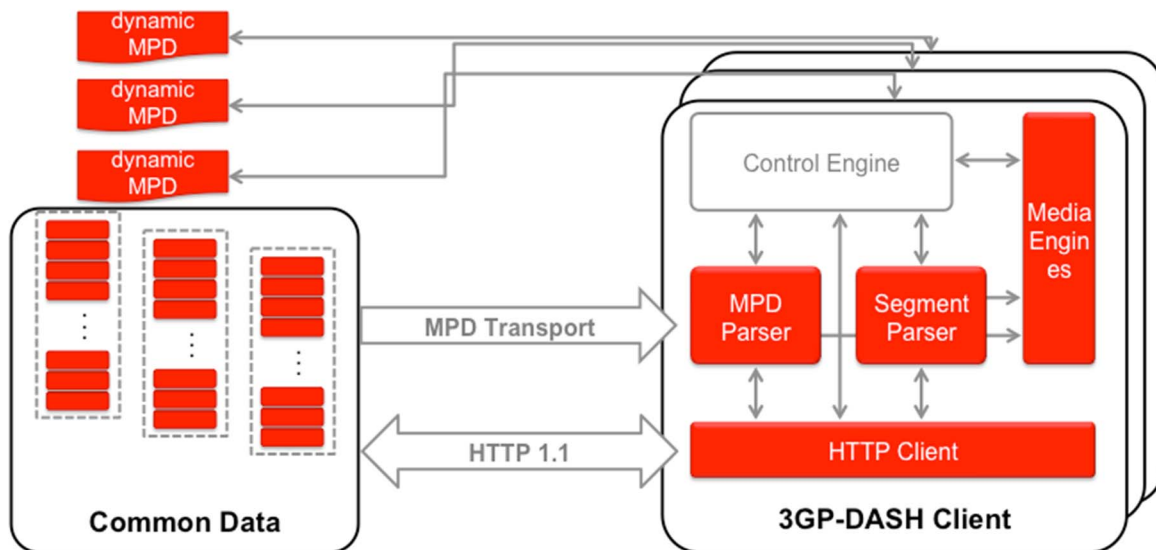


Figure 6.10: Dynamic Services for Controlling client playback

### 6.15.3.2 Tools and Evidence

In order to provide a dynamic service based on existing static content, ISO/IEC 23009-2 [11] includes a tool available here: <http://ec2-54-194-95-240.eu-west-1.compute.amazonaws.com/stattodyn/statodyn.php>. This tool allows playing static content under the control of the service provider.

DASH-IF test vectors here <http://dashif.org/testvectors> may be played using this tool.

This works as follows:

- To instantiate a dynamic service session, run the php script (e.g. a DASH client fetches it) with the following important query attributes.
- `origmpd` : location of the on-demand MPD
- `avail_start` : required start time of the live service (considered "now" if not provided)
- `tsbd` : required time shift buffer depth of the live service (considered infinite if not provided)
- Hence to instantiate a live service from, e.g. Fraunhofer MPD for test case 3a, to be available from just a few minutes ago, with time shift buffer depth of two minutes, pass the following query string to the script:  
`http://cdn.example.com/stattodyn.php?type=mpd&avail_start=1366041646&tsbd=120&origmpd=http://dash.edgesuite.net/dash264/TestCases/3a/fraunhofer/ed.mpd`
- This will instantiate the dynamic DASH service identified by a URL to which the PHP script redirects to. It also responds back to the client with the re-encoded live MPD. The redirection URL uniquely identifies a live service with specific parameters.
- The client uses this provided dynamic MPD to access segments.
- For MPD access/updates, the client is requested to use the redirected URL in step 1 above.

With little modification, the playout of a requesting client can be controlled, as the segments are only available for a short time. The content is only available for a short time shift buffer.

### 6.15.3.3 Just-in-Time Content Insertion

With the above background, dynamic services can now be used for just in time content insertion (e.g. ads, special content) by using a short time-shift buffer such that the client is forced to play the content in the short time-shift buffer.

This requires that new Periods can be added to the live content at arbitrary times. TS 26.247 supports this by MPD updates. The MPD author promises to the client that an instance of the MPD can only be used up to a certain Media

Presentation time. Before moving forward it requires to fetch a new MPD. The server may have added a new Period to a new instance of the MPD. By keeping the @minimumUpdatePeriod small, the MPD can be changed quite frequently.

### 6.15.3.4 Client Behaviour

This client behaviour is not specified in DASH. However, the service provider may track the requests for segments of the client and understand that the segments are requested in playout interval times. Only if segments for special content are requested and fully downloaded (possibly tracked by using cookies or other authentication methods), the content provider has sufficient control.

### 6.15.3.5 More complex rules

If the rules are more complex, then basically the clients instructions on playout, for example moving forward and backward, can be sent back to the server, e.g. using query parameters to MPD URLs and the client takes this into account and offers a new MPD with these parameters. The details of the query parameters may be up to the application, but may also be defined in DASH.

### 6.15.3.6 Addressing the Use Cases

#### 6.15.3.6.1 Repetitive 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.*

This use case is easily fulfilled by always splitting the content in two Periods, one that is the repetitive content and one being the new content of the episode. A client can recognize the repetitive content by identical URLs. The repetitive content may be marked with the same **Period@id**.

Example:

MPD Information	Value
MPD@type	static
MPD@mediaPresentationDuration	30minutes
Period@start	0
Period.BaseURL	"http://repetitive.com/"
Period@id	"disclaimer"
Period@start	30sec
Period.BaseURL	"http://main.com/"

No explicit gaps are identified.

However, for optimizations the following may be provided:

- define a better client behaviour, e.g. to skip repetitive content if already watched
- add the Asset Identifier from ISO/IEC 23009-1 second edition [22] for proper content marking.

#### 6.15.3.6.2 Forced Play-out

##### General

*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.*

For this purpose, a dynamic content offering with MPD updates may be provided. The content provider only provides exactly the content that is played. It will not update the MPD before the content is downloaded.

### Example 1

An example for initial playout behaviour control is shown below:

The client issues a request for the MPD at time *NOW*. The inserted disclaimer is 30 seconds. The server creates a short session for this and responds with an MPD as follows with

MPD Information	Value
MPD@type	dynamic
MPD@availabilityStartTime	NOW
MPD@minimumUpdatePeriod	30sec
MPD@timeShiftBufferDepth	infinite
Period@start	0
Period.BaseURL	"http://disclaimer.com/"

The client plays the content according to the schedule. It may want to trick the server after 15 seconds and the response is still as follows:

MPD Information	Value
MPD@type	dynamic
MPD@availabilityStartTime	NOW
MPD@minimumUpdatePeriod	15sec
MPD@timeShiftBufferDepth	infinite
Period@start	0
Period.BaseURL	"http://disclaimer.com/"

Finally after having played the media for 30 seconds, i.e. it gets to the end of the data, it requests a new MPD, which looks as follows:

MPD Information	Value
MPD@type	dynamic
MPD@availabilityStartTime	NOW
MPD@mediaPresentationDuration	30minutes
MPD@timeShiftBufferDepth	infinite
Period@start	0
Period.BaseURL	"http://disclaimer.com/"
Period@start	30sec
Period.BaseURL	"http://main.com/"
Period.BaseURL@availabilityTimeOffset	30 minutes

At this point the server has given up control and the client can consume the on-demand content.

#### 6.15.3.6.3 Flexible Ad Insertion

*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.*

The client issues a request for the MPD at time *NOW*. The inserted ad is 30 seconds. The server creates a short session for this and responds with an MPD as follows with



MPD Information	Value
MPD@type	dynamic
MPD@availabilityStartTime	NOW
MPD@minimumUpdatePeriod	7sec
MPD@timeShiftBufferDepth	infinite
Period@start	0
Period.BaseURL	"http://ad.com/"

After having played the media for 7 seconds, i.e. it gets to the end of the data, it requests a new MPD, which looks as follows:

MPD Information	Value
MPD@type	dynamic
MPD@availabilityStartTime	NOW
MPD@mediaPresentationDuration	30minutes
MPD@timeShiftBufferDepth	infinite
Period@start	0
Period.BaseURL	"http://ad.com/"
Period@start	30sec
Period.BaseURL	"http://main.com/"
Period.BaseURL@availabilityTimeOffset	30 minutes

At this point the server has given up control and the client can consume the on-demand content.

*In another example, the ad needs only to be watched once and the user can scroll back and forth without having to see an ad for e.g. 30 minutes.*

This is more tricky. It is not clear if the use case means:

- you want to not see it anymore even if you go back; or
- you are allowed to skip the ad but generally you still watch it.

If it is the latter, i.e. skipping only is to be controlled, then one needs to maintain control over the client playout behaviour by continuously updating the MPD, e.g. like this:

MPD Information	Value
MPD@type	dynamic
MPD@availabilityStartTime	NOW
MPD@minimumUpdatePeriod	0
MPD@timeShiftBufferDepth	15 seconds
Period@start	0
Period.BaseURL	"http://ad.com/"
Period@start	30sec
Period.BaseURL	"http://main.com/"

The rewind in the content is actually then handled as follows. Now the client does a rewind on the application level. The DASH client sends back an update request for the MPD with the rewind point being the start before the 30 minutes and at some time NOW+LATER. This is the response.

MPD Information	Value
MPD@type	dynamic
MPD@availabilityStartTime	NOW
MPD@minimumUpdatePeriod	0
MPD@timeShiftBufferDepth	15 seconds

<b>Period@start</b>	30sec
<b>Period.BaseURL</b>	"http://main.com/"
<b>Period@start</b>	LATER
<b>Period.BaseURL</b>	"http://main.com/"

Now the rewind is after 30 minutes at time NOW+MUCHLATER. This is the response:

<b>MPD Information</b>	<b>Value</b>
<b>MPD@type</b>	dynamic
<b>MPD@availabilityStartTime</b>	NOW
<b>MPD@minimumUpdatePeriod</b>	30sec
<b>MPD@timeShiftBufferDepth</b>	15 seconds
<b>Period@start</b>	30sec
<b>Period.BaseURL</b>	"http://main.com/"
<b>Period@start</b>	MUCHLATER
<b>Period.BaseURL</b>	"http://ad.com/"

If the use case targets the earlier, there are some gaps. However with xlink and a re-resolution as defined in MPEG-DASH [22], this may be achieved. Re-resolution will result in resolving this only to an ad if the user operates within the 30 minute time window.

### 6.15.3.4 Gap Analysis and Recommendations

For controlling playout and handling special content, TS 26.247 provides basic tools using dynamic services. However, the approach document in section 3 and 4 has limitations in terms providing a consistent operation. At least the following issues and gaps are identified:

- Controlling the playout requires a customized MPD with regular updates for every client request. Despite the distribution of segments may be scalable, it still results that the MPD is individual and cannot be cached.
- The DASH client is not aware that the control of the playout is due to on segment requests and consumption of the content, but assumes that the availability of the segments depends only on time advancing. Therefore controlling playout may confuse the client and additional information is necessary.
- Consistent and trusted client behaviour including authentication, authorization and session control. However, despite this may be considered relevant, it may not be in scope of 3GPP SA4 to work on these aspects.
- The method cannot be fool-proof unless tracking of all client requests and verification of the client behaviour (which can be done at different levels, ranging from download to decoding and possibly also rendering) is implemented.

## 6.16 Use Cases on DASH Authentication

### 6.16.1 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.

### 6.16.2 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.

### 6.16.3 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.

### 6.16.4 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.

### 6.16.5 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 should 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.

## 6.16.6 Gap Analysis on the Use Cases

### 6.16.6.1 Analysis of Use Cases 1-3 on content access authorization

The Generic Bootstrapping Architecture (GBA) [20] can be used to authenticate the end-user(s) and bootstrap a secure end-to-end connection between the Streaming Server and the end-user(s). It can be used as a basis for user-based security. The Streaming Server as Network Application Function (NAF) may or may not be part of the operator network.

The Generic Bootstrapping Architecture currently does not contain any content access authorization methods for DASH. However, authorization of content access and streaming quality as described in Use Cases 1 to 3 can be additionally implemented in the Streaming Server that takes on the role of NAF. This is an area that SA4 could address in future normative work.

### 6.16.6.2 Analysis of Use Cases 4-5 on application-level content/metadata integrity validation:

Regarding Use Cases 4 and 5, the Generic Bootstrapping Architecture can be used to derive the necessary shared secret(s) required to integrity protect the application level content/metadata between the UE and server serving this DASH content. No further work at SA4 level is required to meet these use cases.

## 6.17 Consistent Quality for DASH users

### 6.17.1 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.

### 6.17.2 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 6.11 and 6.12 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 6.4. The content is encoded into eight representations and the segment length is fixed to 2 seconds (only two representations are depicted in Figures 6.11 and 6.12, 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 6.11 and 6.12, 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).

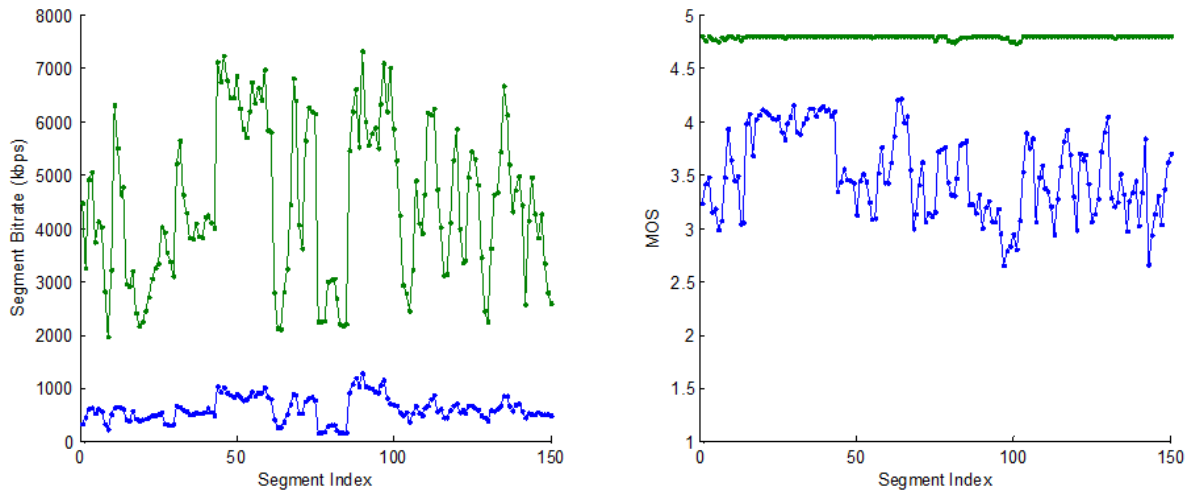
**Table 6.4: Encoding Parameters in x264**

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

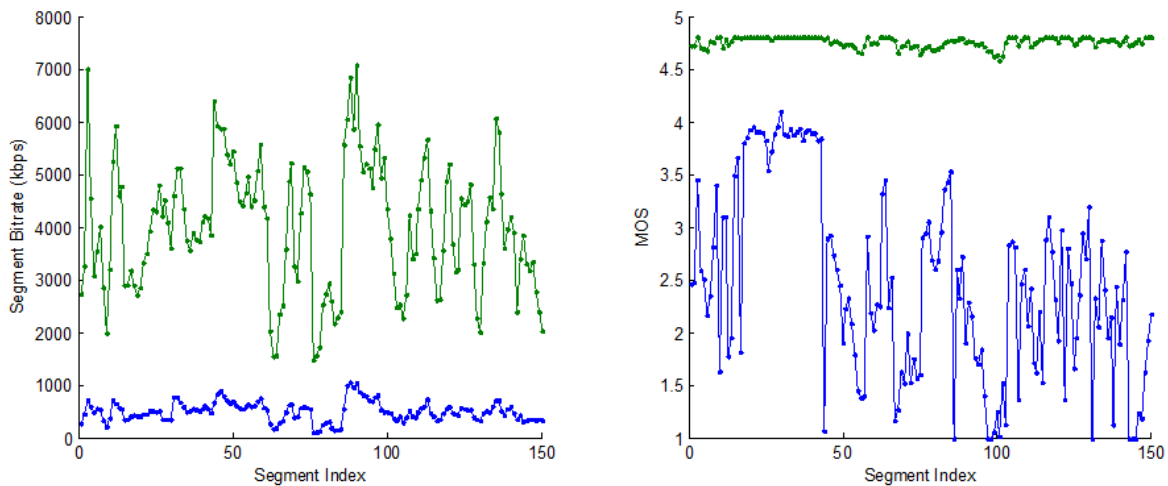
Consequently, DASH-formatted content for a given representation or sub-representation may vary in quality, which is not revealed by the existing @bandwidth attribute or 'sidx' information available at the client. Some potential implications of this may be that it may not be possible to maintain a consistent 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 6.11: Per-segment bitrate and quality for unconstrained VBR**



**Figure 6.12: Per-segment bitrate and quality for constrained VBR**

### 6.17.3 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 agreed (at MPEG#104 in July 2013) toward addressing this use case and a new specification has been created for this purpose [28] toward defining formats for carriage of timed metadata metrics of media in ISO base media file format. 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.

