OpenCable™ Specifications

Adaptive Transport Stream Specification

OC-SP-ATS-I01-140214

ISSUED

DISCLAIMER

This document is published by Cable Television Laboratories, Inc. ("CableLabs®").

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various agencies; technological advances; or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein. CableLabs makes no representation or warranty, express or implied, with respect to the completeness, accuracy, or utility of the document or any information or opinion contained in the report. Any use or reliance on the information or opinion is at the risk of the user, and CableLabs shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

This document is not to be construed to suggest that any affiliated company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any cable member to purchase any product whether or not it meets the described characteristics. Nothing contained herein shall be construed to confer any license or right to any intellectual property, whether or not the use of any information herein necessarily utilizes such intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	OC-SP-ATS-I01-140214			
Document Title:	Adaptive Transport Stream Specification			
Revision History:	I01 – Released 02/14/2014			
Date:	February 14, 2014			
Status:	~~Work in Progress~~	~~Draft~~	**Issued**	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes

Work in Progress	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A generally public document that has undergone Member and Technology Supplier review, cross-vendor interoperability, and is for Certification testing if applicable. Issued Specifications are subject to the Engineering Change Process.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks

Contents

Figures

1 INTRODUCTION

1.1 Overview

There are a variety of Adaptive Streaming wire formats. Some are based on an MPEG-2 Transport Stream container such as HLS (HTTP Live Streaming: Apple) and others on a fragmented MP4 container such as HSS (HTTP Smooth Streaming: Microsoft) and HDS (HTTP Dynamic Streaming: Adobe); whereas DASH (Dynamic Adaptive Streaming over HTTP: MPEG) supports both containers. While different, they utilize common video and audio compression formats; namely: ISO/IEC 14496-10 (AVC) and ISO/IEC 14496-3 (AAC). Additional audio formats, such as Dolby Digital Plus, may also be supported by these or a subset of these Adaptive Bit Rate (ABR) formats.

In a unified ABR encoding and encapsulating system, video and audio data are encoded and conditioned for adaptive streaming purposes and the resultant elementary compressed access units are fed to one or more ABR encapsulators or packagers to be formatted into ABR-specific wire formats.

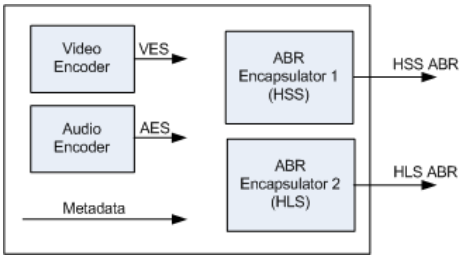

Figure 1 - Unified Transcoder/Encapsulator

The Adaptive Transport Stream (ATS) format allows for streaming/storage of adaptive streaming content originating as Transport Streams in a generic manner without restricting this to a particular adaptive streaming delivery technology (HSS/HLS/HDS). As Figure 2 illustrates, this allows for a separation of the transcoding process from the encapsulation process that produces ABR-specific formats.

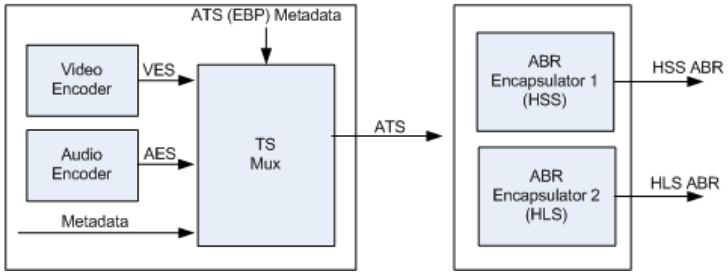

Figure 2 - Separate Transcoder/Encapsulator with ATS between

An ATS is a fully **compliant** continuous MPEG2 Transport Stream. Unlike the HLS format, it is not segmented. HLS is considered to be a Segmented Adaptive Transport Stream (SATS) format. SATS formats may have additional alignment requirements such as an alignment of the physical location of audio packets to some relationship of video packets or the placement of PAT/PMT at boundary locations. ATS has no such requirements. Downstream encapsulation is expected to re-encapsulate an ATS, which may involve partial or complete de-encapsulation (demux) prior to encapsulating into target ABR format. Since this downstream encapsulation does no re-encoding of the media data, the video and audio access units in the ATS are pre-conditioned for adaptive streaming purposes. Additionally, special ATS metadata, called Encoder Boundary Points (EBP), are injected and carried in the transport stream layer to provide adaptive boundary information to downstream processing tasks. An ATS MAY also carry additional PMT descriptors for informative purposes about the various conditioning and boundary points in the ATS. Lastly, structural information on a related set of ATS streams MAY be carried through an ATS Source Description.

1.2 Purpose of Document

The purpose of this document is to describe an ATS stream, the boundary points within it, both explicit and implicit, how boundary points map to various ABR formats such as HSS Fragments and HLS Segments (both in the video and audio domain), and the time stamps, durations and byte ranges of these chunks.

To that end, a significant portion of this document describes the basic requirements for adaptive video and audio conditioning. These sections detail conditioning considerations such as varying frame rates, advertising splice points and input loss handling.

1.3 Organization of Document

This specification is organized into the following sections:

Sections 1 - 4 contain the introduction, references, terms and abbreviations.

Section 5 provides an Adaptive Bitrate (ABR) Functional Overview.

Section 6 provides an ATS Metadata Overview.

Section 7 specifies the Media Presentation Description.

Section 8 specifies ATS stream structure details and boundary points.

Section 9 provides details on Adaptive conditioning for video, audio, and data.

Annex A provides guidelines for Boundary Point creation.

1.4 Scope

This specification describes the requirements and constraints on a single program transport stream (SPTS) that allow it to be used as an Adaptive Transport Stream, including stream conditioning and signaling of segment boundary points. Typically, multiple ATSs will be generated from a single input and sent to a packager, recorder or other device. This specification does not describe how an ATS is stored or how it may be converted to target delivery formats.

In addition, this specification describes both the static and time-sensitive metadata that describes the ATS stream and where this information is located through EBP, PMT, SCTE 35, private metadata, and configuration information. The collection and aggregation of static metadata is defined in a metadata format, the ATS Source Description, that describes the attributes of a set of ATSs. The purpose of the ATS Source Description is to provide information needed to configure a downstream device to package or record the set of ATSs and only carries information related to that task, for example, bitrates, codec parameters, etc. The ATS Source Description is based on the DASH MPD schema, but does not carry any time variant data such as Segment names or Periods. This specification does not describe how a device obtains an ATS Source Description.

1.5 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"SHALL"	This word means that the item is an absolute requirement of this specification.
"SHALL NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

All references are subject to revision, and parties to agreement based on this specification are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

[EBP] Encoder Boundary Point Specification, OC-SP-EBP-I01-130118, January 18, 2013, Cable Television Laboratories, Inc.

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

[MPEG2-TS] ISO/IEC 13818-1:2013, Information Technology - Generic coding of moving pictures and associated audio information - Part 1: Systems.

[SCTE 54] ANSI/SCTE 54 (2009), Digital Video Service Multiplex and Transport System Standard for Cable Television.

2.2 Informative References

This specification uses the following informative references.

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

[DASH264] Guidelines for Implementation: DASH-AVC/264 Interoperability Points, Version 2.0, http://dashif.org/w/2013/08/DASH-AVC-264-v2.00-hd-mca.pdf

[ESAM] Real-time Event Signaling and Management API, OC-SP-ESAM-API-I03-131025, October 25, 2013, Cable Television Laboratories, Inc.

[HDS] ADOBE HTTP Dynamic Streaming, http://www.adobe.com/products/hds-dynamic-streaming.htm

[HLS] HTTP Live Streaming, Apple Inc., http://tools.ietf.org/html/draft-pantos-http-live-streaming-07

[HLS-TM] Timed Metadata for HTTP Live Streaming https://developer.apple.com/library/ios/documentation/AudioVideo/Conceptual/HTTP_Live_Streaming_Metadata_Spec/HTTP_Live_Streaming_Metadata_Spec.pdf

[HSS-1] Smooth Streaming Transport Protocol, http://learn.iis.net/page.aspx/684/smooth-streaming-transport-protocol

[HSS-2] [MS-SSTR]: Smooth Streaming Protocol, http://msdn.microsoft.com/en-us/library/ff469518(v=PROT.10).aspx

[RFC 1521] IETF RFC 1521, MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies, September 1993.

[RFC 2616] IETF RFC 2616, Hypertext Transfer Protocol - HTTP/1.1, June 1999.

[RFC 5646] IETF RFC 5646, Tags for Identifying Languages, September 2009.

[RFC 5905] IETF RFC 5905, Network Time Protocol Version 4: Protocol and Algorithms Specification, June 2010.

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

[SCTE 35] SCTE 35 2013a, Digital Program Insertion Cueing Message for Cable.

[SCTE 104] SCTE 104 2013, Automation System to Compression System Communications Applications Program Interface (API).

[SCTE 128-2] ANSI/SCTE 128-2 2013, AVC Video Constraints for Cable Television Part 2: Transport

[TR 101 290] ETSI TR 101 290 V1.2.1, Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027;
  Phone +1-303-661-9100; Fax +1-303-661-9199; http://www.cablelabs.com
- Internet Engineering Task Force (IETF) Secretariat, 48377 Fremont Blvd., Suite 117, Fremont, California 94538, USA, Phone: +1-510-492-4080, Fax: +1-510-492-4001, http://www.ietf.org
- ISO Central Secretariat: International Organization for Standardization (ISO), 1, rue de Varembé, Case postale 56, CH-1211 Geneva 20, Switzerland; Internet: http://www.iso.ch/
- SCTE - Society of Cable Telecommunications Engineers Inc., 140 Philips Road, Exton, PA 19341
  Phone: +1-610-363-6888 / +1-800-542-5040; Fax: +1-610-363-5898; http://www.scte.org/
- ETSI - http://www.etsi.org

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Acquisition Time	An NTP-derived time which may be carried within an EBP to indicate the time at which the Transcoder/Encoder acquired this Access Unit (AU). Acquisition Time may be carried in EBPs of linear media streams to identify time-based recording boundaries, and provide a record of actual content acquisition time. Acquisition Time has no normative use in packaged or VOD content.
Adaptation Set	Set of interchangeable encoded versions of one or several media content components.
Adaptive Sync	ATS streams that are both Time Synced and Chunk Synced.
Chunk	A discrete section of content that can be independently decoded, possibly given additional initialization information.
Chunk Boundary Point	A specialized Encoder Boundary Point that indicates the beginning of a chunk, and is a stream access point.
Chunk Sync	Chunk sync implies the identical AUs across representations at boundaries indicated by EBP.
Chunk Type	Refers to a specific partition that is contained in the ATS stream. Segment and fragment are examples of different types of chunks.
Conditioned Stream	A transcoded stream that contains independently decodable sections of content (e.g., Fragments and Segments). This stream can be sent to an encapsulator to create fragments and segments in specific ABR formats.
Encapsulator	Processes a conditioned continuous group of elementary streams to create specific ABR-format chunks of mixed or separated elementary streams that are stored in a file or transmitted. Each file is wrapped to be in one or more adaptive streaming formats. An encapsulator does not normally perform any transcoding functions but depends on the conditioned stream to create those independently decodable sections. An encapsulator can also be known as a fragmenter, packager, or segmentor.
Encoder	A subsystem that compresses digital media. The input can be uncompressed digital media or a mezzanine level packetized elementary stream, and output is a compressed stream for delivery to consumers in real-time or via storage media.
Encoder Boundary Point	An indicated point in the stream that is assigned to a PES packet containing one or more access units.
Explicit EBP Stream	The elementary stream carries the EBP structure in the adaptation private data field associated with boundary point AU. Chunks containing an EBP structure in these types of streams can be called "explicit chunks".
Explicit Partition	A partition which references EBP structures carried on its own stream PID.
Fragment	A chunk that is aligned with boundaries in one component stream in the source multiplex. Fragment boundaries are typically explicit for each component. Smooth Streaming is an example ABR format that uses fragments.
Implicit EBP Stream	The elementary stream does not carry the EBP structure in the adaptation private data field associated with a boundary point AU. It may depend on another elementary stream to reference this information. The EBP PMT descriptor may be used to indicate if the elementary stream is an implicit EBP stream. Chunks that do not contain an EBP structure in these types of streams can be called "implicit chunks".
Implicit Partition	A partition that references the EBP structure of an explicit partition carried on a different stream PID (and, possibly, in a different multiplex).
Media Content Component	A single type of media such as audio, video, or text as defined in section 3.1.16 of [DASH].

Partition	A set of continuous chunks within a media stream. A stream can be partitioned in several ways. For example, Partition A corresponds to 2-second chunks, while Partition B corresponds to 5-second chunks. A partition is represented by a series of boundary points across a group of elementary streams.
SAP type	Defines the properties of a Stream Access Point as specified in Annex I of [ISO-BMFF]. SAP Types 1 and 2 correspond to what is known in some coding schemes as a "Closed GOP random access point".
Segment	A chunk with boundaries aligned to include all component streams in the source multiplex across the target presentation time range. Segment boundaries are typically explicit for only one main component (video, for example), and other component boundaries are implicitly derived from this main component. A segment is typically used when packaging content in HLS.
Stream Access Point (SAP)	Enables random access into a media stream as defined in Annex I of [ISO-BMFF].
Time Sync	Indicates identical AUs across representations have the same time stamp.
Transcoder	A subsystem that converts a packetized elementary stream of one bitrate to one or more lower bitrate streams by changing coding parameters, including media resolution. In a broader sense, it can change from one codec format to another.

4 ABBREVIATIONS AND ACRONYMS

This document uses the following abbreviations and acronyms.

