INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29
WG11 N15072
February 2015, Geneva, Switzerland

Title **Technical Note: AAC Implementation Guidelines for DASH**
Source **Communication**
AhG on Responding to Industry Needs on Adoption of MPEG Audio

ABSTRACT

This document describes implementation guidelines for delivering MPEG-4 Advanced Audio Coding (AAC) using MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH). It is closely aligned with the DASH-AVC/264 Interoperability Point Document defined by the DASH Industry Forum (DASH-IF) but adds additional guidelines and background information regarding the recommended usage of AAC-LC, HE-AAC, HE-AACv2 and MPEG Surround. It is intended primarily for encoding equipment vendors and service providers who are offering DASH products and services based on AAC.

Executive Summary

After providing a technology review of DASH and AAC, the document presents guidelines for implementing a “DASH-ready” AAC encoder with seamless switching capability. Most importantly amongst those recommendations are that all Representations within an Adaptation Set are required to keep the Audio Object Type (AOT), Channel Configuration (mono, stereo, 5.1, 7.1) and Sampling Frequency (48 kHz, 44.1 kHz) constant within an Adaptation Set. In addition, Stream Access Points (SAPs) need to be created by the AAC encoder at the beginning of each Segment in a standard compatible way by restricting certain coding tools and options. This includes the selection of the window type and sequence (Start- or Short Window), the adjustment of the core bandwidth for Spectral Band Replication (SBR), and the avoidance of time differential coding for SBR and Parametric Stereo (PS) headers. Alongside those guidelines for segment encoding, the correct signaling in the Media Presentation Description (MPD) is explained in detail, i.e. how to set attributes and elements such as @codecs, AudioChannelConfiguration, @audioSamplingRate, etc.
In the ensuing chapter, the document continues with guidelines for broadcasters and service providers addressing the selection of suitable AAC profiles and bit rates. HE-AAC is recommended as the most suitable AAC profile for DASH because of its high coding efficiency over a broad range of bit rates, its excellent audio quality and universal platform support. Other profiles, such as AAC-LC and HE-AACv2, are only recommended for special use cases. Typical bit rates are provided for DASH streaming of stereo, 5.1 and 7.1 multichannel content, including guidelines for designing Adaptation Sets with multiple bit rates. However, it is frequently the case that audio bit rate adaptation is not needed and only the video bit rate should be adapted. In this case it is sufficient to offer only a single audio Representation in a single Adaptation Set, for instance by using HE-AAC at 160 kbps for 5.1 audio, which simplifies the encoding process. Only if audio takes up a significant share of the total media bit rate or it is an audio-only service may audio adaptation and multiple

Representations be considered. Recommendations about the use of multiple Adaptation Sets and the handling of backward compatibility conclude the document.

Table of content

Abbreviations and Terms

AAC	Advanced Audio Coding Audio codec (family) standardized by MPEG
Adaptation Set	DASH terminology for a set of encodings from the same content at different bit rates. For bit rate adaptation, clients switch between Representations in one Adaptation Set
AOT	Audio Object Type AAC coding tools or algorithms (e.g. SBR or PS)
ASC	Audio Specific Configuration Short byte string defining the AAC encoder config, needed by decoder during init
AU	Access Unit MPEG terminology for an encoded AAC audio frame, typ. 20-80 ms worth of audio
DASH	Dynamic Adaptive Streaming over HTTP Media streaming protocol standardized by MPEG
Fragment	Structure in the MP4 File Format allowing step-by-step storage MP4 fragments correspond to DASH Segments when stored in the MP4 File Format.
HTTP	Hypertext Transfer Protocol Protocol enabling the WWW, typ. used by browsers to fetch web pages, based on TCP
MP4	MPEG-4 File Format File format for storing MPEG-4 codecs (AAC, H.264/AVC), often used in DASH
MPD	Media Presentation Description The DASH manifest, an XML-encoded index-file, defines codecs and URLs of Segments
MPEG	Moving Picture Expert Group Organization standardizing multimedia technology, e.g. MP3, AAC, H.264/AVC, DASH
Period	DASH terminology for a long-lasting content item, e.g. a song or video clip/program Period boundaries mark a discontinuity in content and allow codec re-configuration
Profile	MPEG-4 defines interoperable codecs by combining useful AOTs into one Profile. The most important AAC Profiles are AAC, HE-AAC, and HE-AACv2.
PS	Parametric Stereo (AOT 29) AAC coding tool, parametric extension from mono to stereo using low bit rate side info
Representation	DASH terminology for a specific encoding of a content item Multiple Representations at different bit rates form an Adaptation Set
RTP	Realtime Transport Protocol Protocol used for streaming over UDP, today mainly used in voice over IP
SAP	Stream Access Point DASH terminology for random access point or Intra-frame, support seamless switching

SBR	Spectral Band Replication (AOT 5) AAC coding tool, parametric extension of audio bandwidth using low bit rate side info
Segment	DASH terminology for a short part of a media item, typically 2-10 seconds duration Clients may switch Representations (bit rate) at Segment boundaries
Transparency	A coded audio signal is called transparent if it cannot be distinguished from the original. Audio codecs can typically reach transparency by increasing the bit rate.

1. Introduction

This document describes implementation guidelines for delivering MPEG-4 Advanced Audio Coding (AAC) using MPEG Dynamic Adaptive Streaming over HTTP (DASH). It is closely aligned with the DASH-AVC/264 Interoperability Point [11] as defined by the DASH Industry Forum [12] but adds additional guidelines and background information regarding the recommended usage of AAC. The ISO base media file format is assumed as a transport format, but all guidelines for AAC-encoding also apply to the MPEG-2 Transport Stream profiles of DASH.

2. Technology Review

In this section we review DASH and AAC as the two basic technology components. The intention is to provide enough background information for the reader to understand the design decisions, which are detailed later. The relevant terms and definitions are introduced and the most important concepts are explained. The expert reader may skip directly to Section 3 for the actual implementation guidelines.

2.1 MPEG-DASH

Dynamic Adaptive Streaming over HTTP (DASH) is a media streaming protocol standardized by MPEG [1], which enables high quality streaming of multimedia content over the Internet using conventional HTTP infrastructure and servers [13]. It enables seamless adaptation to changing network conditions, which eliminates the risk of buffering experiences that can frustrate users. A good introduction on MPEG DASH is also available in [13].

The basic idea of MPEG-DASH is to send audio and video as a series of small files, typically containing about 2-10 seconds worth of media, called media Segments. An index file, or playlist, called the Media Presentation Description (MPD) provides the client with the URLs to the Segments. This allows the client to control the media delivery by requesting the Segments using HTTP and splicing them together before decoding and play-out. Because the media is encoded at several bit rates, the client can adapt the download speed to the available bit rate on the channel. As a result, buffer underruns and re-buffering events can be reduced significantly. Since media is delivered as a series of HTTP downloads, DASH can make effective use of existing HTTP infrastructures, with widely deployed HTTP servers able to be reused instead of installing special media servers. In addition, HTTP caches and proxies for efficient content delivery can be reused in existing Content Delivery Networks (CDNs). Finally, problems with firewalls and Network Address Translation (NAT) are greatly reduced compared to RTP based streaming. Fig. 2.1 illustrates the overall system architecture of DASH, which is explained in more detail in the following paragraphs.