## 6.17.4 Additional Simulation Results on DASH over GBR and Non-GBR bearers

The simulation assumptions and results on evaluation of end-to-end DASH users' experience over an LTE network are showed in this section.

Assumptions for LTE are listed in Table 6.5.

**Table 6.5 LTE System Simulation Assumptions**

Parameter	Explanation/Assumption	Comments
Cellular layout	Hexagonal grid, 3-sector sites	
Site to Site distance	500 m	
Antenna pattern	0 degree horizontal azimuth is East 70 degree (-3 dB), 20 dB front-to-back ratio	
Propagation model	$L = 128.1 + 37.6 \text{ Log}_{10}(R)$	R in kilometres
Penetration Loss	20dB	
Std. deviation of slow fading	8.0 dB	Log-Normal Shadowing
Correlation between sectors	1.0	
Correlation between sites	0.5	
Carrier frequency	2000 MHz	
Thermal noise density	-174 dBm/Hz	
Max. # of retransmissions	3	Retransmissions by fast HARQ. Does not include the initial transmission.
Node B antenna gain plus Cable Loss	14 dBi	
Node B noise figure	5 dB	
UE antenna gain	0 dBi	
UE noise figure	9 dB	
BS total Tx power	46 dBm for 10MHz, 43 dBm for 5 MHz	
UE power class	23 dBm (200 mW)	This corresponds to the sum of PA powers in multiple Tx antenna case

Additional assumptions include: downlink transmission scheme - 1x2 SIMO; downlink HARQ – CC; downlink receiver type – MRC; and channel model - SCM urban macro high spread for 3GPP case 1 [21].

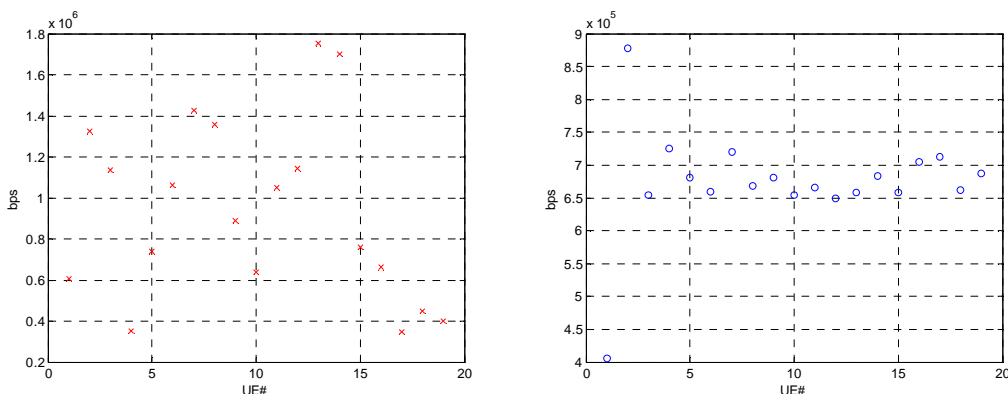
19 UEs in one sector share all the 10MHz frequency band, 2 of them run in the full buffer mode as background traffic while others enjoy DASH services. All DASH UEs are assumed to get initial data buffer of 10 seconds' video segments beforehand. For non-GBR simulation case, UEs choose VLC's adaptation strategy while for GBR simulation case, UEs request segments with the fixed rate of 600kbps. The whole simulation runs for 600 seconds.

For TCP congestion control, the algorithms in RFC 5681 [35] are simulated. Time delay from Server to eNB is assumed to be fixed 20ms. No packet is dropped during the transmission from Server to eNB. The buffer size in eNB is set to be large enough so that drop tail becomes unnecessary.

The original video clip is encoded into 4 AVC/H.264 representations, some of whose parameters are listed below. The quality of experience in terms of MOS score is calculated, in the unit of segment length of 2 seconds, based on the QoE model introduced in ITU-T P.1202.1 [33] specification. Evaluation of original MOS scores considers encoding parameters and average MOS scores for each representation are listed in Table 6.6. At the receiving terminal, evaluation of MOS scores takes both encoding parameters and re-buffering time into account for each segment and it will be shown in the simulation results.

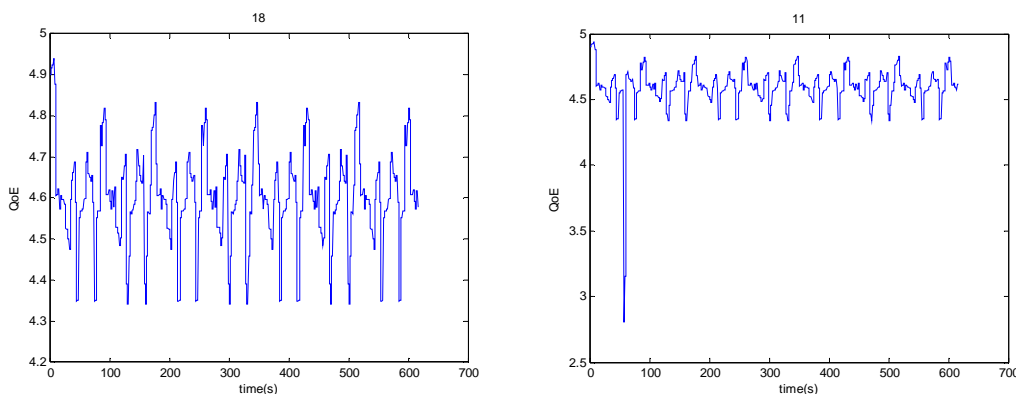
**Table 6.6: Simulation Results**

Average Bitrate	Resolution	Frame Rate	Average Quantization Parameter	Average MOS
400k	480x320	24	40.73	4.35
600k	480x320	24	38.32	4.51
800k	480x320	24	36.36	4.64
1M	480x320	24	34.62	4.73



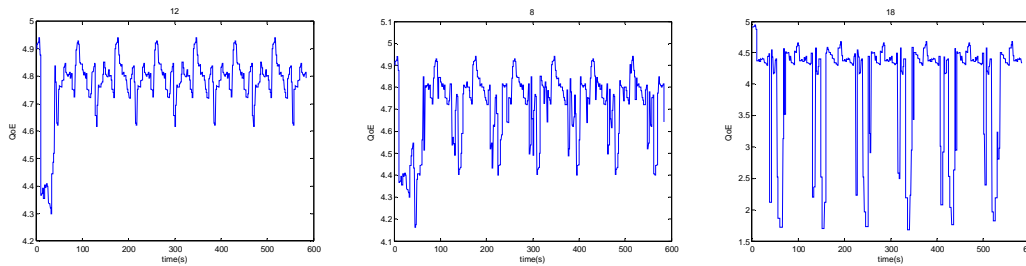
**Figure 6.13: Mean Data Rates for 19 UEs with non-GBR and GBR Bearers**

Figure 6.13 shows the mean data rate for each UE associated with the non-GBR bearer (left) and GBR bearer (right), separately, based on simulation data of 600 seconds. In the right picture, UE #1 and #2 are set to the full-buffer mode (non-GBR). Every other UE achieves a data rate of at least 640 kbps with the GBR setting to over 600 kbps.



**Figure 6.14: QoE for UEs with different GBR settings**

In Figure 6.14, the QoE of UE#18 with GBR setting is shown here. It is observed that most UEs have consistent QoE as UE#18 has. There is only one UE, which is UE #11 depicted in the right picture, experiencing QoE degradation in a very short time period due to re-buffering. It is noted that UE #11 has a data rate of over 600 kbps at most of time.



**Figure 6.15: QoE for UEs with Non-GBR bearer**

In Figure 6.15, videos streamed to all UEs are carried by the Non-GBR bearer. 17 UEs request video segments with appropriate rate using VLC's adaptation strategy as stated before. Among them, 8 UEs experience consistent QoE similar to that shown in the left picture which is the result for UE #12 as a representative, 6 UEs experience QoE as shown in the middle picture represented by the of UE #8; Those 6 UEs play segments at 400 kbps in a large part of the time. There are 3 out of the 17 UEs, e.g. UE #18, can't maintain the lowest data rate, as shown in the right picture. For UEs moving around in the cell, it is observed that they cannot experience any consistent quality when served over the non-GBR bearer

As shown in above simulation figures, observation is summarized below:

- QoE of DASH service can be maintained if DASH service is carried over GBR bearer. Existing QoS mechanism can be used to support consistent QoE of DASH service.
- Considering the current EPC architecture supports QoS feature, consistent QoE of DASH service is supported already from 3GPP architecture point of view.

## 7 Advertisement Insertion Use Cases

### 7.1 Introduction

An ad insertion system can be described in terms of three concepts:

- cues: indicate the locations of ad breaks in the media, which are known at content generation time.
- triggers, information is a cue description, containing at the least some information identifying the upcoming ad break and its start time.
- ad decision: a process typically done in real time (i.e. close to the ad break start time) by an ad server given trigger information. The result of ad decision is the actual content filling the ad break,

Thus ad insertion can be completely described as a simple two-stage process, with an ad break scheduled for media time  $t$  at content generation time, and ad content for this break determined slightly prior to the absolute wall-clock time  $T$  of to which media time  $t$  is mapped.

### 7.2 Use Cases

#### 7.2.1 Use Case 1: Fixed duration advertisement in On-Demand Content

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.



## 7.2.2 Use Case 2: Variable duration advertisement in On-Demand Content

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.

## 7.2.3 Use Case 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.

## 7.2.4 Use Case 4: 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.

## 7.2.5 Use Case 5: Accessing Live Content

A service provider offers a free live service. The content owner has a business relation with service provider, allowing the latter to insert ads in locations marked by the content owner at the content generation time. 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 (or its continuation) is provided.

# 7.3 Working Assumptions

## 7.3.1 Introduction

Two types of ad insertion architectures are considered

- 1) Single Media Presentation: In this case the DASH client controls the access and playout of both, the main content and the inserted ad. Assumptions are provided in section 7.3.3.
- 2) Presentation Layer controlled Ad Insertion: In this case a presentation layer controls the access and playout of both, the main content and the inserted ad and DASH formats are used for the distribution of the different scenes. Assumptions are provided in section 7.3.4.

## 7.3.2 Generic assumptions

- 1) Independent of the ad insertion architecture, the following assumptions apply.
- 2) Some main content is augmented with advertisements.
- 3) Ads may be general or may be targeted/personalized/individual
- 4) The main content may be live. For live services,
  - a) the cues as well as the ad insertion decision is generally done in real-time
  - b) ads breaks are of a pre-defined duration. If different ads are inserted for different viewers, individual ad lengths may differ, but the break duration is still the same for all viewers.

- 5) The main content may be On-Demand. For On-Demand services, inserted ad breaks and ads may be of the same or different duration.
- 6) The main content may be On-Demand that is converted from live content without changing the segments and segment URLs.
- 7) The distribution supports scalability. In addition, the ad insertion opportunities are part of the content as sent to the client. The ad decision is case-by-case and may differ if an ad is added at all as well as the content and the duration.
- 8) Ad decisions may be done close to the time when the content is consumed
- 9) Some default content may be added if no ad is provided or the ad decision decides to not provide. The default content is typically provided by the producer of the main content.
- 10) Live content may be mixed with pre-stored advertisement.
- 11) The ad provider may apply logic on whether the ad can be skipped, or when it is skipped based on information from the DASH client.
- 12) Some tracking of the ad consumption is provided. Tracking information will be made available to the advertisers, either by the operator or by an external entity.
- 13) Different ads may be displayed at the same break for the same user when the same point in an On-Demand content is reached again.

### 7.3.3 Single Media Presentation

Here are working assumptions for use case 1, 2, and 4 based on a single Media Presentation:

- 1) The MPD is the entry point in the service, i.e. ads are inserted in a continuous Media Presentation.
- 2) The functionality that is provided is not an ad insertion system, but functional tools for the DASH client that may be used for other purposes.

### 7.3.4 Presentation Layer controlled Ad Insertion

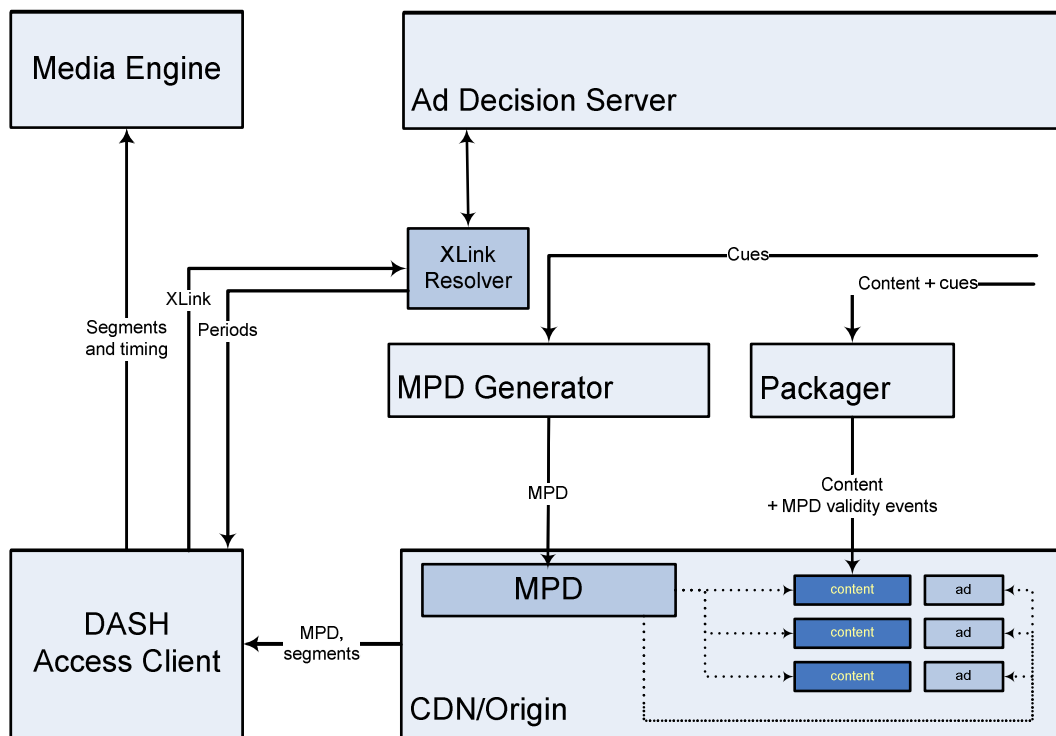
A part of the presentation layer in this case is an application (app) which can control the DASH client. Examples of such an app can be a JavaScript script in HTML5, or a native app running on a smartphone.

