Open IPTV Forum – Release 2 Specification

HTTP Adaptive Streaming

DRAFT V0.04.01 - April 23, 2010

Open IPTV Forum

Open IPTV Forum

Postal address

Open IPTV Forum support office address
650 Route des Lucioles – Sophia Antipolis
Valbonne – FRANCE
Tel.: +33 4 92 94 43 83
Fax: +33 4 92 38 52 90

Internet
http://www.oipf.tv

Disclaimer

The Open IPTV Forum members accept no liability whatsoever for any use of this document.

This specification provides multiple options for some features. The Open IPTV Forum Profiles specification complements the Release 2 specifications by defining the Open IPTV Forum implementation and deployment profiles. Any implementation based on Open IPTV Forum specifications that does not follow the Profiles specification cannot claim Open IPTV Forum compliance.

Copyright Notification

Revision History

edit000	Initial version – based on • OIPF-IPP-AVC-099-R01 • OIPF-MR-AVC-103-Minutes_2010-01-20	2010-01-25
0.02	Major Revision with 3GPP reference- based on: • Dublin meeting • OIPF-CR-AVC-138-R02-OIPF_HAS_using_3GPP • OIPF-CR-AVC-140-3GPP_Profile_and_Extensions • OIPF-CR-AVC-142-Comments_to_profile_lis	2010-03-26
0.03	Includes: • OIPF-CR-AVC-146-The_MP4_System_Layer_for_HAS • Editorials	2010-04-01
0.04	Includes: • Changes base d on discussion of OIPF-CR-AVC-150-HAS_MPEG-TS_generic_support – with changes as agreed per email exchange on AVC reflector, finishing with mail on Fri 23-04-2010 11:29, subject: "RE: [AUDIO_VIDEO_CODECS] THIS IS IT" • OIPF-CR-AVC-158-R03-HAS – Ensuring compliancy with 3GPP MPDs • Notes from: OIPF-MR-AVC-165-Minutes_2010-04-22 • Editorials • 0.04.01 with updated formalities and cosmetics	2010-04-23

Contents

FOREWORD 6
INTRODUCTION 7
1 REFERENCES 8
1.1 NORMATIVE REFERENCES 8
1.1.1 Standard References 8
1.1.2 Open IPTV Forum References 9
1.2 INFORMATIVE REFERENCES 9
2 CONVENTIONS AND TERMINOLOGY 10
2.1 CONVENTIONS 10
2.2 TERMINOLOGY 10
2.2.1 Definitions 10
2.2.2 Abbreviations 10
3 MEDIA PRESENTATION DESCRIPTION 11
3.1 MPD FORMAT 11
3.1.1 Component element 13
3.2 SEGMENTATION CONSTRAINTS 14
4 ADAPTIVE MEDIA FORMATS 15
4.1 MPEG-2 TRANSPORT STREAM SYSTEMS LAYER 15
4.1.1 PID allocation 15
4.1.2 Program Specific Information 15
4.1.3 Access unit signaling 15
4.1.4 Media packaging 16
4.1.5 Content protection 16
4.2 MP4 FILE FORMAT SYSTEMS LAYER 17
4.3 VIDEO 17
4.4 AUDIO 17
4.5 SUBTITLES 17
5 PROTOCOLS 18
5.1 LIVE STREAMING 18
5.2 TRICK PLAY 18
6 CONTENT & SERVICE PROTECTION 19
APPENDIX A. MPD SCHEMA 20
APPENDIX B. COMPONENT MANAGEMENT 21
APPENDIX C. USAGE OF THE MP4 FILE FORMAT 23
C.1 AUDIO/VIDEO SYNCHRONIZATION 23
C.2 PARTIAL REPRESENTATIONS 25

Tables

Table 1: Component Element and Atributes 13
Table 2: Example Audio/Video Synchronization 23

Figures

Figure 1: Content Segmentation for HTTP Adaptive Streaming **7**

Figure 2: Example of the MPD **12**

Figure 3: Example *tfad*-box **24**

Figure 4: Partial Representation Example **25**

Figure 5: Partial Representation Retrieval **25**

Foreword

This Technical Specification (TS) has been produced by the Open IPTV Forum.

This specification provides multiple options for some features. The Open IPTV Forum Profiles specification will complement the Release 2 specifications by defining the Open IPTV Forum implementation and deployment profiles. Any implementation based on Open IPTV Forum specifications that does not follow the Profiles specification cannot claim Open IPTV Forum compliance.

Introduction

This specification defines the usage of and, where necessary, extenstions to the technologies defined in [TS26234] and [TS26244] to enable HTTP based Adaptive Streaming for Release 2 Open IPTV Forum compliant services and devices.

In case of HTTP Adaptive Streaming, a service provides a Content item in multiple qualities/bitrates in a way that enables a client to adapt to (for example) variations in the available bandwidth by seamlessly switching from one bitrate to another while receiving and playing the Content. This is achieved by encoding a Content item in alternative Streams of different bitrates and segmenting these Streams into aligned and independently encoded Segments. This results in a matrix of Segments as depicted in Figure 1.