AAC	Advanced Audio Coding
AAC-LC	Low Complexity AAC
ABR	Adaptive Bit Rate
ASC	Audio Specific Config
ATS	Adaptive Transport Stream
AU	Access Unit
AVC	Advanced Video Coding
DASH	Dynamic Adaptive Streaming over HTTP
DRC	Dynamic Range Control
DVB	Digital Video Broadcasting
EBP	Encoder Boundary Point
ESAM	Event Signaling and Management
FIL	Filler Element
GOP	Group of Pictures
HDS	HTTP Dynamic Streaming
HE AAC	High Efficiency AAC
HLS	HTTP Live Stream
HSS	HTTP Smooth Streaming
IDR	Instantaneous Decoder Refresh
MP4	MPEG-4 Part 14
MPD	Media Presentation Description
PCE	Program Config Element
PCR	Program Clock Reference
PID	Program Identifier or Packet Identifier per [MPEG2-TS]
PES	Packetized Elementary Stream
PMT	Program Map Table
POIS	Placement Opportunity Information System
PS	Parametric Stereo
PTS	Presentation Time Stamp
RAP	Random Access Point
SAP	Stream Access Point
SATS	Segmented Adaptive Transport Stream
SBR	Spectral Band Replication
SDI	Serial Digital Interface
SPTS	Single Program Transport Stream
TS	Transport Stream
VOD	Video On Demand

5 ABR OVERVIEW

In adaptive streaming, a file or linear source asset is encoded into multiple representations, typically uniquely characterized by bitrate. For example, the video of an asset may be encoded into N different bitrates, some of which may be of common picture resolutions. Furthermore, the encoding, while continuous, is broken up into small **chunks**. Each video chunk starts with a Stream Access Point of type 1-3, typically an IDR or closed GOP. Each new audio chunk starts at an audio access unit boundary. These access points are referred to as Chunk Boundary Points.

In some ABR formats, audio is not multiplexed with video; discrete audio streams per language are produced. In some ABR formats, auxiliary data streams such as Subtitles and/or Captions may also be delivered as discrete streams.

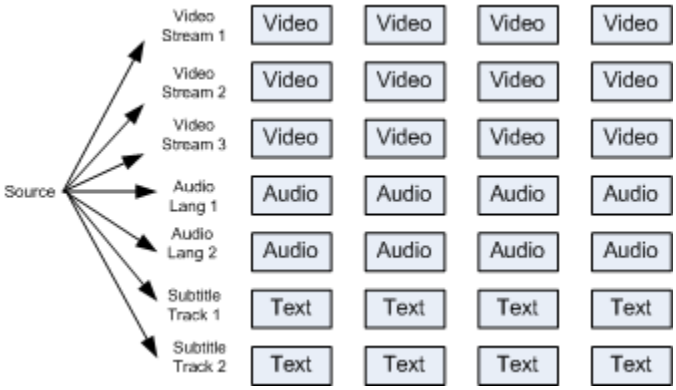

Figure 3 - ABR with discrete audio streams and two subtitle streams

Some formats do not require audio chunks to be of the same duration as the video chunks (shown here with only a single audio language):

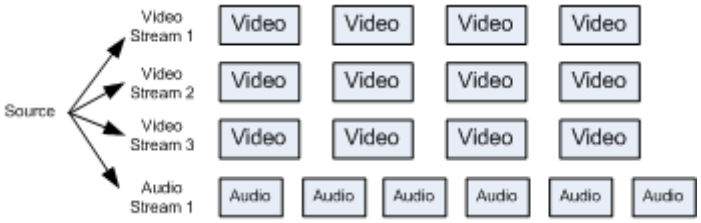

Figure 4 - ABR with audio durations different than video

Other formats (such as HLS prior to iOS-5) have audio multiplexed with video as below, here shown with two audio languages 1 and 2. As shown, all audio languages are to be present in all representations.

Figure 5 - ABR with interleaved audio and video streams

The adaptive nature is such that a client may dynamically switch, for various reasons, such as bandwidth fluctuation, from one representation to another. It does so at chunk boundaries. In Figure 6 below, three streams at different

bitrates are represented. At time A no switch was required. At time B, a change to Stream 3 was deemed required and this continued until time D, at which time a choice was made to switch up to Stream 2.

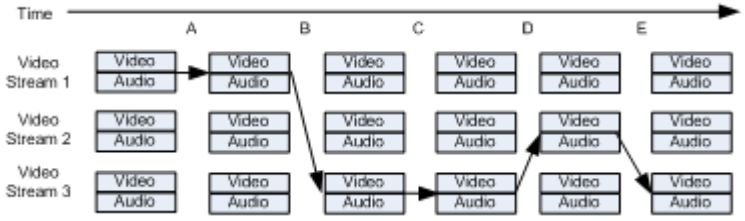

Figure 6 - ABR chunk switching

In order to maintain a seamless experience to the end user, the video and audio data of each of the various streams SHALL be in **time sync** with one another. This implies that the same source video frame V, or audio sample A, in all representations SHALL have the same rendered timestamp. Furthermore, the various streams SHALL be in **chunk sync** with one another. This implies that the same source video frame V, or audio sample A, at the beginning of each chunk in all representations has the same rendered timestamp. If streams were not in both time and chunk sync with one another, then the end user would likely experience some form of discontinuity, a video and/or audio forward or backward disruption. The overall quality of experience would be degraded.

For various reasons, the optimal size of a chunk varies amongst the adaptive formats. HSS and HDS typically use a 2-second chunk duration. HLS typically uses anywhere from 6 to 10 second chunks. An encoding/transcoding system producing adaptive streams for use in multiple formats and also utilizing the same encoded data for the formats SHALL produce chunk alignment for all formats. Ideally, chunks of various formats usually align with one another and thus chunks of longer durations contain an integer number of chunks of shorter durations.

Figure 7 - Chunk alignment of various durations

Chunks in DASH are referred to as Media Segments. Chunks in the HSS/HDS formats are typically referred to as Fragments. Chunks in the HLS format are typically referred to as Segments. Here is another way to view the previous diagram using these terms:

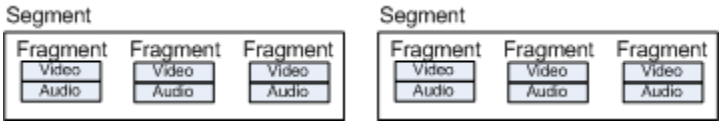

Figure 8 - ABR segments composed of ABR fragments

To further exemplify chunk sync in terms of fragment and segment sync, consider Figure 9 below, which illustrates a number of AVC streams based off of a common source. A switch point in the video domain occurs at the IDR of a chunk boundary. A chunk can contain additional I-pictures, additional GOPs, and the location of these can vary from one stream to another as seen in streams 1 and 2. However, the start of fragment-chunks will be on the same IDR. Therefore, stream 3 is not in adaptive sync with streams 1 and 2. The same is true for segment-chunks. While stream 5's first segment is fragment-aligned with stream 4, it is not segment-aligned.

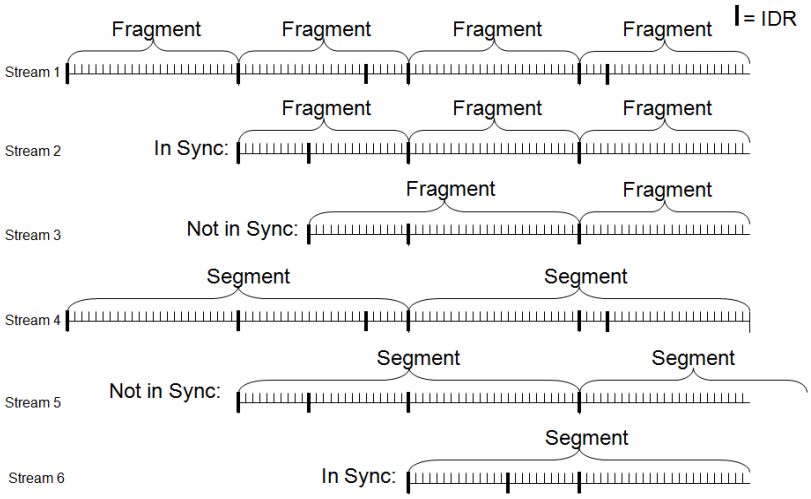

Figure 9 - Fragment and Segment Sync/Alignment

6 ATS METADATA OVERVIEW

ATS Metadata refers to a description of the ATS bitstream. Metadata can be categorized as:

- static metadata, where the description remains the same over the entire bitstream; for example, structural metadata describing the number of ATS streams and their relationships
- time-variant metadata, where the description can vary and change at different points of relative time in the bitstream, (either stream or wallclock time), for example EBP structures describing chunk boundaries
- metadata, such as ad insertion points or program start/end.

ATS supports several mechanisms to carry this metadata. In the elementary stream, metadata can be carried through the EBP structure [EBP], which can convey time variant metadata such as location of chunk boundaries in the stream, types of chunks, and chunk labels through the grouping_id as well as wall-clock media time at chunk boundaries. In the transport stream, the PMT descriptors can describe metadata at the transport level and elementary stream level and can describe PID, stream type (video descriptor, audio descriptor, captioning, language, bandwidth), and supported chunk cadences through the EBP PMT descriptor. The PMT information is usually static until a major event that changes the media experience occurs in the stream. Lastly, event metadata can be conveyed through the SCTE 35 descriptors and triggers (see Section 9.4) or through a private metadata mechanism on a separate pid (see Section 8.13.1). The information carried is used to indicate alternate content such as ad insertion, campaigns, and blackout events. Additionally, static metadata is carried out of band in the ATS Source Description (see Section 7).

7 ATS SOURCE DESCRIPTION

An ATS Source Description is used to describe the properties of a set of related ATSs. The primary use of the ATS Source Description is to provide information about the location of the source ATSs, for example a set of multicast addresses for a group of ATSs that are intended to be combined into a DASH AdaptationSet. It carries static metadata in addition to content source location such as program service metadata, MPEG-2 TS structure and attributes, etc. This metadata may be used by downstream devices to configure packaging or recording tasks.

The ATS Source Description follows the MPD schema of MPEG DASH [DASH] with enhancements to describe sources that are streams rather than files.

The information conveyed by an ATS Source Description SHOULD be created while or before the media transcoding task is started. ATS Source Description may be updated at a programming boundary. The publication and delivery of ATS Source Description is beyond the scope of this specification.

The following sections describe specific elements or attributes of the DASH MPD schema that have different requirements or semantics when used in an ATS Source Description. Elements and attributes not listed below SHALL follow the requirements specified in [DASH].

7.1 MPD requirements

Element or Attribute Name	Use	Description
MPD		
@profiles	M	With value of "urn:mpeg:dash:profile:full:2011".
@mediaPresentationDuration	M	*If MPD@type="static", this is mandatory. It specifies the duration of the entire Media Presentation. For ATS Source Description, the value of 'P100Y" shall be used.*
Legend: For attributes: M=Mandatory, O=Optional. For elements: <minOccurs>…<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and preceded with an @		

For the description of Program Information and Location elements, please see the specification in [DASH].

7.2 Period requirements

Element or Attribute Name	Use	Description
Period		Specifies the information of a Period.
BaseURL	0…N	Specifies a base URL that can be used for reference resolution and alternative URL selection.
Legend: For attributes: M=Mandatory, O=Optional. For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and proceeded with an @.		

The BaseURL element is used to describe where and how to get the media content. In an ATS Source Description, two options are defined to leverage the "xs:anyURL" syntax definition from [DASH] and to support of IP multicast delivery of ATS.

(1) Direct Address Description

Specifies IP multicast address and port number directly in the BaseURL with syntax as below.

<BaseURL>udp://*souce-IP-address***@***multicast-IP-address***:***UDP-port***</BaseURL>**

The italic text shall be replaced with actual value. This option may satisfy most current description of ATS location, but it has limited flexibility for future expansion, such as multiple source specific addresses of IGMP v3, IPv6 protocol, etc.

(2) Indirect JSON File Description

This second option uses the BaseURL to point to a JSON file, in which the media location, as well as other location-based attributes, can be accessed. Its syntax is defined as below.

<BaseURL>*http://www.example.com/directory/ats-stream.json***</BaseURL>**

The JSON file format is based on IETF RFC 4627. Inside the JSON file, the multicast IP address and its port number, as well as IGMP source addresses, etc., can be specified.

The details of JSON file format is TBD.

The BaseURL element MAY be specified in multiple levels, e.g., in Period level, in AdaptationSet level and in Representation level; please refer to [DASH] section 5.6.4 for reference resolution and section 5.6.5 for alternative base URL definition.

7.3 AdaptationSet requirements

Element or Attribute Name	Use	Description
AdaptationSet		Adaptation Set description.
@id	O	Specifies a unique identifier for this Adaptation Set in the scope of the Period. If not present, no identifier for the Adaptation Set is specified.
CommonAttributesElements	-	Specifies the common attributes and elements.
@lang	O	Declares the language code(s) for this Adaptation Set. The syntax and semantics according to [RFC 5646] shall be used. If not present, the language code may be defined for each media component, or it may be unknown.
@contentType	O	Specifies the media content component type for this Adaptation Set. A value of the top-level Content-type 'type' value as defined in [RFC 1521], Clause 4 shall be taken. If not present, the media content component type may be defined for each media component, or it may be unknown.
@par	O	Specifies the picture aspect ratio of the video media component type, in the form of a string consisting of two integers separated by ':', e.g.,"16:9". When this attribute is present, and the attributes @width and @height for the set of Representations are also present, the picture aspect ratio as specified by this attribute shall be the same as indicated by the values of @width, @height, and @sar, i.e., it shall express the same ratio as (@width * *sarx*): (@height * *sary*), with *sarx* the first number in @sar and *sary* the second number. If not present, the picture aspect ratio may be defined for each media component, or it may be unknown.

Element or Attribute Name	Use	Description
@minBandwidth	O	Specifies the minimum @bandwidth value in all Representations in this Adaptation Set. This value has the same units as the @bandwidth attribute. If not present, the value is unknown.
@maxBandwidth	O	Specifies the maximum @bandwidth value in all Representations in this Adaptation Set. This value has the same units as the @bandwidth attribute. If not present, the value is unknown.
@minWidth	O	Specifies the minimum @width value in all Representations in this Adaptation Set. This value has the same units as the @width attribute. If not present, the value is unknown.
@maxWidth	O	Specifies the maximum @width value in all Representations in this Adaptation Set. This value has the same units as the @width attribute. If not present, the value is unknown.
@minHeight	O	Specifies the minimum @height value in all Representations in this Adaptation Set. This value has the same units as the @height attribute. If not present, the value is unknown.
@maxHeight	O	Specifies the maximum @height value in all Representations in this Adaptation Set. This value has the same units as the @height attribute. If not present, the value is unknown.
@minFrameRate	O	Specifies the minimum @frameRate value in all Representations in this Adaptation Set. This value is encoded in the same format as the @frameRate attribute. If not present, the value is unknown.
@maxFrameRate	O	Specifies the maximum @frameRate value in all Representations in this Adaptation Set. This value is encoded in the same format as the @frameRate attribute. If not present, the value is unknown.
@segmentAlignment	OD default: false	When set to "true", this specifies that the EBP is presented for segment boundary in the video stream.
@subsegmentAlignment	OD default: false	When set to "true", this specifies that the EBP is presented for fragment boundary in the video stream.

