OpenCable™ Specifications

Encoder Boundary Point Specification

OC-SP-EBP-I01-130118

ISSUED

DISCLAIMER

Document Status Sheet

Document Control Number:	OC-SP-EBP-I01-130118			
Document Title:	Encoder Boundary Point Specification			
Revision History:	I01 – Released 1/18/13			
Date:	January 18, 2013			
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 stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.

Closed A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks

Contents

Figures

Tables

This page intentionally left blank.

1 INTRODUCTION

1.1 Overview

The Encoder Boundary Point (EBP) Structure is a signaling mechanism that can provide a hinting mechanism for taking continuous streams conditioned for adaptive streaming and creating discrete chunks of decodable content with boundaries in one component stream in the multiplex (Fragment) or across the multiplex (Segment). The EBP is carried as private data of the adaptation private field of an MPEG-2 TS packet for video or audio and can be applied to each video packetized elementary stream (PES) and audio PES packet, resolved down in many cases to a single Video access unit (AU) or a group of audio AUs.

Additionally, it can provide a timing field to indicate the acquisition time of the stream that can be used for synchronization purposes. Lastly, a labeling mechanism is introduced for a single access unit or region of access units in the content stream.

1.2 Organization

This specification is organized into the following sections:

- Sections 1 - 4 contain the introduction, references, terms and abbreviations.
- Section 5 describes the Encoder Boundary Point Structure.
- Section 6 provides design and coding requirements for the EBP structure.
- Section 7 describes the PMT descriptors related to EBP Data.
- Appendix I contains examples of EBP structures and descriptors.
- Appendix II contains information on derivation of acquisition time.

1.3 Scope

This specification describes the EBP structure element that a transcoder inserts in the output transcoded/encoded conditioned MPEG streams. This document is codec independent and may be applied to the carriage of EBP with packetized elementary streams of any type. Reference is made to [SCTE 128] to be compliant with MPEG-2 Systems layer constraints on the use of adaptation field private data.

These created streams are then sent to an encapsulator (also called a fragmentor or packager) directly or stored to be sent to an encapsulator upon request at a later time. Upon receiving such streams, an encapsulator then processes these streams with EBP and produces chunks according to the one or more adaptive streaming encapsulating technologies.

Alternatively, this structure can be inserted at the time of encoding or added during the transcoding process.

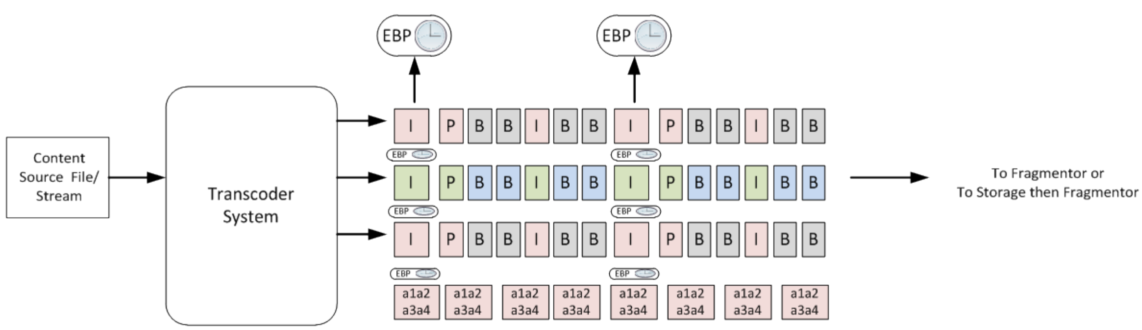


Figure 1 - EBP structure insertion into transcoded conditioned adaptive streams

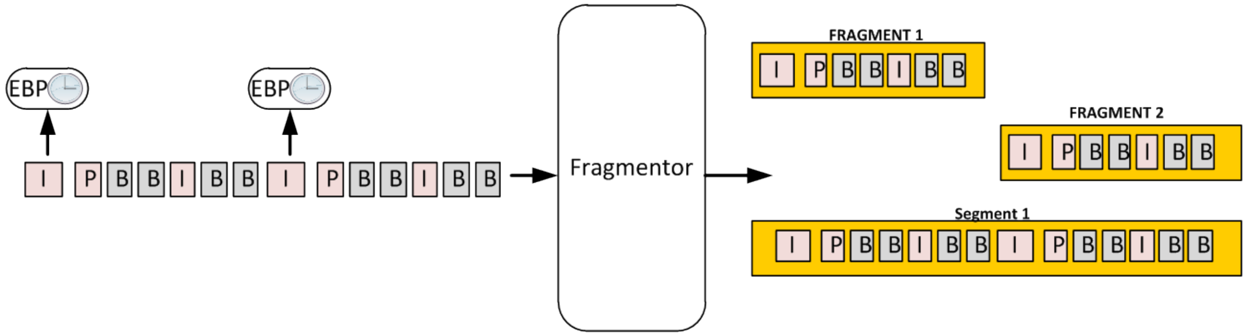


Figure 2 - EBP structure for use in fragmentation processing

1.4 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.

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

[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 13818-1:2007, Information Technology - Generic coding of moving pictures and associated audio information: Systems.

[NTP1] IETF RFC 5905, Network Time Protocol Version 4: Protocol and Algorithm Specification, June 2010.

[SCTE 128] ANSI/SCTE 128 2010-a, AVC Video Systems and Transport Constraints for Cable Television.

2.2 Informative References

The following informative references may be useful in applying this specification.

[AVC] ITU-T Recommendation H.264 | ISO/IEC 14496-10:2012, Information Technology - Coding of audio visual objects - Part 10: Advanced Video Coding.

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

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

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

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

[MPEG2-Video] ISO/IEC 13818-2:2000, Information Technology- Generic coding of moving pictures and associated audio information: Video.

[SCTE 35] ANSI/SCTE 35 2012, Digital Program Insertion Cueing Message for Cable.