- 1) The entry point to the service is a presentation layer. The live service and the advertisement may be different MPDs.
- 2) An app is able to control multiple DASH clients. This implies the existence of an underlying DASH client API, which can be instantiated with an MPD and (optionally) session state.
- 3) A complete presentation may be described by more than one MPD. One possible case would be main content having a single MPD, while inserted ads having their own MPDs or a single MPD for a complete ad break.
- 4) DASH is used as means of triggering ad decision (and consequent context switch). It is possible to preload a complete out-of-band schedule of ad breaks at the beginning of VOD playback..
- 5) The app and the client have been vetted by the operator and/or by the content owner. It is possible to verify that both app and client are the trusted ones.

## 7.4 Analysis in the Context of TS 26.247

### 7.4.1 Single Media Presentation

In the following the set of use cases and working assumptions for use case 1-3 are analysed against the existing TS26.247 3GPP-DASH specification and to some extent also the MPEG-DASH specification. A baseline architecture is provide in Figure 7.1.



**Figure 7.1: Server-based architecture**

In this model, all ad-related information is expressed via MPD and segments, and ad decisions are triggered by client requests for MPDs and for resources described in them (segments, remote periods).

Possible Solutions, gaps and optimizations potentials are identified.

1) *The MPD is the entry point in the service, i.e. ads are inserted in a continuous Media Presentation.*

MPEG-DASH provides this by offering a single MPD which may contain different content. The content is "concatenated/spliced" into a single time line which provides exact timing of the contained media. The exact details on what "concatenated/spliced" means is discussed in the following.

No gaps are identified.

2) *The functionality that is provided is not an ad insertion system, but functional tools for the DASH client that may be used for other purposes.*

The solution looks into methods on how to properly "splice" content. Splicing may be done for other purposes as well and may be used by the service provider without an ad insertion system on the backend.

No gaps are identified.

3) *Some main content is augmented with ads.*

To support simple splicing/concatenation, DASH supports the following functions:

- **Periods:** This allows interrupting main content with content that is completely differently encoded, has different DRM, is accessible at different resources etc.
- **Presentation Time Offset:** specifies the presentation time offset of the Representation relative to the start of the Period. This allows that main content can be interrupted quite arbitrarily and can be continued at a new Period where the internal presentation time of the media that corresponds to the start of the Period is signalled in the value of the @presentationTimeOffset attribute. However, the addition of a Period may not necessarily result in a seamless playout.

The following gaps and optimization opportunities are identified:

- The continuation of content over Period boundaries and the association to main content and ad content is not possible. A solution consideration is the Asset Identifier in the second edition of MPEG-DASH [22]. The

identifier is added on Period level that allows signalling that content in two different Periods belong to the same asset.

- Seamless playout across period boundary and efficient multi period content offering is not yet supported, but is under discussion in further amendments in MPEG.
- The content ought be DASH formatted in order to be presented in a DASH Media Presentation.

#### 4) *Ads may be targeted/personalized/individual*

To support personalized and individual content, a request for a specific resource may result in different responses for different users. In an DASH context this is supported by the following functions:

- When making the HTTP GET call the client may associate a previously delivered Cookie with the request, based on the URL path as defined in RFC 2109 [32].
- This Cookie for example may provide a client identifier.

Cookie support is apparently now a required capability, to be an HTTP user agent. This comes from the relatively recently published RFC 6265 [31]. An excerpt is as follows:

"Origin servers MAY send a Set-Cookie response header with any response. User agents MAY ignore Set-Cookie headers contained in responses with 100-level status codes but MUST process Set-Cookie headers contained in other responses (including responses with 400- and 500-level status codes). An origin server can include multiple Set-Cookie header fields in a single response. The presence of a Cookie or a Set-Cookie header field does not preclude HTTP caches from storing and reusing a response."

Cookies may be used in the MPD requests and then an individual MPD is maintained for each different request. Therefore, personalized content offering can be provided, but the individual MPDs may still have common parts, namely the main content.

The following gaps and optimization opportunities are identified:

- It may be necessary that a system requires the support of RFC 6265 [31]. It needs to be analysed if anything needs to be added in TS 26.247. Other aspects that enable personalization/targeting may be considered.
- While cookie processing ability is a requirement, the time the cookie is stored is not normatively defined and is allowed to be shorter than the intended time. Alternative means for client / session identification may need to be investigated.

#### 5) *The main content may be live. For live services,*

- *the cues as well as the ad insertion decision is generally done in real-time*
- *ads breaks are of the same duration for same content viewed by all viewers, however different viewers may see different ads and have different ad durations within this break.*

The first bullet point requires that new Periods can be added to the live content at arbitrary times during the content distribution. DASH supports this by:

- MPD updates: The MPD author promises to the client that an instance of the MPD can only be used up to a certain Media Presentation timeline. Before moving forward it requires to fetch a new MPD. The server may have added a new Period to a new instance of the MPD. By keeping the @minimumUpdatePeriod small, the MPD can be changed quite frequently. The updated MPD should be fetched by the client early enough to allow for ad decision and buffering, in order to prevent rebuffering and service disruption at the transition from main to inserted content. Frequent poll-based MPD updates may be costly for some environments.
- If MPDs would be used that are of personalized, then the inserted ad breaks need to be of the same or at least very similar duration in order to maintain that all clients playout and request the data at the same time. This is essential to ensure that the segments of the produced live content are available at the announced times in the MPD. If the same `Period@start` value is used each MPD, then for Periods of slightly different durations the note in TS 26.247 applies:

- At the start of a new Period, the playout procedure of the media content components may need to be adjusted at the end of the preceding Period to match the *PeriodStart* time of the new Period as there may be small overlaps or gaps with a Representation at the end of the preceding Period. Overlaps (respectively gaps) may result from Media Segments with actual presentation duration of the media stream longer (respectively shorter) than indicated by the Period duration. Also in the beginning of a Period if the earliest presentation time  $T_P$  of any access unit of a Representation is not equal to  $T_O$  then the playout procedures need to be adjusted accordingly.

The following gaps and optimization opportunities are identified:

- An efficient method for MPD updates. A possible solution are inband event messages with MPD validation expiry events signaled in the event message box as available in second edition of MPEG-DASH [22].
- An efficient and continuous playout mechanism across Period boundaries.

6) *The main content may be On-Demand. For On-Demand services, inserted ads may be of the same or different duration.*

TS 26.247 supports this by:

- the features documented above for inserting main content
- the ability to use **Period@duration** for Periods not being the first one. This enables that a concatenation of On-Demand content can be done by just concatenating Periods in one MPD with different duration.

The following gaps and optimization opportunities are identified:

- the **MPD@mediaPresentationDuration** or the **MPD@minimumUpdatePeriod** should be present which does not take into account that the duration of the Media Presentation can be deduced from the last Period duration, if present. This has been addressed in the second edition of MPEG-DASH [22].

7) *The main content may be On-Demand that is converted from live content without changing the segments and segment URLs.*

MPEG-DASH supports this by one of the following features:

- by designing a new MPD of type static, using the same Segment URLs as in the live distribution and by using the technologies above.
- by changing the live MPD to type static, but then the same advertisement is available that was also available in the live content.

No gaps are identified.

8) *The distribution supports scalability. In addition, the ad insertion opportunities are part of the content as sent to the client. Whether an ad is added and the duration of the ad is decided case-by-case.*

Due to the fact that the Ads are personalized for each session, this means that the Headend should maintain unique MPDs for each client session, which is unsalable and inefficient.

DASH support efficient handling of this issue by the support of xlink. This means that certain part of the content may be offered in one MPD is common and provided directly, whereas content that is specific to a subset of users may be provided only after requesting the content. xlink is supported to resolve content contained in the following elements:

- Period
- AdaptationSet
- SegmentList

Xlink supports two different processing models: onLoad and onRequest. For onLoad, the following is defined in TS 26.247.

- onLoad: an application should dereference the remote element immediately on loading the MPD.

Specifically, the aspect of personalized ads can be fulfilled by using Periods with xlink and the resolution of the xlink is done using the technologies mentioned above, for example cookies.

When using xlink and personalized ads, the following aspects of the previous cases are also fulfilled:

- for live service the content provider adds a Period with xlink at the time when an ad is due and uses as processing model onLoad, i.e. the DASH client immediately requests the xlinked resource.
- for On-Demand service the content provider adds a Period with xlink at each position at which possibly an ad may be inserted and uses processing model onLoad. In addition it only uses **Period@duration** for all Periods that are present in the MPD. When the DASH client loads the MPD, it also resolves all xlink and may get returned a Period with zero duration (empty), one Period or multiple Periods. These Periods also only include **Period@duration** attributes.
- for Live to On-Demand converted data, the same applies as for the previous two cases.

There are cases when http resources are not accessible. Error codes and error cases are defined in section A.7 of TS26.247, but not for xlink.

Based on this, the following gaps and optimization opportunities are identified:

- define the reaction to status codes when an xlink fails, especially fixing the status that an invalid xlink necessarily invalidates the MPD.
- clarify the exact resolution strategies when using xlink

Both aspects are addressed in the second edition of MPEG-DASH [22].

#### 9) *Ad decision close to the time when the content is consumed.*

In this case, the ad decision server wants to provide an ad close to the time when the content is consumed. This may be done in a personalized or non-personalized fashion, e.g. taking into account the time when the resolution is requested.

TS 26.247 provides the following features for this:

- Using a dynamic service, possibly even for VoD content in a personalized manner. Then ads can be inserted close to the time when the content is consumed. However, such an approach is not scalable for VoD and requires heavy involvement of the server.
- For live services with a small time-shift buffer depth the xlink+onLoad based solution mentioned above may be used.

The main issue is that the xlink for onLoad is expected to be resolved at the time when the MPD is loaded.

However, using the onRequest method seems more suitable. For onRequest, the following is defined in TS 26.247.

- onRequest (default): formally, an application should dereference the remote element only on a post-loading event triggered for the purpose of dereferencing. In the context of this Part of ISO/IEC 23009, the application dereferences the link only for those resources it needs (or anticipates it probably will need). Examples include dereferencing a link in a **Period** element when the play-time is expected to enter that Period, dereferencing an Adaptation Set link when it appears to contain Representations that will be needed, and so on.

This means that when an xlink is added, be it in On-Demand or Live services, the xlink is only executed when the play-time is expected. The use case is fulfilled in principle, but the following clarification aspects are identified.

- 1) how can this behaviour be ensured. This is discussed in more details below and is independent of the signal
- 2) what happens when the client visits the same content again?

Both aspects are addressed in the second edition of MPEG-DASH [22].

#### 10) *Live content may be mixed with On-Demand Advertisement.*

Despite the content may be live and segments are offered in a dynamic fashion, interleaved ad Periods may be On-Demand. Despite Segments are available earlier than their regularly computed availability start time, there is no

problem to offer the On-Demand ads as dynamic content. However, clients will only access the content at the time earliest at the computed availability start time.

As a gap analysis,

- the availability of On-demand ads cannot be signalled in TS 26.247. A possible solution is provided in the second edition of MPEG-DASH [22] using the `availabilityTimeOffset`.

11) *The ad provider may apply logic on whether the ad can be skipped, or when it is skipped based on information from the DASH client.*

The logic available in TS 26.247 is that ads are treated as regular content and can be skipped. In addition, it is assumed that the same content is played in the time shift, as the URLs are the same. However, there may be cases where the content provider offers content only if the DASH clients playout is obeying certain rules based on signals in the MPD.

TS26.247 does not currently define mechanisms like this, but may be improved by two orthogonal concepts:

- signaling of simple playout instructions in the MPD for certain content (e.g. do not skip, do not rewind, etc.)
- client authentication, the client authenticates that it will for example obey the playout instructions

12) *Some tracking of the ad consumption may be provided.*

DASH inherently supports tracking by server-side tracking of the HTTP requests. However, this will allow you to only know if a segment has been retrieved but it does not allow you to determine if the segment was viewed. It also does not allow you to track if the segment was viewed in realtime etc. Hence, this is rather "poor mans reporting" and optimizations may be considered.

13) *Different ads may be displayed at the same break for the same user when the same point in an On-Demand content is reached again. In some cases ad break may be skipped altogether.*

This feature is not supported in TS 26.247. A possible solution is provided by MPEG-DASH second edition [22] allowing `@xlink:href` attribute to be returned when a remote Period is dereferenced.

## 7.4.2 Presentation Layer controlled Ad Insertion Analysis

1) *DASH will be used as means of triggering ad decision (and consequent context switch).*

While it is possible to preload a complete out-of-band schedule of ad breaks at the beginning of VOD playback (this is how IAB VMAP [25] operates), however this is impractical in a continuous live service due to unpredictability of ad break start. In existing adaptive streaming systems trigger information is passed by embedding SCTE 35 [26] cue messages, either as XML in an ISO-BMFF box in a media segment, or in as a tag in the manifest. In both cases this is a proprietary extension defined by content producers and operators, rather than enshrined in the specifications themselves.

In DASH case, trigger information can be passed using user-defined DASH events, which are defined in ISO/IEC 23009-1 [22]. In a media-centric approach, all information needed to trigger ad decision is expressed in the media segments themselves. This way, inband ``emsg`` boxes will carry trigger information. The ``emsg`` event timing would indicate the intended splice time in media time. In an MPD-centric workflow MPD Validity Expiration event is inserted into a media segment. This event triggers an MPD update, and rather than an ad decision, hence the Validity Expiration event should appear early enough to allow for both MPD update and ad decision.

No gap is identified in MPEG DASH, however use of events needs to be added to 3GPP-DASH.

In interests of interoperability it may be worth to specify transport of SCTE 35 – while its up to the presentation layer to interpret it, specifying the format may help content producers when ad breaks are marked up at the content generation time.

2) *Some tracking of the ad consumption may be provided.*

An MPD (or media segments) for a single MPD is augmented with additional information using non-DASH format. The standard DASH client is still able to play the presentation, however extended functionality will be available to clients that have modules handling these non-DASH XML-based formats.

The reason for use of non-DASH formats is avoiding reinventing the wheel and making integration with existing ecosystem easier – e.g. there is a widely deployed VAST [27] standard that provides sufficiently rich information. The VAST (not DASH!) client will issue HTTP GET requests that will be understood by existing tracking servers.

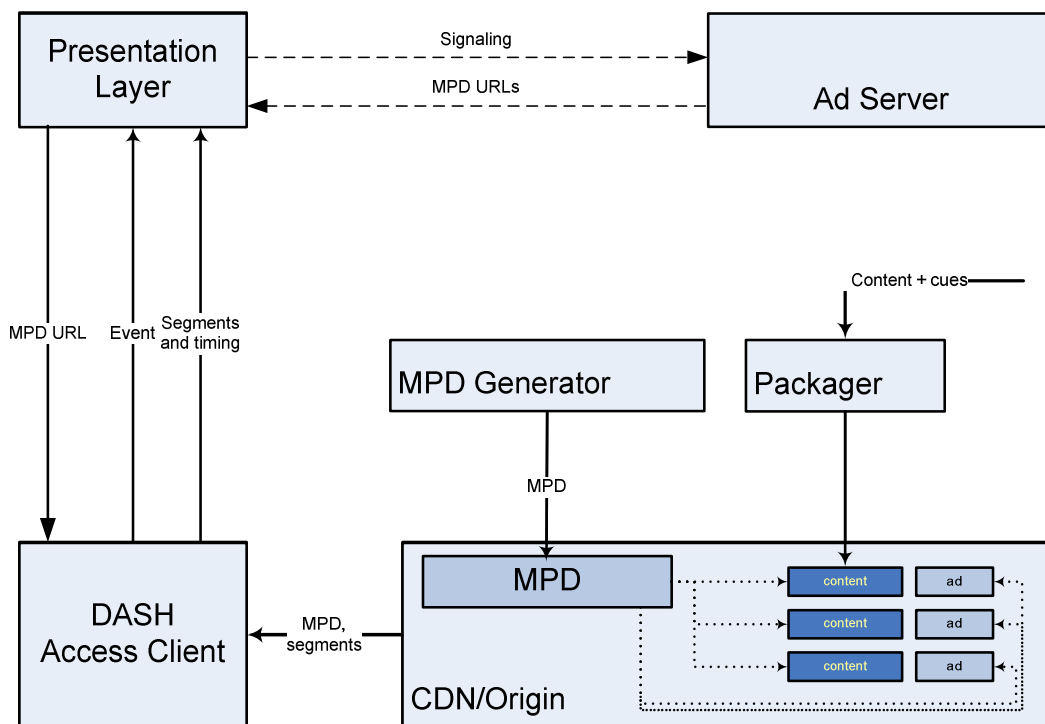
There are no gaps in MPEG DASH – support for DASH events is the enabler for this architecture -- the most reasonable way of conveying e.g. VAST information is DASH events.

