SMPTE ST 430-14:2015

SMPTE STANDARD

D-Cinema Operations — Digital Sync Signal and Aux Data Transfer Protocol

Table of Contents

Approved
August 1, 2015

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual.

SMPTE ST 430-14 was prepared by Technology Committee 21DC.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Engineering Document. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

Introduction

This section is entirely informative and does not form an integral part of this Engineering Document.

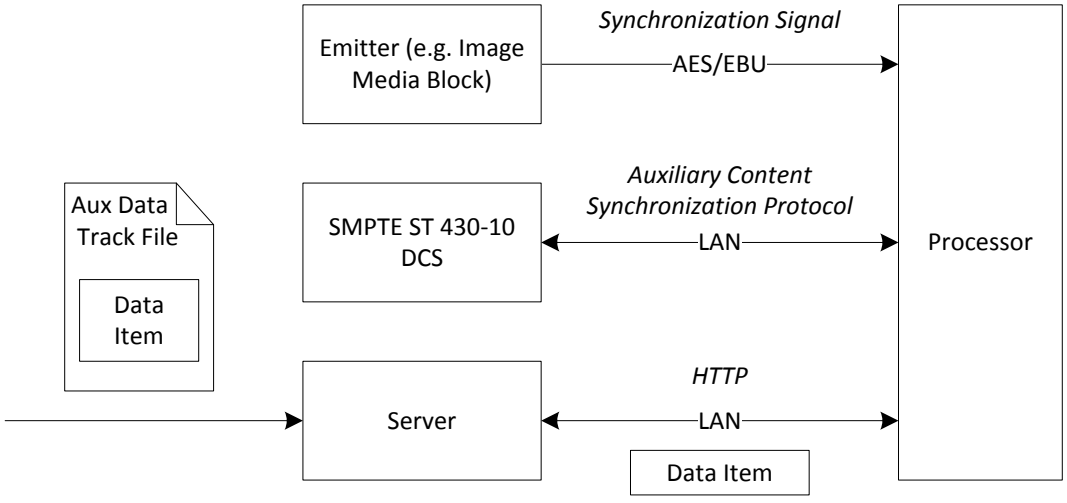

Figure 1 – Overview of the Aux Data Synchronization Signal and Transfer Protocol Architecture

As illustrated in Figure 1, this document specifies a Transfer Protocol that allows Data Items, as specified SMPTE ST 429-14, to be transmitted to a Processor over the Local Area Network (LAN); and a generic Synchronization Signal that carries sufficient information to allow these Data Items to be reproduced by the Processor synchronously with Show playback.

The source of the Data Items transmitted using the Transfer Protocol is called the Server, and the source of the Synchronization Signal is called the Emitter. The Processor discovers the Server by connecting to a Digital Cinema Server (DCS) entity using the Auxiliary Content Synchronization (ACS) protocol — both DCS and ACS are specified in SMPTE ST 430-10.

This specification allows the Server, Emitter and the DCS to be implemented on the same or different devices. A Processor can take many forms, including that of an Immersive Sound Processor or a Motion System Controller.

The Synchronization Signal (see Section 5) is an audio sample-accurate binary signal generated by an Emitter, typically the Image Media Block, at time of playback and compatible with AES/EBU interfaces. It includes the current playback position within a timeline (see Figure 4) that consists of a sequence of contiguous Edit Units corresponding to the currently playing sequence of Compositions; i.e., the Show.

The Transfer Protocol (see Section 6) uses the Hypertext Transfer Protocol (HTTP). It allows the Processor to request from a Server the Data Items of a specified kind (as determined by the Data Essence Coding item of the Data Essence Descriptor) and corresponding to a position with the Timeline (determined from the Synchronization Signal).

Figure 2 illustrates the interaction between Emitter, Server, DCS and Processor in a typical application. When a Show is selected and a valid Synchronization signal is detected (with the Status flag set to Pause or Stopped), the Processor connects to the DCS and, using the Set RPL Location message, determines the Server network address (see Section 6.4). The Processor then buffers Data Items by making a number of requests (see Figure 3) to the Server. While it is buffering, the Processor responds with ACS Busy to Get Status Requests issued by the DCS, indicating that it is not ready for playback.