Legend:
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value.
For elements: <minOccurs>...<maxOccurs> (N=unbounded)
Elements are **bold**; attributes are non-bold and preceded with an @.

7.4 Common Attributes and Elements

Element or Attribute Name	Use	Description
Common attributes and elements		
@width	M	Specifies the horizontal visual presentation size of the video media type on a grid determined by the `@sar` attribute. In the absence of `@sar`, width and height are specified as if the value of `@sar` were "`1:1`". If not present on any level, the value is unknown.
@height	M	Specifies the vertical visual presentation size of the video media type, on a grid determined by the `@sar` attribute. If not present on any level, the value is unknown.
@sar	O	Specifies the sample aspect ratio of the video media component type, in the form of a string consisting of two integers separated by ':', e.g.,`"10:11"`. The first number specifies the horizontal size of the encoded video pixels (samples) in arbitrary units. The second number specifies the vertical size of the encoded video pixels (samples) in same units as the horizontal size. If not present on any level, the value is unknown.
@frameRate	M	Specifies the output frame rate (or in the case of interlaced, half the output field rate) of the video media type in the Representation. If the frame or field rate is varying, the value is the average frame or half the average field rate over the entire duration of the Representation. The value is coded as a string, either containing two integers separated by a "/", ("F/D"), or a single integer "F". The frame rate is the division F/D, or F, respectively, per second (i.e., the default value of D is "1"). If not present on any level, the value is unknown.
@audioSamplingRate	M	Either a single decimal integer value specifying the sampling rate or a whitespace separated pair of decimal integer values specifying the minimum and maximum sampling rate of the audio media component type. The values are in samples per second. If not present on any level, the value is unknown.
@mimeType	M	Specifies the MIME type of the concatenation of the Initialization Segment, if present, and all consecutive Media Segments in the Representation.
@codecs	M	Specifies the codecs present within the Representation. The codec parameters shall also include the profile and level information where applicable. The contents of this attribute shall conform to either the `simp-list` or `fancy-list` productions of [RFC 6381], section 3.2, without the enclosing `DQUOTE` characters. The codec identifier for the Representation's media format, mapped into the name space for codecs as specified in [RFC 6381], section 3.3, shall be used.

Element or Attribute Name	Use	Description
@maximumSAPPeriod	O	When present, specifies the maximum SAP interval in seconds of all contained media streams, where the SAP interval is the maximum time interval between the T_{SAP} of any two successive SAPs of types 1 to 3 inclusive of one media stream in the associated Representations. If not present on any level, the value is unknown.
`AudioChannelConfiguration`	M	Specifies the audio channel configuration of the audio media component type.
@scanType	M	Specifies the scan type of the source material of the video media component type. The value may be equal to one of "progressive", "interlaced" and "unknown". If not specified on any level, the scan type is "`progressive`".
Legend: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>…<maxOccurs> (N=unbounded) Elements are **`bold`**; attributes are non-bold and preceded with an @.		

Element or Attribute Name	Use	Description
ContentComponent		Description of a content component
@id	O	Specifies an identifier for this media component. The attribute shall be unique in the scope of the containing Adaptation Set. The ES PID should be used here to link with sub representation.
@lang	M	Same semantics as in AdaptationSet for @lang attribute.
Legend: For attributes: M=Mandatory, O=Optional. For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are **`bold`**; attributes are non-bold and preceded with an @, list of elements and attributes is in ***italics bold*** referring to those taken from the Base type that has been extended by this type.		

7.5 Representation requirements

Element or Attribute Name	Use	Description
`Representation`		This element contains a description of a Representation.
`@bandwidth`	M	This is used to specify the MPEG-2 TS rate.
`CommonAttributesElements`	-	Common Attributes and Elements.

Element or Attribute Name	Use	Description
SubRepresentation	0 … N	Specifies information about a Sub-Representation that is embedded in the containing Representation. This is used to describe the elementary streams contained within each ATS.

Legend:
For attributes: M=Mandatory, O=Optional. For elements: <minOccurs>...<maxOccurs> (N=unbounded)
Elements are **bold**; attributes are non-bold and preceded with an @, list of elements and attributes is in ***italics bold*** referring to those taken from the Base type that has been extended by this type.

Element or Attribute Name	Use	Description
SubRepresentation		Specifies a Sub-Representation.
@bandwidth	O	Identical to the @bandwidth definition in Representation, but applied to this Sub-Representation. This can be used to specify the ES rate.
@contentComponent	O	If present, specifies the set of all media content components that are contained in this Sub-Representation as a whitespace-separated list of values of **ContentComponent**@id values. If not present, the Sub-Representation is not assigned to a media content component.
CommonAttributesElements	-	Common Attributes and Elements (attributes and elements from base type ***RepresentationBaseType***). For details see Section 7.4

Legend:
For attributes: M=Mandatory, O=Optional. For elements: <minOccurs>...<maxOccurs> (N=unbounded)
Elements are **bold**; attributes are non-bold preceded with an @, List of elements and attributes is in ***italics bold*** referring to those taken from the Base type that has been extended by this type.