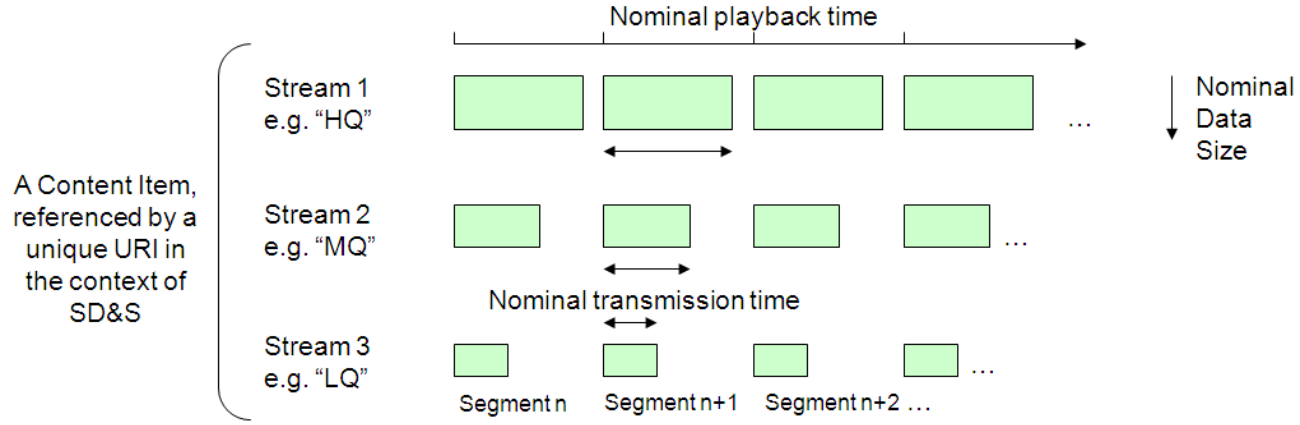

Figure 1: Content Segmentation for HTTP Adaptive Streaming

The Segments are offered for HTTP download from a URI that is unique per Segment. After completion of the download (and playback) of a certain Segment of a certain Stream, a client may switch to an alternate stream simply by downloading (and playing) the next segment of a different Stream. This requires the client to have a description of the available Streams and Segments and URI's from wich to download the Segments. This description is provided as a separate resource: the Media Presentation Description (MPD). The MPD is described in section 3.

The media data in Segments is formatted in compliance with the media formats as defined in [AVC]. However, in the context of HTTP Adaptive Streaming, additional requirements are put on the usage of these formats, especially regarding the systems layers. This "profile" is specified in section 4.

Similarly, the retrieval mechanisms of Segments are in compliance with [PROT], with the usage in the context of HTTP Adaptive Streaming as defined in section 5.

1 References

1.1 Normative References

1.1.1 Standard References

[RFC2119]	RFC 2119 (1997-03), IETF, “Key words for use in RFCs to Indicate Requirement Levels”.
[TS101154]	ETSI TS 101 154 V1.9.1 (2009-09), “Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream”.
[TS102034]	ETSI TS 102 034 V1.3.1 (2007-10), “Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Networks”.
[MPEG2TS]	ISO/IEC 13818-1:2000/Amd.3:2004, “Generic coding of moving pictures and associated audio information: Systems”.
[EN300468]	ETSI EN 300 468 V1.9.1 (2008-11), “Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems”.
[DLNAMEDIA]	DLNA Networked Device Interoperability Guidelines expanded: October 2006. Volume 2: Media Format Profiles. Digital Living Network Alliance.
[MRL BBTS]	Marlin Developer Community, “Marlin Broadband Transport Stream Specification”, Version 1.0, July 2008.
[ISOFF]	ISO/IEC 14496-12:2005, “Information Technology - Coding of Audio-Visual Objects - Part 12: ISO Base Media file format”, International Standards Organization.
[MP4FF]	ISO/IEC 14496-14:2003, “Information Technology – Coding of Audio-Visual Objects – Part 14: MP4 file format”, International Standards Organization.
[AVCFF]	ISO/IEC 14496-15:2004, “Information Technology - Coding of Audio-Visual Objects - Part 15: Advanced Video Coding (AVC) file format”, International Standards Organization.
[OMARLIN]	Marlin Developer Community, “OMArlin Specification”, Version 1.0.1, July 2008.
[MRL FF]	Marlin Developer Community, "Marlin - File Formats Specification", Version 1.1, July 2008, and latest version of “Marlin Errata: Marlin - File Formats Specification V1.1”.
[H264]	ITU-T Recommendation H.264 / ISO/IEC 14496-10:2005: “Information technology – Coding of audio-visual objects- Part 10: Advanced Video Coding”.
[H262]	ITU-T Recommendation H.262 / ISO/IEC 13818-2: “Information Technology – Generic Coding of moving pictures and associated audio information: Video”.
[DVBSUBT]	ETSI EN 300 743 V1.3.1 (2006-11), “Digital Video Broadcasting (DVB) – Subtitling systems”.
[CEACC]	Consumer Electronics Association CEA-708-C (2006), “Digital Television (DTV) Closed Captioning”.
[DVBTTXT]	ETSI EN 300 472 V1.3.1 (2003-05), “Digital Video Broadcasting (DVB); Specification for conveying ITU-R System B Teletext in DVB bitstreams”.
[AAC]	ISO/IEC 14496-3:2009, “Information Technology – Coding of audio-visual objects – Part 3: Audio”.
[AC3]	ETSI TS 102 366 V1.2.1 (2008-08), “Digital Audio Compression (AC-3, Enhanced AC-3) Standard”.
[MPEG1]	ISO/IEC 11172-3:1993/Cor 1:1996, “Information Technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio”.
[ITUT81]	ITU-T T.81 (09/92), "Information Technology - Digital Compression and Coding of Continuous-tone Still Images: Requirements and guidelines”.
[DVB-BB]	DVB BlueBook A001r8: Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in satellite, cable and terrestrial broadcasting applications (draft TS 101 154 V1.9.1)
[MP4V]	ISO/IEC 14496-2:2004: "Information technology - Coding of audio-visual objects - Part 2: Visual"

