APPLICATION BULLETIN

xHE-AAC IMPLEMENTATION GUIDELINES FOR DYNAMIC ADAPTIVE STREAMING OVER HTTP (DASH)

INTRODUCTION

This document describes implementation guidelines for delivering the Advanced Audio Coding (AAC) codec family using MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH). It is closely aligned with the Interoperability Point Documents 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-AAC v2 and xHE-AAC. It is intended primarily for encoding equipment vendors and service providers who are offering DASH products and services based on the AAC codec family.

Fraunhofer Institute for
Integrated Circuits IIS

Management of the institute
Prof. Dr.-Ing. Albert Heuberger
(executive)
Dr.-Ing. Bernhard Grill
Am Wolfsmantel 33
91058 Erlangen
www.iis.fraunhofer.de

Contact
Matthias Rose
Phone +49 9131 776-6175
matthias.rose@iis.fraunhofer.de

Contact USA
Fraunhofer USA, Inc.
Digital Media Technologies*
Phone +1 408 573 9900
codecs@dmt.fraunhofer.org

Contact China
Toni Fiedler
Phone +86 138 1165 4675
china@iis.fraunhofer.de

Contact Japan
Fahim Nawabi
Phone: +81 90-4077-7609
fahim.nawabi@iis.fraunhofer.de

Contact Korea
Youngju Ju
Phone: +82 2 948 1291
youngju.ju@iis-extern.fraunhofer.de

* Fraunhofer USA Digital Media Technologies, a division of Fraunhofer USA, Inc., promotes and supports the products of Fraunhofer IIS in the U. S.

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. The encoding of DASH segments is much easier when using xHE-AAC as it is inherently designed with seamless switching in mind. It provides Immediate Playout Frames (IPFs) as an explicit mechanism for SAP, which eliminates the need for restrictive encoding constraints. The guidelines in this document are fully standard compatible and decoders don’t need to be reinitialized when switching between Representations within an Adaptation Set.

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. Extended HE-AAC is recommended as the most suitable AAC profile for stereo services because it can support seamless switching across the complete bit rate range and extends services down to 12 kbit/s while also providing excellent audio quality above 128 kbit/s. In addition, xHE-AAC provides a build-in solution for loudness and dynamic range control of the audio output. For multi-channel audio and because of the universal platform support, its predecessor HE-AAC is still the most popular AAC profile for current services. Other profiles, such as AAC-LC and HE-AAC v2, 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 will 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 kbit/s 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, audio adaptation and multiple Representations should 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 initialization

AU Access Unit
MPEG terminology for an encoded AAC audio frame, typically. 20-80 ms worth of audio

DASH Dynamic Adaptive Streaming over HTTP
Media streaming protocol standardized by MPEG

CMAF Common Media Application Format
MPEG specification profiling audio-, video-, and file-formats for adaptive streaming

DRC Dynamic Range Control
Technology for controlling the dynamic range of audio signals

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, typically used by browsers to fetch web pages, based on TCP

IPF Immediate Playout Frame
Special AU in xHE-AAC allowing seamless bit stream switching, aka. “Audio I-Frame”

LCM least common multiple

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
ISO Working Group 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, HE-AAC v2, and xHE-AAC.

PS Parametric Stereo (AOT 29)
AAC coding tool, parametric extension from mono to stereo using low bit rate side information

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 (VOIP)

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.

USAC Unified Speech and Audio Coding (AOT 42)
Audio codec combing speech- and general audio-coding technologies

1 INTRODUCTION

This document describes implementation guidelines for delivering the MPEG-4 Advanced Audio Coding (AAC) family of codecs using MPEG Dynamic Adaptive Streaming over HTTP (DASH). It is closely aligned with the DASH-AVC/264 Interoperability Point [12] as defined by the DASH Industry Forum [13] 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 the AAC codec family 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. It enables seamless adaptation to changing network conditions, which eliminates the risk of buffering experiences that can frustrate users.

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.

The example in Fig. 2.1 assumes that audio is encoded at two bit rates: one bit rate for normal operation (green, e.g. 64 kbit/s) and a second one for fallback operation during network congestion (blue, e.g. 24 kbit/s). 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.