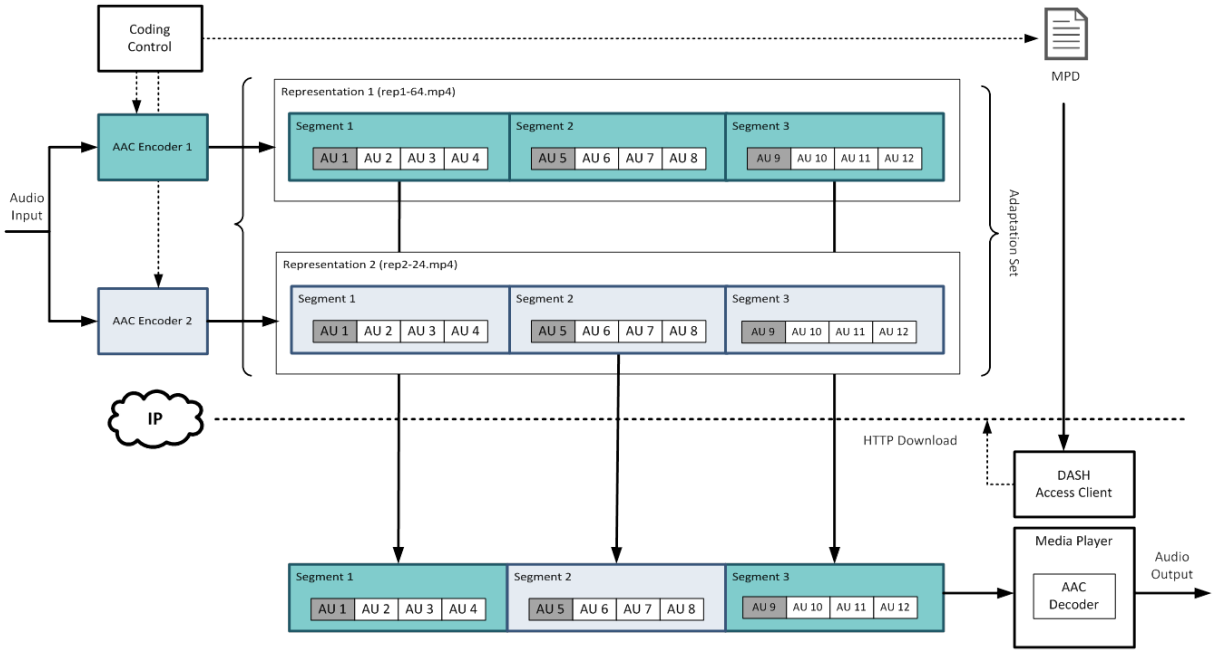

Figure 2.1: DASH System Overview(Image source: Fraunhofer IIS)

The example in Fig. 2.1 assumes that audio is encoded at two bit rates: one bit rate for normal operation (red, e.g. 64 kbps) and a second one for fallback operation during network congestion (blue, e.g. 24 kbps). Those two encodings are called Representations in the DASH terminology and are typically stored in two mp4 files (e.g. rep1-64.mp4, rep2-24.mp4). All Representations, which are encodings of the same content that can be used by the client to dynamically adapt the bit rate during streaming, are grouped into one Adaption Set.

In order to allow switching between Representations, they are divided into Segments with a typical duration of 2-10 seconds. The example in Fig. 2.1 shows three Segments per Representation, each containing four Access Units (AUs), i.e. AAC encoded audio frames (in practice this number is much higher, e.g. ~100 AUs for a segment of 2 seconds duration). Although it is possible to split the mp4 files and store individual Segments in individual files, DASH also allows keeping all Segments in a single mp4 file, accessing them with byte-range requests. We assume the latter option in the following, which reduces the number of files on the server and simplifies signaling in the MPD.

While Segments divide a single media item (e.g. a song or video clip) into small chunks, DASH can also handle extended presentations by concatenating several media items. Those long lasting media items are called Periods in DASH. Since Periods mark a discontinuity in the content, typically accompanied by a fade to silence in audio or a fade to black in video, it is possible to re-configure or switch the media codec at Period boundaries without interfering with a seamless operation.

The encoding of DASH content for seamless streaming requires a common Coding Control and certain capabilities in the AAC encoder that are described in this paper. In particular, it is required to start each Segment with a Stream Access point (SAP). In video, I-frames or IDR-frames are well-known to allow random access and are therefore suitable as SAPs. For audio, however, it is less obvious how to construct an SAP. There is no special frame type in AAC allowing true random access. However, SAPs can be generated by constraining the encoder as detailed in Section 3.1. In Fig. 2.1 the regular insertion of SAPs at the beginning of each Segment is indicated by the gray AUs.

The generation of content on the encoder side comprises two primary stages. Firstly, the actual media is encoded into Segments and stored in the corresponding MP4 files. During a second stage, the MPD is generated which describes how the Segments are encoded and how they can be accessed through HTTP downloads. This signaling information is encoded in XML and stored in the MPD file. The media Segments and their description in the MPD must correspond to each other and form the DASH content package.

On the client side, the content is typically processed as follows. Firstly, the MPD is downloaded by the DASH access client, which is responsible for scheduling the downloads, i.e. when to download which Segment. After analyzing the MPD, it decides which Segments should be downloaded first, e.g. the first Segment of the first Representation as shown in Fig. 2.1. After the download is complete, the Segment is passed to the Media Player API, which will then commence decoding and play-out. The download of the first segment also allows an estimation of the available bit rate on the channel, which is used by the HTTP access client to schedule further downloads. If required, it will switch the Representation in order to better adapt to the channel conditions. Note that the AAC decoder within the Media Player is not re-initialized when switching Representations. Instead, the same instance is running continuously and is unaware of any switching process. For example, the AAC decoder illustrated in Fig. 2.1 decodes AU-4 from Encoder-1 followed by AU-5 from Encoder-2. This switching of bit streams is a unique feature of DASH and requires special consideration during the encoding process. The solution presented in this document is fully standard compatible and avoids decoder changes.

2.2 MPEG-4 AAC

Advanced Audio Coding (AAC) has become one of the most popular audio formats worldwide. In order to better understand how AAC and DASH operate together, some background information on the different profiles of the AAC family and the corresponding signaling are provided below.

2.2.1 AAC Codec Family

AAC is standardized by MPEG as part of the MPEG-4 framework for advanced multimedia systems. Although AAC is also standardized in MPEG-2, this paper focuses on *MPEG-4 AAC*. It is important to understand that AAC is not a monolithic codec but exists in different profiles and levels, such as AAC-LC and HE-AAC, which together construct the *AAC family*. The standard defines a hierarchy of Tools, Audio Object Types and Profiles to specify these codecs.

Audio Object Types

The members of the AAC family share a common coding framework but differ in the specific algorithms that are used to extend the capabilities of the base algorithm. Those algorithms are characterized by the Audio Object Type (AOT), which typically influences the coding efficiency or behavior in case of transmission errors. Tab. 2.1 lists the most important MPEG-4 AOTs with relevance for DASH. For a complete list of audio object types please see [2].

AOT	Abbreviation	Description
2	AAC-LC	AAC Low Complexity
5	SBR	Spectral Band Replication
29	PS	Parametric Stereo
30	MPS	Parametric Surround

Table 2.1: *Audio Object Types (AOT) of MPEG-4 AAC with relevance for DASH*

AAC Low Complexity (AAC-LC) can be seen as the base algorithm on which further extensions of the AAC family are based.