[TS26234]	3GPP TS 26.234 V9.X.Y, Transparent end-to-end Packet-switched Streaming Service (PSS) Protocols and codecs
[TS26244]	3GPP TS 26.244 V9.X.Y 3rd Generation Partnership Project, Transparent end-to-end packet switched streaming service (PSS), 3GPP file format (3GP)

1.1.2 Open IPTV Forum References

[ARCH]	Open IPTV Forum, “Functional Architecture - V2.0”, September 2009.
[SVCS]	Open IPTV Forum, “Services and Functions for Release 2”, V1.0, October 2008.
[OVIEW]	Open IPTV Forum, “Release 2 Specification, Volume 1 – Overview”, August 2010 (TBC).
[AVC]	Open IPTV Forum, “Release 2 Specification, Volume 2 – Media Formats”, August 2010 (TBC).
[META]	Open IPTV Forum, “Release 2 Specification, Volume 3 – Metadata”, August 2010 (TBC).
[PROT]	Open IPTV Forum, “Release 2 Specification, Volume 4 – Protocols”, August 2010 (TBC).
[DAE]	Open IPTV Forum, “Release 2 Specification, Volume 5 - Declarative Application Environment”, August 2010 (TBC).
[PAE]	Open IPTV Forum, “Release 2 Specification, Volume 6 - Procedural Application Environment”, August 2010 (TBC).
[CSP]	Open IPTV Forum, “Release 2 Specification, Volume 7 - Authentication, Content Protection and Service Protection”, August 2010 (TBC).

1.2 Informative References

[SPDIF]	ISO/IEC 60958-3:2006, Digital audio interface – part 3: Consumer applications.

2 Conventions and Terminology

2.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Introduction”, are normative, unless they are explicitly indicated to be informative.

2.2 Terminology

2.2.1 Definitions

In addition to the definitions provided in Volume 1, the following definitions are used in this Volume.

Term	Definition
Content Resource	A Content item that is provided in multiple qualities/bitrates to enable adaptive streaming of that Content item. Service Discovery procedures refer to a Content Resource
Stream	A version of the Content Resource at a specific quality level
Segment	A section of the Content Resource in a specific systems layer format (either TS or MP4), referred to via a unique URI

2.2.2 Abbreviations

In addition to the Abbreviations provided in Volume 1, the following abbreviations are used in this Volume.

Acronym	Explanation
AAC	Advanced Audio Coding
AAC LC	AAC Low Complexity
ATSC	Advanced Television Systems Committee
DVB	Digital Video Broadcasting
DVB-SI	DVB Service Information
ETSI	European Telecommunications Standards Institute
GOP	Group Of Pictures
JPEG	Joint Photographic Experts Group
MPEG	Moving Pictures Expert Group
3GPP-AS	3GPP HTTP Adaptive Streaming solution as specified in [TS26234], section XYZ and [TS26244] section XYZ.

3 Media Presentation Description

3.1 MPD Format

The Media Presentation Description (MPD) SHALL be as specified in [TS26234] with the following extensions and additional requirements:

- The MPD SHALL be an XML file that SHALL validate against the schema in Appendix A,
  Note that the XML schema in Appendix A imports the schema specified in [TS26234]. This means that an MPD that does not use any of the OIPF specific extensions will validate against both the schema defined in [TS26234] as well as Appendix A.

- A <pss:Representation> element may carry the @xsi:type-attribute set to “oipf:RepresentationType”. In this case the <pss:Representation> MAY carry the following additional elements and attributes:

  - @partial. If present and set to true, then this attribute indicates that the <Representation>-element is not necessarily a complete presentation, but one or more individual tracks/components (video, audio, subtitles, etc.) which may be downloaded and provided to the decoder in addition to content being downloaded from other <Representation> elements. In this case the <Representation>-element SHALL contain one or more <Component> elements, as specified in section 3.1.1, for each elementary stream contained in the <Representation> Note that it is the responsibility of the application to select the desired components and to setup the decoder accordingly. Annex B contains an informative description of how this can be done.

  - @switchGroup. The value of this attribute will be the same for representations that are different bitrate versions of the same content. Two representations of the same type but different content (e.g. audio at two different languages) will have different values.