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
- 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/SCTE
- ISO/IEC standards - 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/

3 TERMS AND DEFINITIONS

Adaptive Streaming methods supported by this specification rely upon delivering continuous content as a series of discrete sections, referred to as chunks. 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. 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.
Boundary Point	A Boundary Point is an indicated point in the stream that is assigned to a PES packet containing one or more access units. A Chunk Boundary Point indicates the beginning of a chunk, and is a stream access point.
Chunk	A discrete section of content that can be independently decoded, possibly given additional initialization information.
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	An 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. 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 fragmentor, 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.
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 an "explicit chunk".
Explicit Partition	A partition that 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 an "implicit chunk".
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 in [DASH].
Partition	A partition is 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	A stream access point (SAP) enables random access into a media stream as defined in Annex I of [ISO-BMFF].
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 specification uses the following abbreviations:

ATS	Adaptive Transport Stream
AU	Access Unit
DASH	Dynamic Adaptive Streaming over HTTP
DTS	Decoding Time Stamp
EBP	Encoder Boundary Point
ENC	Encoder
ES	Elementary Stream
GOP	Group of Pictures
HDS	HTTP Dynamic Streaming (Adobe)
HLS	HTTP Live Streaming (Apple)
HRD	Hypothetical Reference Decoder
HSS	HTTP Smooth Streaming (Microsoft)
MBR	Multi-Bit Rate
NTP	Network Time Protocol
PCR	Program Clock Reference
PES	Packetized Elementary Stream
PID	Packet ID
PMT	Program Map Table
PTS	Presentation Time Stamp
SAP	Stream Access Point
TS	Transport Stream
VES	Video Elementary Stream

5 ENCODER BOUNDARY POINT (EBP) STRUCTURE

5.1 Structure Placement in Transport Streams

When applied to a video stream, a boundary point is assigned to a PES packet containing one or more access units and is independent from GOP structure (e.g., SAP_type, IDR, P, B, audio packet).

The boundary indicator can also be applied to the one or multiple audio streams that are contained in the same program. Among other purposes, this mechanism SHOULD be used to indicate chunk boundary points in an adaptive streaming context that can be signaled via the transport stream. At the packager, the information in the EBP data field is used as a hint for processing.

The EBP data, shown in Table 1, SHALL be carried in adaptation field private data in accordance with the format and requirements defined in [SCTE 128] section 6.4.3 with a tag value of 0xDF. The associated format_identifier SHALL have an ascii value of 'EBP0'. When present, the EBP structure SHALL be carried in the transport packet that has the Payload Unit Start Indicator set for the PES packet to which it is assigned.

[SCTE 128] allows carriage of more than one type of private data in the adaptation private field of a TS packet. If there is any other type of private data besides the EBP in the adaptation private field, the requirements specified in [SCTE 128] for ordering of private data SHALL be followed. The sum of all types of private data in the adaptation private field of a TS packet SHALL comply with [MPEG2-TS].

An NTP derived acquisition time MAY also be inserted to indicate the time when the transcoder/ encoder acquired this access unit.

NOTE: In MPEG-2 TS, it is possible for a packet to contain only payload, or no payload and only adaptation field, or both can be present within the same packet. As EBP is carried within the adaptation field, it is possible to carry it in the latter two packet types. If there is no payload in a packet that carries the EBP structure, the structure is interpreted as if it were carried in the nearest following MPEG-2 TS packet with the same PID.

5.2 Structure

Table 1 - Syntax of EBP_info()

Syntax	No. of bits	Format
EBP_info() {		
data_field_tag	8	uimsbf
data_field_length	8	uimsbf
format_identifier	32	uimsbf
EBP_fragment_flag	1	bslbf
EBP_segment_flag	1	bslbf
EBP_SAP_flag	1	bslbf
EBP_grouping_flag	1	bslbf
EBP_time_flag	1	bslbf
EBP_concealment_flag	1	bslbf
reserved	1	'0'
EBP_extension_flag	1	bslbf
If (EBP_extension_flag==1) {		
EBP_ext_partition_flag	1	bslbf
reserved	7	'0000000'
}		
if (EBP_SAP_flag==1){		
EBP_SAP_type	3	uimsbf
reserved	5	'00000'
}		
if (EBP_grouping_flag==1){		
EBP_grouping_ext_flag	1	bslbf
EBP_grouping_id	7	uimsbf
While (EBP_grouping_ext_flag==1){		
EBP_grouping_ext_flag	1	bslbf
EBP_grouping_id	7	uimsbf
}		
}		
if (EBP_time_flag==1){		
EBP_acquisition_time	64	uimsbf
}		
if (EBP_ext_partition_flag==1){		
EBP_ext_partitions	8	uimsbf
}		
For (i=0; i<n; i++) {		
EBP_reserved_byte	8	uimsbf
}		
}		

Uimsbf - unsigned integer translated most significant bit first
Bslbf - bit string left bit first

5.3 Semantics

data_field_tag: The data field tag is an 8-bit field which identifies the type of each data field. This field SHALL have the value 0xDF.

data_field_length: This field is the number of bytes following this length field.

format_identifier: This field SHALL be as defined by ISO/IEC 13818-1 [MPEG2-TS], Section 2.6.9, Section 2.10, and Annex O. This field SHALL carry the registered value 'EBP0' (ascii).

EBP_fragment_flag: Bit flag to indicate the start of a specific type of chunk (a Fragment) in the stream.

EBP_segment_flag: Bit flag to indicate the start of a specific type of chunk (a Segment) in the stream. A segment typically contains one or more fragments and is usually aligned with fragments; in this case, EBP_fragment_flag is set when EBP_segment_flag is set. It is possible that a segment can start within a fragment, in which case EBP_fragment_flag is not set when EBP_segment_flag is set.

EBP_SAP_flag: Bit flag to indicate stream access point type at the start of a boundary point, as defined in [ISO-BMFF]. If bit flag is clear, the boundary point shall meet the requirements for SAP type 1 or 2. If the bit flag is "1", indicates that the EBP_SAP_type is present.

EBP_grouping_flag: Bit flag to indicate the presence of the grouping information. A group indicates a continuous set of AUs which can be independent of the GOP structure. Default is '0' unless grouping information needs to be included.