Spectral Band Replication (SBR) is a coding tool which allows expanding the audio bandwidth using a low bitrate parametric data stream. A core codec is employed to encode the audio signal at a reduced bandwidth, e.g. 12 kHz, and SBR is used to expand this signal to e.g. 24 kHz audio bandwidth. The SBR parameters can be embedded as auxiliary data in a backward compatible way using only a fraction of the total bit rate.

Parametric Stereo (PS) is a coding tool which allows the expansion of a mono signal into stereo. Similar to SBR, it uses a low bit rate parametric data stream, which is transmitted as side information at a fraction of the total bit rate.

MPEG Surround (MPS) is a parametric surround coding tool which allows the expansion of a mono or stereo signal into a multichannel signal. It follows the same principle as PS but pursues the reproduction of surround sound at very low bit rates.

Through these flexible coding tools, the AAC family of codecs supports a wide range of suitable bit rates for any type of application, ranging from maximum efficiency on mobile networks to transparency for download, broadcast or high-quality streaming applications.

Profiles

Since AOTs can be combined in multiple ways, it is necessary to define practical bundles to facilitate testing and assure interoperability. The selection and combination of AOTs from the overall AAC toolbox is achieved in the MPEG-4 standard by defining *Profiles*. Those Profiles are the main means to specify MPEG-4 audio codecs in system specifications or implementations. Tab 2.2 lists the most important AAC Profiles and contained AOTs for the scope of DASH. For a more complete definition of MPEG-4 Audio Profiles we refer to [2].

MPEG-4 Profile	Contained AOTs
AAC	2
HE-AAC	2+5
HE-AAC v2	2+5+29
MPEG Surround	2+5+30

Table 2.2: *MPEG-4 Audio Profiles with relevance for DASH*

Advanced Audio Coding (AAC): The AAC Profile includes only the AAC-LC AOT and defines the baseline AAC codec. This “plain vanilla AAC” is best known for its use in the iPod and iTunes music and movies, and can be implemented for mono, stereo and surround signals up to 48 channels. It can scale up to transparent audio quality. Though the profile name is strictly speaking “AAC”, it is often referred to as “AAC-LC” in order to be more explicit about the used AOT.

High Efficiency AAC (HE-AAC): The HE-AAC Profile is the most relevant profile for DASH and employs AAC-LC as a core codec in combination with SBR. For example, AAC-LC is used to encode an audio signal of 12 kHz audio bandwidth at 48 kbps and SBR is used to expand this signal to 24 kHz audio bandwidth. The SBR parameters can be embedded as auxiliary data in the AAC-LC data stream in a backward compatible way. The HE-AAC

decoder will play AAC-LC and HE-AAC bit streams in best quality while a legacy AAC-LC decoder w/o SBR support is still able to decode the AAC-LC core of a HE-AAC bit stream but will produce a low bandwidth signal.

The HE-AAC audio codec has become one of the most important enabling technologies for state-of-the-art multimedia systems. Thanks to its unique combination of high-quality audio, low bit-rates and audio-specific metadata support, it is the perfect audio solution even over channels with limited capacity, such as those commonly encountered in broadcasting or streaming. HE-AAC's coding efficiency enables it to deliver the same audio quality at one-half or one-third the bit rate of other audio codecs. For example, it can provide high-quality stereo audio at bit rates as low as 32 kbps and also scale up to full transparency when required. The excellent multi-channel audio performance of HE-AAC was confirmed by an extensive, independent listening test conducted by the European Broadcast Union (EBU) that resulted in the "broadcast quality" label for excellent quality at only 160 kbps. As a result, HE-AAC is the ideal surround audio codec for flawless adaptive streaming over MPEG-DASH, with there being no need to switch to stereo when bandwidth is constrained.
HE-AAC decoding is natively supported by the leading mobile and desktop operating systems, streaming platforms and HTML5 browsers, and has been deployed in more than 7 billion consumer electronics devices.

High Efficiency AAC, Version 2 (HE-AAC v2): The HE-AACv2 profile extends HE-ACC with the PS coding tool and allows the transmission of stereo signals at extremely low bit rates, e.g. 24 kbps. The fully backwards-compatible HE-AACv2 decoder will play AAC-LC, HE-AAC and HE-AACv2 bitstreams in best quality while a the HE-AAC decoder w/o PS support is still able to reproduce a mono signal out of an HE-AACv2 bitstream.

Levels
Finally, the MPEG standard defines *Levels*, which typically limit the number of output channels or sampling rate for a given Profile. For example, a Level-2 HE-AACv2 decoder is required to decode stereo up to 48 kHz sampling rate, whilst a Level-4 decoder must be able to decode five channels at the same sampling rate. A Level-6 decoder is capable of decoding up to 7.1 discrete channels of audio whilst the standard supports up to 48 channels.

2.2.2 AAC Transport and Signaling

Independent of the selected profile, any MPEG-4 AAC encoder produces Access Units (AUs) that contain compressed audio data. Those raw AUs need to be multiplexed and packaged before they can be transmitted over DASH. Although DASH can operate with several transport formats, we focus on the MPEG-4 file format [9] as the most common option. Furthermore, the MPEG-4 AAC decoder needs to be initialized correctly before it can decode AUs. For this purpose it needs the so-called Audio Specific Configuration (ASC) that's explained in the next section. For a general overview of AAC transport formats, please refer to [10].

Audio Specific Configuration
An MPEG-4 AAC decoder requires knowledge of the selected coding tools and their exact configuration before it can decode the AU stream. All required information is contained in the Audio Specific Configuration (ASC), which is a short byte string or data structure. Besides the AOTs, the ASC includes the fundamental audio parameters, such as sampling rate, frame length or channel configuration. As illustrated in Fig. 2.2 the AAC encoder typically generates the ASC after it is initialized with input parameters (e.g. bit rate and number of channels etc.). The AAC decoder is then initialized with this ASC before it begins decoding

AUs and producing audio output. Hence, any transport system, including DASH, must convey the ASC to the decoder. It is important that the ASC and AUs match in order to prevent decoder failure. As a consequence, it is not recommended to "hard wire" ASCs.

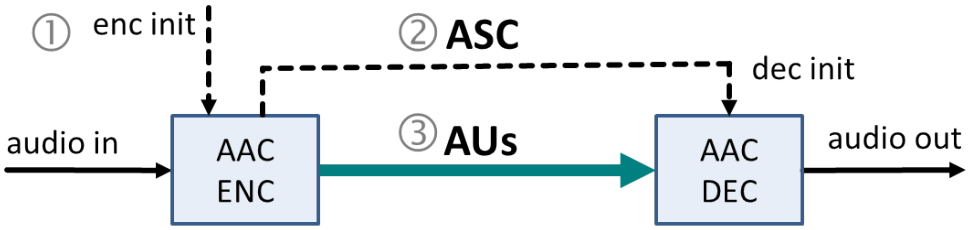

Figure 2.2: *Initialization of AAC decoder with Audio Specific Config (ASC) from AAC encoder (Image source: Fraunhofer IIS)*

MPEG-4 File Format

The most common transport format for AAC in DASH is the MPEG-4 File Format (MP4) [9], which is based on the ISO Base Media File Format (ISOBMFF) [8]. ISO files are structured in a hierarchical, object-oriented manner, utilising the "box" as the basic element. For DASH, each Representation is stored in a single MP4 file, which is structured as illustrated in Fig. 2.3 (building on the example in Fig. 2.1).