- A <pss:SegmentInfoDefault>-element may carry the @xsi:type-attribute set to "oipf:SegmentInfoDefaultType">. In this case the < pss:SegmentInfoDefault > MAY carry the following additional elements and attributes:

  - <pss:InitialisationSegmentURL>. As specified in [TS26234], this element carries a reference to an Initialisation Segment. If present in the <pss:SegmentInfoDefault> in the <pss:Period>, the referenced Initialisarion Segment shall carry the metadata that describes the samples for all representations (partial and complete) e.g. moov box for MP4, PAT/PMT for TS. If the MPD contains <Representation>-elements with the @partial set to “true”, then the MPD SHALL also carry this element

An example instance of the OIPF compliant MPD with the constraints from section 3.2 is depicted in Figure 2.

```
<?xml version="1.0" encoding="UTF-8"?>
<pss:MPD xmlns:oipf="urn:oipf:adaptive_streaming"
 xmlns:pss="urn:3GPP:metadata:2009:PSS:HTTPStreaming"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:3GPP:metadata:2009:PSS:HTTPStreaming file:/C:/KVPrj/OIPF/Streaming/Editor/OIPF-MPD-004.xsd"
pss:minBufferTime="P1Y2M3DT1H10M0S" >
      <pss:Period>
            <pss:Representation xsi:type="oipf:RepresentationType" pss:bandwidth="5000000" pss:mimeType="video/mp4"
            pss:startWithRAP="true" switchGroup="1" partial="true">
                  <pss:SegmentInfo>
                        <pss:Url pss:sourceURL="http://www.aService.com/aMovie/HQ/Seg1.3gs"/>
                        <pss:Url pss:sourceURL="http://www.aService.com/aMovie/HQ/Seg2.3gs"/>
                        <pss:Url pss:sourceURL="http://www.aService.com/aMovie/HQ/Seg3.3gs"/>
                  </pss:SegmentInfo>
                  <component type="video" id="1"/>
                  <component type="audio" id="5" lang="en"/>
            </pss:Representation>

            <pss:Representation xsi:type="oipf:RepresentationType" pss:bandwidth="2500000" pss:mimeType="video/mp4"
            pss:startWithRAP="true" switchGroup="1" partial="true">
                  <pss:SegmentInfo>
                        <pss:Url pss:sourceURL="http://www.aService.com/aMovie/LQ/Seg1.3gs"/>
                        <pss:Url pss:sourceURL="http://www.aService.com/aMovie/LQ/Seg2.3gs"/>
                        <pss:Url pss:sourceURL="http://www.aService.com/aMovie/LQ/Seg3.3gs"/>
                  </pss:SegmentInfo>
                  <component type="video" id="2"/>
                  <component type="audio" id="5" lang="en"/>
            </pss:Representation>

            <pss:Representation  xsi:type="oipf:RepresentationType" pss:bandwidth="125000" pss:mimeType="video/mp4"
                  pss:startWithRAP="true" partial="true">
                  <pss:SegmentInfo>
                        <pss:Url pss:sourceURL="http://www.aService.com/aMovie/french/Seg1.3gs"/>
                        <pss:Url pss:sourceURL="http://www.aService.com/aMovie/french/Seg2.3gs"/>
                        <pss:Url pss:sourceURL="http://www.aService.com/aMovie/french/Seg3.3gs"/>
                  </pss:SegmentInfo>
                  <component type="audio" id="6" lang="fr"/>
            </pss:Representation>

            <pss:SegmentInfoDefault  xsi:type="oipf:SegmentInfoDefaultType">
                  <InitialisationSegmentURL pss:sourceURL="http://www.aService.com/aMovie.3gp"/>
            </pss:SegmentInfoDefault>

      </pss:Period>
</pss:MPD>
```

Figure 2: Example of the MPD

3.1.1 Component element

Element/Attribute	Description	Optionality
Component	This element contains a description of a component.	
@id	the value of this identifier is a container specific identifier for the elementary stream (e.g. the PID for MPEG2-TS or the trackID for MP4).	O
@type	specify the component type. It shall be one of Video, Audio or Subtitles	M
@lang	specify language code according to RFC 5646 [r3] for audio and subtitles stream. It is optional.	O
@angle	specify camera angle for video stream.	O
@channels	specify audio channels for audio stream.	O
@impaired	specify if there is audio description or subtitles for hearing impaired	O

Table 1: Component Element and Atributes

3.2 Segmentation Constraints

The OITF SHALL support at least Segments created as specified in [TS26234] with the following constraints:

- Segments SHALL start with a random access point (RAP) and as a consequence all @startWithRAP attributes in all <Representation> elements in the MPD SHALL be set to 'true'.
- Byte Ranges SHALL NOT be used as a mechanism for identifying Segments. As a consequence the elements 'InitializationSegmentURL' and 'Url' SHALL not include the optional attribute 'range'. Note that this does not preclude the use of HTTP requests with Byte Ranges to retrieve parts of a Segment.
- Segments of representations containing different bitrate variants of the same content will be time aligned, enabling seamless switching between them. As a consequence the attributes 'segmentAlignmentFlag' and 'bitstreamSwitchingFlag' SHALL be present and set to 'True' in all 'Period' elements in the MPD
- If two <InitialisationSegmentURL>–elements have the same value in the sourceURL attribute, then the referenced init-data SHALL be the same. Consequently the client does not need to download the init-data twice.

Note that this means that if a service chooses to segment a Content Resource in a way that does not meet these constraints, then the Content Resource might not be supported on all receivers.