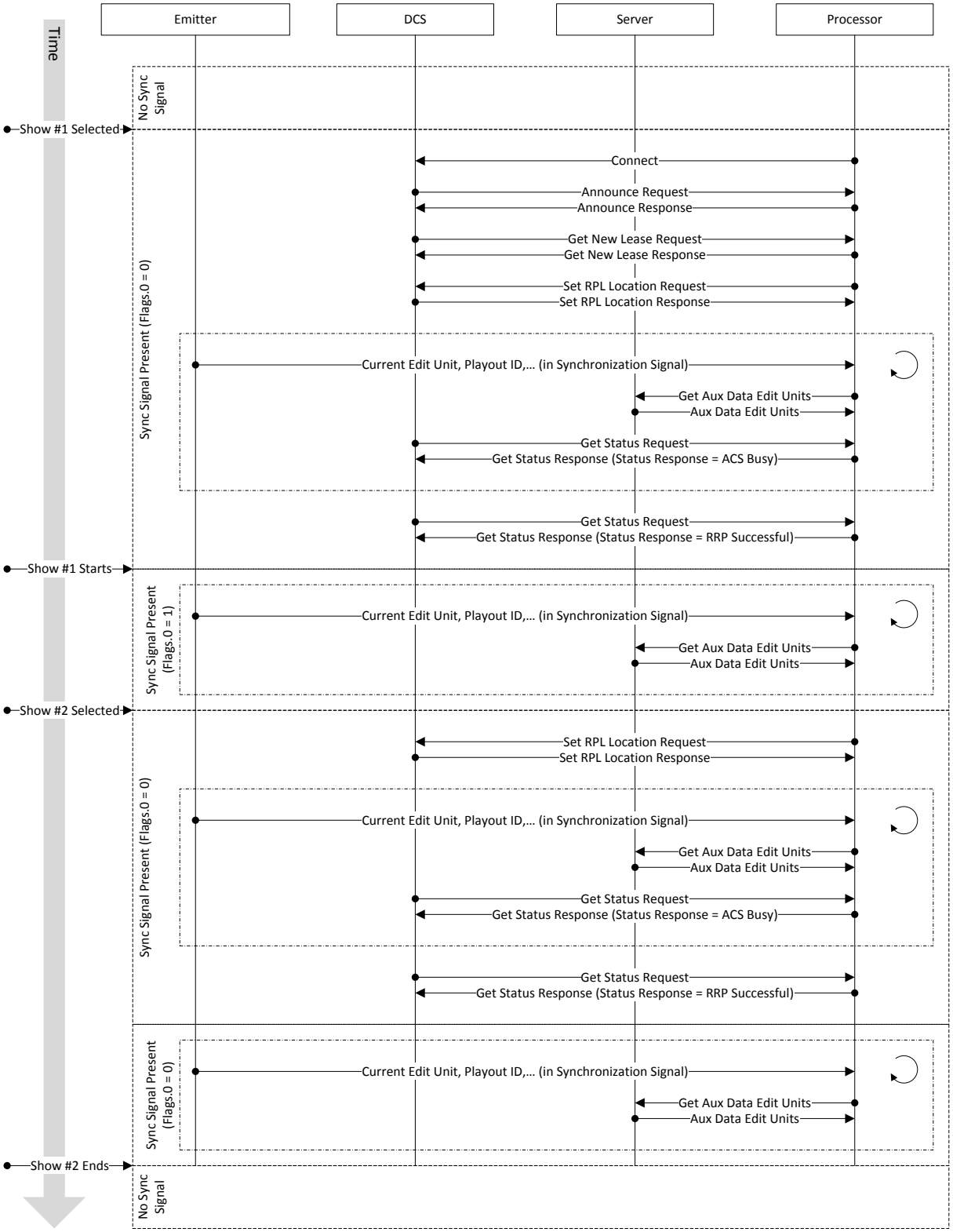

Figure 2 – Transfer Protocol Sequence Diagram

The start of Show playback is signaled by the Status flag of the Synchronization Signal (see Section 5.3.4) transitioning to Playing. While the Status flag remains equal to Playing, the Processor reproduces the Data Items it has obtained from the Server based on the `Timeline Edit Unit Index` carried by the Synchronization Signal.

When playback of the first Show stops and a second Show is selected, the Status flag of the Synchronization Signal transitions to Stopped, and the Processor stops playback and starts buffering. The Processor starts playback along the second Show timeline when the Status flag transitions to Playing again.