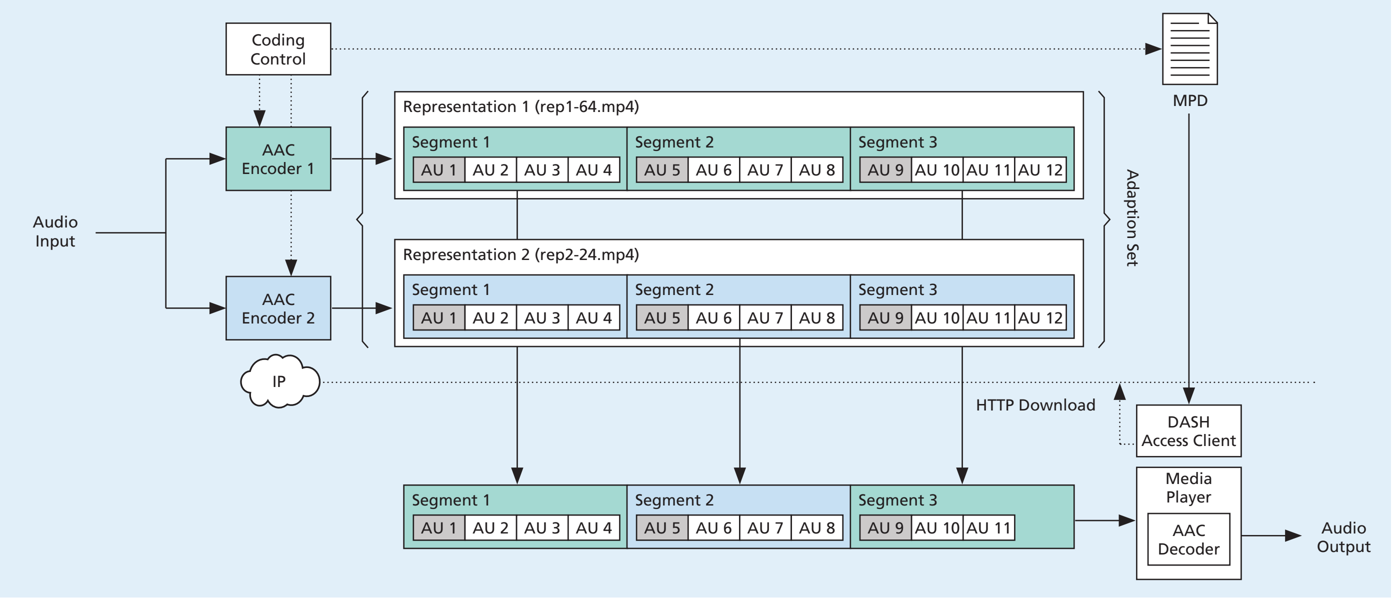

Figure. 2.1: DASH System Overview

In order to allow switching between Representations, they are divided into Segments with a typical duration of 2-10 seconds. The example in Figure 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, only Extended HE-AAC allows the creation of true SAPs. For all other AAC profiles 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 Figure 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 Figure 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 Figure 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 careful consideration during the encoding process or explicit implementations of SAPs. The solutions presented in this document are fully standard compatible and avoid decoder updates.

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. Further information on the availability and licensing of Fraunhofer's "DASH-ready" AAC implementation software can be found in [7] [8].

2.2.1 AAC CODEC FAMILY

AAC is standardized by MPEG as part of the MPEG-4 framework for advanced multimedia systems [2]. Its latest extension, Extended HE-AAC, is based on Unified Speech and Audio Coding (USAC), which is standardized in MPEG-D [14]. 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 Extended HE-AAC, which together construct the AAC family. The standard defines a hierarchy of Tools, Audio Object Types and Profiles to specify these codecs as illustrated in Figure 2.2 and detailed below.

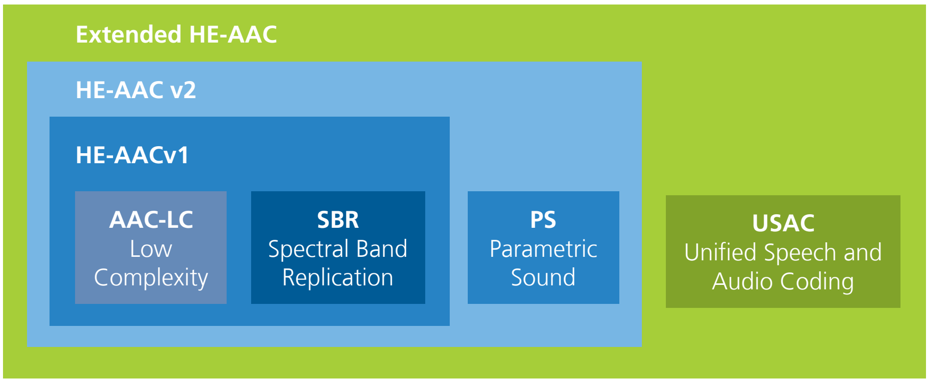

Figure 2.2: Profiles and tools of the AAC codec family

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 AOTs with relevance for DASH. For a complete list of audio object types please see [2].

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

Table 2.1: Audio Object Types (AOT) of AAC with relevance for DASH

AOT	Abbreviation	Description
2	AAC-LC	AAC Low Complexity
5	SBR	Spectral Band Replication
29	PS	Parametric Stereo
42	USAC	Unified Speech and Audio Coding

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.