4 Adaptive Media Formats

4.1 MPEG-2 Transport Stream Systems layer

If the Representation@mimeType attribute equals “video/mpeg” the media of a representation is encapsulated in MPEG2-Transport Stream packets and the carriage of A/V content and related information SHALL be in compliance with the [AVC] requirements on usage of the MPEG2-TS systems layer format, with the remarks, exceptions and additional requirements listed in the following sections

4.1.1 PID allocation

- Each elementary stream that represents a unique alternate version of a specific content component SHALL have the same PID (transport stream header) and the same stream_id (PES packet header) in all representations, global or partial, in which it is included. Obviously, this also includes different bitrate versions of the same component. Example: audio in Spanish is always PID 34, in all representations where it is present, at any bitrate. If elementary streams are encoded with different codecs (e.g. SD content uses a different codec than the HD content), then the representations SHALL use different PIDs and stream_ids.

- Components (Elementary Streams) that represent different A/V content for the user SHALL have different PIDs. Components that represent the same A/V content for the user with the same codec but different qualities (different bitrate, sampling frequency, resoultion, etc.) SHALL have the same PID. Some examples

    - "audio in Spanish" and "audio in English" have different PID

    - "audio in English" and "audio description for impaired in English" have different PID

    - "audio description in English at 64kbps" and "audio description in English at 128kbps" have the same PID

    - "video angle 1 in H.264 at 720x576" and "video angle 1 in H.264 at 320x288" have the same PID.

- When the segments of a representation contain MPEG2 TS packets, the value of the id attribute in each Component element, if present, SHALL be the PID of the Transport Stream packets which carry the component

4.1.2 Program Specific Information

- The PAT and PMT, either contained in the initialization segments or in the media segments, SHALL always contain the full list of all elementary streams, present or not in the representation where that PAT/PMT appear; global representations (those without @partial attribute, or with it set to “false”) SHALL therefore have the same PAT/PMT as any partial representations. It will be responsibility of the application to apply in the decoder the required PID filters for the components which are effectively being retrieved through the HTTP adaptive protocol.

- If the media segments do not contain PAT and PMT tables, the initialization segment SHALL be present and declared in the MPD, pointing to a resource containing transport stream packets with at least a PAT and a PMT

4.1.3 Access unit signaling

- When a transport stream packet is a random access point as defined by ISO/IEC 138181-1 it is STRONGLY RECOMMENDED that the transport stream header includes an adaptation field with the random_access_indicator set to 1.

- When the frame that starts in a transport stream packet is an Intra frame it is STRONGLY RECOMMENDED that the transport stream header includes an adaptation field with the elementary_stream_priority_indicator bit set to 1.

- It is RECOMMENDED that all transport streams packets where a video frame start carry an AU_information data field as defined in document ETSI TS 101 154 v.1.9.1. This data field SHOULD include at least the AU_coding_type_information.

- The inclusion of the above signalling SHALL be used in a consistent manner for all components in all segments for a content item.

4.1.4 Media packaging

- A media segment SHALL contain the concatenation of one or several contiguous PES which are split and encapsulated into TS packets. Media Segments SHALL contain only complete PES packets.
- When packetizing video elementary streams, up to one frame SHALL be included into one PES packet. Larger frames may be fragmented into multiples PES packets. The PES packet where the frame starts SHALL always contain a PTS/DTS header fields in the PES header.
- PTS and DTS values SHALL be time aligned across different representations.

4.1.5 Content protection

- [AVC] specifies two methods to protect (encrypt) MPEG-2 transport streams referred to as BBTS and PF. Each stream has its own Transport Stream and SHALL comply with either BBTS or PF. Thus each Transport Stream can be processed alone.

To facilitate switch from one stream to another and to minimize impact on player:

- PIDs (for PMT, ECM) SHOULD be the same for each TS.
- Crypto-period boundaries SHALL be the same for each TS and Control Words SHALL be the same for each TS.

4.2 MP4 File Format Systems layer

If the Representation @mimeType attribute equals “video/mp4”, then the carriage of A/V content and related information (e.g. subtitles) SHALL be in compliance with the [AVC] requirements on usage of the MP4 systems layer format, with the following remarks, exceptions and additional requirements:

- For every Representation, a [TS26234] Initialisation Segment SHALL be available.
  - For all non-partial Representations, a reference to the Initialisation Segment SHALL be present in a <InitialisationSegmentURL>-element in the <Representation>-element.
  - For all Partial Representation a reference to a single common Initialisation Segments SHALL be present in a <InitialisationSegmentURL> in the <SegmentInfoDefault> element.
  - An Initialisation Segment SHALL be delivered with MIME type “video/mp4”.
  - Initialisation Segments SHALL be formatted as specified in [TS26234], section 12.4.2.2. For every media stream of the (set of partial) Representation (s), the *moov*-box in the Initialisation Segments SHALL contain a *trak*-box describing the samples of the media streams in compliance with [ISOFF].
- Every Representation SHALL consist of Media Segments that are formatted as specified in [TS26234], section 12.4.2.3.
  - A Media Segment SHALL be delivered with MIME type “video/vnd.3gpp.segment” as specified in [TS26244]
  - To allow a client to seek to any Segment with a certain index and start playback with perfect audio/video synchronization, every *traf*-box of a track that contains audio SHOULD contain a [TS26244] *tfad*-box. The contents of the box SHALL be such that if the client starts the playback of the audio samples of a Segment as specified in the box, then the audio and video of the Segment are played in perfect sync.
- The Initialisation Segment and the Media Segments are formatted such that a file that consists of the Initialisation Segment and an arbitrary selection of Media Segments of the (set of partial) Representation(s), stored in order of the *sequence_number* in their *mfhd*-box, is an [ISOFF] compliant file. (Note that this statement assumes that [ISOFF] allows for ‘gaps’ in the *sequence_number*s of consecutive *moof*-boxes; i.e. the difference in the *sequence_number* of consecutive *moof*-boxes may be larger than one.)