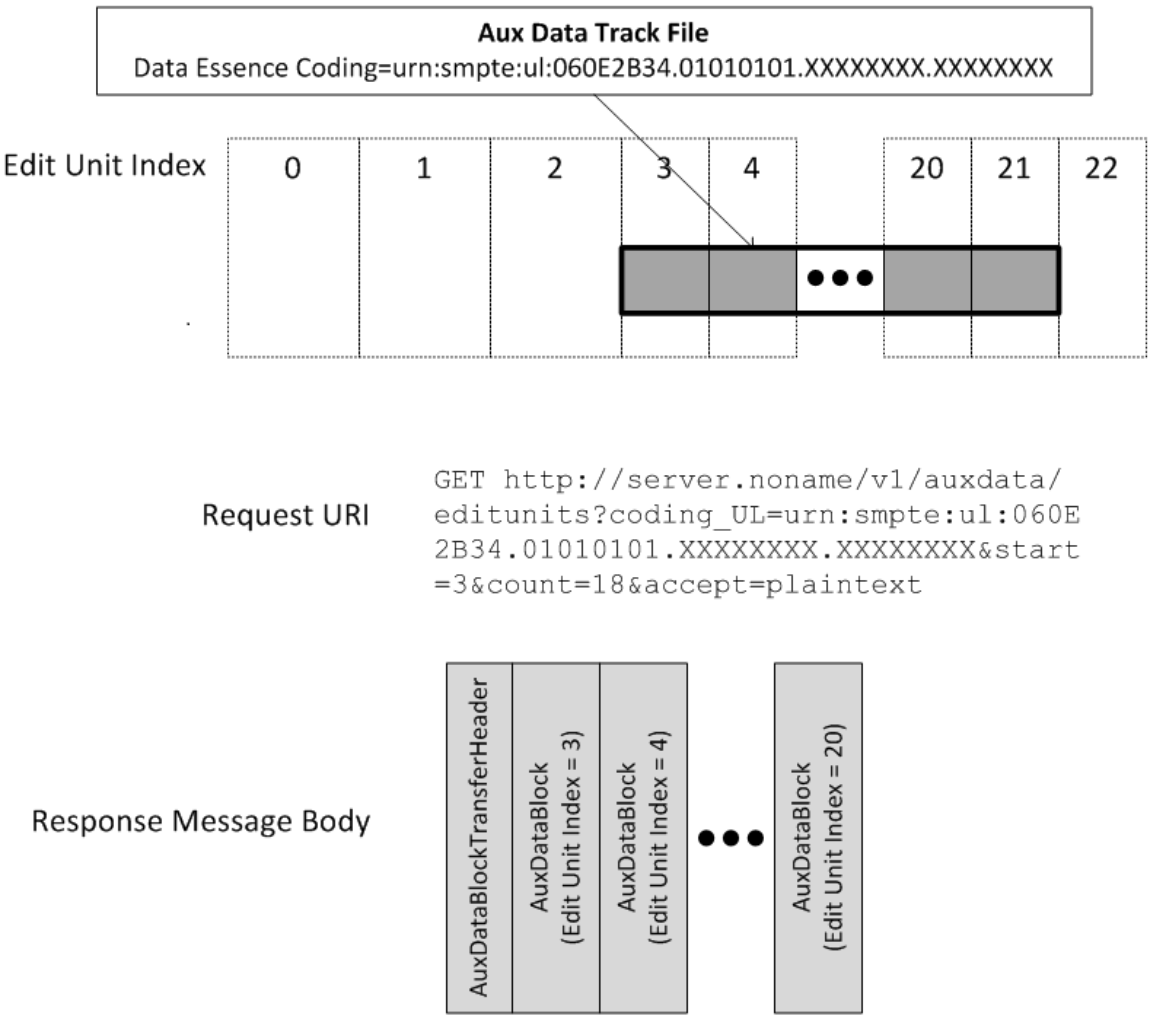

Figure 3 – Sample Transfer Protocol Request-Response

The minimal expected behavior of a Processor is specified in Section 5.7.

The Synchronization Signal and Transfer Protocol are designed to work together. The Synchronization Signal can nevertheless be used without the Transfer Protocol; e.g., a Processor can use the Synchronization Signal to determine playback location within the current Show but use means other than the Transfer Protocol to obtain playback essence. The Transfer Protocol however requires the presence of the Synchronization Signal.

1 Scope

This document specifies (i) a Transfer Protocol that allows Data Items from an Aux Data Track File to be transmitted to a Processor for reproduction, and (ii) a Synchronization Signal that allows a Processor to reproduce Data Items synchronously with the D-Cinema presentation. The Transfer Protocol uses the Hypertext Transfer Protocol (HTTP), while the Synchronization Signal is a binary signal that is accurate to the nearest audio sample of the presentation, intended to be generated by the Image Media Block at time of playback, and for use with AES3 interfaces.

While the Transfer Protocol requires the presence of the Synchronization Signal, the Synchronization Signal can be used in absence of the Transfer Protocol.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE ST 298:2009, Universal Labels for Unique Identification of Digital Data

SMPTE ST 336:2007, Data Encoding Protocol Using Key-Length-Value

[1)] SMPTE ST 377:2004, Television — Material Exchange Format (MXF) — File Format Specification

[1)] SMPTE ST 379:2004, Material Exchange Format (MXF) — MXF Generic Container

SMPTE ST 429-2:2013, D-Cinema Packaging — DCP Operational Constraints

SMPTE ST 429-6:2006, D-Cinema Packaging — MXF Track File Essence Encryption

SMPTE ST 429-14:2014, D-Cinema Packaging — Aux Data Track File

SMPTE ST 430-10:2010, D-Cinema Operations — Auxiliary Content Synchronization Protocol

AES3-2009, AES Standard for Digital Audio Engineering — Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data

IETF RFC 2616, Hypertext Transfer Protocol — HTTP/1.1

IETF RFC 2818, HTTP over TLS

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax

IETF RFC 6597, RTP Payload Format for Society of Motion Picture and Television Engineers (SMPTE) ST 336 Encoded Data

IETF RFC 2234, Augmented BNF for Syntax Specifications: ABNF

IETF RFC 4122, A Universally Unique Identifier (UUID) URN Namespace

IETF RFC 2046, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types

[1)] Note: The reference to SMPTE ST 377:2004 (http://dx.doi.org/10.5594/S9781614827689) and SMPTE ST 379:2004 (http://dx.doi.org/10.5594/S9781614828068) are intentional. SMPTE ST 377-1:2011 and SMPTE ST 379-1:2009 or future versions are not appropriate for use with this document.

4 Terms and Definition

4.1 Timeline

As illustrated in Figure 4, the Timeline is a sequence of N contiguous Timeline Edit Units, resulting from the successive and uninterrupted playback of one or more Compositions in their entirety. Timeline Edit Units can have different duration to accommodate change of edit rates across Composition boundaries.

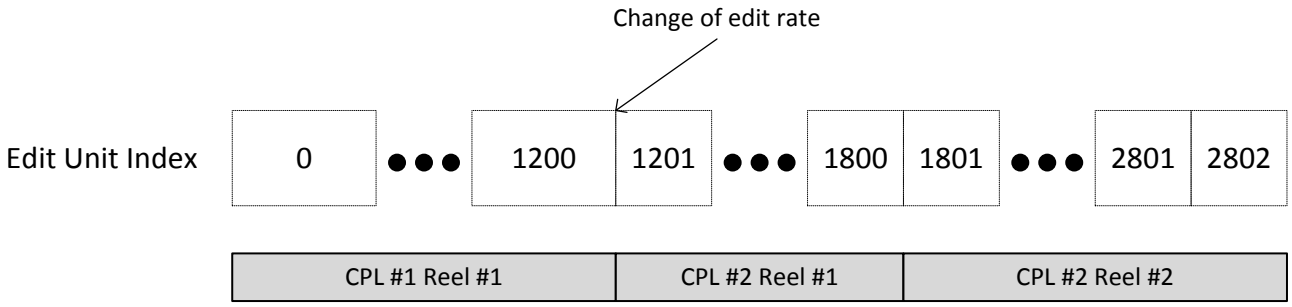

Figure 4 – Timeline

The Timeline Edit Units are indexed from 0 for the first Timeline Edit Unit to N-1 for the last Timeline Edit Unit.

4.2 Primary Picture

The primary picture is the picture projected on the main screen of the auditorium, commonly the output of the Primary Picture Track File.

4.3 Primary Picture Track File

The Primary Picture Track File is the Track File referenced by the Composition Playlist `MainPicture` or `MainStereoscopicPicture` element associated with the Timeline Edit Unit.

4.4 Primary Sound Track File

The Primary Sound Track File is the Track File referenced by the Composition Playlist `MainSound` element associated with the Timeline Edit Unit.