7.6 Source Description Example

```
<?xml version="1.0" encoding="UTF-8" ?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011"
  id="ATS MPD Example"
  profiles="urn:mpeg:dash:profile:full:2011"
  type ="dynamic"
  availabilityStartTime=" 2002-05-30T09:30:10Z"
 minBufferTime="PT0.07S"
  maxSegmentDuration="PT15S"
  maxSubsegmentDuration="PT3S" >

  <ProgramInformation>
    <Title>CNN-EAST-HD</Title>
  </ProgramInformation>

  <Period id="">
    <BaseURL>udp://192.168.4.11@225.1.1.1:</BaseURL>
     <AdaptationSet id="1" mimeType="video/mp2t" codecs="avc1.4D4028,mp4a,ac3" segmentAlignment="true" />
       <ContentComponent contentType="video" id="101" />
       <ContentComponent contentType="audio" id="102" lang="eng" />
       <ContentComponent contentType="audio" id="103" lang="esp" />
```

```
      <ContentComponent contentType="audio" id="104" lang="eng" />
      <ContentComponent contentType="audio" id="105" lang="esp" />

      <Representation id="1" bandwidth="6485000">
        <SubRepresentation bandwidth="5850000", width="1280" height="720" frameRate="180000/3003"
codecs="avc1.4D4028" contentComponent="101"/>
        <SubRepresentation bandwidth="96000" codecs="AC3" contentComponent="102"/>
        <SubRepresentation bandwidth="96000" codecs="AC3" contentComponent="103"/>
        <SubRepresentation bandwidth="96000" codecs="mp4a" contentComponent="104"/>
        <SubRepresentation bandwidth="96000" codecs="mp4a" contentComponent="105"/>
        <BaseURL> 1001 </BaseURL>
      </Representation>

       <Representation id="2" bandwidth="3485000">
        <SubRepresentation bandwidth="2850000", width="1280" height="720" frameRate="90000/3003"
codecs="avc1.4D4028" contentComponent="101" />
        <SubRepresentation bandwidth="96000" codecs="mp4a" contentComponent="104"/>
        <SubRepresentation bandwidth="96000" codecs="mp4a" contentComponent="105"/>
        <BaseURL> 1002 </BaseURL>
      </Representation>

      <Representation id="3" bandwidth="1663000">
        <SubRepresentation bandwidth="1200000", width="768" height="432" frameRate="90000/3003"
codecs="avc1.4D4028" contentComponent="101"/>
        <SubRepresentation bandwidth="96000" codecs="mp4a" contentComponent="104"/>
        <SubRepresentation bandwidth="96000" codecs="mp4a" contentComponent="105"/>
        <BaseURL> 1003 </BaseURL>
      </Representation>

     <Representation id="4" bandwidth="1175000" width="640" height="360" frameRate="90000/3003">
        <SubRepresentation bandwidth="750000", width="640" height="360" frameRate="90000/3003"
codecs="avc1.4D4028" contentComponent="101"/>
        <SubRepresentation bandwidth="96000" codecs="mp4a" contentComponent="104"/>
        <SubRepresentation bandwidth="96000" codecs="mp4a" contentComponent="105"/>
        <BaseURL> 1004 </BaseURL>
      </Representation>

      <Representation id="5" bandwidth="942398" width="512" height="288" frameRate="90000/3003">
        <SubRepresentation bandwidth="480000", width="512" height="288" frameRate="90000/3003"
codecs="avc1.4D4028" contentComponent="101"/>
        <SubRepresentation bandwidth="96000" codecs="mp4a" contentComponent="104"/>
        <SubRepresentation bandwidth="96000" codecs="mp4a" contentComponent="105"/>
        <BaseURL> 1005 </BaseURL>
      </Representation>

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

8 ATS SPECIFICS

8.1 General Makeup

An ATS is a single-program compliant MPEG-2 Transport Stream following restrictions as indicated in [TR 101 290]. Multiple interleaving options are allowable as described below; however, all transcoders SHALL support Method 1.

Video and Audio data SHALL have been conditioned for ABR purposes as described in the previous sections. However, an interleaved ATS is not segmented like SATS. Chunk boundaries on video and optionally audio data SHALL be marked in the transport stream by using Encoder Boundary Points (EBP).

8.2 Interleaving options

8.2.1 Method 1 – All representations carry all audio streams

An ATS SHALL contain a single video representation and all audio representations intended to be used with the video representation. While different audio representations (e.g., different languages, different encoding formats) do not have to be in time sync with one another, the same audio representation (language and bitrate) across all ATS SHALL be in time sync.

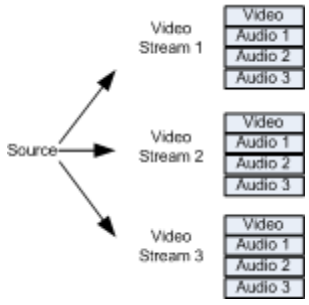

Figure 10 - Interleave Method 1

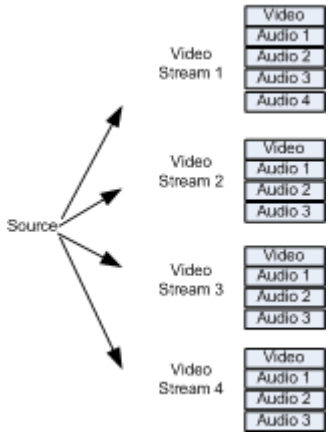

Figure 11 - Interleave Method 1 with Audio 4 used by one Video representation

In Figure 11, audio 4 is only needed for Stream 1. The end use of Streams 2-4 does not use audio 4. Thus according to Method 1, audio 4 does not need to be present in these other streams.

8.2.2 Method 2 – All representations carry a common subset of audio streams

An ATS SHALL contain a single video representation and SHALL contain at least one common audio representation. While different audio representations do not have to be in time sync with one another, if the same audio representation is present in more than one ATS, they SHALL be in time sync with each other.

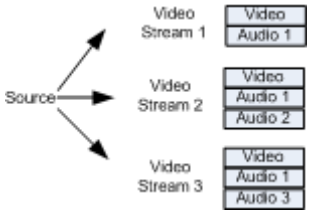

Figure 12 - Interleave Method 2

8.2.3 Method 3 - Audio carried in separate Representations

An ATS MAY contain a single video representation and/or one or more audio representations. Thus, video-only and audio-only ATS streams are supported. While different audio representations do not have to be in time sync with one another, if the same audio representation is present in more than one ATS, they SHALL be in time sync with each other. This is the most efficient bandwidth and storage-wise, as duplication is gated solely on redundancy needs. Non-interleaved encapsulation formats have no need for interleaved ATS assets.

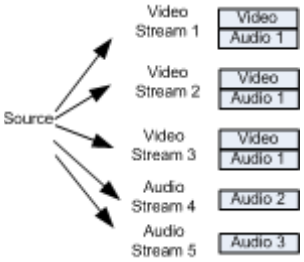

Figure 13 - Interleave Method 3, Hybrid of Method 2

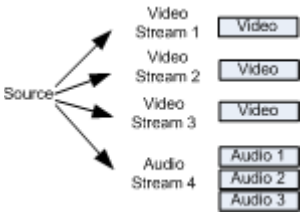

Figure 14 - Another Interleave Method 3 Embodiment

Video-only streams of Method 3 SHALL be playable and include PCR on the video PID. Audio-only streams of Method 3 SHALL include a PCR.

8.3 Late Binding Packaging Implications

Each stream in Method 1 contains all audio representations intended to be used with the video representation. This is not the case with Methods 2 and 3, and thus downstream encapsulation/packaging may need to source multiple source streams in order to bind the desired audio(s) to the desired video component. In the late binding case, there is a potential of multiple PCR tracks that could be associated with different elementary streams due to the need to create compliant transport streams. The PCR track associated with the Main Elementary Stream, usually Video, SHALL be the PCR track used for the bound package.

8.4 Partitions

Media, such as audio and video, within an ATS is conditioned as described given the considerations in Section 9. An ATS is thus a conditioned stream containing independently decodable sections or **chunks** of media. Being a compliant MPEG-2 Transport Stream, there is no physical (packet) alignment within an ATS of discrete media PIDs as a result of this conditioning. For example, the audio chunk from time TA to TB is likely skewed from the video chunk for the same time period.

Media MAY be conditioned to target multiple end ABR formats. The duration of chunks across formats MAY vary, and therefore a boundary for one chunk of media may not be a boundary for a chunk targeting a different duration. An ATS contains one or more **partitions** where each partition defines a set of continuous chunks. For example, one video partition may nominally contain chunks of 2-second durations and another video partition may nominally contain chunks of 10-second durations. Target ABR formats with identical partitioning needs may share a single partition; for example, HSS and HDS with 2-second chunk durations may use the same video partition. However, it is allowable to have multiple identical partitions; e.g., a video partition for HSS and a video partition for HDS, each with exactly the same chunk boundaries.

The chunk boundaries within a partition are delineated by Encoder Boundary Points (EBPs) described further in Section 8.6. A partition may be explicit and carry EBPs. Unique video partitions are required to be explicit. A partition may be implicit and reference or derive boundaries for its partition media from another partition. HLS audio partitions are typically implicit with the boundaries of audio chunks derived from a video partition.

8.5 PMT Descriptors

To describe partitions, an ATS SHOULD include PMT descriptor data for each elementary stream with explicit and implicit partitions. Such information may indicate that, for example, there are two video partitions and three audio partitions with one explicit audio partition and the other two audio partitions are implicitly derived from each of the video partitions. An ATS stream SHOULD carry the EBP PMT descriptor as described in [EBP]. If the stream has closed-captioning service, it SHOULD be preserved with its signaling as mandated by [SCTE 54].

An ATS stream MAY also carry the following PMT descriptors:

1) A maximum_bitrate_descriptor() in the PMT to indicate both program and video and audio elementary stream max bitrates. This informative data can assist downstream encapsulation systems [MPEG2-TS].
2) Audio Descriptor for AC-3, E-AC-3, AAC and others.

8.6 Encoder Boundary Point (EBP)