An informative annex on the use of the MP4 file format systems layer is provided in Appendix C.

4.3 Video

4.4 Audio

4.5 Subtitles

5 Protocols

5.1 Live streaming

Streaming of live content will be done following the rules described in [TS26234]: the MPD may be updated periodically at the interval described in the MPD, and successive versions of the MPD will be guaranteed to be identical in the description of segments that are already in the past. The synchronization of terminals and the live streaming server will be addressed by external protocols such as NTP or equivalent.

If a timeshift window is specified in the MPD, it may be used by the terminal to know at any moment which segments are effectively available for downloading. If this timeshift information is not present in the MPD, the terminal may assume that all segments described in the MPD which are already in the past are available for downloading. This may be used to implement network PVR functionality on events which have not finished yet.

Periods may be used in the live streaming scenario to appropriately describe successive live events with different encoding or adaptive streaming properties. Timeshift will still be possible across the boundaries of such events, provided that the timeshift window is large enough.

5.2 Trick play

6 Content & Service protection

Appendix A. MPD Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:oipf="urn:oipf:adaptive_streaming"  xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:oipf:adaptive_streaming" xmlns:pss="urn:3GPP:metadata:2009:PSS:HTTPStreaming">
    <xs:import namespace="urn:3GPP:metadata:2009:PSS:HTTPStreaming" schemaLocation="3gppMPD.xsd" />

    <xs:complexType name="RepresentationType">
        <xs:complexContent>
            <xs:extension base="pss:RepresentationType">
                <xs:sequence>
                    <xs:element minOccurs="0" maxOccurs="unbounded" name="component" type="oipf:ComponentType"/>
                </xs:sequence>
                <xs:attribute name="switchGroup" type="xs:unsignedInt" use="optional"/>
                <xs:attribute name="partial" type="xs:boolean" use="optional"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="SegmentInfoDefaultType">
        <xs:complexContent>
            <xs:extension base="pss:SegmentInfoDefaultType">
                <xs:sequence>
                    <xs:element maxOccurs="1" minOccurs="0" name="InitialisationSegmentURL" type="pss:UrlType"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="ComponentType">
        <xs:attribute name="id" type="xs:string" use="optional"/>
        <xs:attribute name="type" type="xs:string" use="required"/>
        <xs:attribute name="lang" type="xs:string" use="optional"/>
        <xs:attribute name="angle" type="xs:string" use="optional"/>
        <xs:attribute name="channels" type="xs:unsignedInt" use="optional"/>
        <xs:attribute name="impaired" type="xs:boolean" use="optional"/>
    </xs:complexType>