4.5 Server

Source of the Data Items transmitted using the Transfer Protocol.

4.6 Emitter

Source of the Synchronization Signal.

4.7 DCS

Digital Cinema Server as defined in SMPTE ST 430-10.

4.8 ACS

Auxiliary Content Synchronization as defined in SMPTE ST 430-10

5 Synchronization Signal

5.1 General

Bits are numbered starting at 0 for the least significant bit.

Note: Bit structures are documented using the conventions used in AES3, with the least-significant bit positioned to the left.

Hexadecimal constants are prefixed with "0x" and written most-significant bit and byte first.

EXAMPLE: The unsigned integer constant 0xAAF0 corresponds to the decimal integer 43,760.

5.2 Structure

As illustrated in Figure 5, the Synchronization Signal shall consist of a sequence of 24-bit Samples, grouped into Packets separated by zero or more Fill Samples.

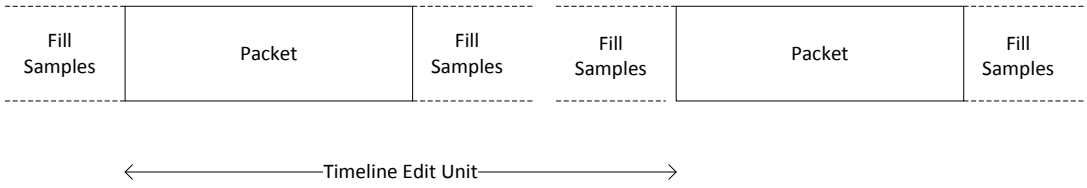

Figure 5 – Synchronization Signal

Each Fill Sample shall be equal to 0x000000.

Each Packet shall consist of the payload of Section 5.3 mapped to a sequence of Sample-Pairs, each consisting of a Lead Sample followed by a Tail Sample.

As illustrated in Table 1, the lower 16 bits of the n^{th} Lead Sample of a Packet shall be equal to the n^{th} Word of the Payload specified in Section 5.3.

Table 1 – Lead Sample Structure

LSB																							MSB
Lead Sample																							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
LSB ..... Payload Word ..... MSB																S	0	0	0	0	0	0	0

Bit 16 (labeled S in Table 1) of the Lead Sample shall be:

- 1 for the first Lead Sample of a Packet; and
- 0 otherwise.

The most significant 7 bits of each Lead Sample shall be 0.

The Tail Sample shall consist of the two's complement of the lead Sample, as illustrated in Table 2.

Table 2 – Tail Sample Structure

LSB Tail Sample bits MSB																							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
LSB ..... 2's complement of Lead Sample ..... MSB																							

Note 1: Restricting the values of Samples to values between 131,071 (0x01FFFF) to -131,071 (0xFE0001) minimizes the possibility of speaker damage should the sync signal accidentally be routed to a speaker. A peak of 0dBFS audio signal represented by the 24 bit number 0x7FFFFF (not 0xFFFFFF since the number is a two's complement signed number). 0x7FFFFF represents 8,388,607 decimal. 131,071/8,388,607 = -36dB. This is 16dB below the standard recording level of -20dBFS. Following each data word with its exact opposite (two's complement negative) results in the sum of all words in the packet being zero. This net DC value of zero protects speakers from a DC offset. Additionally, the transmission of each Payload Word twice (once negated) results in a high degree of error detection.

Note 2: Unless designed specifically to maintain Synchronization Signal integrity, any processing applied to the Synchronization Signal, e.g. sample rate conversion and/or gain, between Emitter and Processor can result in a corrupted signal.

5.3 Payload

5.3.1 General

5.3.1.1 Structure

The Payload shall consist of the sequence of 16-bit Words specified in Table 3.

Table 3 – Payload

Field		LSB Word MSB																Word #
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Marker		0	0	0	0	1	1	1	1	0	1	0	1	0	1	0	1	0
Length		bit 0 ..... bit 15																1
Flags		bit 0 ..... bit 15																2
Timeline Edit Unit Index		bit 16 ..... bit 31																3
		bit 0 ..... bit 15																4
Playout ID		bit 16 ..... bit 31																5
		bit 0 ..... bit 15																6
Edit Unit Duration		bit 0 ..... bit 15																7
Sample Duration	Numerator	bit 16 ..... bit 31																8
		bit 0 ..... bit 15																9
	Denominator	bit 16 ..... bit 31																10
		bit 0 ..... bit 15																11
Primary Picture Output Offset		bit 16 ..... bit 31																12
		bit 0 ..... bit 15																13
Primary Picture		bit 16 ..... bit 31																14

Screen Offset	bit 0 ..... bit 15	15
Primary Picture Track File Edit Unit	bit 16 ..... bit 31	16
	bit 0 ..... bit 15	17
Primary Picture Track File UUID	bit 112 ..... bit 127	18
	bit 96 ..... bit 111	19
	bit 80 ..... bit 95	20
	bit 64 ..... bit 79	21
	bit 48 ..... bit 63	22
	bit 32 ..... bit 47	23
	bit 16 ..... bit 31	24
	bit 0 ..... bit 15	25
Primary Sound Track File Edit Unit	bit 16 ..... bit 31	26
	bit 0 ..... bit 15	27
Primary Sound Track File UUID	bit 112 ..... bit 127	28
	bit 96 ..... bit 111	29
	bit 80 ..... bit 95	30
	bit 64 ..... bit 79	31
	bit 48 ..... bit 63	32
	bit 32 ..... bit 47	33
	bit 16 ..... bit 31	34
	bit 0 ..... bit 15	35
Composition Playlist UUID	bit 112 ..... bit 127	36
	bit 96 ..... bit 111	37
	bit 80 ..... bit 95	38
	bit 64 ..... bit 79	39
	bit 48 ..... bit 63	40
	bit 32 ..... bit 47	41
	bit 16 ..... bit 31	42
	bit 0 ..... bit 15	43
Extension	Zero or more Words	…

5.3.1.2 Integer Fields

An *n*-bit integer shall be encoded in a field such that bit *i* of the field is the i^{th} bit of the integer.