The Encoder Boundary Point Structure [EBP] is a signaling mechanism in the private field of the adaptation field of an MPEG TS packet for video or audio. This structure can be applied to each video PES and audio PES packet although typically it only exists on chunk boundaries. It indicates a hinting mechanism for taking the continuous [MPEG2-TS] stream conditioned for adaptive streaming (the ATS) and creating ABR format specific chunks (fragments, segments, etc.) from them. There are bits in the EBP to indicate discretely different chunk boundaries, for example, fragment boundaries differing from segment boundaries. Additional boundary bits are available to indicate additional boundaries and thus define different partitions. This is to facilitate the different durations of ABR format specific chunks and to allow chunks to occur on different boundaries. A set of continuous chunks for a given media element defines a partition.

In the video domain, however, boundaries of different formats are typically subsets of other boundaries to minimize the bit cost associated with boundary frames. For example, HSS and HDS may both have 2-second chunk durations and are ideally aligned. An HLS chunk may be 6 seconds in duration and align ideally with HDS and HSS chunk boundaries.

When any elementary stream contains explicit EBPs and carries PCR, a PCR SHALL be inserted in any transport packet carrying a boundary EBP according to access unit constraints as indicated by NOTE 2 of section 6.4.2.2 of [DASH].

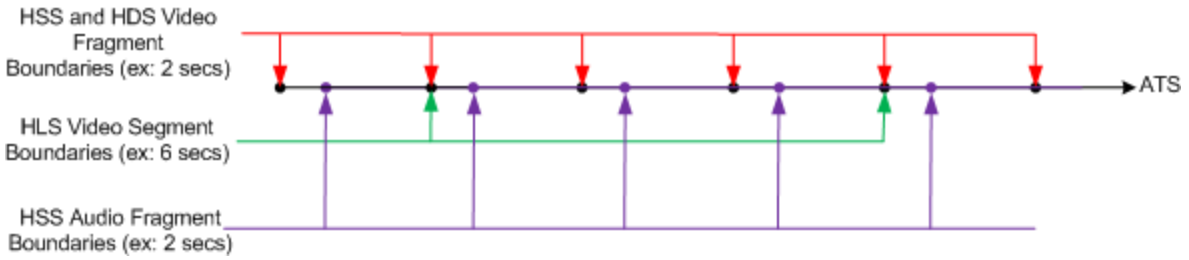

Figure 15 - Boundary Points

8.7 Video Boundary Points

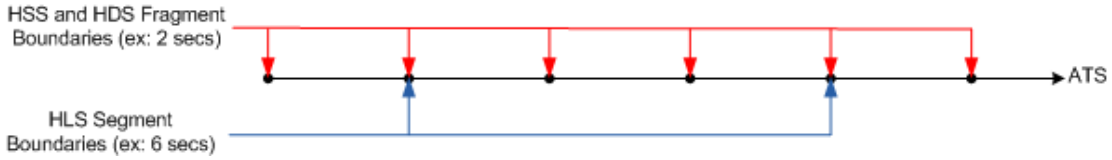

Figure 16 - Boundary Spec Indicating Fragment and Segment Boundaries

Figure 16 illustrates a typical 2-second fragment-chunk (for HSS/HDS purposes) and a 6-second segment-chunk (for HLS purposes). The points on the timeline are video access units and SHALL be marked with an EBP. The video frame at the boundary point SHALL be PES aligned and SHALL begin a new PES packet.

Per Section 9.5, conditioning video at ad boundaries, both OUT and IN points, SHALL occur. A segmentation boundary point SHALL be provided at these locations. For HSS/HDS, these technologies have the option to do splicing by manifest manipulation or by client-side dual decoding. In the case of client-side ad insertion systems, it is possible for splicing to occur mid-fragment. A fragmentation boundary point SHOULD be provided at these locations. This is illustrated in Figure 17.

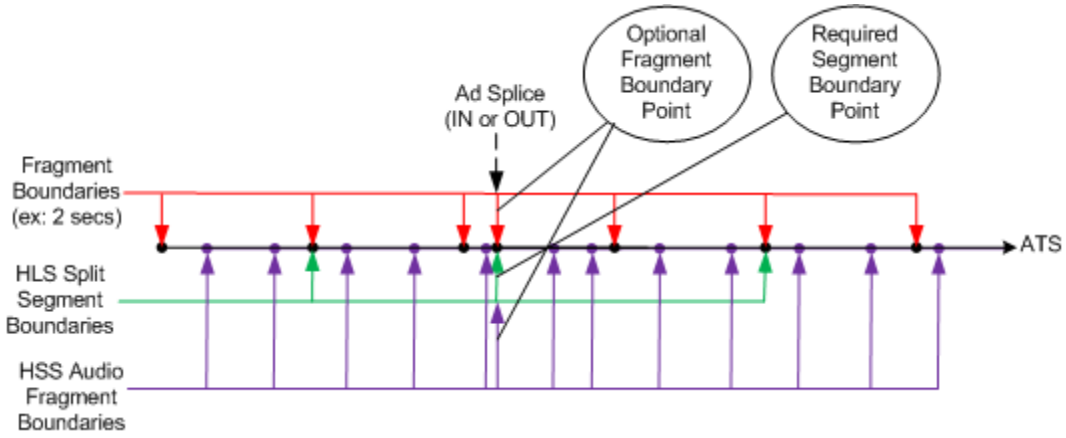

Figure 17 - Boundary Points at Ad Boundaries

8.8 Audio Boundary Points

For HLS and HDS, audio boundaries can be derived from the video boundaries - they have approximately the same start time and end time. The [EBP] specification rule states that the first audio AU of a derived chunk is the first AU where the presentation time is >= presentation time of the video boundary point for that chunk. This works well for both interleaved and non-interleaved HLS and for HDS. In Figure 18, the yellow audio HLS segment boundaries do not require physical EBPs. Furthermore, there is no requirement in an ATS to align the physical location of audio

packets to some relationship of video for segmentation purposes. This segmentation task is the responsibility of downstream encapsulation processes. The same applies to HDS fragments as shown in Figure 19.

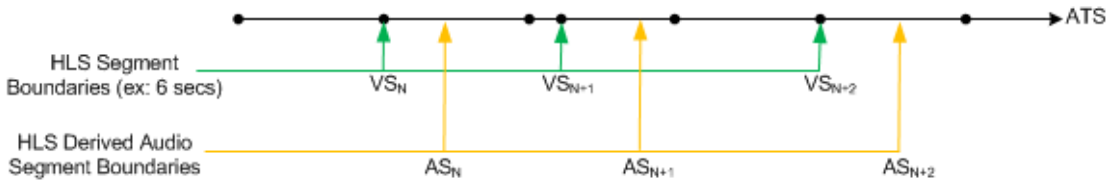

Figure 18 - Implicit / Derived HLS Audio Segment Boundaries

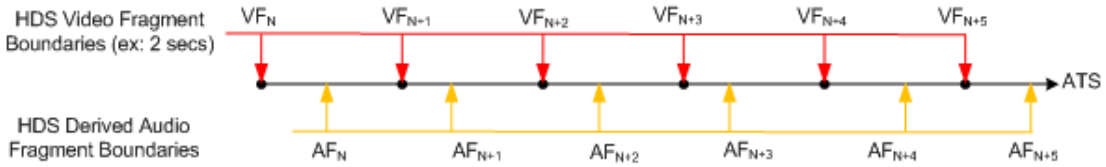

Figure 19 - Implicit / Derived HDS Audio Fragment Boundaries

In some ABR formats, the manifest is optimal if the duration of fragments is constant (i.e., using the t=x, d=y, r=z tags in HSS or DASH time-based segment templates). Using the HLS-derived approach would cause varying fragment durations in the audio domain (there is not a 1:1 relationship between video AUs and audio AUs). So an explicit EBP SHOULD be signaled for audio fragment boundaries so that redundant encapsulators produce the same audio fragments. Figure 20 shows in purple audio AUs that have been explicitly marked with EBP fragment markers. For segmented formats such as HLS, any fragment EBPs (for audio or video) can be ignored.

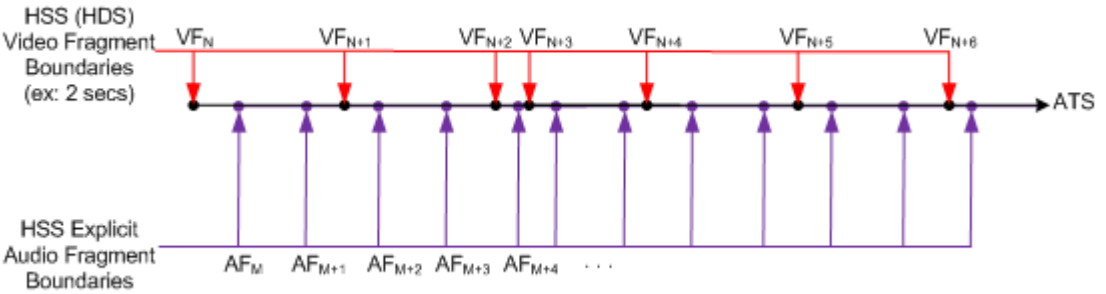

Figure 20 - Explicit Audio Fragment Boundaries

If an audio chunk boundary is indicated via an explicit EBP (an explicit audio EBP), the audio AU at the boundary SHOULD be PES aligned and start a new PES packet. If an EBP is not provided for the audio chunk boundaries (and implicit audio EBP), there is no requirement for PES alignment. Therefore, it is the responsibility of downstream encapsulation systems to split PES packets at boundary points if converting ATS to SATS.

8.9 Audio / Video Skew

Audio and video skew within the continuous ATS is the same as it would be for a normal TS. Figure 21 shows segments in the video and audio domain in different colors. Video/audio in a given color represents media from approximately the same time interval. There are no explicit audio EBPs and thus audio boundaries are implicit from the video segments. The presentation time of the first green A is >= to the first green V and the presentation time of the last red A is < the presentation time of the first green V.

Figure 21 - Audio Skew from Video (e.g., HLS Segments)

Figure 22 below is the same diagram but with explicit audio chunks alternately underlined. So in this example, there are 5 audio AUs per chunk. Note that the audio chunks explicitly delineated by EBPs do not align with the implicit audio chunks that are derived from the video chunk timeline.

Figure 22 - Audio Skew from Video (e.g., HSS Fragments)

8.10 ATS EBP Acquisition Time

A time of day or wall clock time value MAY be carried by the EBP structure to indicate the time that the transport packet carrying the EBP was acquired by the transcoder or other content processing system. The EBP acquisition time SHOULD be placed on chunk boundaries. The time value may be used in downstream processing for various informative purposes. For example, it may be used in linear media feeds to identify time-based recording boundaries, which provide a record of actual acquisition time. The source content to a transcoder may contain EBPs that carry this time field. As per the EBP specification, acquisition time on outgoing EBPs would be derived from this incoming EBP time. Refer to [EBP] for further details on the acquisition and timing of this value.

8.11 ATS Boundaries in Relation to HLS/HDS

In an ATS, HLS audio segment boundaries are derived from video segment boundaries. HDS fragment boundaries may also be derived from video fragment boundaries. Below, HLS segment boundaries are discussed, but the same approach applies to HDS fragment boundaries.

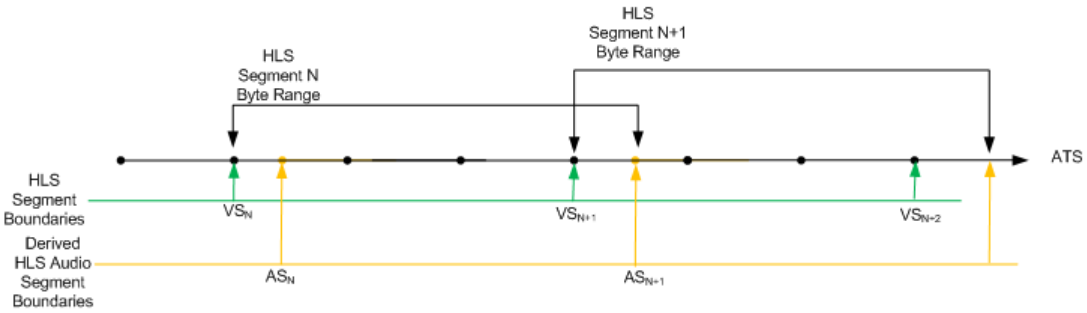

Figure 23 - ATS Segment Byte Ranges

An HLS segment S_N

- has duration TimeStampOf (VS_{N+1}) - TimeStampOf (VS_N)
- starts at byte offset VS_N
- ends at byte offset AS_{N+1}
- size of segment is ByteOffset(AS_{N+1}) - ByteOffset(VS_N)

- byte Ranges for Segment N and N+1 **overlap**
  - ByteOffset(S_N+1) != ByteOffset(S_N) + SizeOf(S_N)

In HLS, URLs refer to file assets (or byte ranges within a file asset) which are themselves complete segments. Segment SN has the following properties in the ATS.

Segment	Time Stamp	Duration	Start Offset	End Offset
0	TS(0)	D(0)	VS(0)	AS(1)
1	TS(0) + D(0)	D(1)	VS(1)	AS(2)
2	TS(1) + D(1)	D(2)	VS(2)	AS(3)
3	TS(2) + D(2)	D(3)	VS(3)	AS(4)
4	TS(3) + D(3)	D(4)	VS(4)	AS(5)

8.12 ATS Boundaries in Relation to HSS

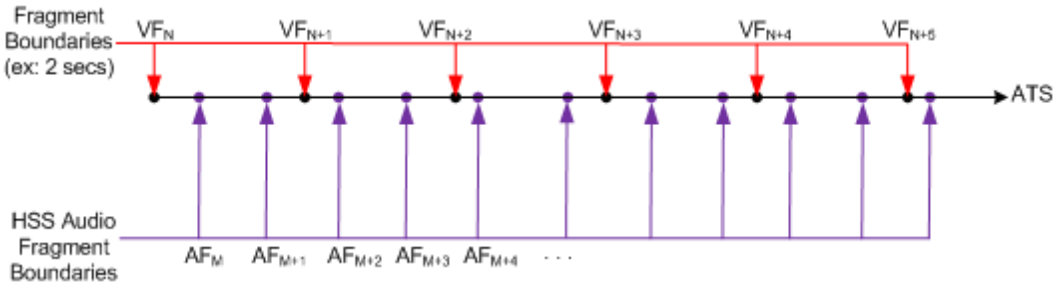

Figure 24 - ATS Video/Audio Fragments

HSS video fragment VF_N

- Has duration TimeStampOf(VF_{N+1}) - TimeStampOf (VF_N)
- Starts at byte offset VF_N
- Ends at byte offset VF_{N+1}
- Size of fragment is ByteOffset(VF_{N+1}) - ByteOffset(VF_N)
- Byte Ranges for Video Fragment N and N+1 **do not overlap**

An HSS audio fragment AF_M

- Has duration TimeStampOf(AF_{M+1}) - TimeStampOf (AF_M)
- Starts at byte offset AF_M
- Ends at byte offset AF_{M+1}
- Size of fragment is ByteOffset(AF_{M+1}) - ByteOffset(AF_M)
- Byte Ranges for Audio Fragment M and M+1 **do not overlap**

For HSS, asset URLs are based on a timestamp.

The TimeStampOf(F_{N+1}) = TimeStampOf(F_N) + DurationOf(F_N)

Fragment	Time Stamp	Duration	Start Offset	End Offset
0	TS(0)	D(0)	S(0)	E(0)
1	TS(0) + D(0)	D(1)	E(0)	E(1)
2	TS(1) + D(1)	D(2)	E(1)	E(2)
3	TS(2) + D(2)	D(3)	E(2)	E(3)
4	TS(3) + D(3)	D(4)	E(3)	E(4)

8.13 Auxiliary data streams

Some applications may require the carriage of private, optionally timed, auxiliary data. Thumbnails, Nielson Metadata, Live Game Stats are such examples. The various ABR formats have specific methods for signaling, formatting and delivering such private data. Clause 2.12 of [MPEG2-TS] details various methods for carrying metadata within a transport stream and any could be used for carrying such data.

8.13.1 Auxiliary data in PES Streams

When auxiliary data is carried in a PES stream with timestamps, the alignment and synchronization requirements for audio streams should be applied. Any descriptors present for this elementary stream in the PMT SHOULD be reflected in the ATS Source Description.

8.13.1.1 HLS Timed Metadata