EBP_time_flag: Bit flag to indicate if an NTP-derived acquisition time is included in this EBP Structure (in the *EBP_acquisition_time* field, below). This flag is expected to be '0' for non-live content.

EBP_concealment_flag: Bit flag to indicate if encoder repaired the source stream that may have lost a frame that was at a boundary point. "Repair" is meant to create a boundary point where it may have been skipped due to a lost boundary point frame. "1" indicates a repaired frame. "0" indicates no repairs done to the stream.

EBP_ext_partition: Bit flag indicating the presence of the extended partition field in this structure.

reserved: Reserved for future use.

EBP_extension_flag: Bit flag that indicates an extension byte is in use for additional bit flags.

***EBP_SAP_type*:** 3 bits to indicate SAP type **(**refer to Section 3, Terms and Definitions).

EBP_grouping_ext_flag: Bit flag used to allow for multiple grouping_ids to belong in the same EBP structure. If flag bit is "1", it would indicate another grouping_id would follow. If the flag is clear, then that grouping ID is the only or last grouping_id.

EBP_grouping_id: 7 bits to indicate group number or group pattern.

EBP_acquisition_time: 64-bit NTP [NTP1] timestamp of the boundary point or AU. Time can be acquired through an NTP mechanism. The content of this field is undefined for non-live content.

***EBP_ext_partitions*:** Bitmask representing all partitions to which this boundary point applies. EBP_ext_partition is a 10-bit mask indexed by partition_id.

EBP_reserved_byte: Reserved byte for future use.

6 DESIGN CONSTRAINTS

6.1 Placement of EBP Structure in MPEG Elementary Stream

When EBP signaling is used, it SHALL be carried on the video PES stream and MAY also be carried on audio and data PES packetized elementary streams. When present in the conditioned stream, an EBP SHALL indicate either start of a video access unit, start of an audio access unit, or start of a data packet. The EBP MAY also be aligned with grouping, SAP, GOP, and chunk boundaries. Additionally, it can indicate an acquisition time value that MAY be separate from the stream time indicated by PTS.

For video, the EBP SHALL exist for the start of a chunk, but it MAY exist for any access unit in the video stream. For audio, the EBP can be used to indicate audio chunks. When the EBP is used to indicate chunk boundary points in the bit stream, it can be used across an aligned set of MBR TS streams (including audio and data) to create specific ABR format fragments and segments used in adaptive streaming technologies. Downstream adaptive streaming fragmentor/encapsulators use the video EBPs to help with encapsulation and can optionally use the EBP on the audio/data to aid in fragmentation of the audio bitstream. EBP is an explicit way of indicating chunk alignment, but dependent elementary streams could alternatively determine alignment in an implicit manner.

NOTE: The duration of the explicit chunk is usually of constant duration for the majority of such chunks. Occasionally the explicit chunk duration can vary, but SHOULD be recoverable within a few chunks to ensure the cadence of the constant duration chunk is minimally interrupted. This is typically the case with Smooth Streaming Fragments.

6.2 ATS Format

The ATS (Adaptive Transport Stream) format allows for streaming/storage of adaptive streaming content in a generic manner without restricting this to a particular adaptive streaming technology (HSS/HLS/HDS). The EBP plays a significant role of signaling the stream/file characteristics to allow for adjusting the stream/file to a specific adaptive streaming technology at the downstream fragmentor/encapsulator device. For a set of MBR streams to be in an ATS format, there are several constraints.

The TS format of each stream SHALL comply to [MPEG2-TS] including the HRD buffering. For AVC/H.264, each TS stream SHALL follow [SCTE 128]. As a result, the video and audio AUs are skewed with audio following video.

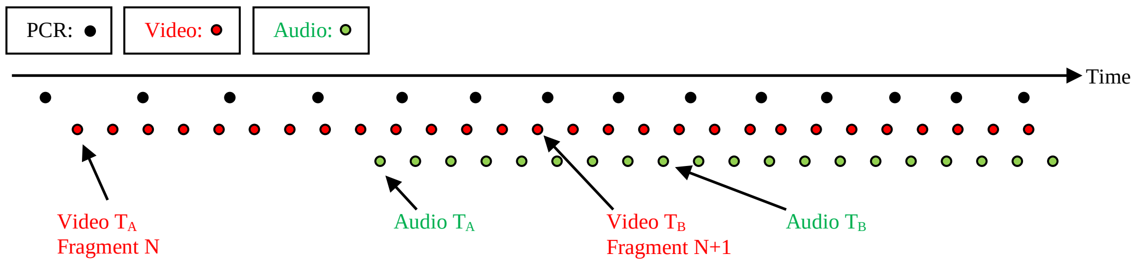


Figure 3 - Example of Audio and Video in a TS stream following HRD model and SCTE 128

The skew between video and audio will affect retrieval of stored files. For example, when retrieving a single stored interleaved (both video and audio) HLS segment, additional byte ranges after or before the segment may need to be requested in order to retrieve all the audio that belongs with the video segment.

For any given media content component in an ATS, the chunks delineated, explicitly or implicitly, by EBPs of a given non-zero partition_id SHALL be non-overlapping as defined in section 4.5.2 of [DASH]. For corresponding media content components in a set of MBR streams, the chunks delineated by EBPs with the same non-zero partition_id SHALL meet the requirements for segmentAlignment=true specified in section 5.3.3.2 of [DASH].

6.3 Video

For video:

- Each video access unit SHALL be completely contained within one PES packet, and the first byte of the PES packet payload SHALL be the first byte of the video access unit.
- More than one EBP structure SHALL NOT be sent with each PES packet header.
- The video AU that is the start of the chunk SHALL be indicated through the EBP signaling mechanism.
- The video AU with EBP that starts the chunk SHOULD be conditioned by a transcoder or encoder to be an intra-coded AU that meets the requirements of SAP type 1 or 2, though it may not be the only intra-coded AU in the chunk. If the video AU starts with a closed GOP (most frequent case), then the EBP_SAP_type in the EBP structure SHOULD NOT be used to indicate SAP type. For all other types of GOP structures, if EBP is present, the EBP_SAP_type SHALL be used to indicate SAP type.