EXAMPLE: Bit 31in Word 3 is the most significant bit (bit 31) of the integer represented in the `Timeline Edit Unit Index` field.

5.3.1.3 UUID Fields

A UUID value shall be encoded in a field as a 128-bit object as specified in Section 4.1.2 of RFC 4122.

EXAMPLE 1: Bit 127 in Word 18 corresponds to the most significant bit of octet 0 of `Primary Picture Track File UUID`.

EXAMPLE 2: Table 4 lists the Composition Playlist Id value `urn:uuid:65bfa8d3-5765-4c19-83bf-74ce29e5b47f` encoded in `Composition Playlist UUID` field.

Table 4 – Sample encoding of a UUID value `urn:uuid:65bfa8d3-5765-4c19-83bf-74ce29e5b47f` in the `Composition Playlist UUID` field

LSB Word MSB																Word #
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	1	1	1	1	1	0	1	1	0	1	0	0	1	1	0	36
1	1	0	0	1	0	1	1	0	0	0	1	0	1	0	1	37
1	0	1	0	0	1	1	0	1	1	1	0	1	0	1	0	38
1	0	0	1	1	0	0	0	0	0	1	1	0	0	1	0	39
1	1	1	1	1	1	0	1	1	1	0	0	0	0	0	1	40
0	1	1	1	0	0	1	1	0	0	1	0	1	1	1	0	41
1	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	42
1	1	1	1	1	1	1	0	0	0	1	0	1	1	0	1	43

5.3.2 Marker

The `Marker` field is a constant bit pattern used to detect the start of the Packet.

Note: The value of the Marker field expressed as an unsigned integer is 0xAAF0.

5.3.3 Length

The `Length` field shall be the number of Words in the Payload, including the `Extension` field, but not including the `Marker` and the `Length` fields.

`Length` shall be an unsigned integer.

5.3.4 Flags

The `Flags` field structure shall be as specified in Table 5.

Table 5 – Flags Field Structure

LSB Flags Field MSB															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LSB . Status..MSB				....................Reserved											

The Emitter shall set the value of bits 4-15 of the `Flags` field to 0.

The Processor shall ignore the value of bits 4-15 of the `Flags` field.

The values of the `Status` bits shall be as specified in Table 6.

Table 6 – Status Values

Value	Meaning
0	Stopped
1	Paused
2	Playing
3-15	*Reserved*

The Emitter shall not set the value of `Status` to a reserved value.

5.3.5 Timeline Edit Unit Index

The `Timeline Edit Unit Index` shall be the index of the Timeline Edit Unit associated with the Packet.

`Timeline Edit Unit Index` shall be an unsigned integer.

When status is Stopped or Pause, `Timeline Edit Unit Index` should correspond to the Timeline Edit Unit that will be reproduced when playback resumes.

5.3.6 Playout ID

The `Playout ID` shall be the value of the Playout ID reported by the Auxiliary Content Synchronization Protocol, as specified in SMPTE ST 430-10, for the Timeline Edit Unit associated with the Packet.

`Playout ID` shall be an unsigned integer.

5.3.7 Edit Unit Duration

The `Edit Unit Duration` field shall be equal to the number of Samples in the Timeline Edit Unit associated with the Packet.

`Edit Unit Duration` shall be an unsigned integer.

Note: If the Synchronization Signal is mapped to an AES3 interface sampled at 48 kHz, per Section 5.6.1, and Edit Units have a rate of 24 Hz, the Edit Unit Duration is 2000.

5.3.8 Sample Duration

The ratio of the `Sample Duration Numerator` and the `Sample Duration Denominator` fields shall be equal to the duration in seconds of one Sample of the Synchronization Signal.

The `Sample Duration Numerator` and the `Sample Duration Denominator` fields shall each be unsigned integers.

5.3.9 Primary Picture Output Offset

The `Primary Picture Output Offset` field shall be the temporal offset between:

a) the leading edge at the output of the media block of the first Sample of the Packet; and

b) the leading edge at the output of the media block of the primary picture edit unit associated with the Timeline Edit Unit with index `Timeline Edit Unit Index`.

`Primary Picture Output Offset` is positive if a) occurs earlier than b).

`Primary Picture Output Offset` shall be expressed as an integer multiple of the duration of a Sample and shall be a two's complement signed integer.

`Primary Picture Output Offset` when represented in milliseconds shall be in range [-500, 500] ms.

Annex A provides guidelines for setting `Primary Picture Output Offset`.

5.3.10 Primary Picture Screen Offset

The `Primary Picture Screen Offset` shall be the temporal offset between:

- the leading edge at the output of the media block of the first Sample of the Packet; and
- the leading edge at the auditorium screen of the primary picture edit unit associated with the Timeline Edit Unit with index `Timeline Edit Unit Index`.

`Primary Picture Screen Offset` shall be expressed as an integer multiple of the duration of a Sample and shall be an unsigned integer.

`Primary Picture Screen Offset` when represented in milliseconds shall be in range [0, 500] ms.

Annex A provides guidelines for setting `Primary Picture Screen Offset`.

5.3.11 Primary Picture Track File Edit Unit

The `Primary Picture Track File Edit Unit` shall be the index of the Edit Unit of the Primary Picture Track File associated with the Timeline Edit Unit, with the first Edit Unit of the Primary Picture Track File having an index of 0.

`Primary Picture Track File Edit Unit` shall be an unsigned integer.