Unified Speech and Audio Coding (USAC), is an audio codec combining speech- and perceptual audio-coding technologies and therefore provides a consistent high audio quality for all signal types. Especially for content including speech signals, USAC can significantly improve audio quality at very low bit rates, i.e. down to 12-24 kbit/s stereo. 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 standards by defining Profiles. Those Profiles are the main means to specify MPEG 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 Audio Profiles we refer to [2].

Table 2.2: MPEG Audio Profiles with relevance for DASH

MPEG Profile	Contained AOTs
AAC	2
HE-AAC	2+5
HE-AAC v2	2+5+29
Extended HE-AAC	2+5+29+42

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 widely used 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 kbit/s and SBR is used to expand this signal to 24 kHz audio bandwidth.

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 most popular audio solution for 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 kbit/s 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 kbit/s. As a result, HE-AAC is the ideal surround audio codec for flawless adaptive streaming over MPEG-DASH, without the 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 8 billion consumer electronics devices. Fraunhofer IIS is one of the co-developers of the HE-AAC standard, which can be used with any adaptive streaming technology including MPEG-DASH, Apple HLS, Adobe HDS and Microsoft Smooth Streaming.

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

Extended High Efficiency AAC: This profile extends HE-AAC v2 with the USAC audio codec and allows the transmission of stereo signals at even lower bit rates, e. g. 16 kbit/s. Just as the previously described AAC profiles, Extended HE-AAC is a hierarchical extension and comprises all of the above coding tools: An Extended HE-AAC decoder can decode AAC-LC, HE-AAC and HE-AAC v2 bit streams.

Extended HE AAC has a unique feature with particular relevance for DASH streaming: With the introduction of Immediate Playout Frames (IPFs) it is possible to switch seamlessly between streams of different configurations. Extended HE-AAC therefore allows to cover the complete bit rate range with one single Adaptation Set, which greatly simplifies the design of DASH services.

Additionally, and in contrast to previous AAC profiles, xHE-AAC includes a built-in solution for loudness and dynamic range control specified by the MPEG-D DRC standard [17]. MPEG-D DRC defines two profiles that are both supported by xHE-AAC. For achieving a consistent user experience across different devices and services, it is mandatory to include basic metadata sets in the xHE-AAC bit stream. This is also defined in the CMAF specification [16] and explained in more detail in Section 3.1.5 below. For further information on xHE-AAC and its applications see [15].

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-AAC v2 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 [10] 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 [11].

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. 2.3 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 allowed to "hard wire" ASCs.

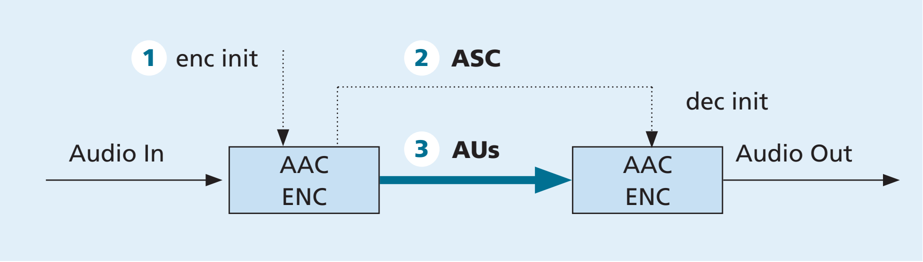

Figure 2.3: Initialization of AAC decoder with Audio Specific Config (ASC) from AAC encoder.

MPEG-4 File Format

The most common transport format for AAC in DASH is the MPEG-4 File Format (MP4) [10], which is based on the ISO Base Media File Format (ISOBMFF) [9]. ISO files are structured in a hierarchical, object-oriented manner, utilizing the "box" as the basic element. For DASH, each Representation is stored in a single MP4 file, which is structured as illustrated in Fig. 2.4 (building on the example in Fig. 2.1).