6.4 Audio

There MAY be more than one audio ES associated with the video ES. All audio elementary streams in the ATS SHALL comply with this section.

For audio:

- The PES packet MAY contain multiple audio AUs.
- The PES packet SHALL contain an integral number of audio AUs.
- More than one EBP structure SHALL NOT be sent with each PES packet header.
- Audio AU with an explicit EBP that indicates the start of the audio chunk SHALL start the PES packet.
- The audio AU that indicates the start of the fragment MAY be indicated through the EBP signaling mechanism.
- The audio AU that starts the chunk SHOULD be conditioned by a transcoder or encoder to be an intra-coded AU that meets the requirements of SAP type 1 or 2, though it may not be the only intra-coded AU in the chunk.
- In absence of audio EBP points (i.e., an implicit audio EBP), the first audio AU of a chunk SHALL be the first AU where the presentation time is greater than or equal to presentation time of the corresponding video boundary point.
- Audio AU SHALL NOT lag the aligned video AU by greater than 3 seconds in the MPEG-2 transport stream.

6.5 Acquisition and Timing in EBP structure

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 value MAY be used in linear media feeds to identify time-based recording boundaries, which provide a record of actual acquisition time and SHOULD be placed on chunk boundaries. The EBP acquisition time has no normative use in packaged or VOD content.

The value placed in the EBP acquisition time field is derived by the transcoder or other device from an external time reference and SHALL meet the following requirements:

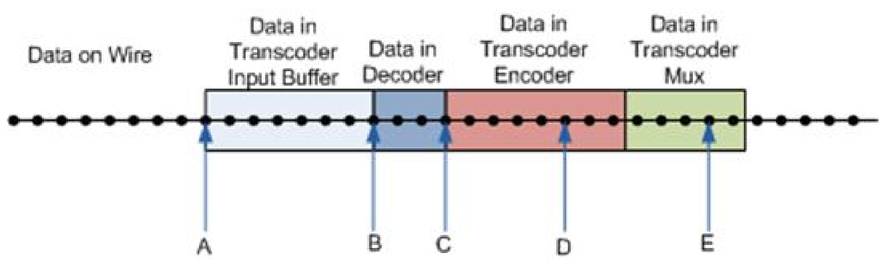
- EBP acquisition time at the start of chunk SHALL be the same across all streams of the MBR being generated from the same source, within the sampling accuracy limit described below.
- EBP acquisition time SHALL be continuous even if PTS of the stream is in rollover mode or has a time discontinuity.
- EBP acquisition time among different transcoders for the same AU in the same input source SHALL NOT differ by more than +/- 300ms.

- When acquisition times are included in:
  - Chunk boundary point partition EBPs, they SHALL be included in every chunk boundary point partition EBP.
  - Other types of EBPs, they SHALL be included at a frequency of at least every 2 seconds.
- The difference of NTP value between two EBP points SHALL reflect the temporal distance between those boundary points with jitter no more than +/- 10 ms. For example, if a fragment chunk is two seconds, it is expected that the difference of NTP at the beginning of this fragment and the next fragment will be two seconds with jitter no more than +/- 10 ms.
- In the case where the incoming streams (arriving at the transcoder) *already* contain EBPs carrying timestamps, the transcoder SHALL preserve the incoming timeline. This means that:
  - When outgoing EBPs remain on the same AU as the incoming, they have the same EBP time value.
  - When outgoing EBPs do not remain on the same AU (e.g., due to re-encoding), they have EBP time values derived from the incoming timeline.
- In the case where the transcoder was using the incoming timeline (EBPs arriving from upstream with acquisition times), but then incoming streams stopped including acquisition times, the transcoder SHALL continue to preserve the incoming timeline.
- In the case where the transcoder was using the local timeline (EBPs arriving from upstream without acquisition times), but then incoming streams started including acquisition times, the transcoder SHALL continue to preserve the local timeline.

Generation of adaptive transport streams may introduce varying delays among the different formats. In order to provide a consistent value of acquisition time among the different formats and different transcoder implementations, this specification describes where the time shall be sampled in terms of reference points in a nominal transcoder model that consists of

- An input buffer to remove network jitter
- A decoder that conforms to the T-STD buffer model
- Video processing and encoding
- Construction of the output multiplex

Figure 4 illustrates these reference points for the nominal transcoder model. It is required that NTP value for each boundary point represents the time at point C when the corresponding access unit exits the decoder buffer of a decoder that conforms to the T-STD buffer model described in [MPEG2-TS]. This allows acquisition times to be based on the presentation order of the access units.



A= enters transcoder dejitter buffer, B= AU enters decoder buffer, C=AU exits decoder output buffer, D= video processing and encoding, E=output multiplexer

Figure 4 - Candidate Reference Time Sample Points in Signal Chain in Nominal Transcoder

Transcoders are required to use time values derived from NTP or an internal clock that is synchronized to NTP. Implementations MAY sample NTP at different points in the signal chain, but they SHALL compensate accordingly to reflect time at point C. If the implementation does not sample NTP for each access unit, it SHALL derive an interpolated value for each access unit.

Transcoder implementations may introduce some amount of dejitter delay between points A and B in the figure above. If a transcoder introduces more than 300 ms of dejitter delay, it SHALL adjust the value derived at point C to compensate for the average delay between points A and B.

Appendix II of this specification contains a more detailed discussion of the derivation of acquisition time.

When NTP timestamps (acquisition time) are inserted into EBPs, the transcoder or encoder SHALL acquire UTC time via NTP originating from a low stratum server (lower than stratum 3), such as a GPS source. Inserted time SHALL use the NTP 64-bit timestamp format [NTP1], a 32-bit unsigned seconds field counting from the prime epoch of 1-Jan-1900 00:00:00 and a 32-bit fraction field.

If the transcoder or other processing device is unable to lock to the external reference, it SHALL follow [NTP1].

Alternatively, the source encoder MAY insert a time value in an EBP structure on pre-designated video frames during creation of the original content stream.