A common example of auxiliary data carried in timed PES packets is the [HLS-TM] Timed Metadata specification which leverages a specific option from 2.12.3 of [MPEG2-TS], the synchronous delivery of metadata carried in TS PES packets, for the carriage of timed metadata in the HLS ABR format. For these purposes, the HLS-TM can be disassociated from the HLS ABR format. Utilizing the HLS-TM approach for delivering timed metadata in ATS makes it easier on downstream packagers converting ATS to HLS as it is no further work other than re-positioning timed metadata packets into the segment that contains the video time referred to by the timed metadata. The re-positioning is similar to the alignment of audio AUs within an ATS to HLS segments. A packager for other ABR formats would need to extract the timed metadata payload and convert to the appropriate ABR format. Specifications for specific forms of timed metadata, such as thumbnails, Nielson Metadata, etc., are outside the scope of this document.

HLS-TM utilizes ID3 metadata payloads, which are self-describing and require no further configuration information. However, as downstream processing of ATS, such as encapsulating ATS into other ABR formats, may require *a priori* knowledge of the different types of metadata carried within a single private metadata stream, the ID3 self-describing information SHOULD be detailed out of band, for example through the ATS Source Description. The use of [MPEG2-TS] may also require detailing of metadata types through an out of band mechanism such as the ATS Source Description.

9 ADAPTIVE STREAMING CONDITIONING

9.1 Video Conditioning and Synchronization

As described in the ABR overview section, media content in an ABR stream will be chunked such that seamless switching can occur at the chunk boundaries from one representation to any other. The task of video encoding for ABR may be accomplished in a single system with one or more encoders. It may be distributed across more than one system. Figure 25 illustrates two transcoding systems, each with multiple encoders.

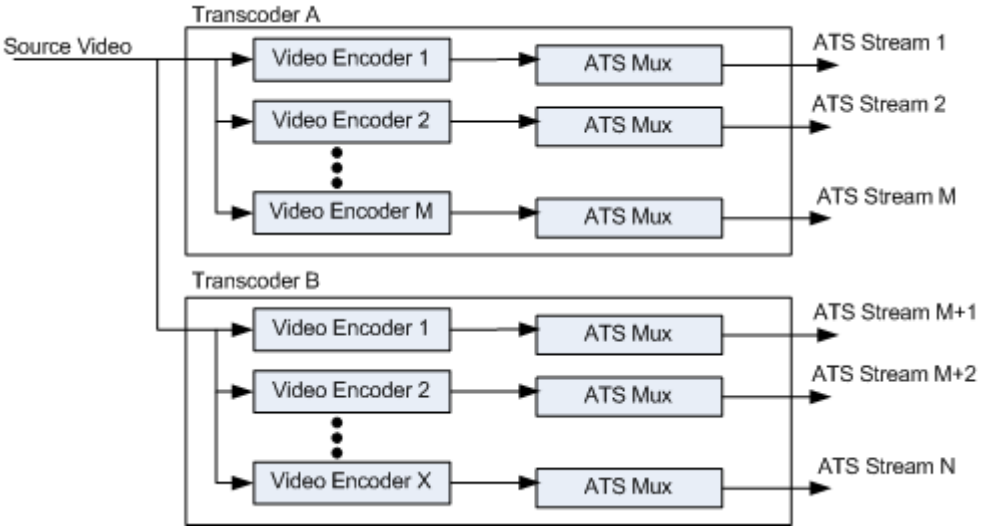

Figure 25 - Video transcoding across one or more systems

9.2 Video Chunk Sync: Start-up Considerations

ABR conditioning for video requires that, regardless of when any given encoder starts, it SHALL produce chunk boundaries on the same source frame even across streams of differing frame rates. This is referred to as **Chunk Sync**. Chunks targeted for a specific ABR format SHALL be in sync with one another. For example, HSS streams SHALL be in fragment-sync with one another. HLS streams SHALL be in segment-sync with one another. As described previously, boundaries of longer duration chunks, such as HLS segments, SHOULD be in sync with boundaries of shorter duration chunks such as HSS fragments. Since chunk boundaries in the video domain are IDR frames, this cross-format chunk sync minimizes the number of IDR frames, which are bandwidth expensive.

Figure 26 illustrates both fragment and segment chunks. As shown, an encoder cannot arbitrarily start encoding on any given frame. The encoder SHALL choose a source frame that is fragment and segment aligned to all other streams for the given presentation. Subsequently, all future fragments/segments SHALL also be aligned to the same source frame.

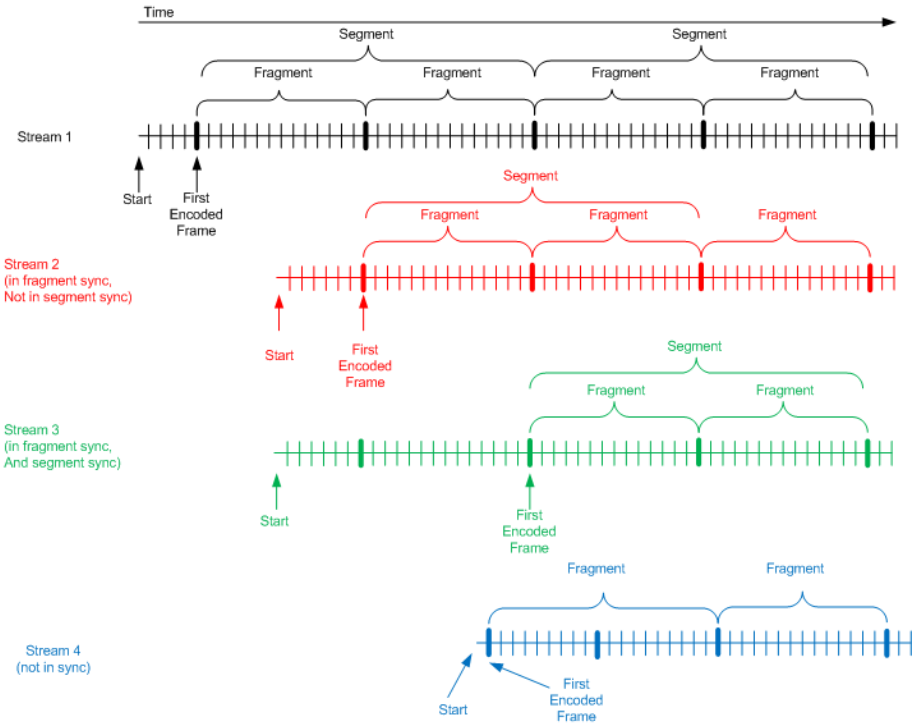

Figure 26 - Conditioning Video Start

9.3 Video Adaptive Sync

When identical AUs across representations have the same timestamp, this is known as being in Time Sync. When Chunk Synced streams are also in Time Sync, the output presentation timestamp of the first chunk boundary SHALL be identical across all encoders. Future chunk boundary frames SHALL also have identical presentation times. This is referred to as being in Adaptive Sync.

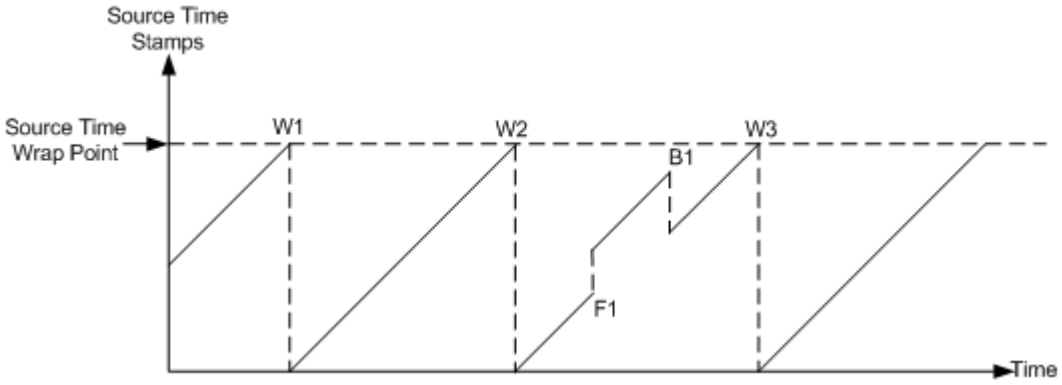

Figure 27 - Source Time Discontinuities

Figure 27 illustrates various source time discontinuities. Many source time bases have wrap points due to the limited precision of the timestamp. For example, a 33-bit MPEG-2 TS PTS will wrap about every 26.5 hours. SDI sources with timecode may wrap every 24 hours. Such wraps are shown as W1, W2, W3. Sources may also exhibit forward discontinuities (F1) or backward discontinuities (B1). If two encoders start on either side of any number of discontinuities, not only SHALL they be in chunk sync, but the output timestamps of the frames SHALL be the same.

9.4 SCTE 35 and Splice Points

The ATS stream MAY contain SCTE 35 descriptor and trigger points that were either carried through on the source stream of the ATS transcoder or inserted at the time of the transcoding of the source stream. SCTE 35 descriptors may be identifying parts of the stream. SCTE 35 triggers indicate a point in the stream where conditioning of the ATS stream SHALL occur. SCTE 35 triggers can indicate AD insertion points. It can also indicate other types of conditioned inserted content. In order to handle single decoder end client systems, a conditioned point in the ATS stream SHALL occur at the beginning of a chunk.

9.5 Ad Breaks

The replacement of ads from source content with alternate ad content may take place at downstream ABR encoders or encapsulators. The method for some ABR formats, such as HLS, involves the complete replacement of encapsulated chunks with alternate chunks. For this reason, source ad content for ABR formats SHOULD be chunked discretely. This means that a new chunk will be started at the first frame of an advertising break. Also, a new chunk will be started at the first frame returning from an advertising break.

Figure 28 illustrates a break Out of programming (Ad start) and a return In to programming (Ad end). At the boundaries, the previous segment ends and a new one is started. This produces a split in what would have been a single segment.

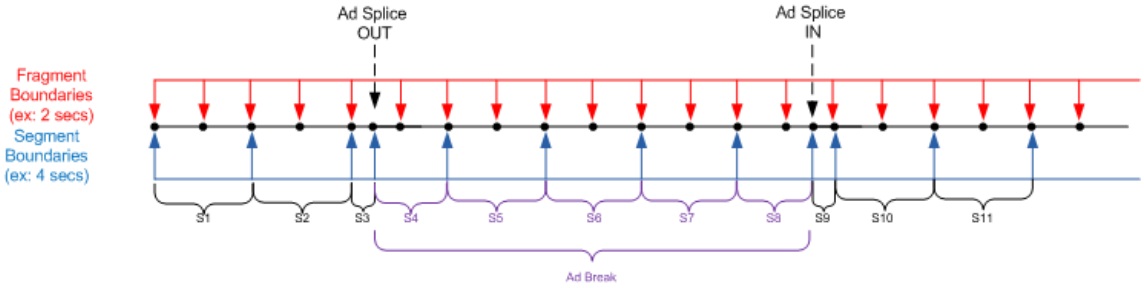

Figure 28 - New Segment at Ad Boundaries

Figure 29 illustrates the same break boundaries. Here, both new fragments and segments were created at the boundaries.

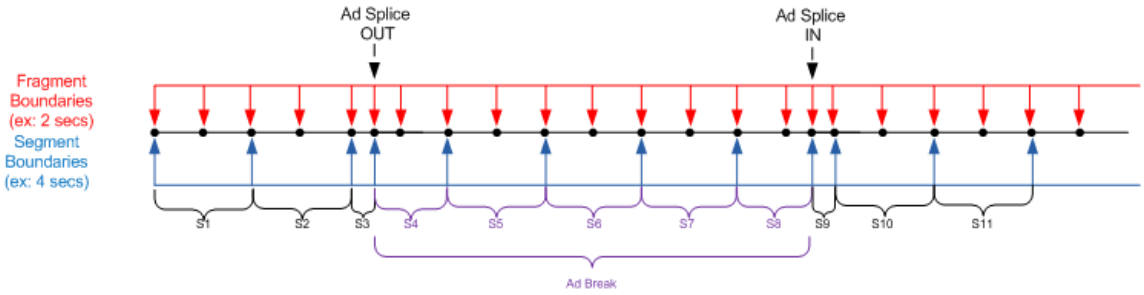

Figure 29 - New Segment and Fragment at Ad Boundaries

An Out point and an In point MAY be co-located; that is, a single boundary MAY serve as both a safe place to leave and a safe place to return. Additionally, an Out point MAY be followed by another Out point. As shown in Figure 30, this allows an ad break to be composed of multiple breaks and to specify at which break boundaries it is safe to return to programming.

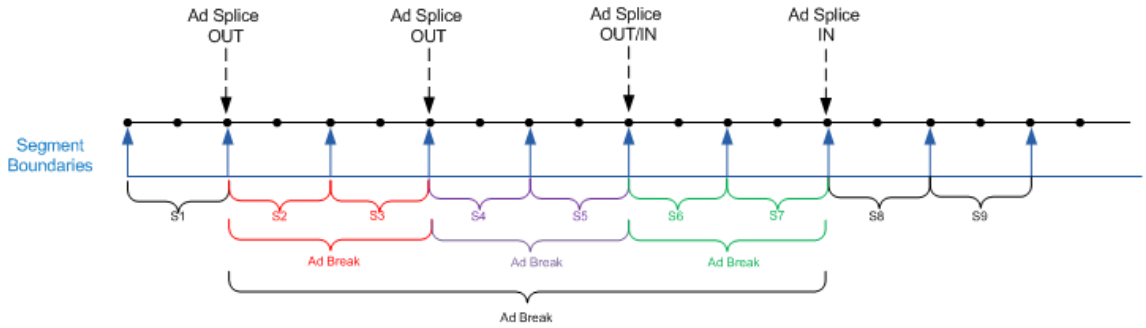

Figure 30 - Multiple Break Boundaries

9.6 Data, Audio, Video Stream Alignment Considerations

There is no time alignment expectation between PES encapsulated audio or data AUs with video AUs, i.e., the start time of an audio or data AU is rarely exactly the same as a corresponding video AU. For instance, in the case of 48-KHz AAC-LC audio, there are approximately 47 AU/sec compared to approximately 30 AU/sec for video. Consider Figure 31, which shows a timeline of video and audio AUs (note this is not meant to represent how AUs may be interleaved in a container format).

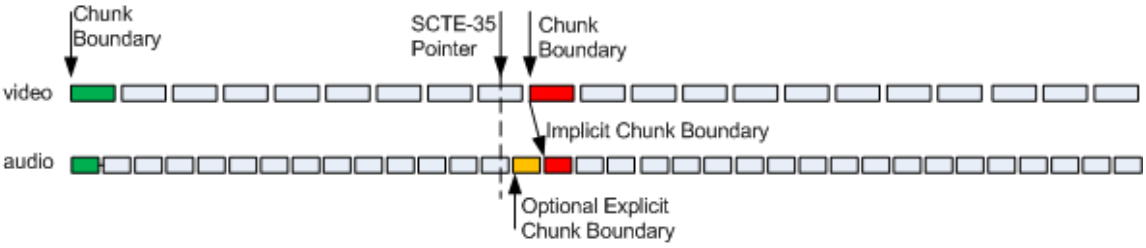

Figure 31 - Splice point and Video and Audio AUs

An advertising splice point is further shown. As described in the previous section, the first video AU >= splice point is chosen as the first AU of the new chunk. This is shown in red.

However, as we'll learn later in this document, the EBP specification details that the first audio AU of an **implicit** chunk is the first AU with sap type 1 or 2 with a PTS >= to the corresponding video AU for the chunk. This is shown as the red audio AU. This is not the first audio AU >= the splice point, shown as the yellow audio AU in the figure.

The first AU of an **explicit** audio chunk does not have to have this relationship to video. Thus the first audio AU of an explicit chunk can be the one >= the splice point, shown in yellow. Ideally, however, an encoder typically chooses to align implicit and explicit audio chunk boundaries at some boundaries, namely advertising and program delineation boundaries. DASH's tendency to create new Periods at such boundaries is one such reason for this ideal alignment.

9.7 Input Frame Loss

If one encoding system experiences input loss that is not experienced by another encoding system at the frame anticipated to be a boundary point, it may choose to conceal the loss by some means. Concealment produces a boundary point frame with a timestamp equivalent to what would have been used without the loss. It is highly recommended that concealment is used because it will produce switchable points since all streams will continue to have Time Sync and Chunk Sync. The EBP specification contains a bit-field to indicate when concealment is used on a boundary point.

If concealment is not used, according to [MPEG2-TS] the bitstream is to be marked with a time discontinuity in the transport adaptation field. How downstream encapsulation systems handle such non-concealed discontinuities is out of the scope of this document. Such systems may choose to further conceal or not; if not, then the chunks may not be obtainable by the client and/or may not be switchable (i.e., client may not be able to switch to it from another stream).

9.8 Audio Conditioning and Synchronization

The base interleave method for an ATS (refer to Section 8.1) is an interleave of a given video presentation and all audio representations. In this method, each ATS SHALL carry the same audio content. Other methods may allow for carrying a subset of languages multiplexed with video as well as allowing audio only ATS streams where audio is not multiplexed with video. In any method, audio SHALL be conditioned. The following describes audio conditioning.

In a single transcoding system, audio may be encoded once and provided to multiple ATS muxes as shown in Figure 32. In this design case, audio data and their presentation timestamps in each ATS will be identical.

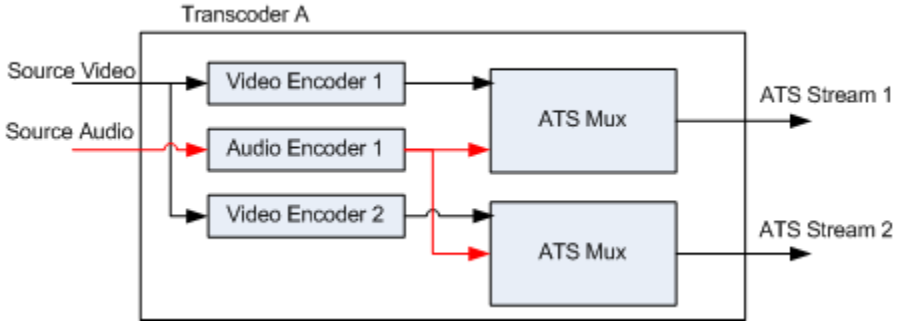

Figure 32 - ATS Transcoder with Audio distributed to Multiple Muxes