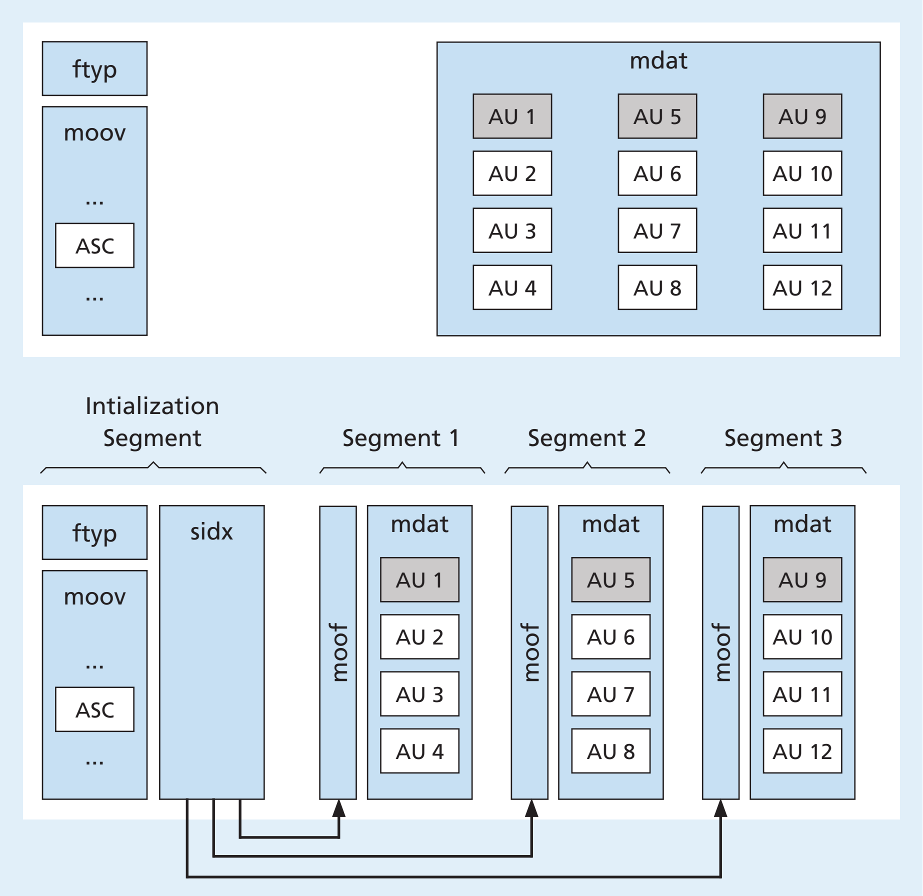

Figure 2.4: MP4 file structure for a single Representation, before (top) and after (bottom) fragmentation and segmentation

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. Our description follows the DASH-IF Implementation Guidelines [12], 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. For AAC-LC, HE-AAC and HE-AAC v2, 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. For xHE-AAC, signaling of SAPs is done similar to video, i.e. SAPs are signaled using the sync sample table and additional pre-roll information may be carried in Sample Groups. For more detail, see Section 10.6 of the ISOBMFF specification [9] and Section 3.1.5 below. This allows encoding the xHE-AAC bit stream into a “flat” (non-fragmented) MP4 file, which contains all information necessary for fragmentation at a later stage. Hence, no immediate fragmentation is necessary, which can ease the integration into existing content generation workflows.

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 bit streams that originate from two different encoder instances, see Fig. 2.1. This process of bit stream 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 bit stream switching is not a common requirement for other transport technologies, special care must be taken during codec design and/or 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. With the introduction of IPFs into the xHE-AAC profile, the switching between bit streams is supported through an explicit implementation of SAPs. This provides optimal control on the switching process but does not eliminate the need for generating synchronous SAPs 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. The AAC encoder implementation developed by Fraunhofer IIS does comply with those requirements and is therefore "DASH-ready". In addition, all test vectors available at [13] 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 bit stream 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 identical AOT, i.e. all Representations use either:

- AAC-LC (i. e. AOT 2, thus complying with the AAC profile) or
- AAC-LC with SBR (i. e. AOT 5, thus complying with the HE-AAC profile) or
- AAC-LC with SBR and PS (i.e. AOT 29, thus complying with the HE-AAC v2 profile) or
- USAC (i. e. AOT 42, thus complying with the Extended HE-AAC profile).

For example, it is not recommended to use AOT 2 (AAC-LC) in one Representation and AOT 5 (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. xHE-AAC deviates from this behavior slightly in that it allows varying frame lengths (single, double and quadruple duration).

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 kbit/s 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 [6], 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 bit streams 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 precompensated, 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).

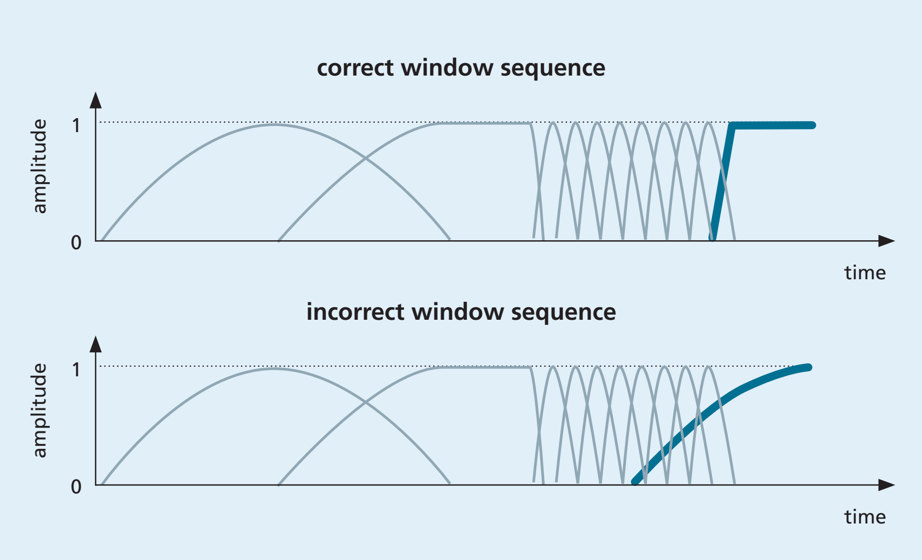

Figure 3.1: Correct and incorrect windowing sequence for seamless switching of AAC-LC

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.