A value of 0xFFFFFFFF shall indicate that no Primary Picture Track File is associated with the Timeline Edit Unit and that the value of `Primary Picture Track File UUID` shall be ignored.

5.3.12 Primary Picture Track File UUID

The `Primary Picture Track File UUID` shall be the UUID of the Primary Picture Track File.

5.3.13 Primary Sound Track File Edit Unit

The `Primary Sound Track File Edit Unit` shall be the index of the Edit Unit of the Primary Sound Track File associated with the Packet, with the first Edit Unit of the Primary Picture Track File having an index of 0.

`Primary Sound Track File Edit Unit` shall be an unsigned integer.

A value of 0xFFFFFFFF shall indicate that no Primary Sound Track File is associated with the Timeline Edit Unit and that the value of `Primary Sound Track File UUID` shall be ignored.

5.3.14 Primary Sound Track File UUID

The `Primary Sound Track File UUID` shall be the UUID of the Primary Sound Track File.

5.3.15 Composition Playlist UUID

The `Composition Playlist UUID` shall be the UUID of the Composition Playlist associated with the Timeline Edit Unit.

5.3.16 Extension

The `Extension` field is reserved for use by future revisions of this specification.

5.4 Packet Validity

A Packet is invalid if:

- any Sample-Pair does not conform to the provisions of Sections 5.2; or
- the Payload does not conform to the provisions of Sections 5.3.

In particular, a Packet is invalid if the `Marker` field does not conform to Section 5.3.2.

5.5 Synchronization Signal Validity

A Synchronization Signal is invalid if no valid Packet is received within a period of 3 s, and otherwise valid.

Note: Receiving at least one single valid Packet every 3 s is sufficient to preserve Synchronization Signal Validity.

5.6 Emitter Operation

5.6.1 Mapping to AES3 Interfaces

This Section shall apply when the Synchronization Signal is mapped to an AES3 interface.

The Synchronization Signal shall be mapped to a single AES3 channel, with successive Samples mapped to audio sample words of successive sub-frames.

The `Sample Duration Numerator` and the `Sample Duration Denominator` fields shall be set according to Table 7 for the AES3 nominal sampling rates listed.

Table 7 – `Sample Duration Numerator` and `Sample Duration Denominator` Values

AES3 Nominal Sampling Rate	Sample Duration Numerator	Sample Duration Denominator
48,000 Hz	1	48000
96,000 Hz	1	96000

5.6.2 Constant Output

The Emitter should output a valid Synchronization Signal whenever a Show timeline exists, i.e. a Show is loaded, and only when Aux Data Track Files exists.

Note: Absence of a valid Synchronization Signal, as defined in Section 5.5, is an indication that no Show timeline is available to the Processor, i.e. no Show is loaded.

5.6.3 Main Sound Alignment

The first Sample of the Packet shall be output simultaneously with the first audio sample of the Primary Sound Track File of the Timeline Edit Unit associated with the Packet.

5.6.4 Minimizing Playout ID Changes

`Playout ID` should change only on Show (and not Composition) boundaries.

Note: Changing the value of `Playout ID` on a Composition boundary can result in the Processor interrupting playback to buffer Data Items (see Section 5.7.3.4.3). This is likely to cause, for instance, an interruption in audio playback.

5.7 Processor Operation

5.7.1 Processor Configurable Delay

In order for the essence output by a Processor to be perceived by the audience simultaneously with the primary picture, a Processor should allow the reproduction of a Timeline Edit Unit to be delayed by at most 500 ms from the reception of the corresponding Synchronization Signal Packet.

Annex A provides guidelines for setting the Processor configurable delay.

5.7.2 Invalid Packets

A Processor shall ignore invalid Packets as specified in Section 5.4.

5.7.3 State Machine

5.7.3.1 General

The Processor shall conform to the state machine illustrated in Figure 6 when using the Synchronization Signal.

The communications between the DCS, as defined in ST 430-10, and the Processor specified in this Section shall conform to the Auxiliary Content Synchronization Protocol specified in SMPTE ST 430-10.