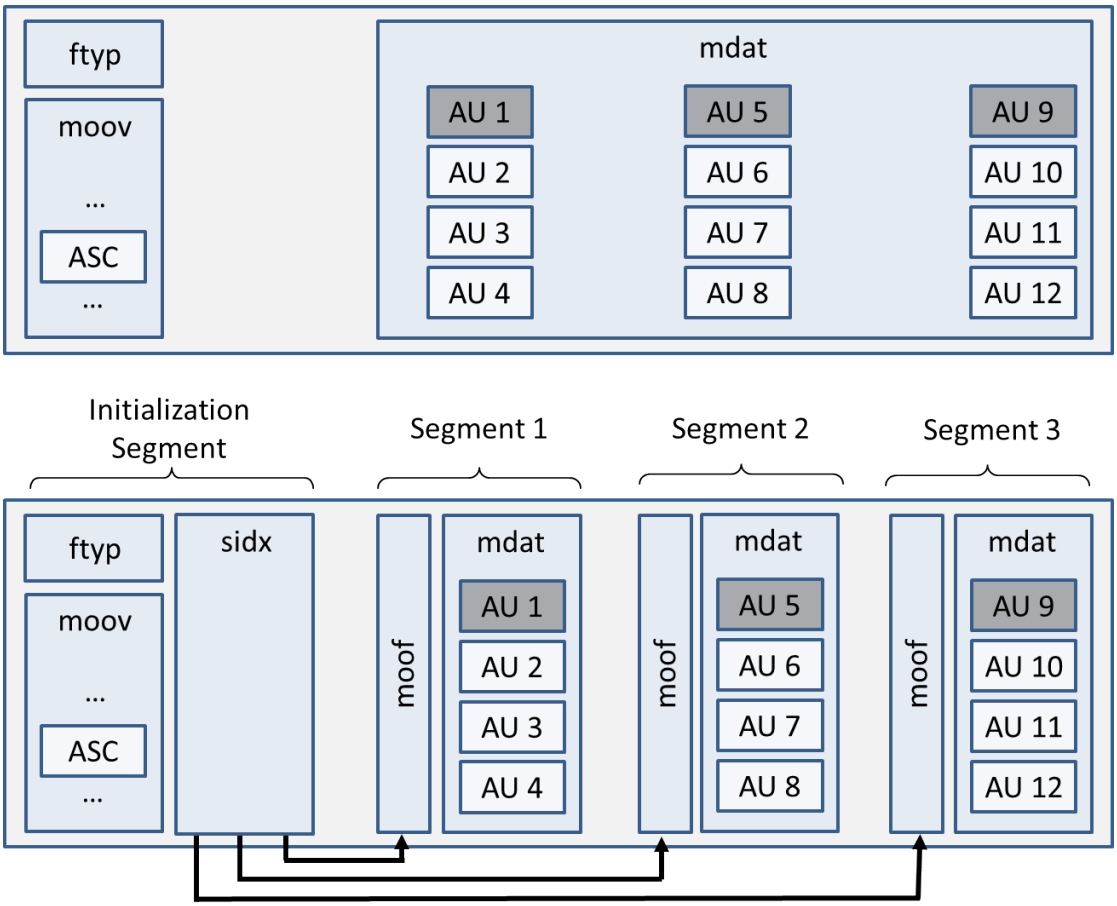

Figure 2.3: *MP4 file structure for a single Representation, before (top) and after (bottom) fragmentation and segmentation. (Image source: Fraunhofer IIS)*

In summary, the File Type Box (ftyp) specifies the file type and compatibility. The Movie Box (moov) contains all metadata and can be understood as the header of the file. As an example, the ASC is stored in the moov box (in lower levels of the box hierarchy). For a regular MP4 file, the only remaining box on the top level is the Media Data Box (mdat),

which contains all media data, i.e. AAC AUs. For DASH, however, the mdat is split into fragments, where each MP4 fragment corresponds to one DASH Segment. In addition, each fragment is preceded by a corresponding Movie Fragment Box (moof), which contains extra header information for each fragment. The process of splitting an MP4 file into several fragments is called “fragmentation”. Finally, DASH defines an additional Segment Index Box (sidx), which contains the byte offsets to each fragment so that those can be accessed directly without sequential parsing. Adding the sidx box to an MP4 file is a process known as “segmentation”.

It should be noted that DASH enables several options for storing Segments in MP4 files. For example, it is also possible to store each fragment in a separate file, which makes the sidx Box superfluous. The description in this document follows the DASH-264/AVC guidelines [11], which currently seem to be the most accepted in the industry. With this approach, the first part of the MP4 file (‘moov’ + ‘sidx’) is called the “Initialization Segment”, which is loaded before any media segments.

The generation of content is often achieved in two steps. Firstly, a regular MP4 file is generated with a single mdat box. In a second step, the file is fragmented and segmented. As fragments and SAPs must be aligned across all Representations of an Adaptation Set, this two-step process must be executed with care. In particular, it is required to identify the SAPs in the original MP4 file and then generate the fragments accordingly. It is therefore recommended to include at least the fragmentation in the first step, which makes it easier to align SAPs with the start of a fragment. A more elegant approach is the use of "sample groups" to explicitly signal SAPs in the MP4 file format. In general, sample groups allow the assignment of common attributes to a set of AUs which are scattered across the time-line. In this case, the common attribute is the property of being an SAP. Support for SAP sample groups is specified in Amendment 4 of the current ISOBMFF specification [15].

3. Implementation Guidelines

As illustrated in Fig. 2.1, a DASH content package consists of two main components: the media segments, which are stored in several mp4 files, and the MPD describing and referencing them. Both must comply with certain constraints and match each other in order to enable interoperability and seamless switching. As a first step, the encoding of AUs and their encapsulation into MP4 files has to assure segment alignment and SAP generation. In a second step, the encoded and encapsulated segments must be referenced correctly in the MPD. The following two sections explain what needs to be considered in each step.

3.1 Segment Encoding

As explained in Section 2.1, MPEG-DASH assumes that segments from different Representations can be spliced together and processed by the decoder. From the perspective of the AAC decoder, this is equivalent to switching between two bitstreams that originate from two different encoder instances, see Fig. 2.1. This process of bitstream switching shall be as seamless as possible, i.e. the user shall not hear any audible artefacts when the DASH client decides to adapt the bit rate of the stream. Because bitstream switching is not a common requirement for other transport technologies, special care must be taken during encoding. In particular, the configuration of all encoder instances must be consistent and adhere to specific constraints. In addition, all encoder instances have to generate SAPs synchronously at segment boundaries.

The guidelines in this section are mainly relevant for codec developers implementing an AAC encoder and assume detailed knowledge of the AAC standard beyond the background material provided in Section 2.2. Application developers considering the codec as a 'black box' may use this information for education and for clarifying the requirements with third party AAC vendors. In addition, all test vectors available at [14] and [12] can be used as a reference for implementation. It is important to note, that all guidelines are standard-compatible and do not require any changes in the operation of the decoder; only the encoding process should be optimized. Broadcasters and service providers may focus on the content generation guidelines described in Section 3.3.