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 six time slots (6 x 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.

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 [3] 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.

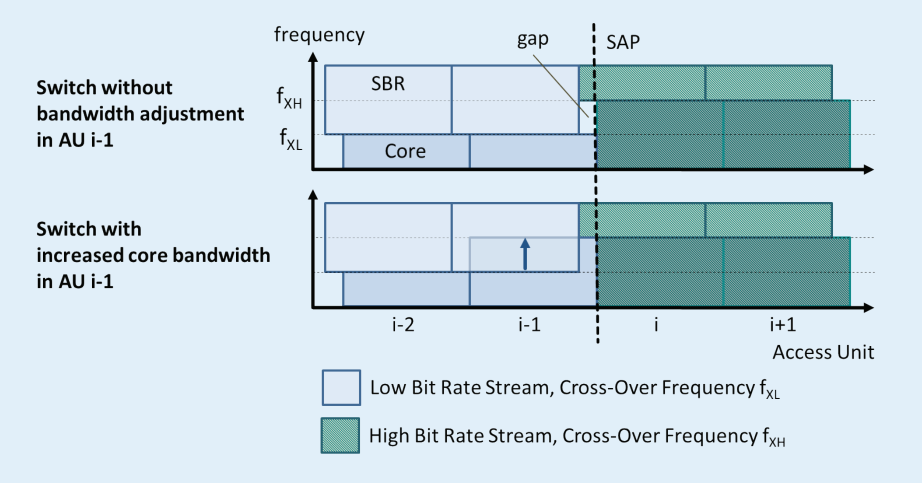
frequency
gap
SAP
Switch without bandwidth adjustment in AU i-1
f_{XH}
f_{XL}
SBR
Core
Switch with increased core bandwidth in AU i-1
i-2
i-1
i
i+1
Access Unit
Low Bit Rate Stream, Cross-Over Frequency f_{XL}
High Bit Rate Stream, Cross-Over Frequency f_{XH}

Figure 3.2: Schematic illustration of core bandwidth adjustment for HE-AAC switching

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-AAC v2

As HE-AAC v2 relies on a HE-AAC core, all restrictions listed for seamless switching of AAC-LC and HE-AAC streams are also valid for HE-AAC v2. 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-AAC v2 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-AAC v2 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 xHE-AAC

The approach for supporting bit stream switching in xHE-AAC is very different from AAC profiles because this special requirement was considered from start. Instead of imposing constraints on the encoding process and by that ensuring that SAPs are generated in an implicit way, xHE-AAC introduces Immediate Playout Frames (IPFs) as an explicit implementation of SAPs. This is very similar to I-frames or IDR-frames in video coding, which is why IPFs are sometimes also referred to as "Audio I-Frames".

Having an explicit implementation of SAPs makes it easier to identify them in the bit stream for verification and compliance but also helps in assuring optimal audio quality during the switching process. Most importantly, however, IPFs allow switching of coding tools while maintaining seamless transitions. This allows covering the complete bit rate range from 12-128 kbit/s or above in a single Adaptation Set without compromises in audio quality, where all other profiles only cover a certain range of bit rates. This greatly simplifies the design of DASH services as described in more detail in Section 3.3.

Immediate Playout Frames: The basic principle for constructing an IPF is illustrated in Fig. 3.3. In order to allow immediate playout of audio samples after reception of the current AU(n) it is necessary to include the previous AU(n-1) as an "Audio Pre-Roll" within the extension payload. This is necessary because in a regular decoding process it is assumed that the xHE-AAC decoder is in a certain state when decoding the current AU. This state information can be reconstructed when decoding the Audio Pre-Roll but would be missing otherwise. In addition, it has to be assured that the bit stream of the current AU and its Audio Pre-Roll can be decoded independently, i.e. without prediction or other dependencies to previous AUs. This is achieved by setting the usacIndependencyFlag („indepFlag" in Fig. 3.3), which resets the bit stream parsing process. As it is also possible to include the ASC within the extension payload, it is possible to change the configuration during a switch. Hence, the optimal configuration for each bit rate can be selected without negative (i.e. noticeable) impact on the stream transition. Because the construction and decoding of an IPF are specified in the MPEG standard it is not necessary to go into more detail here. It is sufficient to note that a standardized and explicit mechanism for bit stream switching is defined for xHE-AAC, which makes it the ideal choice for DASH services.