However, there may be designs where audio is encoded discretely for some set of ATS streams. In the single transcoding system in Figure 33, audio is encoded by two different encoders.

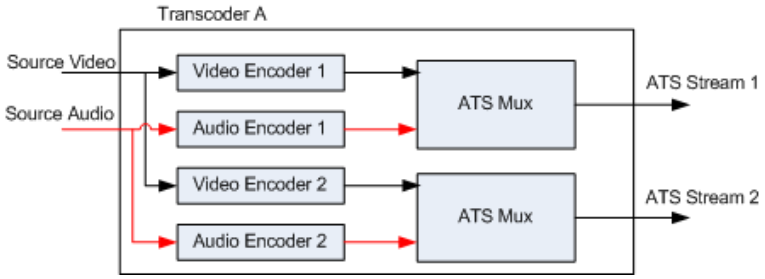

Figure 33 - ATS Transcoder with Audio encoded for each output

In Figure 34, audio is encoded only once in a given transcoder; however, two transcoders are used for stream density or redundancy purposes.

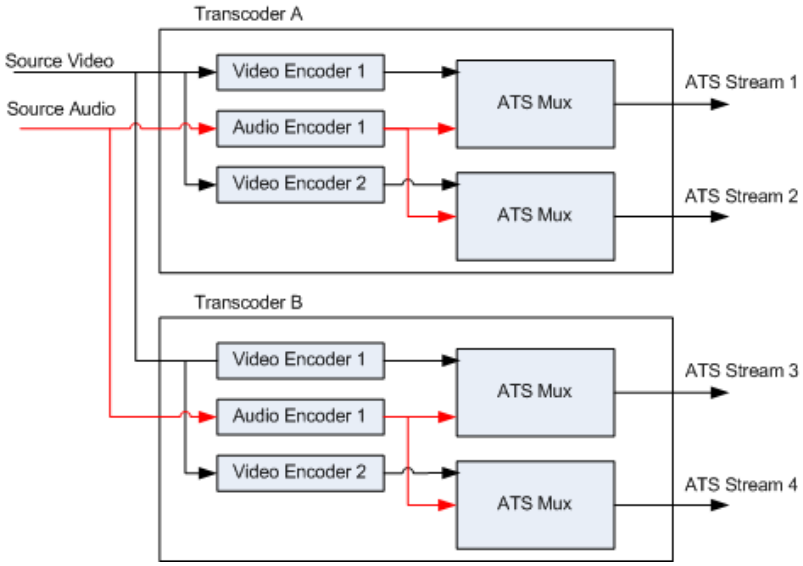

Figure 34 - Audio encoded in multiple ATS Transcoders

For adaptive purposes, audio data SHALL be time synchronous across streams and across transcoding systems independently of the time at which transcoding starts.

Uncompressed audio data takes the form of discrete samples. A sample rate of 48,000 samples per second (48 KHz) is common. In contrast to video, audio encoders do not output discrete audio samples. Some audio encoders produce discrete output access units (AUs) given some number of fixed input samples. For example, an AAC audio encoder produces an output AU for exactly 1024 input samples. Thus, the duration of an AAC AU is defined as: 1024 / input sample rate. A 48 KHz audio source flowing into an AAC audio encoder produces AUs of 21.33 millisecond durations (1024/48000), 46.875 AUs per second.

Adaptive Streaming stream-switching requires AUs across streams to be in time sync with one another in order to provide seamless transitioning from one stream to another. In the audio domain, this means that the time stamps of AUs outputted from different audio encoders need to be the same.

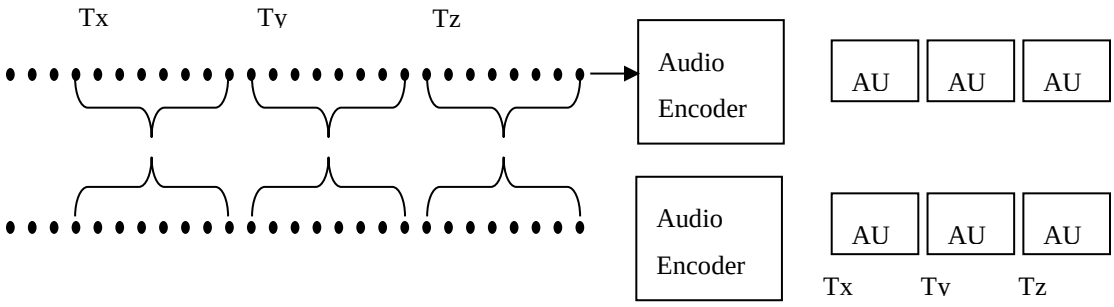

Figure 35 - Audio Access Units: Groups of Samples

If two audio encoders start at exactly the same instant, the set of samples entering the audio encoders will be exactly the same time; more specifically, the initial sample entering each encoder has the same time stamp. In Figure 35, this time stamp is Tx. This initial sample and N-1 further samples will result in the output of a single encoded audio AU. The time stamp of this AU is defined as the time stamp of the first sample it represents, which would again be Tx in this example. Subsequently, another N samples starting at time Ty enter the encoder and another AU with time Ty is generated. It follows that another N samples starting at time Tz enter the encoder and another AU with time Tz is generated.

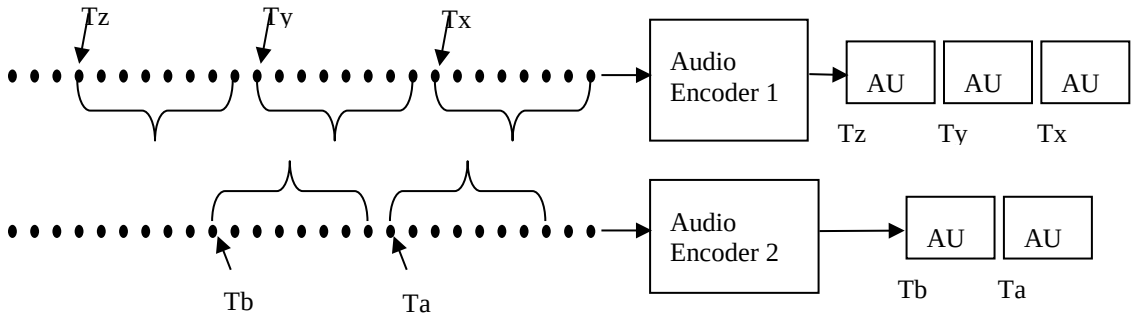

Figure 36 - Nonaligned Audio Access Units

Figure 36 illustrates the case when two encoders do not start at the same time. Encoder 2 started later than encoder 1. The first audio sample entering audio encoder 2, having time stamp Ta, did not align with the first sample of the (depicted) second AU of encoder 1 having time stamp Ty. Because of this discrepancy, the discrete AUs from the encoders are not in sync with one another. If an Adaptive Streaming solution switched from a stream from audio encoder 1 to a stream from audio encoder 2, some number of discrete audio samples would be repeated due to the overlap in AUs. This could cause an audio distortion (pop) or pause, or some other audible manifestation to the end user.

In order to achieve audio synchronization for Adaptive Streaming, the time stamps of audio AUs have to be the same and therefore the audio data represented by the AUs have to be the same. Thus the first audio sample entering an audio encoder cannot be arbitrary. It SHALL fall on a boundary such that, independent of when an audio encoder starts, this first sample is synchronous with the first sample of an AU of any other audio encoders, regardless of when these encoders started.

9.9 Audio Chunk Boundary requirements

For most audio codecs, each Access Unit is fully decodable and may be used as a chunk boundary. However, for AAC family codecs (AAC LC, HE AAC, HE AAC v2), Access Units used as chunk boundaries should meet the requirements for AAC Random Access Points as not every AU is required to carry metadata required to reproduce the encoded audio with full fidelity.

In particular, the following headers are not required to be carried on every audio AU:

- Audio Specific Config (ASC) to configure the audio decoder, containing a Program Config Element (PCE) to describe the channel configuration.
- Spectral Band Replication (SBR) header. When SBR is in use, all AUs will carry SBR data in a FIL (filler) element, but an SBR header is required to start decoding the data.
- Parametric Stereo (PS) header. When PS is in use, all AUs will carry PS data in a FIL element, but a PS header is required to start decoding the data.
- Loudness, Downmix and Dynamic Range Control information, carried in one or more of PCE (subsequent to the one present in the ASC), Data Stream Element (DVB ancillary data) and FIL element.

Loudness and Downmix information typically do not change within a program, but may be changed as the result of an ad insertion. DRC information will vary with program content.

When switching between audio streams with different encoding parameters, or entering a new stream, the lack of PS and SBR header data may result in a reduction of audio quality until that header data is acquired. Lack of loudness and downmix metadata may result in unacceptable changes in perceived loudness, particularly if the change occurs at an ad insertion point.

To enable switching between HE AAC audio streams, all Representations within an Adaptation Set SHALL meet ONE or more of the following requirements:

1. Identical audio encoding parameters are used for corresponding audio components in each Representation.
2. The Access Unit aligned with a chunk boundary in each audio PES packet SHALL meet the requirements for an audio RAP. Subsequent AUs carried in the same PES packet do not need to meet these requirements.

An audio Random Access Point for AAC family codecs SHALL meet the following requirements:

- For AAC-LC, HE AAC and HE AAC v2, if channel configuration 0 is used, the **ProgramConfigElement()** (PCE) containing the actual channel configuration SHALL be present.
- For AAC-LC, HE AAC V1 and HE AAC V2, both **dynamic_range_info()** and **MPEG4_ancillary_data()** SHALL be present. In addition, the program reference level SHALL be present in **dynamic_range_info()**, i.e., **prog_ref_level_present** SHALL be set to 0x1. In the **MPEG4_ancillary_data()** structure, **downmixing_levels_MPEG4_status** SHALL be set to 1.
- For HE AAC V1 and HE AAC V2, **bs_header_flag** SHALL be set to 1, i.e., **sbr_header()** SHALL be present.
- For HE AAC V2, **enable_ps_header** SHALL be set to 1, i.e., the PS decoder configuration data SHALL be present.

9.10 Audio Alignment Across Representations

Audio streams of the same format, language, sample rate and channels, thus differing only in bitrate, SHALL be aligned across streams as described in the previous sections.

Annex A Guidelines For Boundary Creation

A.1 Adjusting Chunk Durations

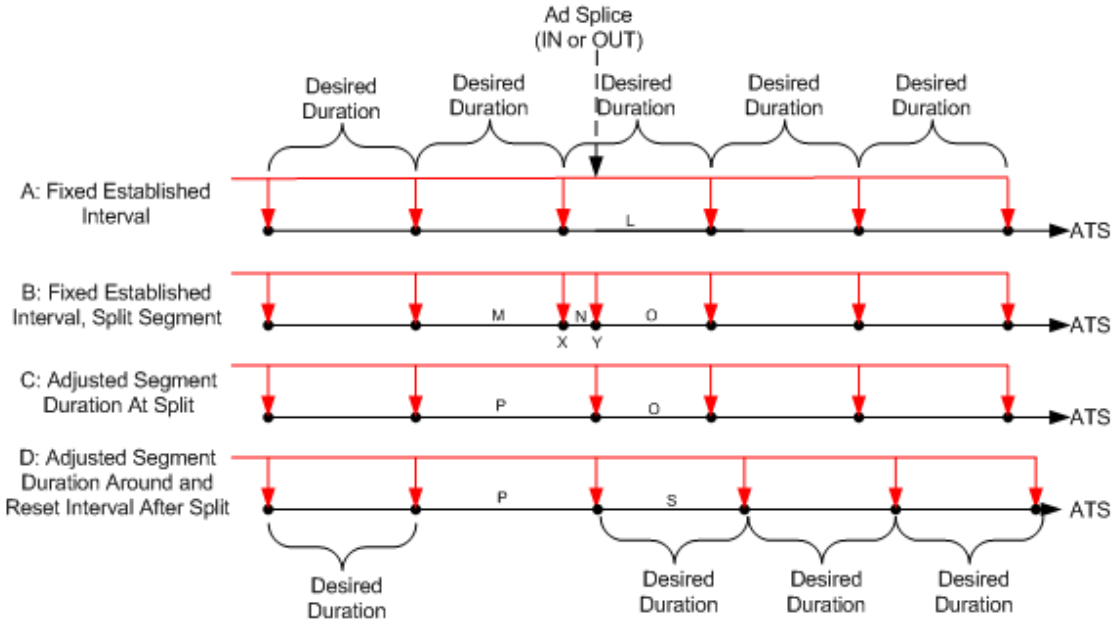

Figure 37 - Adjusting Durations Around/After Splice Points

For Figure 37, Timeline A shows an ATS stream with chunk boundaries of a fixed duration. An advertising splice point occurs within a chunk (L). Per Section 9.4, a new chunk needs to start at this point. This is shown initially in timeline B. Prior to the splice point (Y), a new chunk was started at X due to the creation of chunks at fixed established intervals. The result of starting a new chunk at Y splits what was chunk L in timeline A into two smaller chunks, N and O.

When the size (duration, number of frames) of N is small, a transcoder can choose to adjust the size of a chunk as shown in timeline C. In this example, both a longer (P) and shorter (O) chunk is created around the splice point. However, the duration of the splice point chunk (O) is set so that future chunks align with the original timeline A.

In timeline D, only a single chunk (P) is adjusted (longer in this case) to align with the splice point. The duration of the splice point chunk (S) is set to the desired target chunk duration. Because of this, the boundaries of future chunks no longer align with timeline A.

In Figure 38, a similar example is shown but with a splice point leading up to (versus trailing in as in Figure 37) a chunk boundary. Timeline E is similar to the approach of Timeline C, creating a shorter (T) and longer (U) chunk around the splice point and maintaining chunk boundaries with timeline A. Timeline F is similar to the approach of Timeline D, creating a shorter chunk (T), in this case around the splice point, but then maintaining desired chunk durations afterwards, subsequently becoming unaligned with timeline A.

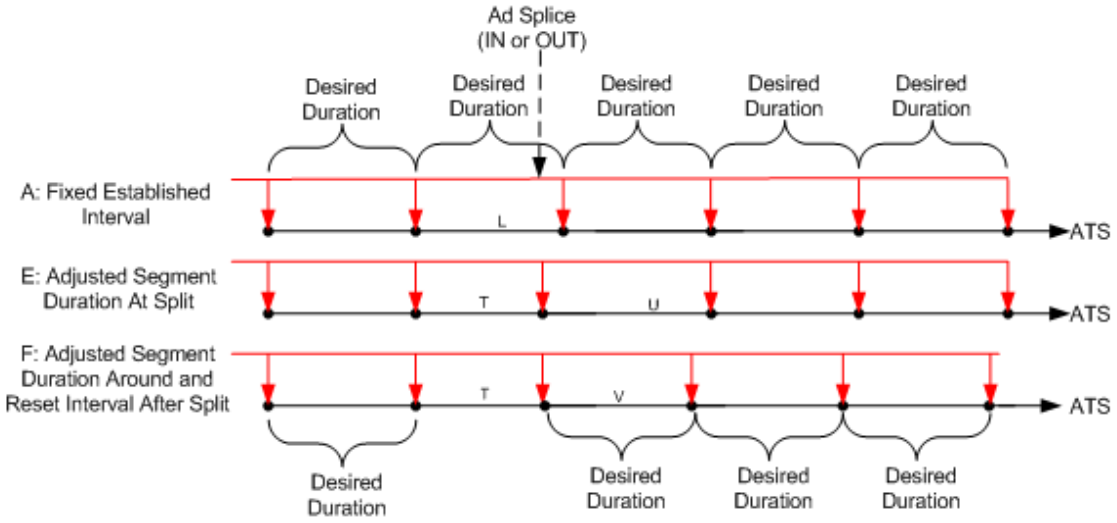

Figure 38 - Example 2, Adjusting Durations Around/After Splice Points

These techniques can apply to scene changes or other conditions that force the placement of an Intra picture in close proximity to a desired boundary position.

All approaches are acceptable. However, if N is less than 1/2 second, then approaches C/E or D/F are highly desired. Furthermore, [DASH264] specifies a +/- 50% tolerance on target durations except for the last chunk of a period, which has no duration restriction. An ATS follows this restriction in most circumstances. For example, the smallest chunk in an ATS partition with a 2-second target duration is 1.5 seconds and the largest chunk is 2.5 seconds. Because of this restriction, depending on the location of a splice point, it could not always be possible to achieve approaches C/D above where alignment to the original timeline past the splice point occurs at the next boundary, although alignment to the original timeline would be possible at the subsequent boundary. Refer to Figure 39.