This section mentions several external standards, such as SCTE 35 and VAST.

SCTE 35 is a standard that provides inband information on the upcoming ad breaks. Some of the information provided is timing (start / duration) and break identification. The parameters extracted from the SCTE 35 message are passed to the ad server and are used in process of ad decision by the ad decision servers. SCTE 35 is a binary format defined for MPEG-2 TS and its use is ubiquitous in the N. American market. An older version of SCTE 35 is also known as ITU-T J.181 [36]. An XML representation of SCTE 35 will be available in SCTE 35 2013.

The Interactive Advertising Bureau (IAB) provides standards for online advertisement. IAB's Video Ad Serving Template (VAST) specification specifies ad server response instructing the client on ad content it is expected to play. IAB Video Multiple Ad Playlist (VMAP) provides out-of-band information on ad breaks, with functionality similar to SCTE 35. The major difference between the two is the fact that SCTE 35 is tightly synchronized with media and appears inband, while VMAP is loaded at the beginning of a presentation.

A reference Ad insertion system architecture is shown below:



**Figure 7.2: Example system Architecture for Ad insertion for DASH**

The content publisher controls when and where to add ad in the content stream.

The content publisher delivers the content with in-band signalling to the packager for content distribution,

The presentation layer interacts with DASH clients to play DASH content. The Presentation layer also interacts with Ad server for Ad MPD URLs access, other non-DASH Ad content access and tracking report.

The in-band signalling conveyed as event in main content will trigger DASH client to send the trigger information to the presentation layer to initiate the interaction between the presentation layer and Ad server. The Ad MPD URL and other non-DASH Ad content will be retrieved by the presentation layer from the Ad server. The Ad MPD URL will be passed to DASH client by the presentation layer. The Ad DASH client will play the Ad accordingly.



## 7.4 Examples

### 7.4.1 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="200000"/>
      <Representation id="ad2" width="960" height="720" bandwidth="400000"/>
    </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>
```

### 7.4.2 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"/>

<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>

```

The `xlink:href="http://adserver.com/movie_11/ad_1.period"` document may be the below.

```

<!-- 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="200000"/>
    <Representation id="ad2" width="960" height="720" bandwidth="400000"/>
  </AdaptationSet>
</Period>

```

Once the resolution is applied it results in the following MPD.

```

<?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 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="200000" />
        <Representation id="ad2" width="960" height="720" bandwidth="400000" />
    </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>
    
```

### 7.4.3 Example call for Ad insertion for DASH

The example call flow for Ad insertion according to clause 7.3.2 for DASH is illustrated in Figure 7.3.

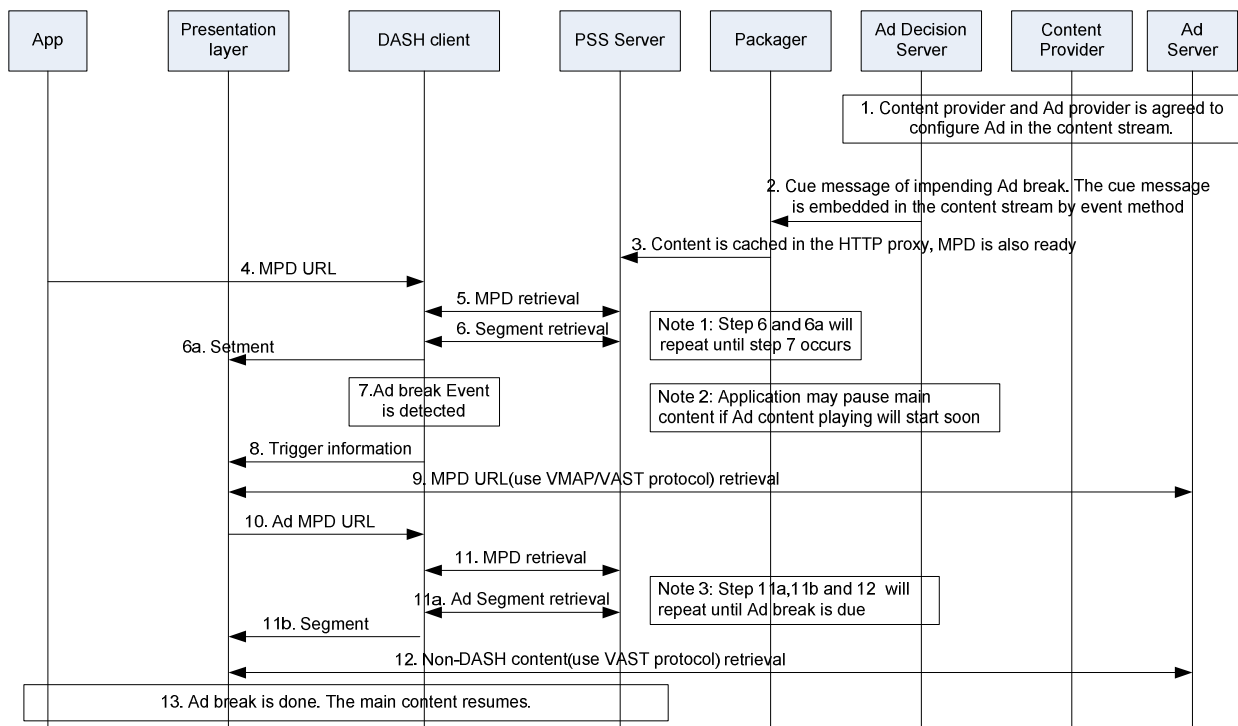


Figure 7.3: Example call flow of target Ad insertion for DASH

In-stream Ad normally is classified as liner Ad type, non-linear Ad type and companion Ad type. The liner Ad is DASH format compliant for DASH support. The non-linear Ad and companion Ad include text, graphic, rich media, etc. those content format is non-DASH compliant. The application interacts with Ad server by VAST protocol support. The whole solution relies on the support which is outside of scope of 3GPP.

## 7.6 Advanced Advertisement insertion in the operator network

### 7.6.1 Description

#### 7.6.1.1 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 ad break in the middle of the DASH video (especially for longer content) before the user can resume watching the video.

Another 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).

#### 7.6.1.2 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 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).

#### 7.6.1.3 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.

### 7.6.2 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.

### 7.6.3 Solution based on TS 26.247 and Gap Analysis

On use cases in clauses 7.6.1.1 and 7.6.1.2: Based on information exchange with 3GPP SA2, the use of Sh from a PSS server is not restricted in TS 23.228 [24] and therefore the use cases in clauses 7.6.1.1 and 7.6.1.2 can be fulfilled by using the Sh interface. This is a deployment choice.

On use case in clause 7.6.1.3:

- According to clause 6.12.3 and based on feedback from SA2 on their discussions during the work on UPCON in Rel-12 and feedback they received from RAN groups, the access to accurate information about the frequently changing radio conditions at the RAN node in a timely manner is seen as almost impossible. Information about

user plane congestion at a RAN node should be expected to represent a mean value over a period of time (e.g. a congestion level for a couple of minutes).

- However, a DASH content provider can add low-bitrate content in an Adaptation Set to be played in case the client observes issues on the available access bandwidth. If the client observes low bitrate, it automatically switches to the low bitrate content.

Other options are to provide a GBR bearer to provide a minimum bitrate even in case of congestion.

Generally, no gaps are identified, but proper documentation is encouraged of how the use cases can be fulfilled with today's architectures.

---

## 8 Conclusions and Recommendations

3GPP's Dynamic Adaptive Streaming over HTTP (DASH) specification was developed in and Rel-10 and is available in TS 26.247 [2] in alignment with MPEG's ISO/IEC 23009-1 [9]. During 2012 and 2013 MPEG has conducted work on a second edition of DASH [22] that includes additional technologies. These technologies are currently not part of 3GP-DASH, but at least some of them may be considered useful also for 3GPP-based DASH distribution as identified in this TR.

Clause 5 of the present document provides deployment guidelines for content authoring, client operation as well as operational guidelines. These guidelines provide an initial overview, but may be expanded further in order to support implementers in choosing appropriate parameters.

In terms of use cases and technologies, the following conclusions can be drawn:

- For Live Services (see clause 6.2), TS 26.247 supports all basic aspects to offer a live service. Additional tools may be considered to optimize efficiency and robustness.
- For Content Protection (see clause 6.3), 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] there is explicit support for common encryption nor key rotation.
- For Fast Media Startup (see clause 6.4), no gaps are identified and the tools included in TS 26.247 and described in clause 6.6 are considered sufficient.
- For Advanced Trick Modes (see clause 6.5), no gaps are identified and the tools included in TS 26.247 and described in clause 6.7 are considered sufficient.
- For Content and Device Interoperability (see clause 6.6), it was identified that 3GP-DASH as defined TS26.247 lacks a profile that is aligned with existing industry practices.
- For Advanced Support for Live Services 3GP-DASH (see clause 6.7), TS 26.247 lacks tools and guidelines for low-latency live services. MPEG-DASH second edition [22] defines certain tools for this purpose that may be considered to improve latency. In addition, the use of certain HTTP delivery modes may be considered.
- For Consistent QoE/QoS for DASH users (see clause 6.8), the use of the existing 3GPP PCC architecture as defined in TS 23.203 [15] and providing the Rx reference point provides sufficient means to fulfil the use case.
- For DASH as download format (see clause 6.9), the DASH formats may also be used in the progressive download context using a restricted subset of DASH. No specific gaps are identified, but the usage may be documented.
- For Efficiency of HTTP-caching infrastructure (see clause 6.10) on DASH the use case is fully supported since Rel-11 of TS26.247 and no gaps are identified.
- For Multiple Spectator Views offered with DASH (see clause 6.11), TS26.247 provides certain tools to support the use case, but gaps are identified to provide location/orientation information in the MPD and in timed metadata tracks in the 3GPP File Format as well as providing MPD indication of highlights and capturing parameters.
- For Operator control of video streaming services (see clause 6.12) at least a certain amount of use cases may be solved by solutions introduced in 6.15.3. Additional optimizations may be considered, but may require consultation with other 3GPP groups.

- For DASH Operation with Network Proxy Caches (see clause 6.13), certain gaps have been identified including the inability to provide proactive or reactive information to the DASH client about the location of Representations/Segments or information from the PSS server to the network proxy cache.
- For Services with caching of DASH content at UE functions (see clause 6.14), 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.
- For handling special content (see clause 6.15), TS 26.247 provides basic tools using dynamic services. However, the approach has limitations in terms providing a consistent operation. Identified gaps include the playout requires customized MPDs, no explicit signalling is available on special content, consistent and trusted client behaviour including authentication, authorization and session control is missing and tracking of all client requests and verification of the client behaviour may need to be implemented.
- For DASH Authentication the Generic Bootstrapping Architecture (GBA) [20] (see clause 6.16) can be used to authenticate the end-user(s) and bootstrap a secure end-to-end connection between the Streaming Server and the end-user(s). The Generic Bootstrapping Architecture currently does not contain any content access authorization methods for DASH. However, authorization of content access and streaming quality can be additionally implemented in the Streaming Server that takes on the role of NAF. The Generic Bootstrapping Architecture can also be used to derive the necessary shared secret(s) required to integrity protect the application level content/metadata between the UE and server serving this DASH content. No further gaps are identified.
- For Consistent Quality for DASH users QoE of DASH service (see clause 6.17) can be maintained if DASH service is carried over GBR bearer. Existing QoS mechanism can be used to support consistent QoE of DASH service. Considering the current EPC architecture supports QoS feature, consistent QoE of DASH service is supported already from 3GPP architecture point of view. In the mean time, signalling of quality-related information about the content (via the DASH MPD or other means) to the DASH client can be desirable as demonstrated by performance evaluation results documented in Annex A.
- For Ad Insertion (see clause 7), TS 26.247 supports a significant amount of tools, but certain optimizations are provided in ISO/IEC 23009-1 second edition [22] to enable more comprehensive and more feature rich ad insertion use cases. It was also highlighted that ad insertion in operator's network is enabled by existing architectures.

For all potential solution and optimizations, it is recommended to investigate whether existing technologies defined by MPEG in ISO/IEC 23009-1 second edition [22], other MPEG technologies as well as other organization such as the IETF fulfil the use cases.

---

# 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 clause 6.17.

---

### 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](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](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.

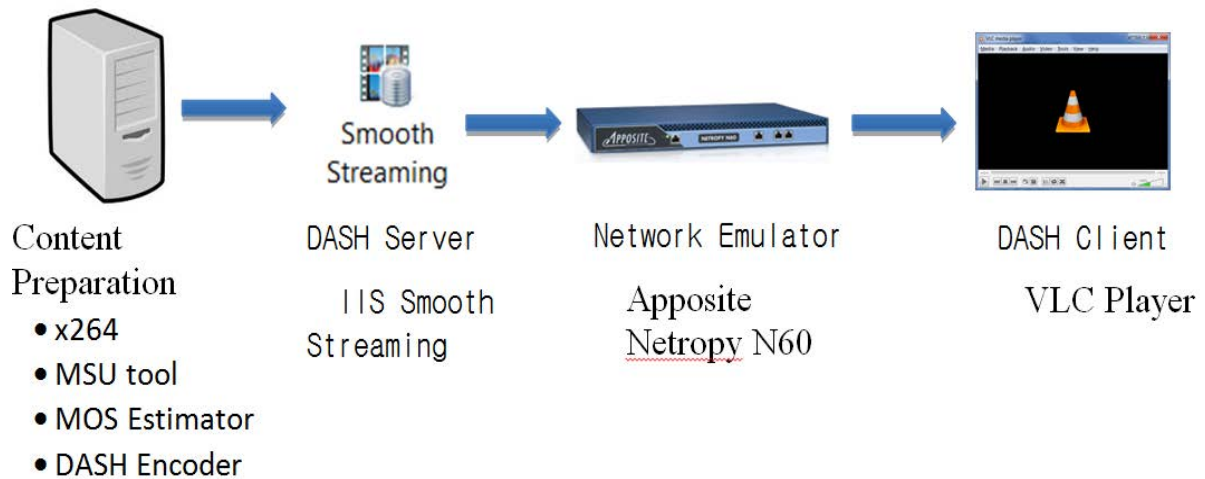


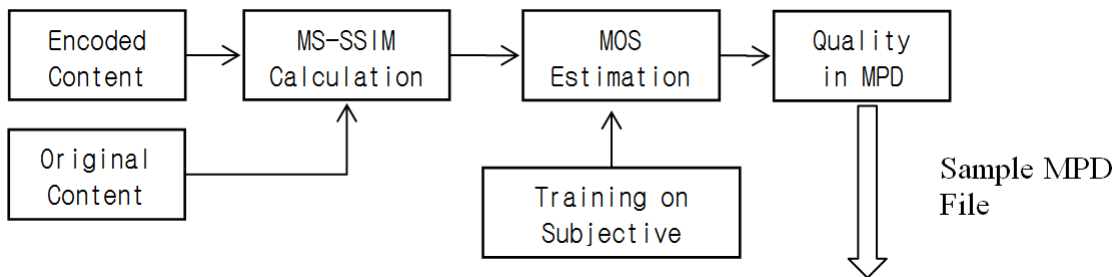
Figure A.1: Simulation Setup

### 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 1: 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  $\alpha$  and  $\beta$  are functions of video content characteristics (videos classified based on spatial details, motion levels, bitrate, resolution) and device characteristics (display size and resolution).

$$MOS = \alpha MSSIM + \beta \quad (1)$$

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 2: 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.