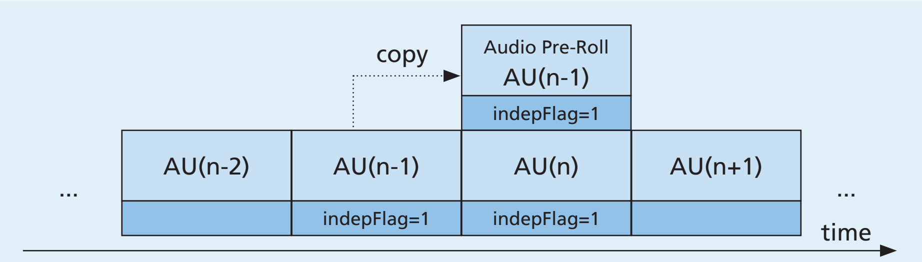

Figure 3.3: Construction of an Immediate Playout Frame (IPF).

Framing: xHE-AAC can operate in various operation modes which differ in audio frame size. This frame size is roughly correlated with the bit rate and will vary between Representations. In order to keep the segment borders time-aligned across all Representations, the length (in audio samples) of any Segment must be a multiple of the least common multiple (LCM) of all frame sizes in that Adaptation Set.

For example in a typical Adaptation Set there may be a high rate Representation with 1024 audio sample framing as well as 2048 and 4096 framing for mid and low rate Representations. Therefore all segment lengths must be a multiple of 4096, because LCM(1024, 2048, 4096) = 4096.

Formally xHE-AAC can also operate in a 768 audio sample frame mode. That mode is not recommended, not least because it increases the above mentioned segment length granularity up to 12288 audio samples.

Loudness and Dynamic Range Control: Service providers have the need to control the loudness of the overall program, e. g. when switching from/to commercials or when users are listening on different devices. In addition, it is often desired that the dynamic range of the audio signal can be adapted to the listening environment, e. g. to enhance the intelligibility of the dialog when listening in a noisy environment.

Such functionality can be provided by MPEG-D DRC [17], which defines specific loudness and DRC metadata to be included during encoding and applied during decoding. Recognizing the need for loudness and dynamic range control in practical applications and following the CMAF specification [16], xHE-AAC is bundled with MPEG-D DRC for DASH services: xHE-AAC segments must contain basic metadata sets conforming to the Loudness Control Profile or to the Dynamic Range Control Profile, Level 1 or higher as specified by MPEG-D DRC.

It is up to the service provider to include additional metadata that provides flexibility in the trade-off between functionality and bit rate overhead. However, if DRC metadata is included in one Representation, consistent DRC metadata must also be included in all other Representations of that Adaptation Set. For example, if one Representation includes a specific DRC sequence, all other Representations must also include a DRC sequence for the same use case to allow for smooth audio transitions when switching between Representations while DRC is active. Otherwise, the decoder might produce loudness discontinuities when switching between Representations.

It is important to note that xHE-AAC decoders must implement the Dynamic Range Control Profile of MPEG-D DRC to guarantee consistent behavior of the DASH service.

3.2 MPD SIGNALING

This section describes how MPEG AAC codecs are correctly signaled in the Media Presentation Description (MPD). The relevant parameters, their meaning and appropriate values for MPEG AAC are explained in this chapter. Fig. 3.4 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>