3.1.1 General Considerations and Requirements

Period Boundaries: Firstly, it should be noted that the constraints documented below only apply to bitstream switching within a Period. At Period boundaries no constraints apply and the service provider can freely select and change the codec and/or configuration. This is the case because a Period boundary usually signals a change in content, e.g. when the next song starts or an advertisement is inserted ("ad insertion"). As this content change involves a natural discontinuity of the signal, typically including a fade to black and silence, any codec reconfiguration is possible, i.e. it is possible to change codec parameters that might also trigger discontinuities in the decoded signal through reconfiguration. Within Period boundaries, however, a switch of Representations shall be seamless, leading to the following constraints:

Codec Constraints: In order to guarantee seamless switching, the following parameters should not be changed within an Adaptation set:

1. Audio Object Type (AOT)
2. Channel Configuration
3. Sampling Frequency

This means in particular that all Representations within an Adaptation Set shall use the same AOTs, i.e. all Representations use either AAC-LC or HE-AAC or HE-AACv2. It is e.g. not recommended to use AAC-LC in one Representation and HE-AAC in another Representation of the same Adaptation Set. However, it is possible to offer several Adaptation Sets, e.g. one for HE-AAC and another one for AAC-LC, see Section 3.3.

The requirements about channel configuration and sampling frequency assure that the raw PCM output format remains constant and does e.g. not require a re-configuration of the audio device or sound card. Furthermore, a constant sampling frequency also assures identical temporal framing of AUs. As a result, the segments are aligned and no overlap or gap has to be compensated when switching.

Switching between Surround and Stereo: Switching from surround to stereo and back should be avoided within a Period for the following reasons. Firstly, a switch from surround to stereo is not perceived as seamless but will be disturbing to the listener, especially if the configuration is switching back and forth frequently. Hence, service providers typically want to have control over this behavior and not leave it up to the adaptation logic of the DASH client. Secondly, a switch of the channel configuration may cause a re-configuration of the output device and therefore a discontinuity as described above. Finally, the coding efficiency of the AAC family enables surround sound at bit rates as low as 64 kbps with HE-AAC and therefore allows the service to maintain surround output even under extreme bandwidth constraints. Hence, the need to switch from surround to stereo for the reason of bit rate

adaptation is eliminated. Note that an HE-AAC multichannel encoder can allocate the bit rate to the stereo channels if considered advantageous and seamlessly switch to a stereo configuration internally. Hence, there is no need to enforce this switch through an external channel configuration. The encoder can do this internally in the most efficient way while keeping the external channel configuration constant.

A more intelligent client may be able to apply resampling and upmix/downmix after decoding to keep the output format constant. It may also compensate for gaps/overlaps in the decoded PCM signal and handle a switch of AOTs. However, unless such post-processing steps are well defined in the client, they should not be relied upon. Furthermore, the Media Player API may make it difficult to integrate such post-processing steps in practice, as is e.g. the case for the Media Source Extension API of the W3C [7], which is implemented in modern HTML5 browsers (e.g. Chrome v23+ and Internet Explorer v11+). Hence, it is recommended to keep the above-mentioned parameters constant within one Adaptation Set.

Delay Alignment: All bitstreams shall be delay-adjusted. Encoder implementations may add additional delay depending on their configuration (e.g. bit rate dependent Low Pass filter with varying tap count). The delay for all configurations needs to be pre-compensated, so that all segments in an Adaptation Set have the same framing.

In addition to those general requirements for seamless switching, there are additional constraints for each AOT, which are detailed in the following sections.

3.1.2 AAC-LC

To guarantee seamless switching between different AAC-LC bit streams the following restrictions have to be taken into account.

Window Type and Window Sequence: In order to avoid artefacts from not cancelled Time Aliasing Components, the window type and window shape of the AAC-LC streams need to be synchronized across all bit streams at Segment boundaries. Hence, all streams should use a defined overlap and window shape at each Segment boundary (right window half of last frame in a Segment and left window half of the first frame in a Segment). It is recommended to use a short overlap (i.e. a Start or Short Window), as this allows for the use of either a Short or a Long Block in the SAP Frame (depending on signal characteristics). The correct/incorrect usage of the windowing sequence is illustrated in Fig. 3.1.

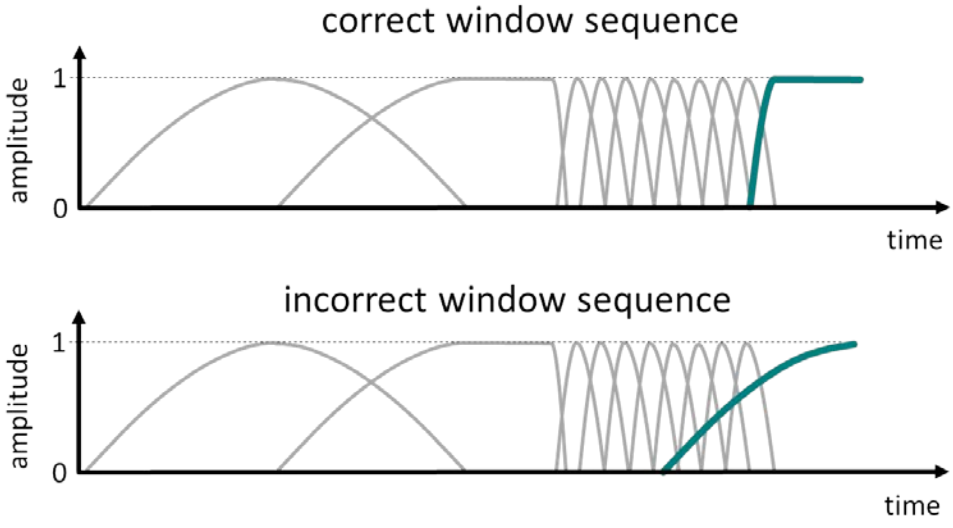

Figure 3.1: *Correct and incorrect windowing sequence for seamless switching of AAC-LC (Image source: Fraunhofer IIS)*

3.1.3 HE-AAC

As HE-AAC is based on an AAC-LC core coder, all restrictions for AAC-LC also apply to HE-AAC. However, some additional adaptations for the AAC-LC core are required to avoid drop-outs when switching between streams with different AAC/SBR cross over frequencies. Additional restrictions apply to the SBR tools.

Core Bandwidth Adjustment: The framing of the SBR decoder analysis is delayed by 6 time slots (6*64 samples) compared to the framing of the AAC core decoder. In addition, the analysis QMF adds another 320 samples. This time shift between core and SBR framing may result in an energy gap when switching between bit streams with different crossover frequencies. This typically happens when switching from a low bit rate stream to a high bit rate stream as illustrated in Fig. 3.2. To avoid this gap in the frequency range, the AAC core bandwidth of the last frame of a Segment needs to match the highest AAC/SBR crossover frequency of the supported stream configuration. To properly encode the additional bandwidth extra bits are necessary. The bit reservoir control should be adapted accordingly.