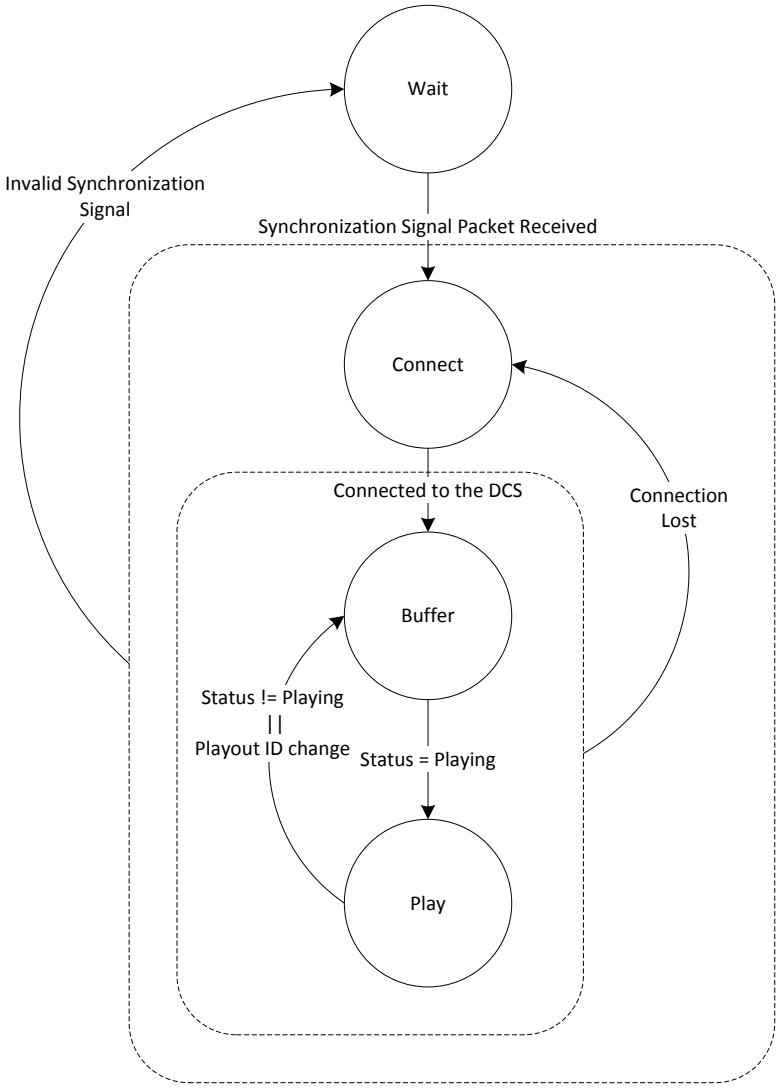

Figure 6 – Processor State Machine

5.7.3.2 Wait State

The Processor may disconnect from the DCS, if previously connected. When a valid Packet is received the Processor shall transition to the Connect State.

5.7.3.3 Connect State

Unless already connected to the DCS, the Processor shall, in order,

1. connect to the DCS
2. accept and respond to an Announce Request from the DCS
3. accept and respond to a Get New Lease Request from the DCS
4. transition to the Buffer State

In case of an ACS Error, the Processor shall disconnect from the DCS and transition to the Connect State.

If the Synchronization Signal is invalid, as specified in Section 5.4, the Processor shall transition to the Wait State.

5.7.3.4 Buffer and Play States

5.7.3.4.1 General

The Processor may buffer essence based on the values of `Timeline Edit Unit Index` and `Playout ID` fields from the most recently received Synchronization Signal Packet.

The Processor shall respond to Get Status requests.

As specified in Section 7.2.4 of SMPTE ST 430-10, in response to a Get Status request, the Processor shall return:

- `Processing`, if not able to reproduce essence due to incomplete buffering; and
- `RRP Successful`, otherwise.

5.7.3.4.2 Buffer State

When the `Status` field is:

- `Paused`, the Processor may reproduce essence corresponding to the `Timeline Edit Unit Index` and `Playout ID` fields from the most recently received Synchronization Signal Packet.
- `Stopped`, the Processor shall not reproduce essence.

The Processor shall transition to the Play State when:

- the `Status Flag` of the Synchronization Signal is equal to `Playing`; and
- buffering is complete.

If the connection is lost with the DCS, the Processor shall transition to the Connect State.

If the Synchronization Signal is invalid, as specified in Section 5.4, the Processor shall transition to the Wait State.

5.7.3.4.3 Play State

The Processor shall reproduce essence synchronously with the primary picture edit unit at the auditorium screen.

Note: The `Primary Picture Screen Offset` field of the Synchronization Signal specifies the offset between the Synchronization Signal and the primary picture edit unit at the auditorium screen – see Section 5.3.10.

When the `Status Flag` of the Synchronization Signal indicates a state other than `Playing`, the Processor shall transition to the Buffer State.

If the `Playout ID` changes while in the Play State, the Processor shall transition to the Buffer State.

If the connection is lost with the DCS, the Processor shall transition to the Connect State.

If the Synchronization Signal is invalid, as specified in Section 5.4, the Processor shall transition to the Wait State.

6 Transfer Protocol

6.1 General

The server and client shall support:

- the HTTP 1.1 protocol as specified in RFC 2616; and
- the HTTP over TLS 1.0 protocol specified in RFC 2818.

A persistent connection, as specified in Section 8.1 of RFC 2616, should be maintained.

When receiving the request specified in Section 6.5, the server shall respond according to Section 6.6.

The server may support other HTTP requests not specified in this specification.

Note: Section 4.4 of RFC 2616 specifies the various means by which the length of a message can be transmitted.

6.2 Aux Data Timeline

As illustrated in Figure 7, each Timeline Edit Unit shall be associated with zero or more Data Items (as specified in SMPTE ST 379), each associated with an Aux Data Track File (as specified in SMPTE ST 429-14) identified by a Data Essence Coding UL.

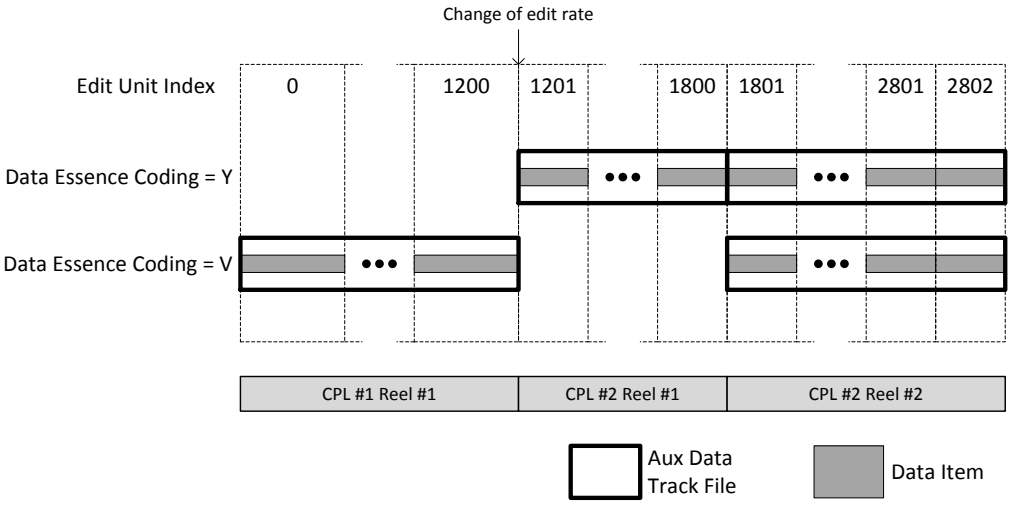

Figure 7 – Aux Data Timeline

Note: This specification assumes that an Aux Data Track File can have more than one Data Item in an Edit Unit and that multiple Aux Data Track Files with identical Data Essence Coding UL values can be associated with an Edit Unit.