      </AdaptationSet>

    </Period>

  </MPD>
```

Figure 3.4: 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. Two Representations are used, i. e. the content is encoded two times at 24 and 64 kbit/s, and stored in two independent MP4-files named sintel-24. mp4 and sintel-64.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 Adaptation Set 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 as described in RFC 6381 [5].

@codecs: This attribute describes the audio codec in more detail as defined in RFC 6381 [5]. The AAC codecs are defined by an MPEG 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.

Table 3.1: @codecs attributes for common AAC codecs

MPEG Profile	AOT	@codecs
AAC	2	mp4a.40.2
HE-AAC	2+5	mp4a.40.5
HE-AAC v2	2+5+29	mp4a.40.29
Extended HE-AAC	2+5+29+42	mp4a.40.42

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 [4]. The values for common channel configurations are summarized in Table 3.2.

Channel Configuration	Speakers	Front/Surr.LFE	@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". For the case of xHE-AAC this means that each segment must start with an IPF.

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. 3.4 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 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 kbit/s HE-AAC 5.1) and only adapt the video bit rate (e. g. in the range 800 – 2000 kbit/s). 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, HE-AAC v2 and Extended HE-AAC Profiles. As explained above, those profiles are designed as backward compatible in the sense that an Extended HE-AAC decoder can always decode AAC-LC, HE-AAC and HE-AAC v2 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 Extended HE-AAC for Best Performance: The Extended HE-AAC Profile is the most advanced AAC Profile for DASH with dedicated support for seamless switching. For stereo audio, it provides good quality down to 16 kbit/s and can improve quality consistently by adding more bits – up to 256 kbit/s, at which point the audio quality is excellent for typical items. For the most critical content, bit rates can be increased further with continuous gains until audio quality for professional requirements is achieved. For channels with severe bit rate constraints it is possible to operate at a fallback of 12 kbit/s (or 8 kbit/s mono), which is e.g. of particular interest for mobile streaming over 2G-networks in emerging markets. The audio quality at each bit rate is equal or superior to any other AAC profile. As a unique feature with particular relevance to DASH, Extended HE-AAC can provide seamless switching between the complete range of bit rates while other profiles can only support a limited range. Hence, the complete range from lowest bit rate to highest quality can be covered with a single Adaptation Set and full support for seamless switching. As Extended HE-AAC has its particular strength at very low bit rates, its main application area are stereo services for which those can be achieved.

Use HE-AAC for Universal Platform Support: Though Extended HE-AAC is technically superior, its predecessor HE-AAC currently provides the advantage of universal platform support. Hence, HE-AAC is the second best choice in case Extended HE-AAC is not yet available on a given platform. Though the bit rate range of the HE-AAC Profile is reduced compared to Extended HE-AAC, it still supports a wide range over which it can provide excellent audio quality for mono, stereo and surround signals. For stereo audio, it provides good quality down to 32 kbit/s and can improve quality consistently by adding more bits - up to 128 kbit/s. Due to the usage of SBR, the audio quality saturates beyond this point and adding bit rates provides only diminishing returns. For 5.1 surround audio, good audio quality can be maintained down to 64 kbit/s and broadcast quality is commonly provided at 160 kbit/s.

Use of HE-AAC v2: The "Version 2" of the HE-AAC Profile is an extension based on the Parametric Stereo (PS) tool, which allows lower bit rates for stereo services. Though it does have an advantage over HE-AAC (Version 1) for bit rates below 32 kbit/s, it is now superseded by xHE-AAC, which has the same advantage but additional benefits. The inherent problem of HE-AAC v2 is that it cannot scale to excellent audio quality when increasing the bit rate beyond 48 kbit/s. Therefore, HE-AAC v2 is only recommended if Extended HE-AAC is not available and even then it should not be automatically preferred over HE-AAC (Version 1) which has greater flexibility and broader support on platforms and devices. In summary, HE-AAC v2 can be seen as an outdated profile with a very limited use for DASH services.

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 kbit/s can be guaranteed and audio quality beyond consumer needs is a primary requirement. For the vast majority of services today, 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, HE AACv2 and Extended HE-AAC 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 fallback 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. Note that the legacy members of the AAC codec family (AAC-LC, HE-AAC, HE-AAC v2) have a limited bit rate range for optimal operation, while Extended HE-AAC can cover the full range without any drawbacks in audio quality and further extends the range towards lower bit rates.

MPEG Profile	AOT	@codecs	bit rate [kbit/s] for 44.1/48 kHz
HE-AAC v2	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
Extended HE-AAC	42	mp4a.40.42	12 16 24 32 48 64 96 128 160 256

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

MPEG Profile	AOT	@codecs	bit rate [kbit/s] for 44.1/48 kHz
HE-AAC	2+5	mp4a.40.5	64 96 128 160 192 256 320 384
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)

MPEG Profile	AOT	@codecs	bit rate [kbit/s] 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

For DASH services based on xHE-AAC a single Adaptation Set is sufficient to cover the whole range of bit rates, which makes the design of the service very simple. However, if xHE-AAC is not available for all clients then multiple Adaptation Sets can be used to cover a broader range of bit rates and/or to achieve backwards compatibility to legacy devices.