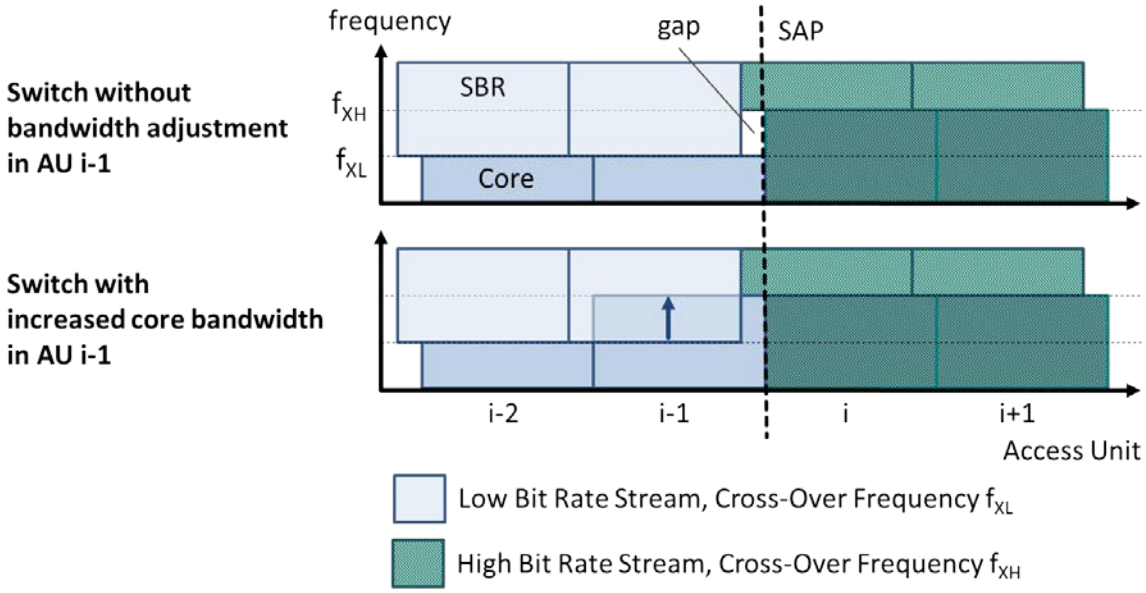

Figure 3.2: *Schematic illustration of core bandwidth adjustment for HE-AAC switching (Image source: Fraunhofer IIS)*

SBR Header and Time Differential Coding: In contrast to the AAC configuration that is completely signaled inside the audio specific configuration (ASC), the SBR decoder requires additional configuration parameters. These parameters are transmitted inside the SBR Header, which may not be contained in every access unit (AU). The MPEG-4 standard recommends a transmission interval of 500 ms or whenever an instantaneous change of header parameters is required (see [2] chapter 4.5.2.8.2.1). To allow for a seamless switching of HE-AAC bit streams it is necessary to transmit an SBR header with each SAP frame. As the MPEG-4 Audio Conformance [4] forbids the use of tools that rely on preceding frames for frames containing an SBR Header, this restriction also assures that the SAP frame can be completely decoded and processed.

SBR Frame Class: SBR envelopes can reach over frame borders, i.e. “VARVAR” and “FIXVAR” frames may overlap the SBR frame border. A SAP Frame should always start with a FIX border (“FIXVAR” or “FIXFIX”) to make sure all necessary information is available to fully decode the audio contained in that frame. Consequently the last frame in a

Segment (the frame before the SAP) should end with a “FIX” border (i.e. a “FIXFIX” or “VARFIX” frame).

3.1.4 HE-AACv2

As HE-AACv2 relies on a HE-AAC core, all restrictions listed for seamless switching of AAC-LC and HE-AAC streams are also valid for HE-AACv2. In addition, the following requirements apply to the usage of the PS tool.

PS Header and Time Differential Coding: As with the SBR payload, the PS payload Configuration Parameters may not be transmitted with every frame, but on a less regular basis. In addition, time differential coding of certain parameters can be used to increase compression efficiency. To allow for a seamless switching of HE-AACv2 bit streams, it is necessary to transmit a PS header with each SAP frame. For frames containing a PS header, the MPEG-4 Audio Conformance forbids the use of tools that rely on past information, i.e. time differential coding of parameters. With the above restriction it is therefore assured that the SAP frame can be completely decoded and processed. As HE-AACv2 conformance requires a PS header with each SBR header, this requirement is also implicitly inherited from the HE-AAC requirements.

PS Tools and Parameters: All PS Tools are designed to allow for continuous remapping of different configurations (e.g. frequency resolution of parameter bands). This is especially true for the baseline version of the PS Tool, which is the only relevant version in practice. Hence, no special care has to be taken at SAPs considering PS tools and parameters.

3.1.5 MPEG Surround

As MPEG Surround is based on an AAC-LC or HE-AAC core coder, all restrictions for AAC-LC and HE-AAC are valid. Further requirements are as follows; for details see [3].

The MPEG Surround data shall be conveyed in the AAC extension payload providing implicit signaling. Each SAP frame must contain the syntactic element SpatialSpecificConfig that contains the MPEG Surround configuration data. Additionally, the lossless coding of SAP frames shall be independent of previous frames so the bitstream payload element bsIndependencyFlag shall be set to 1. The MPEG Surround tool residual coding employs a representation of differential signals using the AAC-LC syntax. As described in the “Window Type and Window Sequence” section of this document, the window type of the residual signal shall be synchronized at the SAP.

3.2 MPD Signaling

This section describes how MPEG-4 AAC codecs are correctly signaled in the Media Presentation Description (MPD). The relevant parameters, their meaning and appropriate values for MPEG-4 AAC are explored in this chapter. Fig. 3.3 shows an example MPD for HE-AAC, which is used in the following to illustrate the relevant parameters.

```
<?xml version="1.0" ?>
<MPD
   mediaPresentationDuration="PT887.957333333S"
   minBufferTime="PT2S"
   profiles="http://dashif.org/guidelines/dash264,urn:mpeg:dash:profile:isoff-on-demand:2011"
   type="static"
   xmlns="urn:mpeg:dash:schema:mpd:2011"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011 DASH-MPD.xsd">

  <BaseURL>./</BaseURL>

  <Period>

    <AdaptationSet
       contentType="audio"
       mimeType="audio/mp4"
       codecs="mp4a.40.5"
       lang="en"
       subsegmentAlignment="true"
       subsegmentStartsWithSAP="1">

      <AudioChannelConfiguration
         schemeIdUri=" urn:mpeg:mpegB:cicp:ChannelConfiguration"
         value="2" />

       <Representation audioSamplingRate="48000" bandwidth="24000" id="sintel-24">
         <BaseURL>sintel-24.mp4</BaseURL>
         <SegmentBase indexRange="606-2776">
           <Initialization range="0-608" />
         </SegmentBase>
       </Representation>

      <Representation audioSamplingRate="48000" bandwidth="64000" id="sintel-64">
        <BaseURL>sintel-64.mp4</BaseURL>
        <SegmentBase indexRange="606-2776">
          <Initialization range="0-608" />
        </SegmentBase>
      </Representation>

      <Representation audioSamplingRate="48000" bandwidth="96000" id="sintel-96">
        <BaseURL>sintel-96.mp4</BaseURL>
        <SegmentBase indexRange="606-2776">
          <Initialization range="0-608" />
        </SegmentBase>
      </Representation>

    </AdaptationSet>

  </Period>

</MPD>
```

Figure 3.3: *Example MPD for HE-AAC Stereo*

The example assumes that an audio track (in this case, extracted from the movie “Sintel”) is encoded using HE-AAC stereo. Three Representations are used, i.e. the content is encoded three times at 24, 64 and 96 kbps, and stored in three independent MP4-files named sintel-24.mp4, sintel-64.mp4 and sintel-96.mp4, respectively. See Section 3.1 for additional requirements on generating those files and the encapsulated AAC AUs. For simplicity, no video is signaled in the example, which would require a second AdaptationSet element. The syntax of the MPD is based on XML and allows additional white space characters for formatting. In addition, the order of elements and attributes is flexible.

3.2.1 Attributes and elements for signaling AAC

@contentType: This attribute shall be set to “audio” and describes the general content type (independent from coding and encapsulation).