6.6 Grouping ID

The grouping ID construct MAY be used in any stream using EBP and placed on any PES packet whether the AU represents a boundary point or not. If the stream is part of an MBR set, then the construct SHOULD be aligned across the MBR set of streams. The grouping ID can be triggered by event or information in the content source stream or during the transcoding process.

The purpose for the EBP_grouping_ID is to provide an informational label for the boundary to mark that AU in the bitstream. Alternatively, a continuous region in the bitstream can be labeled as well. The start and end flags MAY be used to represent a region with the grouping_id marked for each EBP construct defined within the region.

The grouping ID MAY also be used to label an alternate set of boundary points (other than the default set, which is automatically assumed to be grouping_id=0). This may be useful for alternate sets of downstream devices reading the EBP structure (but following the alternate sets) or providing a repeated counter.

Multiple grouping IDs can be sent within a single EBP structure. This is accomplished by using the EBP_grouping_ext_flag, which when set to "1" allows for another grouping ID to be carried in the next successive byte.

For the start indicator for a specific grouping ID, a grouping ID=126 SHALL be sent as the next immediate grouping ID. For the end indicator for a specific grouping ID, a grouping ID=127 SHALL be sent as the next immediate grouping ID. Between start and end indicators, the specific grouping ID MAY be inserted repeatedly on successive chunk boundaries or alternate frequency repetition on successive AUs as a way to indicate region of the specific grouping ID.

2 Start (or 2 END) indicators for the same grouping ID SHALL NOT be allowed within the same EBP structure. The same grouping ID SHALL NOT be sent within the same EBP structure more than once except for the start and end indicators. The grouping_ID is identified by 7 bits that can represent numbers **0-127**.

Current assignments of grouping_ID are as follows:

0: Default group. If no EBP_grouping byte exists, then it is assumed to be group 0.

1-31: Private

35: Ad Insertion **-** Represents an AU that represents a Splice Out and/or In point. Splice Out point SHALL be identified by EBP_grouping_ID = 35 followed by EBP_grouping_ID = 126 (Start Indicator). Splice In point SHALL be identified by EBP_grouping_ID =35 followed by EBP_grouping_ID = 127 (End Indicator). When EBP_grouping_ID is both a start and end indicator (EBP_grouping_ID = 35 followed by EBP_grouping_ID = 126 and lastly followed by EBP_grouping_ID = 127), then it indicates that this particular AU is both a splice out and splice in point. The "35" grouping ID label can help to identify a marked chunk point as a special ad insertion case which can be indicative of an irregular chunk duration. This is an informational label only and does not replace the [SCTE 35] tags in the transport stream. This label may be triggered by SCTE 35 tags in the content source stream or by system input during the transcoding process.

126: Start indicator of the immediately preceding non-indicator EBP_grouping_ID in the EBP structure order.

127: End indicator of the immediately preceding non-indicator EBP_grouping_ID in the EBP structure order.

6.7 On the Use of Partitions

As defined in Section 3, partitions are sets of continuous chunks (in case partition is used for carrying boundary information) or sets of marked packets (if used, e.g., to carry acquisition times).

Boundary partitions allow multiple chunking timelines to be associated with a content stream to meet the chunking requirements of different ABR delivery formats. Thus a typical application of boundary partitions would be to associate one or more ABR formats with a unique partition_id.

Three reserved values of boundary partition_id are defined:

- partition_id=1 corresponds to Segments
- partition_id=2 corresponds to Fragments
- partition_id=0 is used to indicate non-boundary EBPs, e.g., EBPs carrying only acquisition times.

It is possible to use more partitions than defined above; extended partitions 3-9 can be used for any purpose.

All non-boundary EBP structures always implicitly belong to partition 0; however, an implementer MAY choose not to use partition 0 and use instead an extended partition for non-boundary use.

The purpose and frequency of a partition are specified in the optional EBP descriptor. For boundary partitions, buffering requirements can be derived from the optional max_bitrate_descriptor.

For a given PID, a partition can be explicit or implicit. An explicit partition is one which is used in the EBP structures carried on this PID, while an implicit partition is a reference to an explicit partition carried on a different PID (and, possibly, in a different multiplex).

7 PMT DESCRIPTORS RELATED TO EBP DATA

7.1 Descriptors

The PMT structure SHOULD include one or more descriptors to signal the presence of EBP data in the stream to assist a downstream device in quickly processing a stream that may have EBP information.

7.1.1 SCTE_adaptation_field_data_descriptor

When EBP is present, the PMT SHALL carry the SCTE_adaptation_field_data_descriptor() as defined in [SCTE 128] to signal the presence of private data in adaptation private field of a TS packet. This SHALL be included in the PMT for the elementary stream(s) carrying private data and follows ES_info_length field. In the absence of such private data in adaptation private field, this descriptor SHALL NOT be used. Even if an elementary stream may carry more than one type of private data in adaptation private field, only one descriptor is required. This descriptor SHALL precede any descriptor that may provide information about the private data.

7.1.2 maximum_bitrate_descriptor

The PMT structure SHOULD include a maximum_bitrate_descriptor() at the program level and video and audio elementary stream level. This descriptor is specified in [MPEG2-TS].

7.1.3 EBP_descriptor

The PMT structure SHOULD also include one EBP_descriptor () for each of the elementary streams. Descriptor length MAY be zero if and only if EBP structure is present on PID. The presence of EBP_descriptor() signals the fragmentor that EBP data MAY be carried as adaptation field private data of the MPEG-2 TS packets. The EBP_descriptor is provided in Table 2.