Extend bit rate range without xHE-AAC: In this approach, each Adaptation Sets contains 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-AAC v2 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 DASH clients needs to be well defined. When using xHE-AAC, this can be simplified.

Extend service to legacy clients not yet supporting xHE-AAC: Multiple Adaptation Sets can also be used to achieve backwards compatibility to legacy clients. For example, a service provider may want to use xHE-AAC for best performance but also wants to reach legacy clients that only support HE-AAC. In this case, he may offer an xHE-AAC Adaptation Set as the default but adds a separate HE-AAC Adaptation Set for backward compatibility to HE-AAC devices. The legacy player will then automatically select the

HE-AAC Adaptation Set as it must ignore codecs it does not support. A DASH client supporting xHE-AAC in theory does have the choice to select either of the two Adaption Sets. However, it is recommended that the client selects the xHE-AAC Adaptation Set because of the superior performance of xHE-AAC with respect to quality and bit stream switching.

3.3.3 Backward Compatible Signaling

The parametric coding tools of the HE-AAC profile (SBR) and HE-AAC v2 profile (PS) can beused in theory to achieve backwards compatibility to older AAC profiles. However, use of this feature for designing DASH services is strongly discouraged and should be strictly avoided. More specifically, signaling of the SBR and PS coding tools in the HE-AAC(v2) profiles should always be either "explicit hierarchical" or "explicit backward compatible". The legacy "implicit" signaling shall be strictly avoided and is not supported in any application standard such as DASH-IF, DVB DASH or MPEG CMAF. The concern with this signaling mode is that consistent client behavior cannot be guaranteed and that DASH-clients may initialize the decoder incorrectly during session setup.

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. xHE-AAC's suitability for seamless switching over a broad range of bit rates combined with its excellent coding efficiency and built-in loudness and dynamic range control makes it the ideal audio component for DASH streaming. Until xHE-AAC is broadly available on platforms and devices, HE-AAC can be used as a transitional solution. 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. Fraunhofer IIS, as a member of the DASH-Industry Forum, provides an extensive set of DASH test vectors, "DASH-ready" AAC encoder implementations and know-how to all its licensees and partners in the streaming media ecosystem.

REFERENCES

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

[2] ISO/IEC 14496-3 (MPEG-4 Part 3),
Information technology – Coding of audio-visual objects – Part 3: Audio.

[3] ISO/IEC 14496-26 (MPEG-4 Part 26),
Information technology – Coding of audio-visual objects –
Part 26: Audio
conformance.

[4] ISO/IEC 23091-3,Information technology – Coding independent code points –
Part 3: Audio.

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

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

[7] HE-AAC – The codec of choice for broadcast and streaming, Fast Facts, available at
http://www.iis.fraunhofer.de/en/bf/amm/download/whitepapers.html

[8] Fraunhofer IIS MPEG Audio Software, White Paper, available at
https://www.iis.fraunhofer.de/en/ff/amm/impl.html

[9] ISO/IEC 14496-12:2015 (MPEG-4 Part 12),
Information technology – Coding of audio-visual objects – Part 12: ISO base
media file format

[10] ISO/IEC 14496-14:2012 (MPEG-4 Part 14),
Information technology – Coding of audio-visual objects –
Part 14: The MP4 File Format

[11] AAC Transport Formats, Application Bulletin, available at
http://www.iis.fraunhofer.de/en/bf/amm/download/whitepapers.html

[12] DASH-IF Interoperability Documents, available at http://dashif.org/guidelines/

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

[14] ISO/IEC 23003-3 (MPEG-D Part 3),
Information technology – MPEG audio technologies –
Part 3: Unified speech and audio coding

[15] Extended High Efficiency AAC (xHE-AAC) product page, available at
http://www.xhe-aac.com

[16] ISO/IEC 23000-19 (MPEG-A Part 19),
Information technology – Multimedia application format –
Part 19: Common media application format (CMAF)

[17] ISO/IEC 23003-4 (MPEG-D Part 4),
Information technology – MPEG audio technologies –
Part 4: Dynamic range control

INFORMATION IN THIS DOCUMENT IS PROVIDED ‘AS IS‘ AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
INFORMATION IN THIS DOCUMENT IS OWNED AND COPYRIGHTED BY THE FRAUNHOFER-GESELLSCHAFT AND MAY BE CHANGED AND/OR UPDATED AT ANY TIME WITHOUT FURTHER NOTICE. PERMISSION IS HEREBY NOT GRANTED FOR RESALE OR COMMERCIAL USE OF THIS SERVICE, IN WHOLE OR IN PART, NOR BY ITSELF OR INCORPORATED IN ANOTHER PRODUCT.

ABOUT FRAUNHOFER IIS

The Audio and Media Technologies division of Fraunhofer IIS has been an authority in its field for more than 25 years, starting with the creation of mp3 and co-development of AAC formats. Today, almost all consumer electronic devices, computers and mobile phones are equipped with Fraunhofer’s media technologies. Besides the global successes mp3 and AAC, the Fraunhofer technologies that improve consumers’ audio experiences include Cingo® (spatial VR audio), Symphoria® (automotive 3D audio), xHE-AAC (adaptive streaming and digital radio), the 3GPP EVS VoLTE codec (crystal clear telephone calls), and the interactive and immersive MPEG-H TV Audio System.

With the test plan for the Digital Cinema Initiative and the recognized software suite easyDCP, Fraunhofer IIS significantly pushed the digitization of cinema. The most recent technological achievement for moving pictures is Realception®, a tool for light-field data processing.

Fraunhofer IIS, based in Erlangen, Germany, is one of 72 institutes and research units of Fraunhofer-Gesellschaft, Europe’s largest application-oriented research organization.

For more information, contact amm-info@iis.fraunhofer.de or visit www.iis.fraunhofer.de/amm.