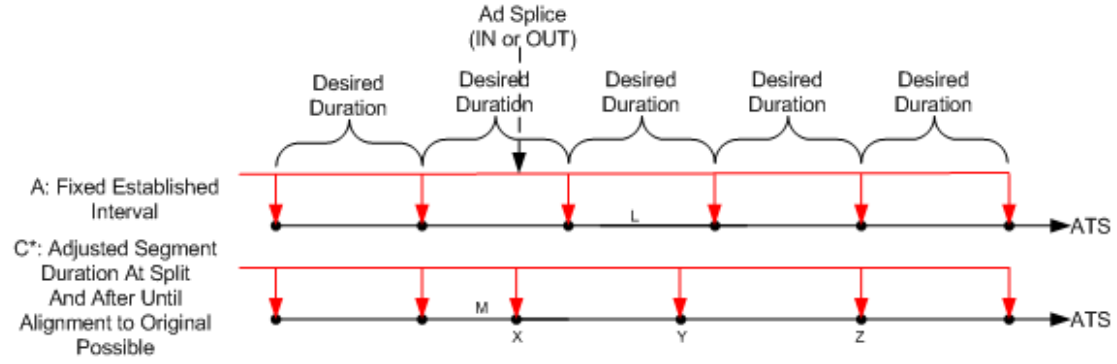

Figure 39 - Adjusting Durations with DASH264 25% Tolerance to Align to Original Timeline

A.2 Frame Rate Decimation

In order to maintain alignment across representations with differing frame rates, restrictions on chunk sizes need to be imposed. For example, consider a 25 fps source encoded to both a 25 fps stream and a 12.5 fps stream where every other source frame is skipped.

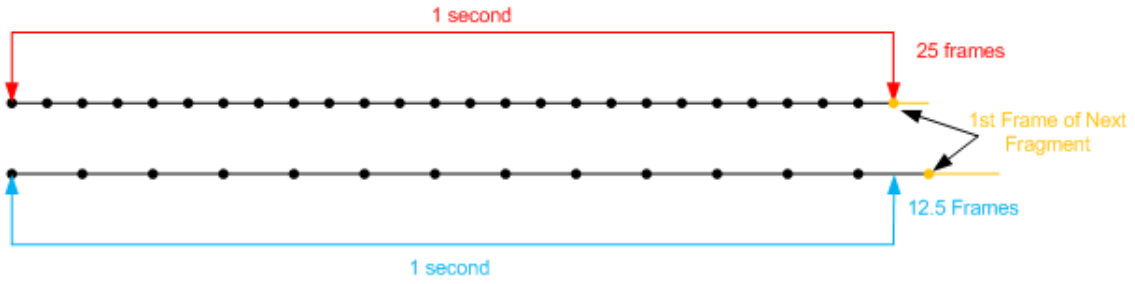

Figure 40 - One Second of Frame Rate Decimation (25 to 12.5)

As Figure 40 illustrates, there is not a common frame between these two streams at the 1-second mark. If a chunk boundary was made after the 1st second as shown above, the two streams would not be in chunk sync nor time sync with one another.

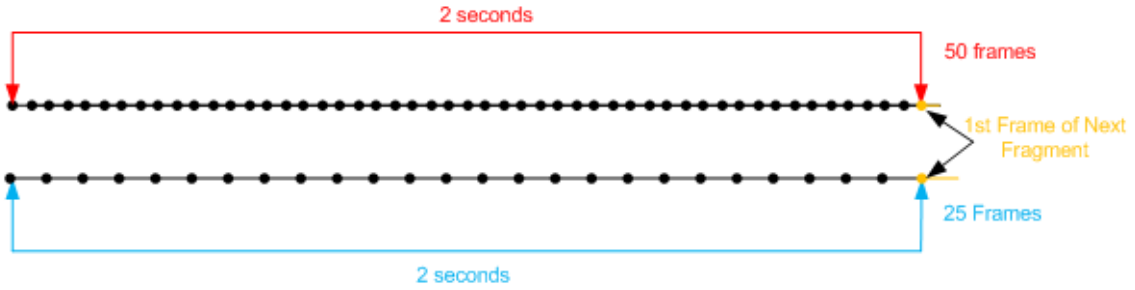

Figure 41 - Two Seconds of Frame Rate Decimation (25 to 12.5)

If we extend the timeline to two seconds as shown in Figure 41, we can see there is an alignment across the two representations at the two-second mark. As this example illustrates, an ABR representation with multiple frame rates is restricted in possible chunk sizes.

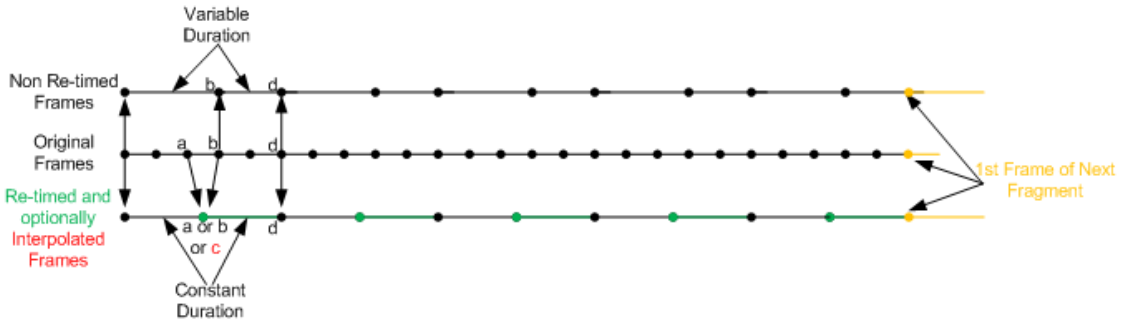

Figure 42 - Non integer frame rate reduction

Figure 42 illustrates non-integer frame rate decimation (i.e., where source frame rate is not divided by an integer). Here the source frame rate is 25 fps and an output frame rate is 10 fps. An encoder can choose multiple approaches to producing frames and the times of these frames.

- Select a source frame and use its timestamp. This is shown in the top timeline. Using this approach will produce variable frame durations, but the decimated frame timestamps will match non-decimated frame timestamps. Note that even with this approach, two different frame rate reduced streams will not have corresponding matching frame times.

- Select a source frame and interpolate its timestamp. This is shown in the bottom timeline. The green samples are re-timed but the frames themselves (a or b) are identical to one of the closest source frames.
- Interpolate an output frame from multiple source frames and interpolate its timestamp. This is also shown in the bottom timeline. The green samples are re-timed but the frames themselves (c) are interpolated.

A.3 Boundaries Desired at Unaligned AUs

The previous section describes creating aligned chunks across multiple frame rates. In Figure 43, a naturally occurring chunk boundary is illustrated. Four different streams are represented; the first (1x) has every AU represented. The second is at 1/2 the frame rate and thus each AU is 2x the duration of the first stream. The third is at 1/3 the frame rate and thus each AU is 3x the duration of the first stream. The fourth is at 1/4 the frame rate and thus each AU is 4x the duration of the first stream. In this example the least common multiple for alignment is 12 frames and thus the target duration for chunks would be a multiple of 12 frames.

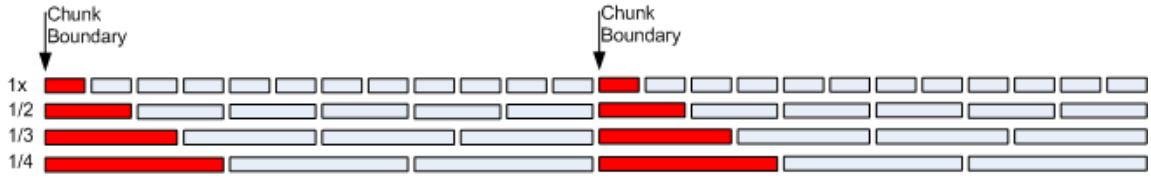

Figure 43 - Naturally Aligned Chunk Boundary

Now consider the need to start a new chunk at an arbitrary AU. This may be because of an advertising in or out point, a program delineation point, a scene change in close proximity to an anticipated boundary, etc. Figure 44 shows the same diagram as Figure 43 but with the desire to start a new chunk due to an [SCTE 35] marker.

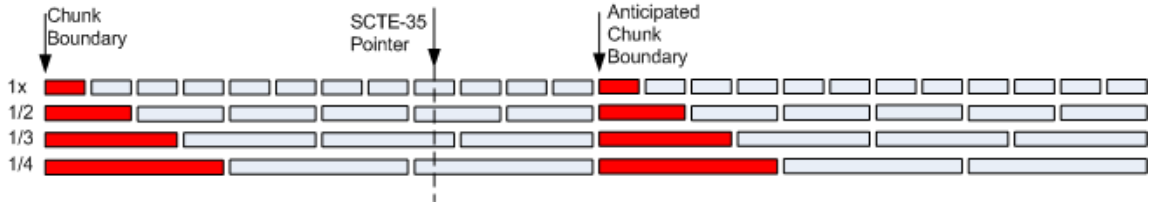

Figure 44 - Desire for new Chunk

In this case, the SCTE 35 metadata is pointing to a time position in the middle of an AU. As a result, the desire is to start a new chunk on the first AU following this point. However, as seen in Figure 45 for this example, this first AU for stream 1 only aligns with an AU from frame rate decimated stream 3. This does not align with AUs with streams 2 and 4.

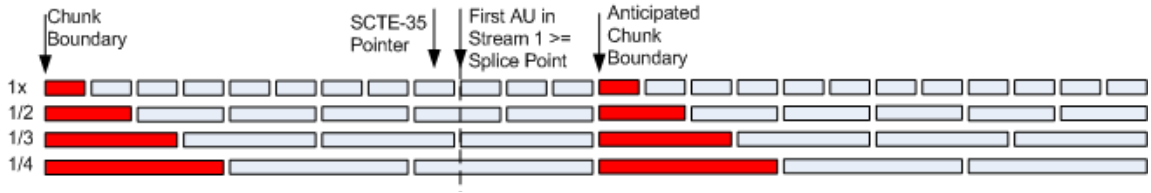

Figure 45 - First AU >= Splice Point

If one were to simply start a new chunk in each stream with that stream's first AU that was >= to the splice point, one may end up with something like Figure 46. In this figure we see for each stream a new chunk (in red) starting with that stream's first AU that was >= to the splice point.

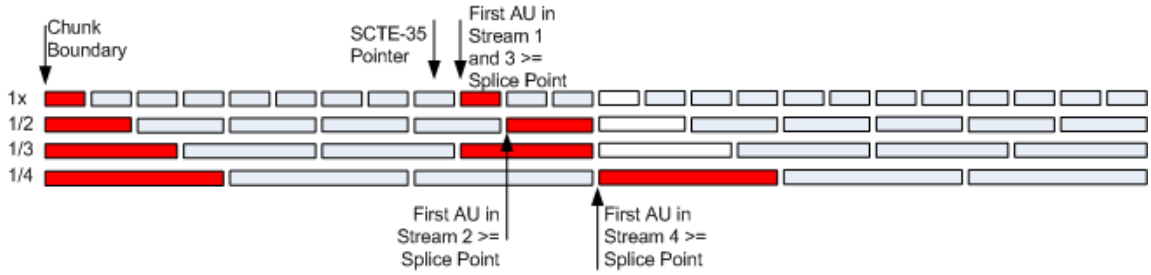

Figure 46 - Chunk simply starting at next frame rate decimated frame AU >= splice point

The issue with this approach is that the streams are not in adaptive sync with one another at this location. Specifically in this example, stream 1 and 3 are still in sync but neither of these is in sync with streams 2 and 4, which in turn are not in sync with each other. To address this problem, the first source AU prior to frame decimation that is >= the splice point needs to be used as the first AU for the new chunk in each stream. In order to get alignment across various chunk sizes with differing frame rates, restrictions on chunk sizes needs to be imposed. This is illustrated in four different methods in the following figures. Recall that Stream 1 represents the unaltered or highest frame rate of all representations.

In the following three cases, a new chunk is started in each stream, regardless of frame rate, on the first source AU >= the splice point.

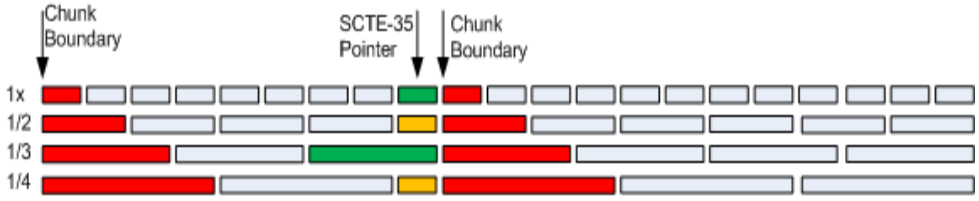

Figure 47 - Using AUs to Complete Chunk

Option 1: In Figure 47, prior to the new chunk boundary, each stream encodes its next AU as it normally would have done. However, in some cases, the duration of this AU in the frame rate decimated streams is shorter than normal. These are shown in yellow (streams 2 and 4). In this example, stream 3's AU, shown in green, is of normal duration prior to the new chunk boundary. Stream 1's AU, the unaltered or highest frame rate stream, will always be of normal duration.

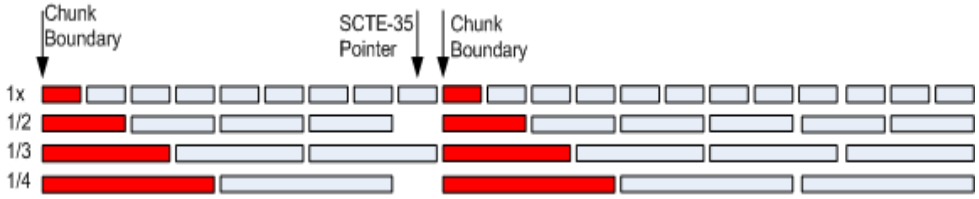

Figure 48 - Skipping AUs leading up to the next Chunk

Option 2: In Figure 48, prior to the new chunk boundary, each stream does not encode its next AU if the duration of this AU would extend beyond the new chunk boundary. This would result in a time discontinuity in some streams. In this example, this occurs in streams 3 and 4. The figure represents this with gaps in the timeline.

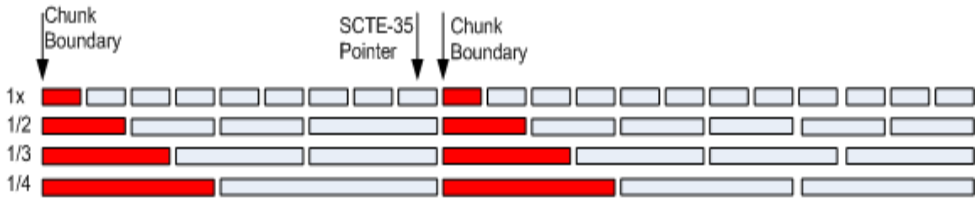

Figure 49 - Skipping AUs leading up to the next Chunk and Extending previous AU's Duration

Option 3: Figure 49's method is similar to Figure 48, content identical: prior to the new chunk boundary, each stream does not encode its next AU if the duration of this AU would extend beyond the new chunk boundary. However, the duration of the AU prior to this skipped AU is extended to the new chunk boundary. While the same number of AUs is represented here as in Figure 48, there is no time discontinuity due to extending the duration of AUs. In this example, this occurs in streams 2 and 4, which were both extended by one AU time interval.

Option 4: In this use case, a new chunk is not started on the first source AU >= the splice point. Instead the first AU >= splice point that is aligned across all streams is chosen. In the example, depicted in Figure 50, we see that the first AU that is aligned across all streams does not occur until three frames after the first AU >= splice point.

Figure 50 - Selecting first AU aligned across four bitrates

In the example depicted in Figure 51, there are only two frame rates and thus, in the worst case, the first AU that is aligned across all streams will occur not more than one frame after the first AU >= splice point.

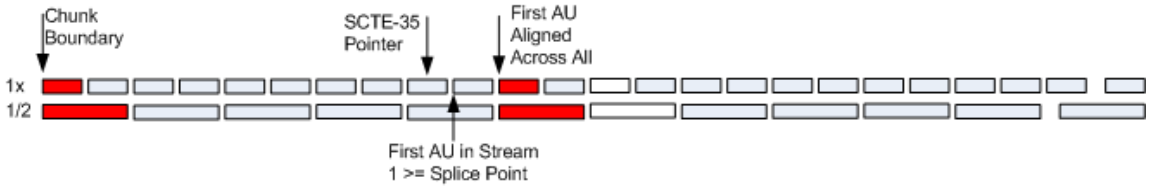

Figure 51 - Selecting first AU aligned across two bitrates

While Option 4 does not create a boundary on the first AU >= splice point, if the number of AUs following this AU, as in Figure 51, is small, this option is ideal because it does not influence the frame rates of the low-rate streams by encoding extra frames as in Option 1 or not encoding frames as in Options 2 and 3. However, it is not anticipated that the [SCTE 35] message(s) for this splice point would be modified to account for the new location. If any downstream processing could not accept this, then this option is probably not to be used.

A.4 Boundaries for Scene Changes in Multi Framerate Streams

The examples in the previous section considered an advertising splice point (in or out) but the same considerations and techniques apply to other factors creating new chunks at boundaries that do not naturally align across all frame rate streams. For example, consider Figure 52, which shows a scene change occurring in close proximity to an anticipated chunk boundary.

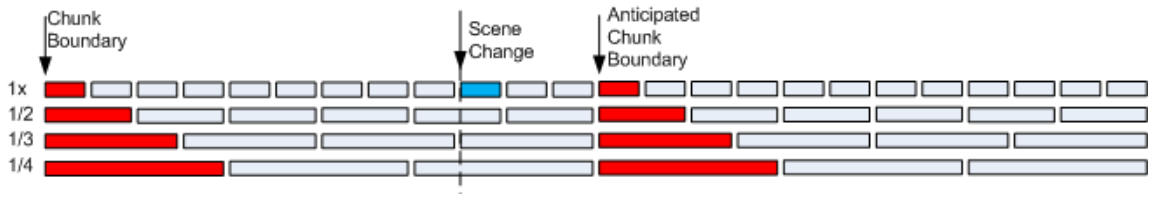

Figure 52 - Scene change in close proximity to anticipated chunk boundary

With no special handling, the resultant encode could produce something like Figure 53, where the scene change frames in each stream (blue) are encoded as I-frames and the boundary frames (red) are encoded as IDR-frames. This approach can impact picture quality due to the excessive bits used to encode the intra encoded frames.

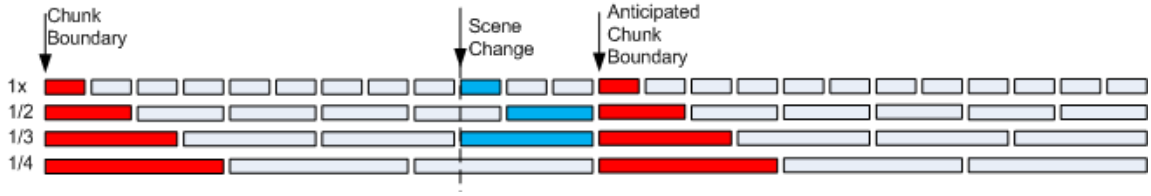

Figure 53 - I (blue) and IDR (red) frames due to Scene Change and Chunk Boundaries

Ideally, the upcoming anticipated boundary is moved up and relocated to the AU of the scene change. The end result could be something like Figure 54, which is similar to the approach shown in Figure 47. Approaches similar to Figure 48 and Figure 49 are also possible.

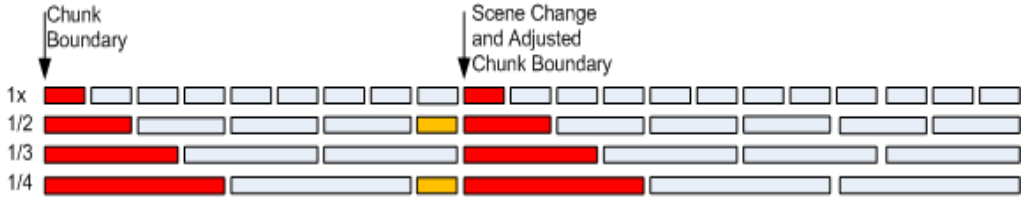

Figure 54 - Adjusted Chunk Boundary at Scene Change