Table 2 - Syntax of EBP_descriptor()

Syntax	No. of bits	Format
`EBP_descriptor(){`		
`descriptor_tag`	8	uimsbf
`descriptor_length`	8	uimsbf
`if (descriptor_length > 0)`		
`{`		
`num_partitions`	5	uimsbf
`timescale_flag`	1	bslbf
`reserved`	2	bslbf
`if (timescale_flag == 1 )`		
`{`		
`ticks_per_second`	21	bslbf
`EBP_distance_width_minus_1`	3	bslbf
`}`		

Syntax	No. of bits	Format
for (i = 0; i < num_partitions; i++)		
{		
EBP_data_explicit_flag	1	bslbf
representation_id_flag	1	bslbf
partition_id	5	uimsbf
if (EBP_data_explicit_flag == 0)		
{		
reserved	1	bslbf
EBP_PID	13	bslbf
reserved	3	uimsbf
}		
else		
{		
boundary_flag	1	uimsbf
EBP_distance	N	bslbf
if (boundary_flag == 1)		
{		
SAP_type_max	3	uimsbf
reserved	4	bslbf
}		
else		
{		
reserved	7	uimsbf
}		
acquisition_time_flag	1	bslbf
}		
if (representation_id_flag == 1)		
{		
representation_id	64	uimsbf
}		
}		
}		
}		

7.2 Semantics

***descriptor_tag*:** This 8-bit unsigned integer SHALL have the value 0xE9, identifying this descriptor as EBP_descriptor().

***descriptor_length*:** This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

timescale_flag: Bit flag to indicate the presence of timescale information. If "0", ticks_per_second is assumed to be "1" and EBP_distance_width_minus_1 is assumed to be "0".

ticks_per_second: precision, in ticks per second, of the EBP_distance field, e.g., 0.1 second precision is 10 ticks/sec, 0.01 second precision with 100 ticks/sec, etc.

***EBP_distance_width_minus_1*:** length, in bytes (minus one), of the EBP_distance field. This means that the value 0 indicates 8-bit EBP_distance field, 1-16 bit, etc. The expectation is that in the vast majority of cases the value will be 0.

num_partitions: number of partitions in the loop below.

***representation_id_flag*:** Bit flag to indicate that a representation ID is used.

EBP_data_explicit_flag: if set to 0, this ES is a dependent stream, and boundary data for it is provided on a *reference partition* on a different PID, specified by EBP_PID; otherwise the current PID carries EBP structures.

***partition_id*:** id of the partition described. If EBP_data_explicit_flag is set to 0, this is the partition_id of a reference partition on the EBP_PID.

EBP_PID: Associated PID carrying EBP structures that are used to partition this particular PID elementary stream.

boundary_flag: Bit flag that if set to '0', the partition is used for marking a frame, otherwise it is used for indicating a chunk boundary point.

representation_id: 64-bit opaque label identifying the representation. If EBP_data_explicit_flag is set to 0, this is the representation_id of a reference partition on the EBP_PID.

EBP_distance: expected distance in time slice ticks between two EBP structures with the same partition. This is an N-byte integer, where N = EBP_distance_width_minus_1 + 1.

SAP_type_max: maximum possible value of EBP_SAP_type field in the EBP structure. If EBP_sap_flag is 0 in all EBP structures associated with this partition, the value of SAP_type_max is 2.

acquisition_time_flag: Bit flag that if set to 1, the value of EBP_time_flag in any EBP structure associated with this partition SHALL be '1'.

Appendix I Informative Examples

This section describes several informative examples of how the EBP structure and the PMT descriptor can be used.

I.1 EBP Use Case Examples

I.1.1 Indicate a Base Fragment Boundary Point

The EBP structure can be used to indicate fragment boundaries used in AVC video (e.g., 2-second fragments). EBP is associated with an IDR AU that starts the fragment. The fragment boundary point should be on and other flags and bytes should be set according to the table below.

Table 3 - EBP Structure for Fragment Boundary Point

BP Frag Flag	BP Seg Flag	SAP Flag	Grouping Flag	Time Flag	Conceal Flag	Reserve	Extension Flag
1	X	0	X	X	X	0	X

1= Flag on, 0= Flag off, X= Any Value, NA= Not Applicable, NTP= 64 bit binary NTP count relative counter

I.1.2 Indicate a Segment Boundary Point

A single EBP data structure can indicate the video IDR AU is the start of both a fragment and/or segment as used in AVC video. This can be used to indicate larger segment boundaries for use in HLS fragments. The same AU can indicate both a segment boundary and fragment boundary (which would be the most common case) by turning on both the fragment and segment flags as indicated in the table below. One can, however, indicate a segment boundary that is not a fragment boundary if that need occurs, and the stream should be conditioned at the start of segment.

Table 4 - EBP Structure for Segment Boundary Point

BP Frag Flag	BP Seg Flag	SAP Flag	Grouping Flag	Time Flag	Conceal Flag	Reserve	Extension Flag
X	1	0	X	X	X	0	X

1= Flag on, 0= Flag off, X= Any Value, NA= Not Applicable, NTP= 64 bit binary NTP count relative counter

I.1.3 Signaling timing and partitions for HSS, HLS, and HDS

The EBP signaling can indicate to the encapsulator how to locate boundary points for HSS, HDS, HLS. For instance, explicit fragment boundary points for both video and audio can be used for HSS fragmentation. Retrieving consecutive fragments in HSS mode can be done without overlapping byte ranges. For HSS, the use of boundary points on both the video ES and audio ES is ideal. For HLS, the use of segments and implicit audio EBP signaling will provide enough information for the encapsulator to segment for HLS. Retrieval of consecutive segments for HLS requires overlap in byte ranges. For HDS, implicit audio should be used as well.

Table 5 - EBP Structure for HSS, HLS, HDS

BP Frag Flag	BP Seg Flag	SAP Flag	Grouping Flag	Time Flag	Conceal Flag	Reserve	Extension Flag
1 or 0	1 or 0	0	X	X	X	0	X

1= Flag on, 0= Flag off, X= Any Value, NA= Not Applicable, NTP= 64 bit binary NTP count relative counter

I.1.4 Indicate Time on an AU

NTP time can be indicated through the EBP structure. This time can be applied to a PES header, so it could be put on any video AU or group of audio AUs. The time could be placed at an encoder or transcoder. A transcoder may generate its own NTP time from an NTP signal or derive this from the NTP time in the EBP structure of the input content stream, if it exists.