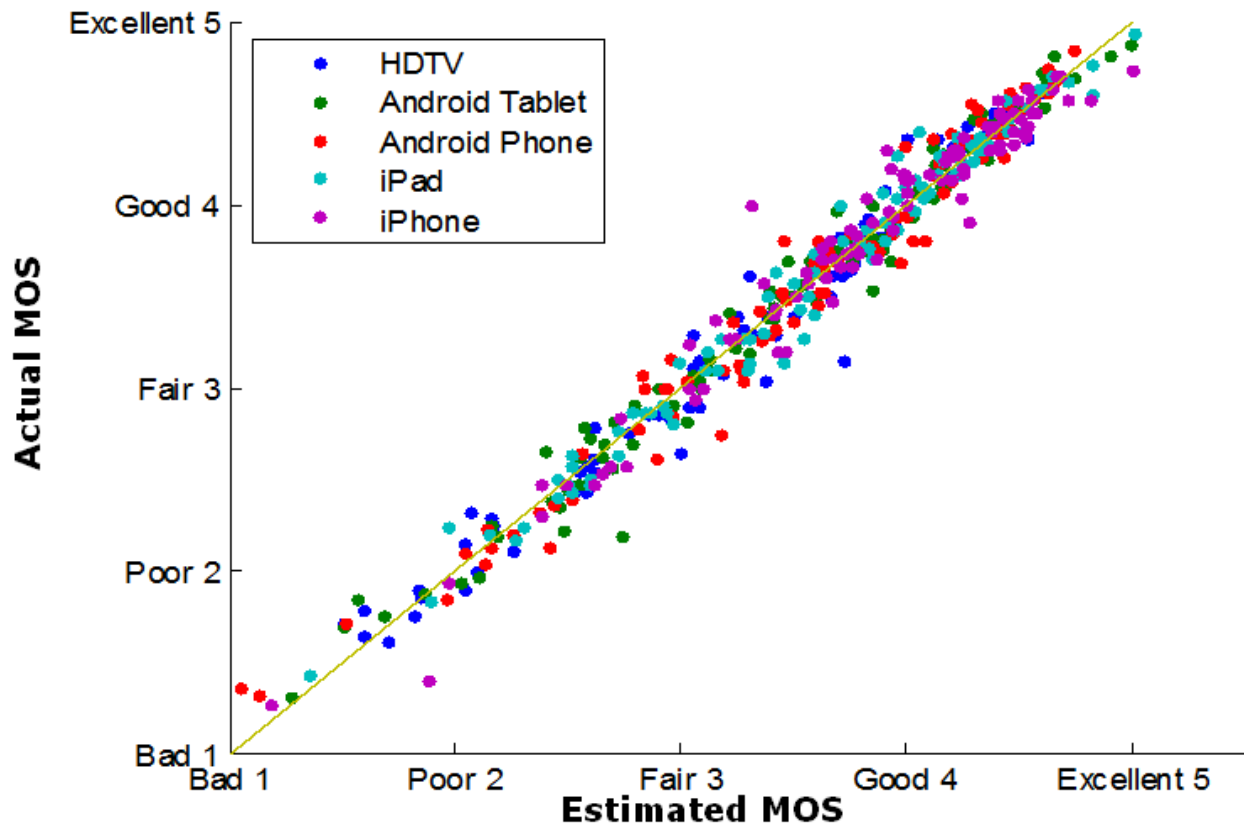


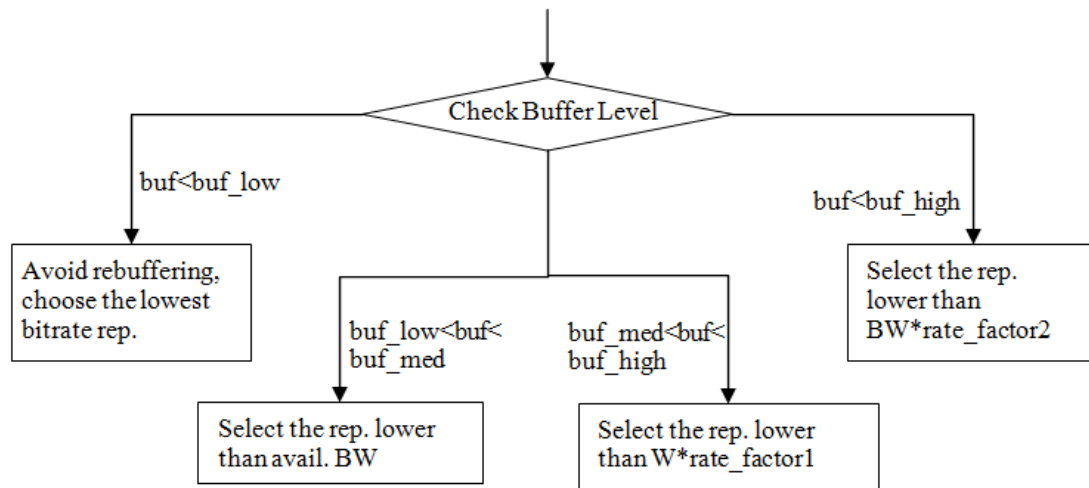
Figure A.3: MOS Estimation based on MS-SSIM, Video Content and Device Information

## 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:**  $\text{buf} < \text{buf\_low}$

Select lowest bitrate representation

**Condition 1:**  $\text{buf\_low} < \text{buf} < \text{buf\_med}$

Select highest bitrate representation lower than the available BW

**Condition 2:**  $\text{buf\_med} < \text{buf} < \text{buf\_high}$

Select highest bitrate representation lower than the available  $\text{BW} * \text{rate\_factor1}$

**Condition 3:**  $\text{buf} > \text{buf\_high}$

Select highest bitrate representation lower than the available  $\text{BW} * \text{rate\_factor2}$

## 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:  $\text{buf\_low}$ ,  $\text{buf\_med}$ ,  $\text{buf\_high}$
- Quality threshold:  $\text{quality\_min}$ ,  $\text{quality\_max}$
- Bitrate threshold factor:  $\text{rate\_factor1}$ ,  $\text{rate\_factor2}$

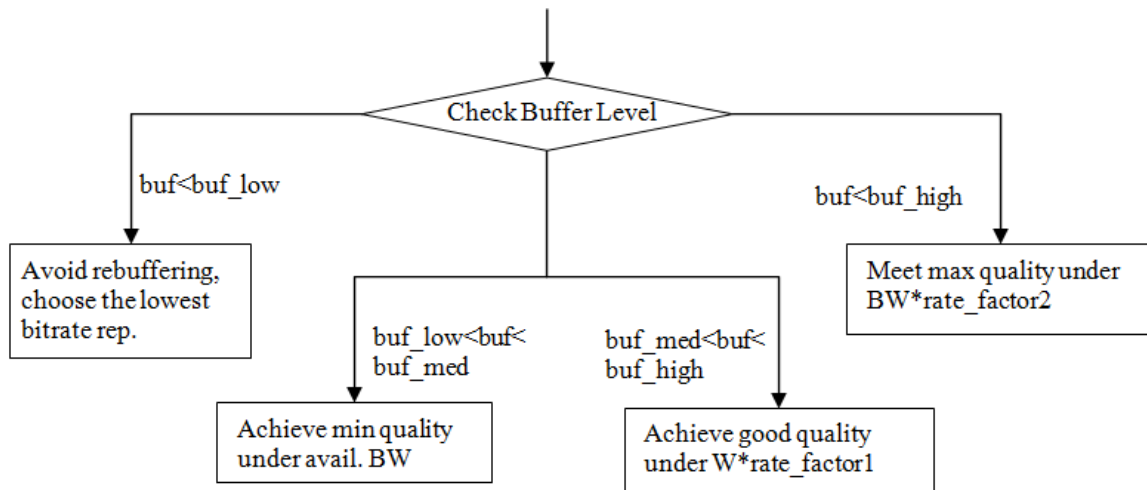


Figure A.5: Flowchart of Quality-based Rate Adaptation Strategy

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 2 Mbps and 200 kbps every 5 s.

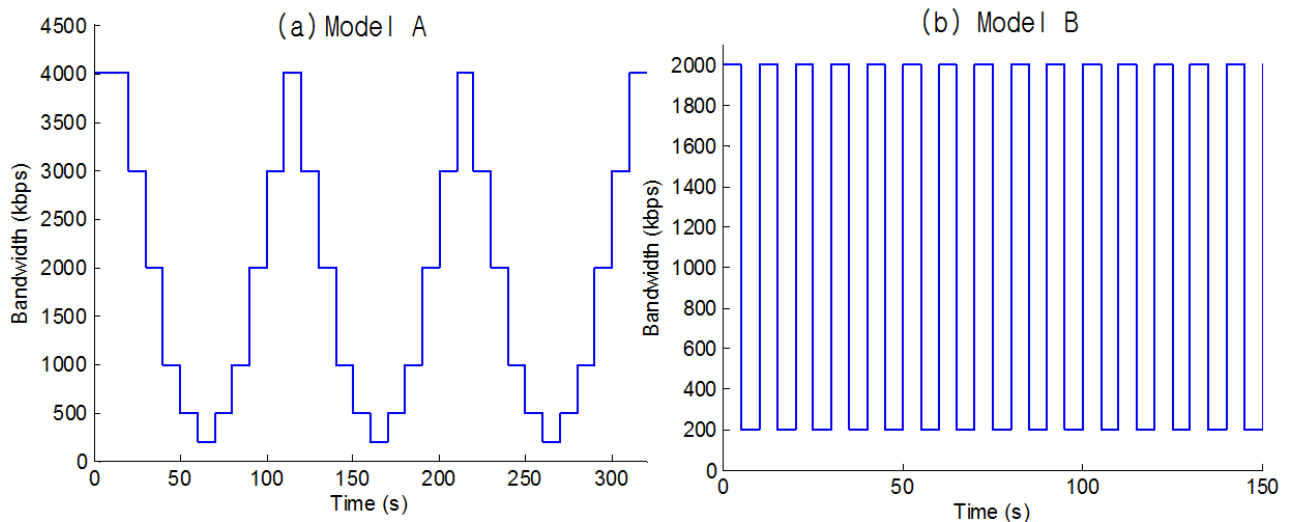


Figure A.6: Bandwidth Models A and B

## 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 30 s. 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 <sub>min</sub>	Q <sub>max</sub>
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)

Constrained VBR							
Algorithm	Metric	Avg. Bitrate (Mbps)	Avg. MOS	MOS<3	Avg. Buffer %	Rebuffering Time (s)	No. Rebuffer
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	26.2 %	45.5 %	14.3	1
Quality-based (Q <sub>min</sub> =3.0, Q <sub>max</sub> =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 <sub>min</sub> =3.0, Q <sub>max</sub> =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.

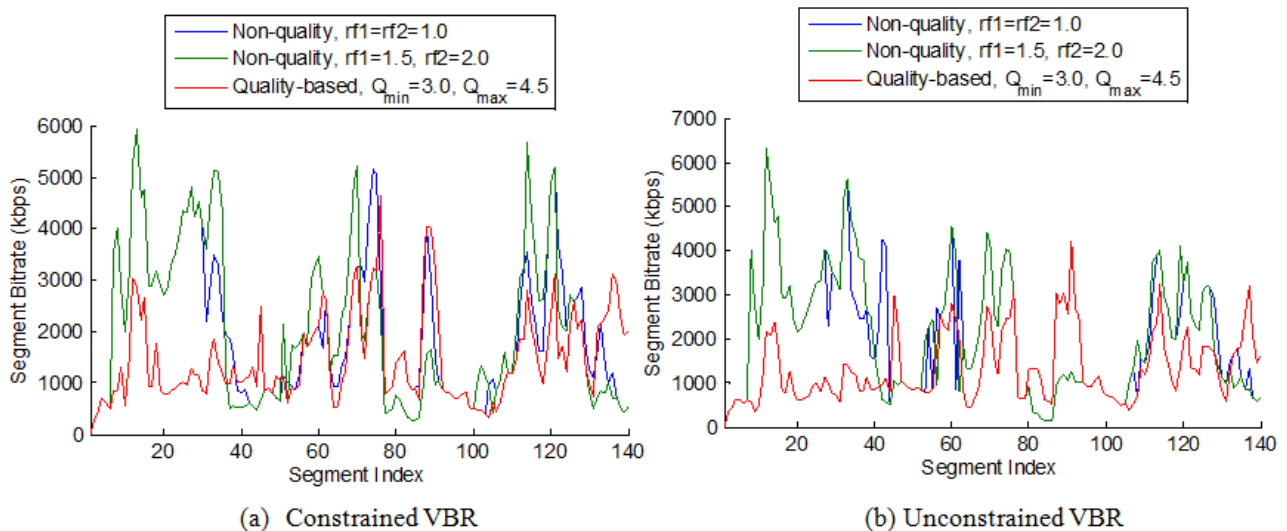


Figure A.7: Segment Index vs. Segment Bitrate (Bandwidth Model A)

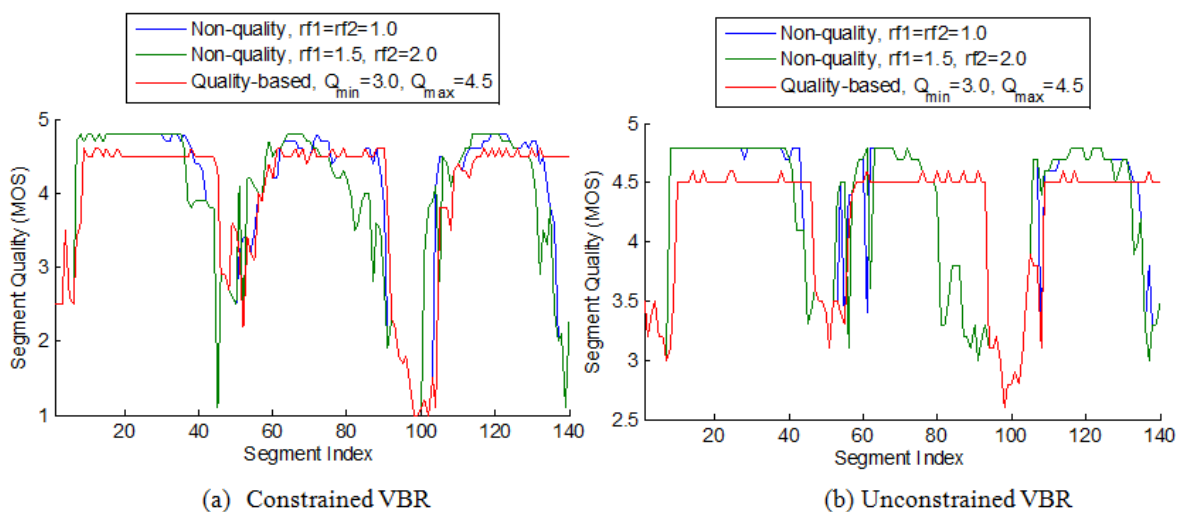
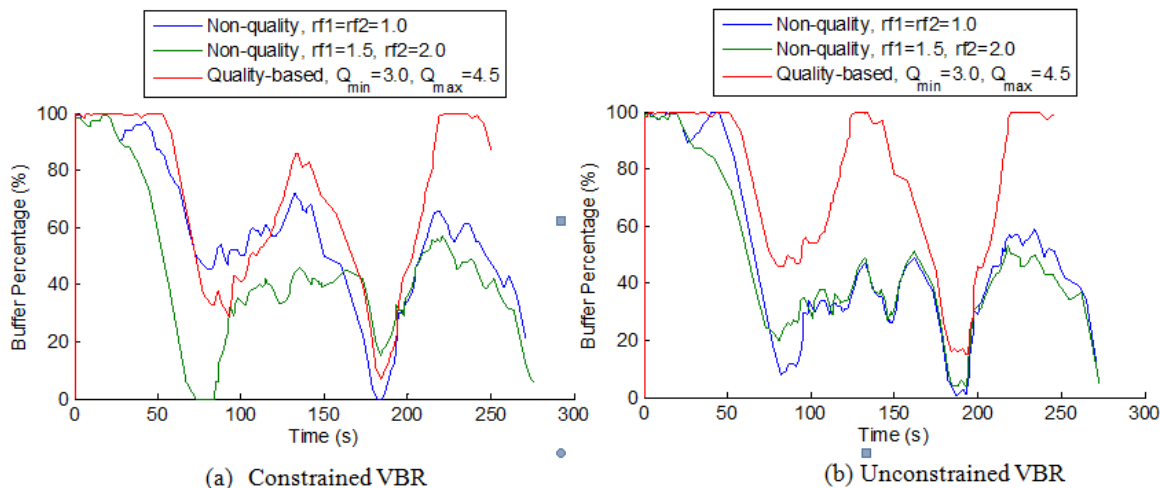


Figure A.8: Segment Index vs. Segment Quality (Bandwidth Model A)

**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	16.2 %	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 B: Evaluation of QoS-Driven DASH Client Adaptation

This annex describes simulation methodology and results on the evaluation of end-to-end capacity and QoE over an LTE-based system-level simulation platform in the presence of enforcement of dedicated QoS bearer policies for each DASH client. In particular, an analysis is presented quantifying the performance benefits of QoS-driven DASH adaptation techniques based on operator's selection of the maximum bitrate (MBR) and consequent adaptation at the DASH client. Furthermore, additional simulation results are provided demonstrating the quality improvement from the availability guaranteed bitrate (GBR) information at the DASH client during start-up.