@mimeType: This attribute shall be set to “audio/mp4” and indicates that an MPEG-4 file is used to encapsulate the audio stream. This follows the MIME type registration for MPEG-4 as described in RFC 6381 [6].

@codecs: This attribute describes the audio codec in more detail as defined in RFC 6381 [6]. The AAC codecs are defined by an MPEG-4 Profile, which may contain multiple AOTs. In this case the “highest” AOT is used in the @codecs attribute for signaling the profile. The values for the most common AAC codecs are summarized in Table 3.1.

MPEG-4 Profile	AOT	@codecs
AAC	2	mp4a.40.2
HE-AAC	2+5	mp4a.40.5
HE-AAC v2	2+5+29	mp4a.40.29
MPEG Surround	2+5+30	mp4a.40.30

Table 3.1: *@codecs attributes for common AAC codecs*

AudioChannelConfiguration: This element describes the channel configuration, e.g. mono, stereo or surround. The @schemeIdUri attribute defines the encoding scheme of the @value attribute and is set to “ urn:mpeg:mpegB:cicp:ChannelConfiguration”. For legacy devices the attribute can also be set to: "urn:mpeg:dash:23003:3:audio_channel_configuration:2011". However, the former signaling method using the Codec Independent Code Points (CICP) is preferred because it can be extended more easily to future channel configurations (e.g. 22.2 or 7.1+4) and is therefore more future proof. The @value is then equivalent to the ChannelConfiguration as defined in ISO/IEC 23001-8 [5]. The values for common channel configurations are summarized in Table 3.2.

Channel Configuration	Speakers	Front/Surr.LTE	@value
mono	C	1/0.0	1
stereo	L,R	2/0.0	2
5.1	C, L, R, Ls, Rs, LFE	3/2.1	6
7.1	C, L, R, Ls, Rs, Lsr, Rsr, LFE	3/4.1	12

Table 3.2: *@value attribute in the AudioChannelConfiguration element*

@audioSamplingRate: This attribute describes the output sampling rate of the AAC codec as an integer value in units of Hz. Typical values are e.g. 48000 or 44100.

@bandwidth: This attribute describes the bit rate of the audio stream as an integer value in units of bit/s. Typical values are e.g. 24000 or 160000.

@subsegmentAlignment: This attribute indicates if the (sub)segments of all Representations are aligned in time and therefore no overlap or gap is introduced when switching. Though DASH can also operate w/o segment alignment it makes the implementation of clients much more complex and unreliable. As a consequence we follow DASH-264/AVC and recommend

that segments be aligned during the encoding process (see Section 3.1). Consequently, the @subsegmentAlignment attribute is then set to “true”.

@subsegmentStartsWithSAP: This attribute indicates the type of Stream Access Point (SAP) which is used for starting each (sub)segment. For seamless switching it is required that a “type 1” SAP be used during the encoding process. In alignment with DASH-264/AVC we consequently recommend encoding the AUs at segment boundaries according to the guidelines in Section 3.1 and set the attribute to “1”.

3.2.2 Seamless Switching Requirements

In order to assure seamless switching between the Representations of an Adaption Set, certain requirements must be fulfilled as described in Section 3.1 above. On the MPD level this has the consequence that the following attributes must remain constant across all Representations of an Adaptation Set:

1. Audio Object Type (as defined in the @codecs attribute)
2. Channel Configuration (as defined in the AudioChannelConfiguration element)
3. Sampling Frequency (as defined in the @audioSamplingRate attribute)

In the example of Fig. 3.1 this is automatically assured for the former two by specifying them on the Adaptation Set level, which means that they apply to all included Representations. However, they could also be specified on the Representation level as it is achieved for the @audioSamplingRate. Although the syntax of an MPD would allow different values for the above three attributes, this shall be avoided.

As discussed above, another constraint for seamless switching is that the segments are aligned between Representations and start with an SAP of type 1. The corresponding attributes @subsegmentAlignment and @subsegmentStartsWithSAP are used to signal that the segments comply with this requirement.

3.3 Content Generation Guidelines for Broadcasters & Service Providers

Although the above two sections describe how segments shall be encoded when using AAC and how those are signaled in the MPD, there is still a lot of flexibility in terms of generating DASH content. This begins with selecting the AAC Profile and appropriate bit rates but also includes the approach to backward compatibility. In the following we therefore provide additional guidelines for encoding DASH content. It should be noted that those are informal guidelines, which can be adapted to fit the need of a particular service or Standard Defining Organizations (SDOs).

Adapt video before audio: Very often, audio bit rate adaptation is not needed and it is sufficient to have a single audio Representation in a single Adaptation Set. As long as the video bit rate is significantly higher than the audio bit rate, it is more effective to adapt the video bit rate and keep the audio bit rate constant. Audio can be perceived as the base layer that is most important for the user experience and should remain undisturbed if possible. For many video services it may therefore be sufficient to use a single audio Representation (e.g. 160 kbps HE-AAC 5.1) and only adapt the video bit rate (e.g. in the range 800 – 2000 kbps). Only in instances where audio occupies a significant share of the total media bit rate or the service is audio-only should audio adaptation and multiple Representations be considered.

Services that only use a single audio Representation have no need for generating SAPs according to Section 3.1. Hence, content generation is greatly simplified.

Even if a single audio bit rate is sufficient, the service provider still has to select the preferred AAC Profile and corresponding bit rate. The following guidelines are provided to support content providers with this selection but also address the use case of multiple Representations and Adaptation Sets.

3.3.1 Selection of AAC Profiles and Bit Rates

The following section describes usage of the AAC, HE-AAC and HE-AACv2 Profiles. As explained above, those profiles are designed as backward compatible in the sense that an HE-AACv2 decoder can always decode AAC-LC and HE-AAC bit streams. However, for the encoding of a particular DASH bit stream (or Representation) the content provider has to select the most appropriate profile and bit rates.

Use HE-AAC as Default: The HE-AAC Profile can be seen as the default AAC Profile for DASH. The HE-AAC Profile can be used over a wide range of bit rates and achieves excellent audio quality for mono, stereo and surround signals. For stereo audio, it provides good quality down to 24 kbps and can improve quality consistently by adding more bits - up to 128 kbps, at which point the audio quality is excellent even for the most critical items. For 5.1 surround audio, good audio quality can be maintained down to 64 kbps and broadcast quality is commonly provided at 160 kbps. For the most critical content, bit rates can even be increased further. Hence, the complete range from lowest bit rate to highest quality can be covered with a single Adaptation Set allowing seamless bit rate switching by following the requirements described in Section 3.1.

Use of HE-AACv2: Although the “Version 2” (v2) suffix indicates advanced technology, the HE-AACv2 Profile has to be used with care and is not automatically preferred over HE-AAC (Version 1) for DASH. The benefit of HE-AACv2 is that it allows even lower bit rates than HE-AAC (v1) by using the Parametric Stereo (PS) tool. As the name implies, this tool is only applicable for stereo services. Compared to HE-AAC there is an advantage for bit rates below 32 kbps. However, it does not scale to excellent audio quality when increasing the bit rate beyond 48 kbps. Therefore HE-AACv2 is mainly recommended for audio-only services (in stereo) with strict bandwidth limitations. For video services and channels with sufficient bandwidth, HE-AAC is recommended for its greater flexibility and availability in a broad range of platforms and devices.

Use of AAC-LC: The AAC Profile is mainly of interest to music streaming services targeting the audiophile community. It may be considered if channel bit rates above 128 kbps can be guaranteed and audio quality beyond consumer needs is a primary requirement. For the vast majority of services, however, HE-AAC is the preferred AAC Profile.