</xs:schema>
```

Appendix B. Component Management

The <Representation> element defined in [TS26234] corresponds to a particular version of the full content with all its elements (video, audio, subtitles, etc). The different representations listed in the MPD correspond also to full, alternate versions that differ in a particular aspect (bitrate, language, etc). This means that the terminal will at every moment download and present segments of only one representation. While this provides a quite simple and straightforward model it has an important lack of flexibility in the following sense: if there are many alternatives for a particular component (e.g. audio at different languages) and there are also a number of different bitrate alternatives, all combinations shall be available at the server.

For instance, if a service provides 3 audio languages and the video in 4 bitrate levels, then it will need to provide 12 different representations; however, there will be groups of 3 representations which share exactly the same bulky video (they only differ in audio). This causes an important waste of storage space in the server. Even if the server can be optimized with respect to this (e.g. to build the segments in real time from the elementary streams stored separately in its disks), this cannot be done in the HTTP caches, which in general are not aware of the protocol and the similarities between the content pointed by different URLs.

In order to solve this problem, OIPF includes the concept of Partial Representations in the MPD. Presence of the @partial attribute in the <Representation> element indicates that the representation does not include all elements available for presentation at the terminal, but only a subset of them (e.g. “audio in Spanish”). An OIPF terminal will be able to identify the representations that it requires and will download their segments independently, composing the final combination at terminal side. In case of the example service above, the server may serve 4 representations with 4 different bitrate versions of a movie with English audio, and separately it can serve 2 additional representations, each with just the audio in a different language. This way, all the 12 combinations mentioned above will be possible (all bitrates at all languages) but with roughly a third of the required storage in the server and the HTTP caches.

As the multiplicity of representation happens in two dimensions (bitrate versions and alternate content), a new attribute ‘SwitchGroup’ has been added to the <Representation> element. The terminal uses this attribute to identify representations which contain effectively the same alternative version of the content, only at different bitrates. Representations that have different values in this element have different content.

This component-aware scenario relates to the process for selecting and displaying the desired set of components. This process also takes place for content that is delivered through other mechanisms, and is not specific for content that is delivered via the HTTP adaptive protocol described in this document. In the context of OIPF (for example using the DAE “Extensions to video/broadcast for playback of selected components”), this process operates based on the information from the MPEG2-TS or MP4 meta-data. This information is also available in the context of HTTP Adaptive Streaming in the Initialisation Segment.

This leads to the following informative process for component selection in case of content that is delievered using this specification:

1. Retrieve the MPD. If the MPD includes both, decide if you want to play the partial or non-partial Representations. It is RECOMMENDED to use the partial representations.
2. In case of a non-partial Representation:
    a) Based on meta-data in the MPD (typically the @bandwidth-attribute), select an initial Represenation.
    b) If present, retrieve the Init Segment of the Representation.
    c) Retrieve Media Segments of the chosen Representation.
    d) Find the elementary streams in the downloaded Init Segment / Media Segments. Typically select one video and one audio stream. If there are options, select from those.
    e) Setup the "player" to play the selected elementary streams. Play them.
    f) While playing, allow the user to select from other/additional elementary streams in the Init Segment / Media Segments.
    g) If you want to switch to a different bitrate, select an alternate non-partial Representation and continue from step 2b.
3. In case of a Partial Representation:
    a) Based on meta-data in the MPD (typically the @bandwidth-attribute and the <Component>-element) select the initial Represenations.
    b) If present, retrieve the Init Segment of the Period.
    c) Retrieve Media Segments of the chosen Representations.
    d) Based on the @id's of the <Components>-elements, or using information from the Init Segment, setup the "player" to play the selected elementary streams. Play them.
    e) While playing, allow the user to select from other/additional Components/elementary streams. If other/additional streams are selected, continue from step c.
    f) If you want to switch to a different bitrate select an alternate partial Representation in the same switchGroup and continue from step C.

Notice that the initialization segments will always contain the full description of all component alternatives, so it will be guaranteed that there is no identifiers conflicts between them (e.g. two languages with the same MPEG-TS PID or MP4 trackID). The parsing of this initialization segment and the corresponding settings of the chipset to select the appropriate components will be responsibility of the application (the media player).

Appendix C. Usage of The MP4 File Format

C.1 Audio/Video Synchronization

Unlike MPEG2-TS, the MP4 system layer ([ISOFF]) does not define a system clock or global timestamps that link the various elementary streams to the system clock. Instead, every track has its own independent timeline, specified based on the durations of each sample. The decoding and composition time of a sample is calculated by summing up the durations of all samples since the start of the track.