Re-buffering has been identified as one of the most critical QoE metrics for streaming video. In a 3GPP DASH-based implementation of QoE metrics in the client device, this metric can be computed via monitoring the buffer status and/or play list metrics. Given the key importance of re-buffering in dictating the QoE delivered to the user, the service capacity of an LTE system is defined based on an outage criterion that is centered around the re-buffering percentage, i.e. the percentage of the total presentation time in which the user experiences re-buffering due to buffer starvation. In particular, a user is designated to be satisfactorily supported if its re-buffering percentage is smaller than a re-buffering outage threshold  $A^{\text{out}}$ . The service capacity is then defined as the maximum number of users that can be supported in the network such that the percentage of satisfied users is greater than the network coverage threshold  $A^{\text{cov}}$ . i.e.

$$C_{\text{out}}^{\text{rebuf}} = \mathbf{E} \left[ \arg \max_K \left\{ \frac{\sum_{i=1}^K \mathbf{1}(p_{\text{rebuf},i} \leq A^{\text{out}})}{K} \geq A^{\text{cov}} \right\} \right]$$

where  $\mathbf{E}[\cdot]$  denotes the expectation over multiple user geometry realizations and  $\mathbf{1}(\cdot)$  denotes the indicator function.

Five VBR-encoded video clips (Sony, Citizen Kane, Die Hard, NBC News, Matrix Part1) are considered with different bitrate requirements hosted at the HTTP server with multiple versions of each video clip available at different quality levels in the PSNR range of 26-39 dB, as shown in Figure B.1 and Table B.1. Two video traces for each video representation level contain content information with regards to – i) size and quality information for each video frame and ii) offset traces which give information of the video quality obtained by concealing lost video frames with previous frames. PSNR was used to model video quality as a representative although other advanced metrics could also be used.

A cellular deployment is assumed based on an IMT-Advanced urban macro-cell (UMa) test environment with an inter-site distance (ISD) of 500 m, where each user in the LTE network randomly requests one of the five available video clips. A 19-cell scenario is considered, where the center cell generating video traffic is surrounded by two layers of interfering cells generating full buffer traffic. Users are randomly dropped in the center cell. The simulation parameter settings and assumptions on the LTE air interface are provided in Table B.2 below. The additional assumptions include the following: 1) For the link to system mapping, Mutual Information Effective SINR Metric (MIESM) is used, 2) AWGN PER versus SINR curve corresponding to that modulation, code rate are used to determine the probability of error, 3) Channel Quality Indicator (CQI) are delayed by 5ms, 4) HARQ retransmissions are delayed by 8 ms with a maximum of 4 retransmissions. 5) The base stations in all other cells generate interference patterns corresponding to a full buffer mode of operation. 6) 100,000 sub-frames were simulated to generate LTE link statistics, 7) Users were picked randomly from a user population of 684 dropped uniformly in the sector. 8) For each configuration, statistics were collected from thirty different random drops of users in the network. 9) Packet fragmentation based on the maximum MTU size of 1500 bytes is considered, and HTTP/TCP/IP layer protocol behaviour and overheads are also incorporated in the analysis - 40 bytes of header was included in each TCP segment (10 bytes for NALU prefix + 12 bytes for HTTP header + 8 bytes for TCP header). 10) All the main features of TCP Reno flavour were implemented in the simulator including flow control, slow start, congestion avoidance, RTT estimation, timeout, re-transmission, fast re-transmit and fast-recovery to account for the presence of TCP. 11) The Backhaul Network (BN) between the eNodeB (eNB) and S-GW is modelled with a fixed bandwidth of 1 Gbps. 12) Core Network (CN) from video servers to the S-GW was modelled using a fixed delay of 50 ms. 13) Core and back-haul networks are assumed to lossless and radio access network is considered as the main bottleneck. 14) Uplink transmissions are assumed to be errorless. 15) The delay involved in establishment of the dedicated bearer (e.g. GBR bearers) was not included in the assumed system model.

Multiuser resource allocation over the OFDMA-based downlink LTE air interface is performed based on the well-known proportional fair scheduling principles. Only half of the available bandwidth of the 10 MHz LTE system is



assumed to be reserved for the DASH-based video streaming service while the remaining half is assumed to be dedicated for other services, e.g. voice and data services.

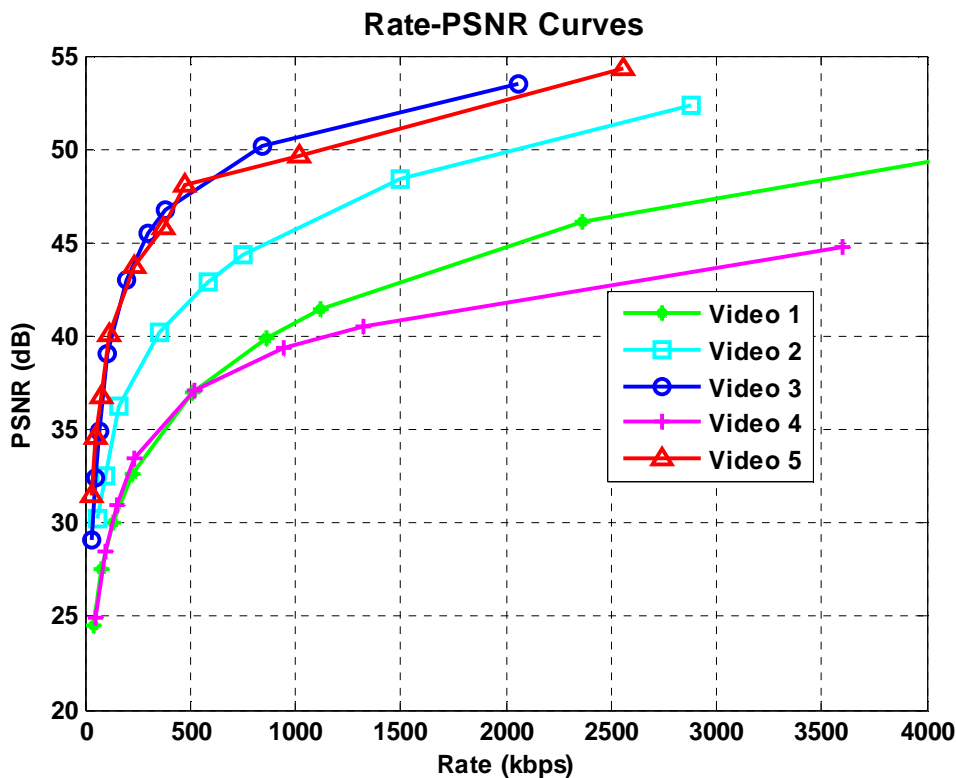


Figure B.1: Rate-PSNR Curves of Sample Videos

Table B.1: Details on the video content used in the evaluation

Video Source	Quantization Parameter Range	PSNR Range (dB)	Average Bitrate Range (kbps)
Sony_1080	28 - 48	24.5 – 36.94	44.23 – 508.24
Citizen Kane	28 - 42	30.25 – 40.25	60.11 – 351.91
Die Hard	34 - 48	29.00 – 39.00	32.38 – 103.24
NBC News	28 - 48	24.90 – 37.07	54.08 – 519.82
Matrix-1	34 - 48	31.45 – 40.05	30.98 – 118.64

Table B.2: LTE Air Interface configuration

Parameters	Assumption
Channel Model	Video requests are sequential: subsequent request is made after receiving previous video segment
Downlink Transmit Power	46 dBm
MIMO Mode	4x2 SU-MIMO for the downlink
Cellular Layout	Hexagonal grid, 19 cell sites, 3 sectors per site
Distance-dependent path loss	Loss $L = I + 37.6 \log_{10}(.R)$ , R in kilometers, $I = 128.1$
Lognormal Shadowing	Similar to UMTS 30.03, B 1.141
Shadowing standard deviation	8 dB
Number of antennas at UE	2
Number of antennas at cell	4
Antenna configuration at UE	Co-polarized antennas
Antenna configuration at eNB	Co-polarized (0.5λ spacing)
Outer-loop for target FER control	10% FER for 1 <sup>st</sup> HARQ transmission
Link adaptation	MCSs based on LTE transport formats according to TR 36.213
HARQ scheme	Chase combining
DL overhead	3 for PDCCH
UE speed	3km/h
Scheduling granularity	5 RB sub-band
Receiver type	MMSE-IRC
Feedback mode	Wideband PMI based on LTE 4-bit CB, subband CQI
Inter-site Distance	500 m
User distribution	Users dropped uniformly in the entire cell

According to the DASH-based adaptive streaming framework, users may consume varying qualities of video based on the working of the assumed adaptation algorithm, which selects the optimal quality/bitrate representation among the available video clips based on monitoring of user experience via 3GPP-based QoE metrics, i.e. particularly the playback buffer level. The different representations of the video requested by a representative client are indexed using letter  $k$ . In particular,  $k=1$  represents the lowest bitrate representation level and  $k = N$  represents the highest representation level and  $b_k$  represents the bitrate of encoded video of representation level  $k$ ,  $b_1 \leq b_2 \leq b_3 \leq \dots \leq b_N$ . Rate adaptation is client-driven and is done at segment level where each video segment might contain one or more GOPs (Group of Pictures).

The DASH-based adaptive streaming framework monitors the LTE link throughput and client buffer state and requests the video representations accordingly to realize the highest possible quality but also making sure to avoid playback buffer starvation. The DASH client starts playback with initial start-up delay of one second. It requests the video at a higher fetch rate during the buffering mode (playback buffer under a specified threshold) while the fetch rate is lower during the streaming mode (playback buffer above the specified threshold). Encountering playback buffer starvation, the client enters re-buffering mode while stalling the playback. The playback resumes after a certain targeted amount of media (i.e. 1 second) is aggregated in the media buffer.

A typical DASH-level throughput estimate is the average segment Throughput which is defined as the average ratio of segment size to the download time of the segment.

$$R_i^{\text{seg}} = \frac{1}{F} \sum_{s=S_i-F+1}^{S_i} \frac{S_{\text{seg}}(s)}{T_{\text{dwnd}}^{\text{seg}}(s) - T_{\text{fetch}}^{\text{seg}}(s)}$$

where  $S_{\text{seg}}(s)$ ,  $T_{\text{fetch}}^{\text{seg}}(s)$ ,  $T_{\text{dwnd}}^{\text{seg}}(s)$  are the size, fetch time, and download time of the  $j^{\text{th}}$  video segment,  $S_i$  the number of segments downloaded until frameslot  $i$ , and  $F$  is the number of video segments over which the average is computed.

The best video representation level possible in frameslot  $i$ ,  $Q_i^{\text{sup}}$ , is determined based on the current average segment throughput estimate as well as the GBR and MBR signaled by the network as follows:

$$Q_i^{\text{sup}} = \arg \max_k b_k ;$$

$$s.t. \quad b_k \leq \max(GBR, \min(R_i^{\text{seg}} \cdot MBR)); \quad k = 1, 2, \dots, N$$

Therefore, if the client is in steady state, then the best representation level is chosen as determined by the minimum of the DASH-level throughput estimate and MBR. Furthermore, in the start-up phase, the best representation level is chosen as determined by the maximum of the DASH-level throughput estimate and GBR.

Based on these simulation conditions, an evaluation of the cumulative distribution function (CDF) of re-buffering percentage was conducted for different system loading conditions in terms of the number of users the re-buffering outage threshold  $A^{\text{out}}$  is set to 2 % and the network coverage threshold  $A^{\text{cov}}$  is set to 95 %.

Figure B.2 shows the CDFs of re-buffering percent for various loads (number of users in the system) when the MBR is set to 2000 Kbps (GBR=0). Only the curve with  $N_{ue} = 35$  has 95 % of users with re-buffering less than 2 %. Figure B.3 and Figure B.4 plot the CDFs of re-buffering percent for various number of users when the MBR is set to 500 Kbps and 250 Kbps. As the MBR is reduced, the curves of more number of users have 95 % users with re-buffering less than 2 %.

Figure B.5 plots the percentage of users with re-buffering less than 2 % as the load is varied for different settings of MBR (GBR=0). For a given MBR, as loading is increased, the percentage of users that experience re-buffering less than 2% decreases. For the same loading, as the MBR is decreased the number of users with re-buffering less than 2 % is higher.

Figure B.6 plots the mean video quality obtained in terms of PSNR as the load is varied for different settings of MBR (GBR=0). For a given MBR, as loading is increased, mean video quality decreases. Also setting an MBR sets an upper limit on the video quality that could be obtained. Also note that the curves for different MBR values converge after the loading is increased beyond a certain point.

Figure B.7 plots the service capacity as the MBR is varied (GBR=0). As the MBR is increased, the service capacity decreases drastically initially and then gradually after a certain limit.

Figure B.8 plots the mean quality obtained when operating at the system capacity for a given MBR versus MBR. As the MBR is increased the improvement in quality initially increases drastically but then increases very gradually at higher MBRs.

The evaluations indicate that there exists a tradeoff between service capacity and achievable video quality that could be obtained by varying the MBR, and that the service capacity could be enhanced by properly setting MBR limits in the system while offering a reasonable maximum video quality. Therefore the network MBR limit has to be properly set depending on the capacity for which the system has to be designed and the target video quality desired. In that sense, the operator can effectively accommodate various levels of user loading and still deliver satisfactory QoE to the end users by controlling the MBR provided that the DASH clients can receive and use the MBR information in their adaptations.

The evaluation on the impact of GBR deals with a 100-user loading scenario with 25 % of the users each on a dedicated bearer with a certain GBR (premium users) and the rest of the users served by a default bearer (MBR is infinity for all bearers). A delay-bounded quality optimization in the start-up phase is considered with a target delay value of 2 seconds. For the premium users with a certain GBR, a scheduler that ensures the enforced GBR is used. There are two schemes of interest, (i) GBR is enforced by the network but not known to the DASH clients of the premium users (referred to as 'QASch only') and (ii) GBR is enforced by the network and is known by the DASH clients of the premium users (referred to as 'QASch + QASig'). Figures B.9 and B.10 plot the average user throughputs of premium and default users after scheduler's enforcement of GBR=250 kps and 500 kbps, respectively.