Table 6 - EBP Structure for time only

BP Frag Flag	BP Seg Flag	SAP Flag	Grouping Flag	Time Flag	Conceal Flag	Reserve	Extension Flag
0	0	X	X	1	X	0	X

1= Flag on, 0= Flag off, X= Any Value, NA= Not Applicable, NTP= 64 bit binary NTP count relative counter

I.2 EBP descriptor use case examples

I.2.1 Signaling timing and partitions for HSS and HLS

Table 7 shows a use case that shows a video PID with 10-second segments for HLS and 2-second fragments for HSS. Each EBP structure contains an acquisition time.

Table 7 - Example of EBP_descriptor() for HLS and HSS video PID

EBP_descriptor(){ ...		
num_partitions	2	
timescale_flag	0	
for (i = 0; i<num_partitions; i++)	i = 0	i=1
EBP_data_explicit_flag	1	1
representation_id_flag	1	0
partition_id	1	2
boundary_flag	1	1
EBP_distance	10	2
SAP_type_max	1	1
acquisition_time_flag	1	1
representation_id	0x0000E1E1E1E11389	n/a

I.2.2 Explicit and implicit partitions

Table 8 shows a use case with partition 0 for acquisition times and 2-second fragments for HSS, as in the previous example. The example in Table 8 shows an audio PID with explicit 2-second fragments and implicit timing based on video PID 481.

Table 8 - Example of EBP_descriptor() for HLS and HSS audio PID

EBP_descriptor(){		
...		
num_partitions	2	
timescale_flag	0	
for (i = 0; i<num_partitions; i++)	i = 0	i = 1
EBP_data_explicit_flag	0	1
representation_id_flag	1	0
partition_id	1	2
EBP_PID	481	n/a
boundary_flag	n/a	1
EBP_distance	n/a	2
SAP_type_max	n/a	1
acquisition_time_flag	n/a	0
representation_id	0x0000E1E1E1E11389	n/a

Table 9 and Table 10 show a use case (HDS Method 1) for HDS video fragments using the same partition as HSS video fragments, and HDS audio fragments being implicitly derived from HSS video fragments (PID 481).

Table 9 - Example of EBP_descriptor() for HLS, HSS, HDS (implicit to partition=2) video PID

EBP_descriptor(){		
...		
num_partitions	2	
timescale_flag	0	
for (i = 0; i<num_partitions; i++)	i = 0	i=1
EBP_data_explicit_flag	1	1
representation_id_flag	1	1
partition_id	1	2
boundary_flag	1	1
EBP_distance	10	2
SAP_type_max	1	1
acquisition_time_flag	1	1
representation_id	0x0000E1E1E1E11389	0x0000E1E1E1E1138A

Table 10 - Example of EBP_descriptor() for HLS, HSS, HDS (implicit partition=2) audio PID

EBP_descriptor(){			
...			
num_partitions	3		
timescale_flag	0		
for (i = 0; i<num_partitions; i++)	i = 0	i = 1	i = 2
EBP_data_explicit_flag	0	1	0
representation_id_flag	1	0	1
partition_id	1	2	2
EBP_PID	481	n/a	481
boundary_flag	n/a	1	n/a
EBP_distance	n/a	2	n/a
SAP_type_max	n/a	1	n/a
acquisition_time_flag	n/a	0	n/a
representation_id	0x0000E1E1E1E11389	n/a	0x0000E1E1E1E1138A

Table 11 and Table 12 show a use case (HDS Method 2) where HDS video fragments use their own partition and HDS audio fragments are implicitly derived from this partition.

Table 11 - Example of EBP_descriptor() for HLS, HSS, and HDS (explicit partition=3) video

EBP_descriptor(){			
...			
num_partitions	3		
timescale_flag	0		
for (i = 0; i<num_partitions; i++)	i = 0	i=1	i=2
EBP_data_explicit_flag	1	1	1
representation_id_flag	1	0	0
partition_id	1	2	3
boundary_flag	1	1	1
EBP_distance	10	2	2
SAP_type_max	1	1	1
acquisition_time_flag	1	1	1
representation_id	0x0000E1E1E1E11389	n/a	n/a

Table 12 - Example of EBP_descriptor() for HLS, HSS, and HDS (implicit partition=3) audio PID

EBP_descriptor(){			
...			
num_partitions	3		
timescale_flag	0		
for (i = 0; i<num_partitions; i++)	i = 0	i = 1	i = 2
EBP_data_explicit_flag	0	1	0
representation_id_flag	1	0	1
partition_id	1	2	3
EBP_PID	481	n/a	481
boundary_flag	n/a	1	n/a
EBP_distance	n/a	2	n/a
SAP_type_max	n/a	1	n/a
acquisition_time_flag	n/a	0	n/a
representation_id	0x0000E1E1E1E11389	n/a	0x0000E1E1E1E11389

Appendix II Derivation of Acquisition Time (Informative)

Since the processing required to generate Adaptive Transport Streams in different formats may introduce delays that vary by format type and across implementations, implementations should ensure that the acquisition time represents a consistent point in the signal processing chain.

Consider the signal chain in an ISO/IEC 13181-1 compliant transcoder model given in Figure 5 below.