In the context of adaptive streaming (and especially in case of live streaming), a client may want to start playback at any point in the content without having access to the durations of all samples since the start of the track. This would not be a problem if at the start of each segment, audio and video would be perfectly aligned. This however is not possible, because video frames and audio frames are typically unequal in duration. Consequently, a segment that contains in integer number of audio and video frames will not have equal durations of audio and video data. This problem is illustrated in the following example.

Say for example that a movie consists of an audio and a video elementary stream, where the video is sampled at 25fps and the audio is sampled at 48 Khz and framed using 1024 audio samples per frame. This means that the duration of a video frame is 40 ms and the duration of an audio frame is 21,33 ms. Say also that these elementary streams are delivered using this specification and the MP4 system layer, with the following parameters:

- timescale as specified in the *mvhd*-box: 25 (“ticks” per second)
- timescale as specified in the *mdhd*-box of the video track: 25
- timescale as specified in the *mdhd*-box of the audio track: 48000
- segment duration as specified in the MPD: 2 seconds

For this case, **Table 2** gives an overview of the allocation of audio and video frames to the first 12 segments of this movie:

Table 2: Example Audio/Video Synchronization

Segment index	0	1	2	3	4	5	6	7	8	9	10	11	12
Video start time (ticks)	0	50	100	150	200	250	300	350	400	450	500	550	600
Audio start time (ticks)	0,00	50,13	100,27	149,87	200,00	250,13	300,27	349,87	400,00	450,13	500,27	549,87	600,00
#Video frames	50	50	50	50	50	50	50	50	50	50	50	50	50
#Audio frames	94	94	93	94	94	94	93	94	94	94	93	94	94
Video duration (ticks)	50	50	50	50	50	50	50	50	50	50	50	50	50
Audio duration (ticks)	50,13	50,13	49,60	50,13	50,13	50,13	49,60	50,13	50,13	50,13	49,60	50,13	50,13

So in this example audio and video are perfectly aligned in segments 0, 4, 8 and 12. However if a client seeks to for example segment 5, then it would need to delay play-out of the audio for 0.13 ticks or 5ms compared to the video to achieve perfect audio video synchronization.

To signal this to the client, [TS26244] specifies the *tfad*-box, which this specification recommends to insert into the audio track. **Figure 3** depicts a close up of the situation at the start of segment 5 in this example:

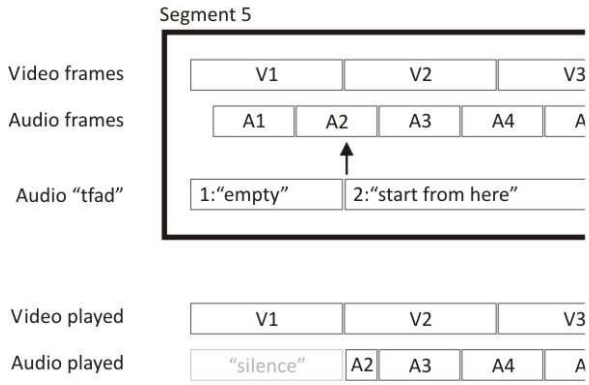

Figure 3: Example *tfad*-box

The *tfad*-box allows to specify to add empty-time into a track at the accuracy of the timescale of the *mvhd*-box, which in this example is 25 and equal to the video. The *tfad*-box also allows specifying to skip certain samples of a track at the timescale of the track, which is this example is 48000. To achieve perfect AV sync in this example, segment 5 would therefore need to include a *tfad*-box with the following contents:

- Entry 1 ("1:empty" in **Figure 3**):
  - segment_duration= 1
  - media_time= -1 (i.e. "empty" time)
- Entry 2 ("2:start from here" in **Figure 3**):
  - segment duration=99
  - media_time=1664

A client that starts playing at Segment 5 could use this box to synchronize audio and video, which will result in the playing of the samples as depicted in the bottom half of **Figure 3**. A client that continues playing the content from Segment 4 assumingly already has synchronized the audio and video tracks and should ignore the *tfad*-box and add the samples of Segment 5 back to back with the samples of Segment 4.

C.2 Partial Representations

Via Partial Representations, this specification allows services to offer the various elementary streams of a presentation as separate downloads/streams (see Appendix B). In this case it is required that there is one single Initialisation Segment describing the samples in all Media Segments of all Partial Representations and that the concatenation of the Initialisation Segment and the Media Segments is an [ISOFF] compliant file. This section illustrates how such requirement can be met by working out the example of Appendix B in combination with the MP4 system layer.

In this example a service offers a video in four bitrates and audio in three languages (English, French and German), where two languages (French and German) are offered for separate retrieval as separate Representations. Figure 2 depicts a potential allocation of movie and track fragments to segments and representations for the first few segments of this example.

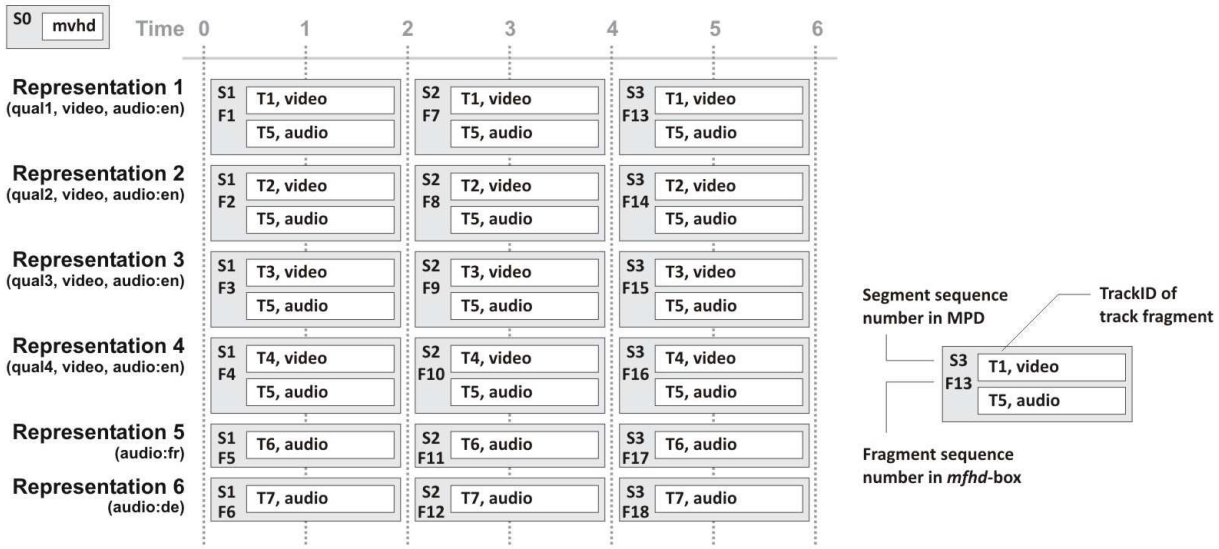

Figure 4: Partial Representation Example

Each Segment has a sequence number in the MPD and contains a single movie fragment with a certain sequence number in the *mfhd*-box. Segments of Representations 1 though 4 contain both audio (English) and video track fragments. In this example the service has chosen to put each video track on its own TrackID, such that each track can have its own set of configuration parameters in the *mvhd*-box.

If a client selects to retrieve German audio and starts playing the video at a quality 3 while increasing the quality with each consecutive segment, then it will retrieve the sequence of Segments as depicted in Figure 3.

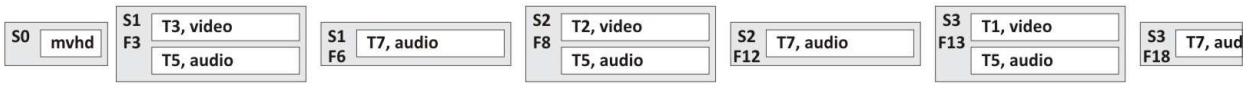

Figure 5: Partial Representation Retrieval

When stored as depicted (Initialisation Segment first, Media Segments in sequence of the Fragment sequence number) this is a valid [ISOFF] file. Note that to play this file, the client will need to switch to those video tracks that were selected for download.

Note that the MPD could also include four additional non-partial Representations, that reference the same Media Segments as Representations 1 through 4 in this example, and the same (or a different) Initialisation Segment. In this way the same service (and the same HTTP caches!) can be used for Clients that do not support partial representations.