Figure B.11 plots the start-up quality CDFs of the premium users for the comparison of 'QASch only', and 'QASch + QASig' schemes at various GBR values. It is seen that the startup quality improves significantly when the GBR information is available at the DASH client. The intuition for this result is clear; in the buffering mode, the DASH client without GBR knowledge will start with the lowest quality and try to fill up its buffer as fast as possible, while the DASH client with GBR knowledge can start from a higher quality level as long as its target start-up delay requirement is satisfied. For verification purposes, Figure B.12 plots the corresponding start-up delay CDFs of the premium users; clearly showing that all schemes do meet the imposed delay requirement.

Figure B.13 plots the overall quality CDFs of the premium users for the comparison of 'QASch only', and 'QASch + QASig' schemes at various GBR values. As expected, it is seen that the knowledge of the GBR at the DASH client provides little improvement to the overall quality, since client is able to rely on its average segment throughput estimates to optimally utilize the available bandwidth after the start-up phase is over.

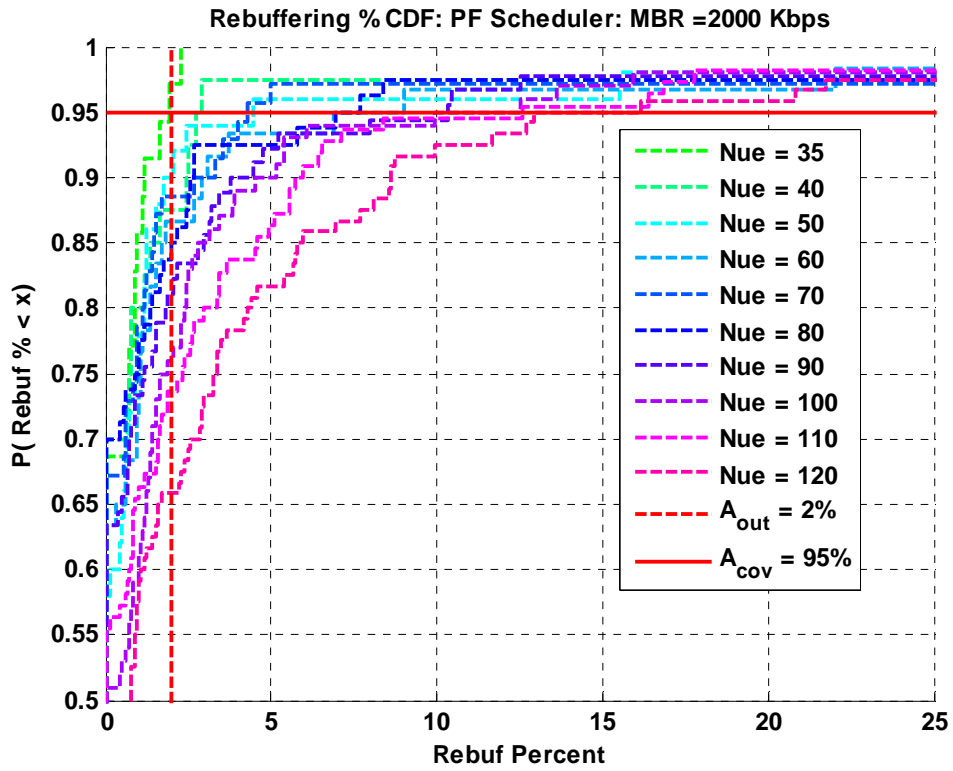


Figure B.2: CDF of Re-buffering percentage with MBR = 2000 Kbps

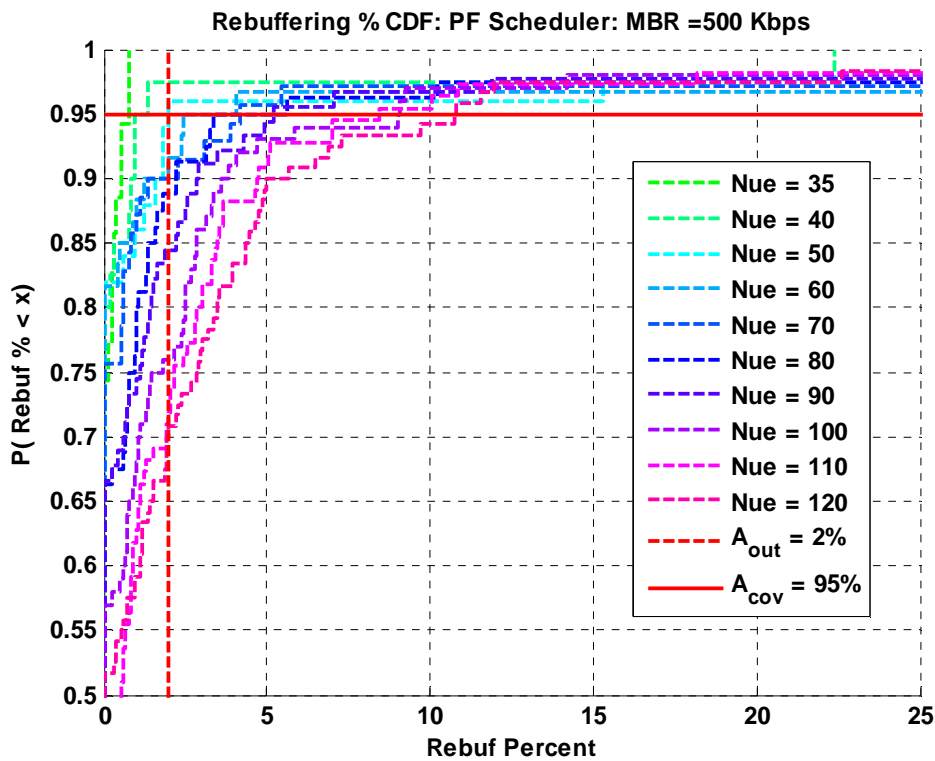


Figure B.3: CDF of Re-buffering percentage with MBR = 500 Kbps

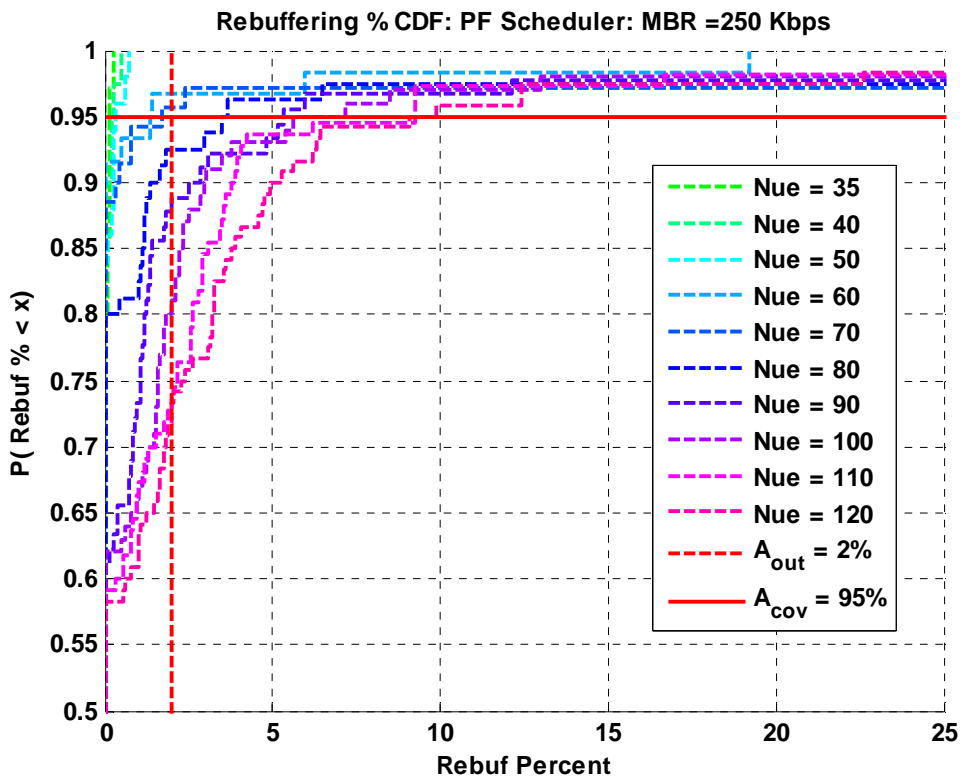


Figure B.4: CDF of Re-buffering percentage with MBR = 250 Kbps

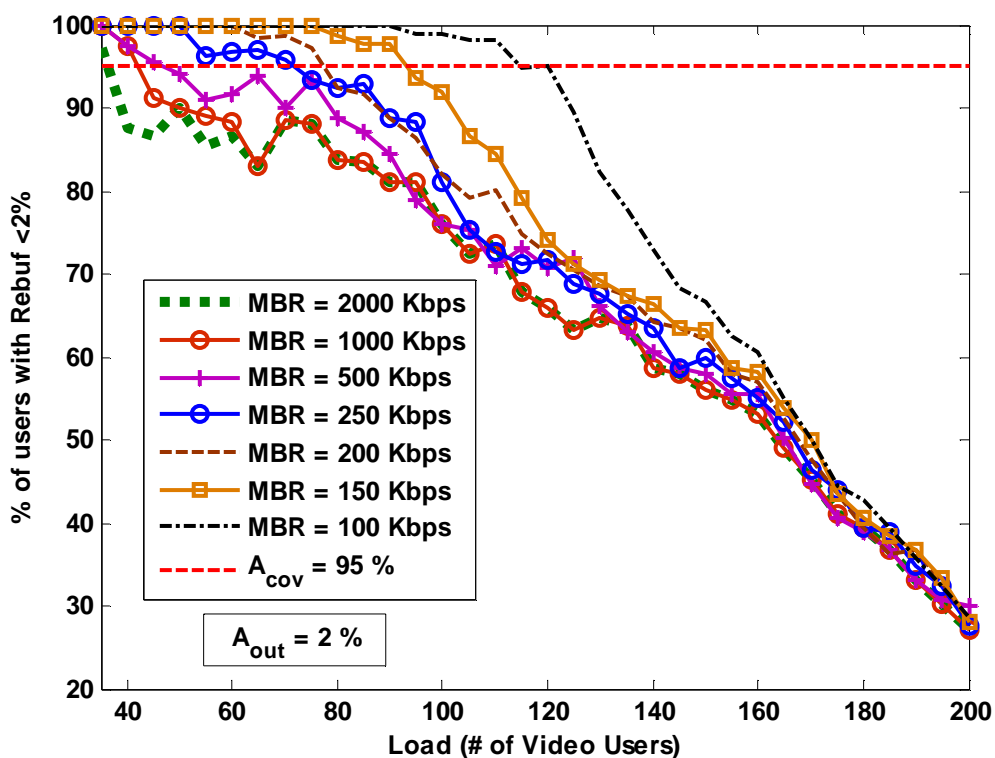
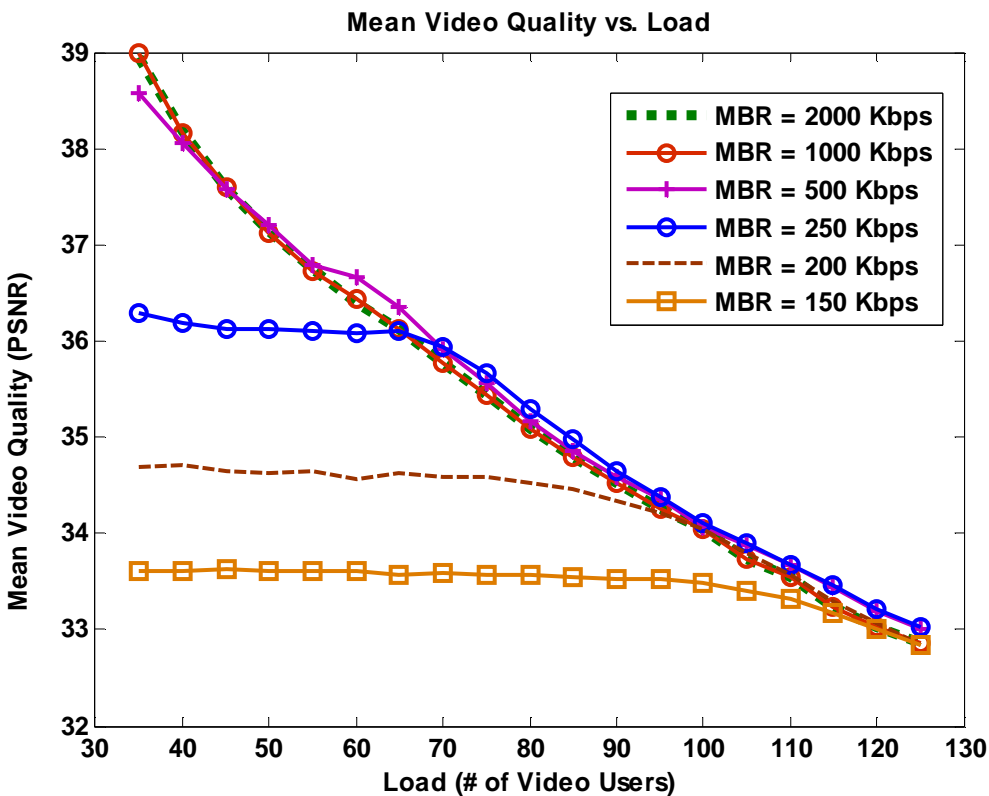