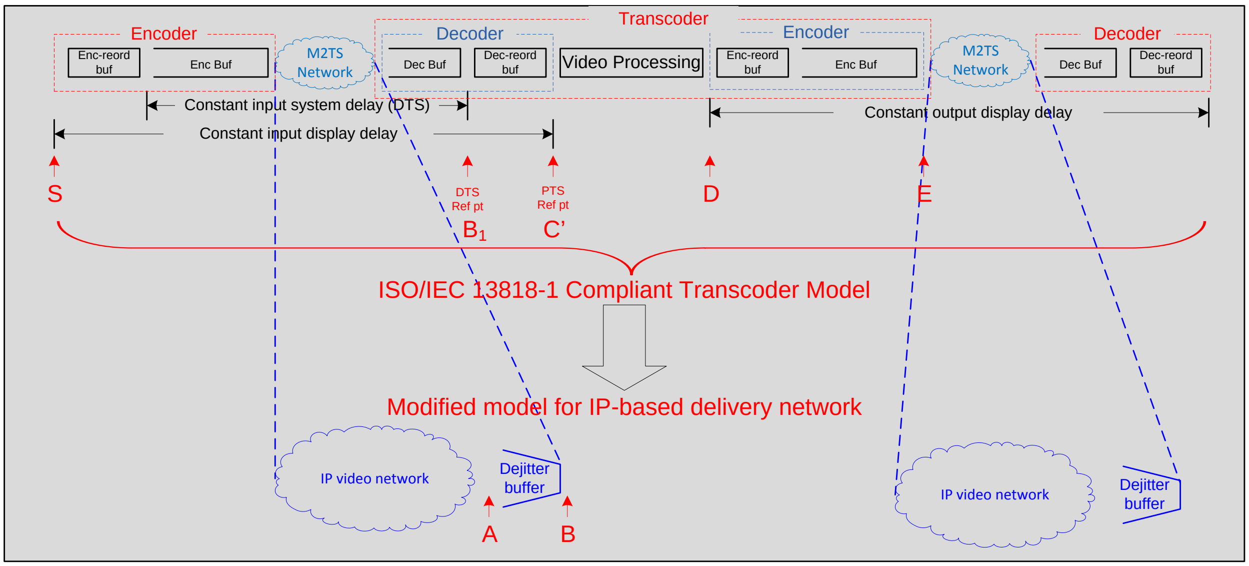


Figure 5 - ISO/IEC 13818-1 compliant transcoder model

The upper half of Figure 5 illustrates a transport network compliant to ISO/IEC 13818-1. Here, the encoder is modeled with an encoder reordering buffer and an encoder data buffer, and the decoder is modeled with a decoder buffer and a decoder reordering buffer. Both the encoding operation and decoding operation are assumed to be instantaneous. Regardless of implementation details, a transcoder can be modeled with a decoder, an encoder, and optionally a baseband video processing unit. As shown in Figure 5, a transcoder receives compressed video from an upstream MPEG-compliant video encoder and is expected to generate an MPEG-compliant stream as its output.

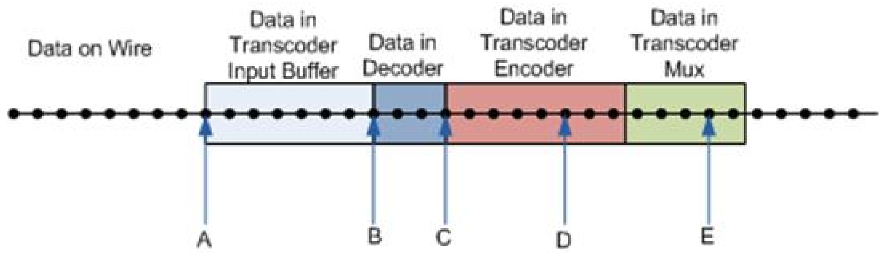
To determine a common sampling point for EBP acquisition time, several points are marked in Figure 5. Point S is the time when a source video frame is about to enter the reordering buffer at the upstream encoder. Point B_1 is when the decoder within the transcoder decompresses the access unit and moves it into the decoder reordering buffer. This is the time referred to as DTS. Point C is when the access unit is reordered and displayed. This is referred to as PTS. Point D is when the decoded picture is about to enter the encoder reordering buffer at transcoder. Point E is when the re-encoded access unit is about to be multiplexed with audio and other data.

The lower half of Figure 5 extends the traditional MPEG2-compliant network into an IP-based delivery network, by adding a dejitter buffer at the decoder input. It is expected that TS traffic at the output of a dejitter buffer complies with ISO/IEC 13818-1. For the purpose of discussion, the lower half of Figure 5 introduces two more reference points, A and B. Point A is when the IP packet containing the first byte of an access unit is received at the transcoder side, and point B is when the same TS packet is out of the dejitter buffer.

Among all the points given in Figure 5, point S is obviously the best reference point to sample NTP. However, it is quite impractical to expect sampled NTP at point S due to unknown encoding-decoding delay chosen by the upstream encoder. If jitter induced by the IP network is negligible, point C would be another good reference point, since delay between point S and point C is fixed. However, this is not the case for IP-based transport network. Delay between point S and point C can vary by the amount of network jitter.

Figure 6 illustrates the possible reference points at a nominal transcoder. While it is expected that a transcoder will sample NTP based on its display time referred to as point C, a different decoder may choose a different display time.

To avoid this potential discrepancy, it is expected that the NTP value for each boundary point is sampled corresponding to point C signaled by PTS in TS stream. All transcoders with the same input stream are expected to implement this theoretical timing model defined by ISO/IEC 13818-1 [MPEG2-TS].



A= enters transcoder dejitter buffer, B= AU enters decoder buffer, C= exits decoder output buffer, D= video processing and encoding, E= gets multiplexed

Figure 6 - Candidate Reference Time Sample Points in Signal Chain in Nominal Transcoder

An implementation of a transcoder should sample NTP regularly. It may sample NTP at a different point mentioned in Figure 6, but it should compensate accordingly to reflect time at point C. If the implementation does not sample NTP for each access unit, it should derive the appropriate value for each access unit.

As an example, one can sample NTP at point B and use or derive PCR value and PTS value of the corresponding TS packet that contains the first byte of access unit, according to operation defined in [MPEG2-TS]. With those data, NTP corresponding to point C can be derived with (sampled NTP + (PTS – PCR)). PTS is equal to (DTS + reordering delay). (DTS – PCR) is known as vbv_delay defined in the ISO/IEC 13818-2 [MPEG2-Video] or cpb_removal_delay in ISO/IEC 14496-10 [AVC]. Another implementation may choose to sample at point A. In this case, it is expected that NTP values will be filtered so that it reflects NTP values at point B. After this is done, it should compensate in a similar way to point B.

Similarly, an implementation that samples NTP at point D of E should compensate the values back to point C before inserting into EBP field.

It is possible that the transcoder system will introduce extra delay inherent in its deployment architecture. For the purpose of EBP specification, this delay is counted as part of the +/- 300 ms jitter budget. However, a specific implementation may choose to compensate this extra delay to its sampled NTP value to improve accuracy of NTP value.