Typical Bit Rates: The tables below show typical stereo, 5.1 and 7.1 bit rates for AAC, HE-AAC, and HE-AACv2 Adaptation Sets. The bit rates are suitable for a sampling rate of either 44.1 kHz or 48 kHz. Note that each profile has a sweet spot for normal operation (highlighted in grey) but can also be operated at lower and higher bit rates. The lower bit rates are fall-back modes that should only be used temporarily to cope with severe network congestion. The higher bit rates are saturation modes that may not yield significant gains in quality except for the most critical content. The bit rates are recommendations only and may be adapted for specific service requirements. In particular, a subset of the bit rates can be selected to reduce the number of the Representations resulting in a more granular adaptation.

Profile	AOT	@codecs	bit rate [kbps] for 44.1/48 kHz								
HE-AACv2	2+5+29	mp4a.40.29	18	**24**	**32**	48					
HE-AAC	2+5	mp4a.40.5		24	**32**	**48**	**64**	**96**	128	160	
AAC	2	mp4a.40.2					64	**96**	**128**	**160**	256

Table 3.3: *Typical stereo bit rates for AAC Adaptation Sets (normal operation range highlighted).*

Profile	AOT	@codecs	bit rate [kbps] for 44.1/48 kHz								
HE-AAC	2+5	mp4a.40.5	64	96	**128**	**160**	**192**	**256**	320		
AAC	2	mp4a.40.2				160	192	**256**	**320**	**384**	448

Table 3.4: *Typical 5.1 bit rates for AAC Adaptation Sets (normal operation range highlighted).*

Profile	AOT	@codecs	bit rate [kbps] for 44.1/48 kHz								
HE-AAC	2+5	mp4a.40.5	96	128	**192**	**224**	**288**	**320**	448		
AAC	2	mp4a.40.2				224	288	**320**	**448**	**512**	640

Table 3.5: *Typical 7.1 bit rates for AAC Adaptation Sets (normal operation range highlighted).*

3.3.2 Multiple Adaptation Sets

If a broad range of devices and bit rates is to be covered it is possible to offer multiple Adaptation Sets, each containing a single AAC Profile with constant configuration. The MPD will then include several Adaptation Sets with the @codecs parameter as given in Section 3.2.1. For instance, a DASH client may select the HE-AACv2 Adaptation Set when connected via 3G while using the HE-AAC Adaptation Set for WiFi or Broadband. However, this selection needs to be maintained for the duration of the current Period. At Period boundaries however, a switch of Adaptation Set is of course possible. Although the use of multiple Adaptation Sets is possible and increases flexibility in system design, it has to be done with care as the selection of the "correct" Adaptation Set by the DASH client needs to be well-defined. In general it is therefore recommended to use a single HE-AAC Adaptation Set unless there is a compelling reason to do otherwise.

3.3.3 Backward Compatibility

Backward compatibility can be supported in several ways. Firstly, it will be recalled that a DASH client supporting the HE-AACv2 Profile can also decode HE-AAC and AAC-LC, as it is for example defined in the DASH-ACV/264 Interoperability Points. In this case, the content provider can offer multiple Adaptation Sets (see Tab. 3.3) and can be certain that the DASH client will reproduce the intended audio experience.

If support for legacy clients that only support AAC-LC is desired, the content provider must create a corresponding AAC-LC Adaptation Set. As an example, he may offer an HE-AAC Adaptation Set as the default but will need to add a separate AAC-LC Adaptation Set for backward compatibility to AAC-LC devices. The legacy player will then select the AAC-LC Adaptation Set. A DASH client supporting the HE-AACv2 Profile in theory does have the choice to select either of the two Adaption Sets. However, it is recommended that the client selects the HE-AAC Adaptation Set as it supports lower bit rates at equivalent quality and therefore increases stability during network congestion.

In rare cases it may be desirable to have a single Adaptation Set and still provide backward compatibility. For example, a content creator may want to offer only a single HE-AACv2 Adaptation Set but still desires to reach legacy players supporting only HE-AAC. Even though this is possible in principle, it has certain drawbacks and should therefore be considered only as a second choice for DASH. For the given example this becomes possible through the backward compatibility of the Parametric Stereo (PS) tool, which expands a mono representation into stereo. Hence, the same bitstream can be decoded by a HE-AACv2 decoder to stereo or by a legacy HE-AAC decoder to mono. Another requirement for this mode of operation is that "Explicit Backward Compatible" signaling is used in the Audio Specific Config (ASC) to initialize the decoder (which is the case for DASH-AVC/264). Therefore, a content creator may offer an HE-AACv2 Adaptation Set and signal it in the MPD using @codecs=mp4a.40.29. The legacy player can access this Adaptation Set and feed it to the HE-AAC decoder. The drawback is that the player will only produce a mono signal. Hence, different users will experience different behavior depending on the capabilities of their clients (some hear stereo while some hear mono). The preferred alternative is to offer a second HE-AAC Adaptation Set with stereo as described in the paragraph above.

4. Conclusions

The MPEG-DASH standard in combination with the AAC Family of audio codecs and HEVC as the forthcoming standard for video coding paves the way for a bright future for adaptive streaming over HTTP, making proprietary protocols and browser plugins increasingly irrelevant. HE-AAC's suitability for seamless switching combined with its excellent coding efficiency and system intelligence makes it the ideal audio component for DASH streaming. Its native decoding support in all leading operating systems, HTML5 browsers and connected CE devices such as the Google Chromecast enables broadcasters and service provider to stream premium content to the leading platforms with premium audio quality free of licensing fees for content distribution and playback.

By following the recommendations for AAC profiles and bit rates, and by applying the coding constraints and client implementation guidelines explained in this document, seamless dynamic switching is possible and straightforward using existing decoders. Mpeg-audio.org provides an extensive set of DASH test vectors to support interested parties in the streaming media ecosystem.

References

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

[2] ISO/IEC IS 14496-3:2009, Information technology - Coding of audio-visual objects - Part 3: Audio.

[3] ISO/IEC 23003-1, Information technology - MPEG audio technologies - Part 1: MPEG Surround.

[4] ISO/IEC IS 14496-26, Information technology - Coding of audio-visual objects - Part 26: Audio conformance.

[5] ISO/IEC 23001-8, Information technology – MPEG systems technologies — Part 8: Coding-independent code points.

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

[7] W3C Working Draft, “Media Source Extensions”, http://www.w3.org/TR/media-source/

[8] ISO/IEC 14496-12:2012, Information technology – Coding of audio-visual objects – Part 12: ISO base media file format

[9] ISO/IEC 14496-14:2012, Information technology – Coding of audio-visual objects – Part 14: The MP4 File Format

[10] “AAC Transport Formats”, White paper, available at http://www.mpeg-audio.org/faq.html#whitePapers

[11] Guidelines for Implementation: DASH-AVC/264 Interoperability Points, DASH-IF, June 4, 2013

[12] DASH Industry Forum, homepage available at http://dashif.org/

[13] “MPEG-DASH: The Standard for Multimedia Streaming Over Internet”, White paper on MPEG-DASH Standard, Iraj Sodagar, April 2012

[14] MPEG-audio.org, homepage available at http://www.mpeg-audio.org/

[15] ISO/IEC 14496-12:2012, Information technology – Coding of audio-visual objects – Part 12: ISO base media file format, AMENDMENT 4: Improved audio support