Figure B.5: Percentage of users with Rebuf < 2 % vs. load



FigureB.6: Mean Video Quality (PSNR) vs. Load

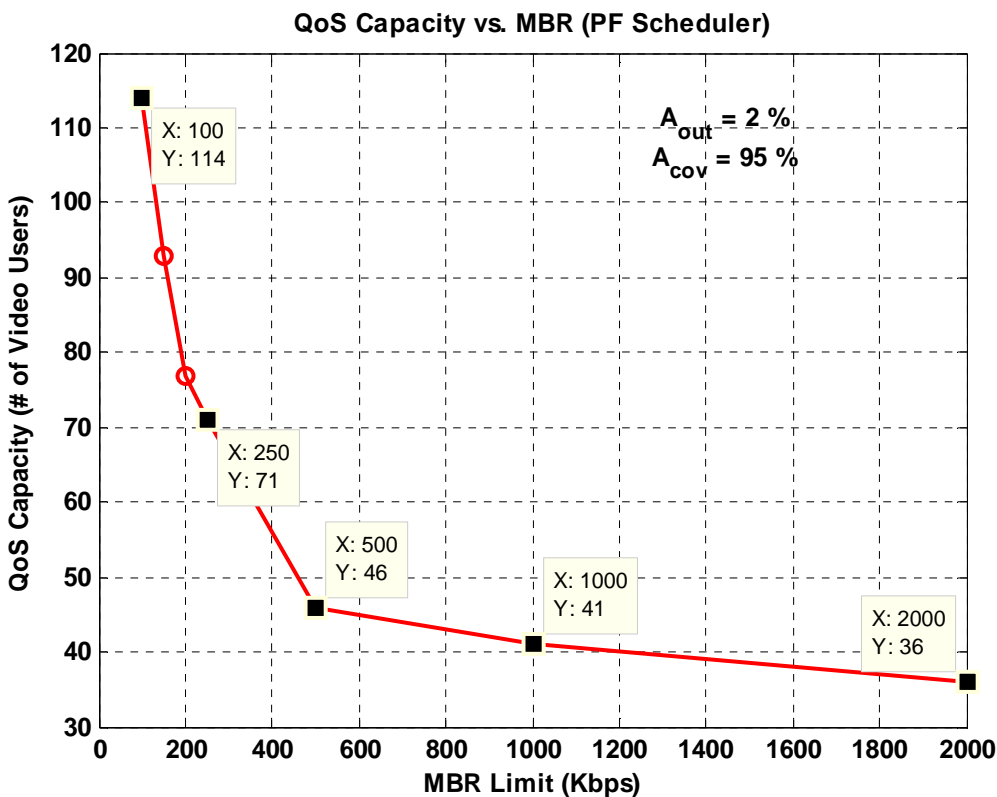


Figure B.7: Service Capacity vs. MBR

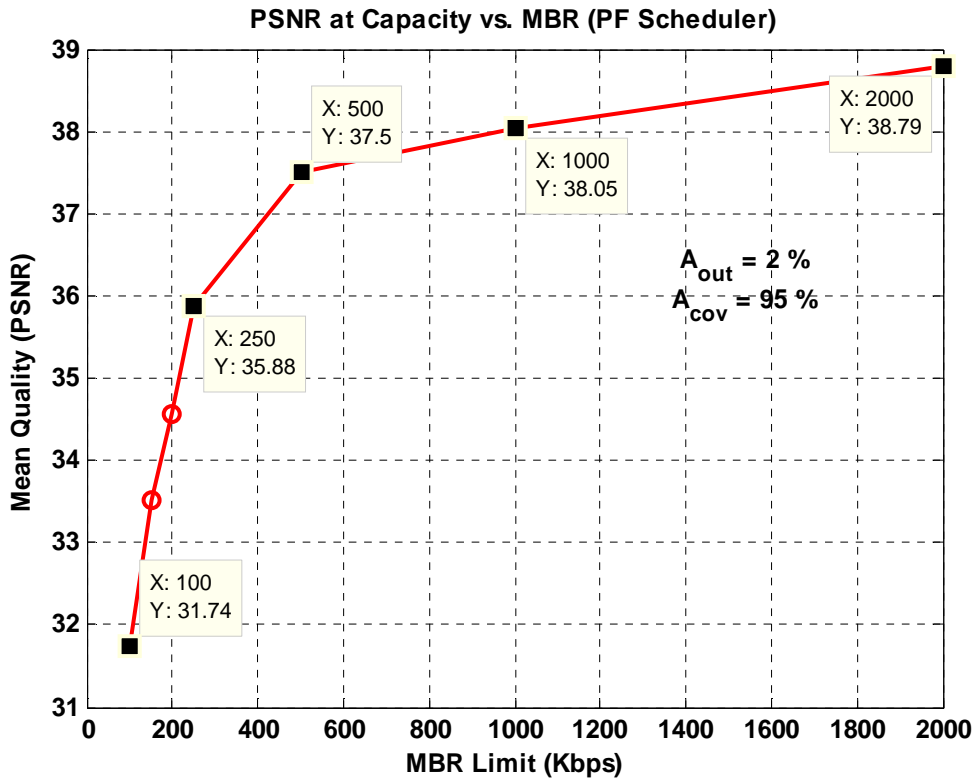


Figure B.8: Mean Quality at Capacity vs. MBR

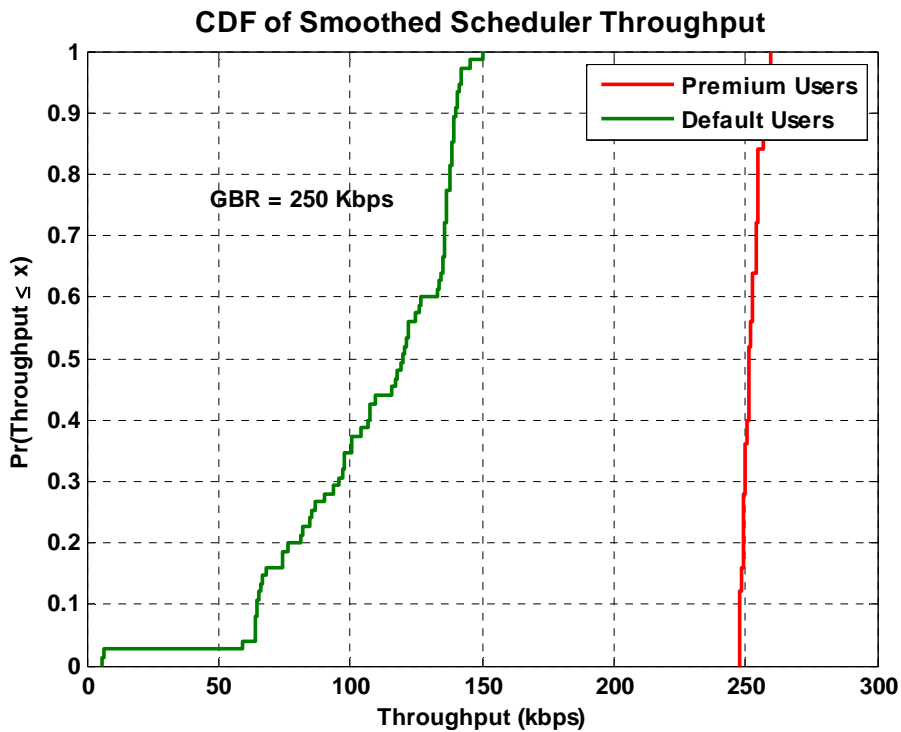


Figure B.9: CDFs of average user scheduler throughput for GBR=250 kbps

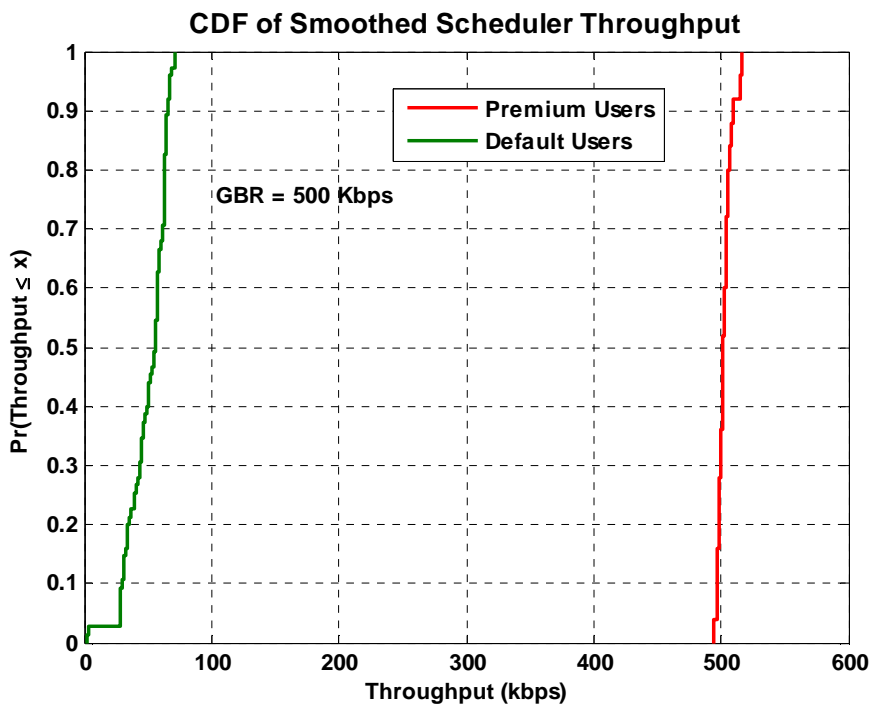


Figure B.10: CDFs of average user scheduler throughput for GBR=500 kbps

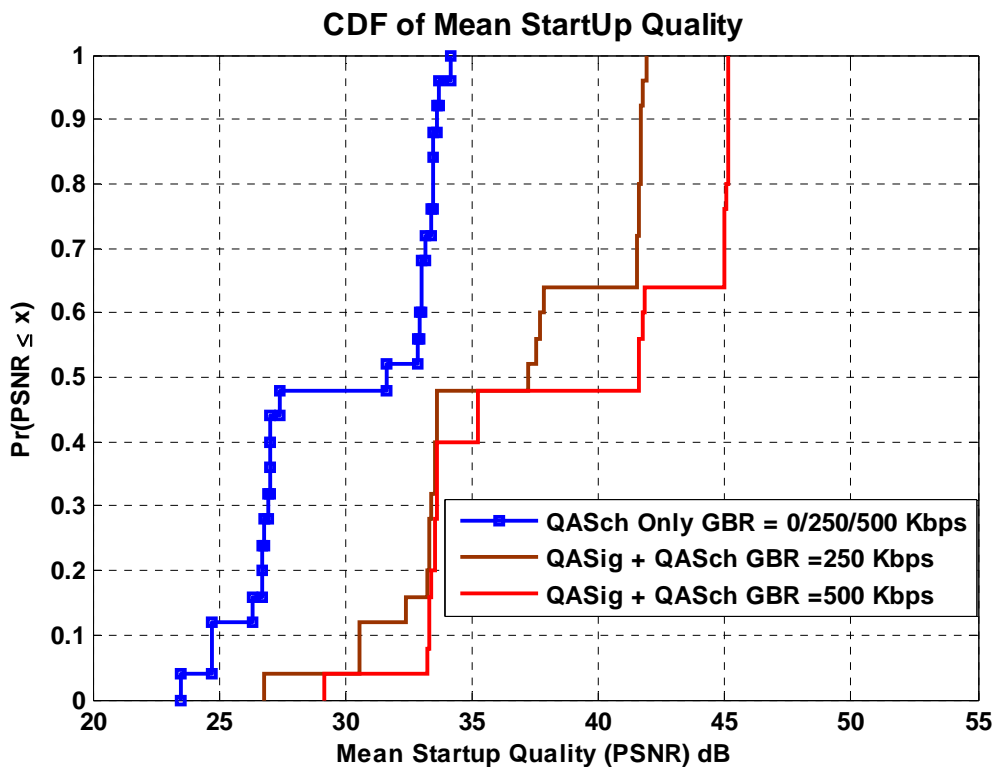


Figure B.11: Mean Start-up Quality CDF comparison at various GBRs



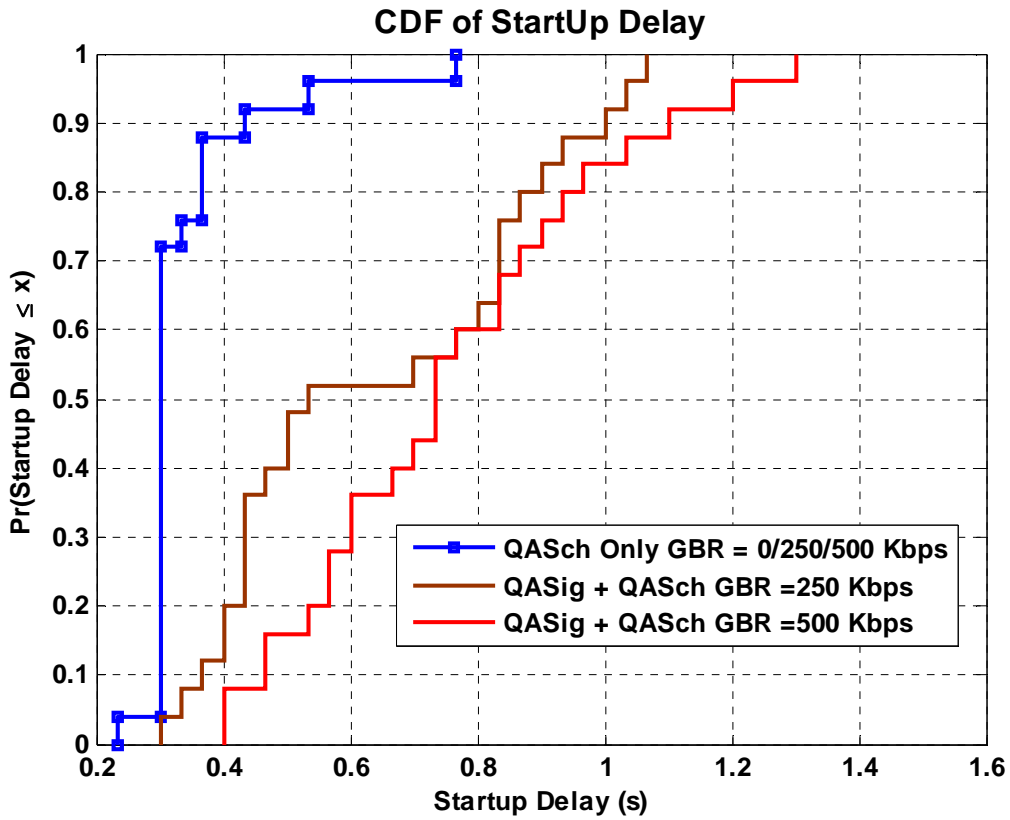


Figure B.12: Mean Start-up Delay CDF comparison at various GBRs

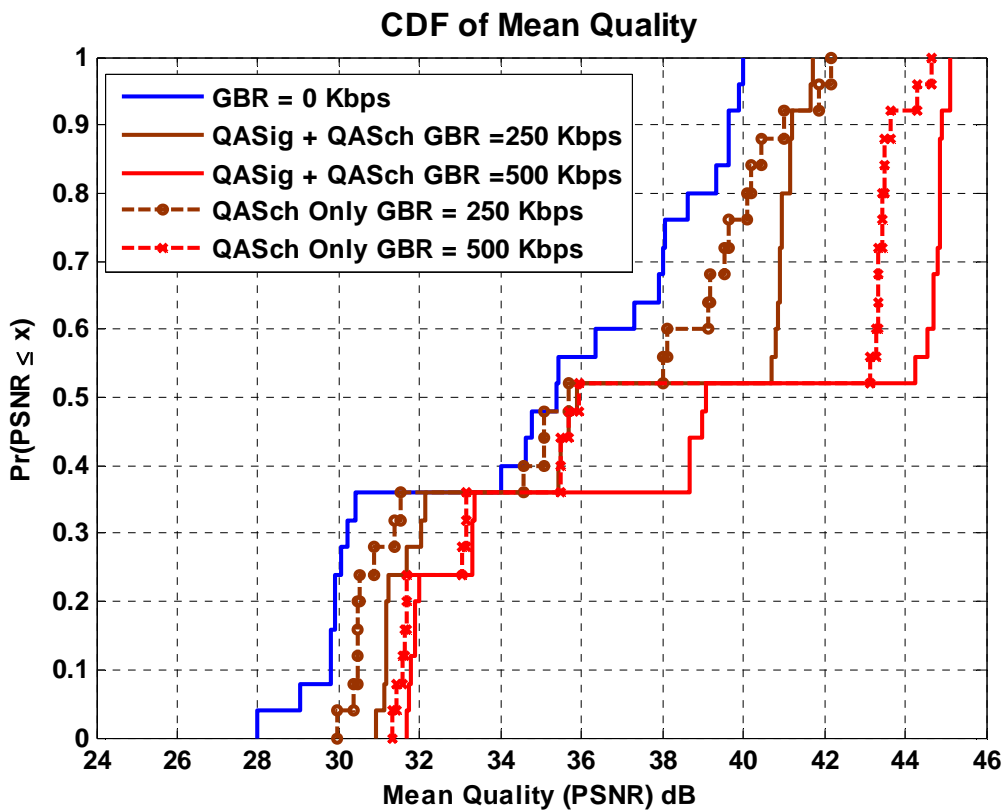


Figure B.13: Mean Overall Quality CDF comparison at various GBRs

## Annex C: 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
09-2014	65	SP-140479			Presented for approval at TSG SA#65	1.0.0	2.0.0
09-2014	65				Approved at TSG SA#65	2.0.0	12.0.0
12-2015	70				Version for Release 13	12.0.0	13.0.0

Change history							
Date	Meeting	TDoc	CR	Rev	Cat	Subject/Comment	New version
2017-03	75					Version for Release 14	14.0.0
2018-06	80					Version for Release 15	15.0.0
2020-07	-	-	-	-	-	Update to Rel-16 version (MCC)	<b>16.0.0</b>

---

# History

<b>Document history</b>		
V16.0.0	September 2020	Publication