6.3 Additional Processor Operation

6.3.1 General

This Section specifies operations in addition to those specified in Section 5.7.

6.3.2 Wait and Connect States

In the Wait and Connect States, the Processor shall not request Data Items using the Transfer protocol.

6.3.3 Buffer and Play States

The Processor shall accept and respond to a Set RPL Location Request from the DCS.

Note: As specified in Section 6.4, the Processor determines the protocol, network address and port number used to transfer Aux Data Items from the most recent Set RPL Location Request.

In the Buffer and Play States, the Processor may request Data Items using the Transfer protocol, based on the most recent value of the `Timeline Edit Unit Index` and `Playout ID` fields of the Synchronization Signal.

6.4 Protocol, Network Address and Port Number

The specification allows different combinations of protocol, network address and port number to be specified depending on the Data Essence Coding UL of the Data Items being requested. For instance, some types of essence can require transmission using HTTP over TLS 1.0, while others can use HTTP.

For a given Data Essence Coding UL value, the client shall initially use the following protocol, network address and port number:

- The protocol shall be HTTP.
- Using the Resource URL item of the Set RPL Location message specified in SMPTE ST 430-10:
    - the network address shall be the host component of the Resource URL, if present; otherwise it shall be the network address of the DCS;
    - the port component shall be the port component of the Resource URL, if present; otherwise it shall be the default port specified by the protocol.

As provided in Section 6.6.3, the server may modify the protocol, network address and port number associated with a given Data Essence Coding UL.

The client may initiate a Control Protocol with the server as specified in SMPTE ST 430-10 with the server if one was not previously initiated or one does not already exist.

6.5 Request

6.5.1 Method

The method shall be `"GET"`.

6.5.2 Request URI

6.5.2.1 General

The request URI shall be an absolute-path reference as specified in Section 4.2 of IETF RFC 3986.

6.5.2.2 Path Component

The path component shall be "/v1/auxdata/editunits".

The path component is uniquely associated with the normative provisions of this specification.

6.5.2.3 Query Component

The query component contains the parameters of the request. It shall conform to the QUERY syntax expressed in ABNF as specified in IETF RFC 2234:

```
NAME = 1*ALPHA
VALUE = *( UNRESERVED / PCT-ENCODED / DELIMS / ":" / "@" / "/" )
PCT-ENCODED = "%" HEXDIG HEXDIG
UNRESERVED = ALPHA / DIGIT / "-" / "." / "_" / "~"
DELIMS = "!" / "$" / "'" / "(" / ")"
                / "*" / "+" / "," / ";"
PARAMETER = NAME "=" VALUE
QUERY = PARAMETER *("&" PARAMETER)
```

A given NAME value shall appear at most once in a given query component.

PARAMETER with NAME values specified in Table 8 shall be present. PARAMETER values with other NAME values shall be ignored.

Table 8 – Aux Data Request Parameters

Name	Value
coding_UL	UL encoded as specified in Annex B of SMPTE ST 298.
start	Base-10 integer in the range [0..2^{32}-1] represented according to the syntax: `START = 1*DIGIT`
count	Base-10 integer in the range [0..2^{32}-1] represented according to the syntax: `COUNT = 1*DIGIT`

accept	The kind of AuxDataBlock accepted by the client.
	`ACCEPT = ACCEPT-KIND *("," ACCEPT-KIND)`
	`ACCEPT-KIND = 1*ALPHA`
	A given ACCEPT-KIND value shall appear at most once.

At least one of the ACCEPT-KIND values of the accept parameter shall be equal to one of the values listed in Table 9. Other ACCEPT-KIND values may present but shall be ignored by the server. Section 6.6.2.3 specifies Server behavior in the presence of the values listed in Table 9.

Table 9 – Defined ACCEPT-KIND values

Value
plaintext
encrypted

Note: The encoding of the query component is consistent with `application/x-www-form-urlencoded` as specified in the W3C HTML 4.01 Specification.

6.5.2.4 Fragment

A fragment component may be present and shall be ignored by the server.

6.5.3 Message Body

A request message body may be present and shall be ignored by the server.

6.6 Response

6.6.1 General

The server shall return all Data Items with Data Essence Coding UL equal to `coding_UL` in the timeline interval [`start`, `start` + `count`) , with `coding_UL`, `start` and `count` being the values of the parameters received in the request, as specified in Table 8.

The number of Data Items returned may be smaller than `count` if

- `start` + `count` exceeds the timeline duration; or
- if data blocks Data Items with Data Essence Coding UL equal to `coding_UL` do not exist for one or more timeline edit units; or
- if the server does not have the resources to return all requested Data Items.

If the server completes the request successfully, the server shall respond as specified in Section 6.6.2.

The absence of Data Items with Data Essence Coding UL equal to `coding_UL` in the requested interval, or the overall timeline, shall not cause an error.

Sections 6.6.3 - 6.6.6 specify responses when the server does not complete the request successfully. The server may return other status codes, as specified in IETF RFC 2616.

6.6.2 No Error

6.6.2.1 Content type

The content type shall be `"application/smpte336m"`, as specified in IETF RFC 6597.

6.6.2.2 Status code

The status code shall be `"200 OK"`.

6.6.2.3 Message Body

The message body shall consist of a sequence of contiguous and complete KLV triplets, as specified in SMPTE ST 336. The KLV encoding of item types shall be as specified in SMPTE ST 336 and SMPTE ST 377.

The first KLV triplet shall be an `AuxDataBlockTransferHeader` triplet as specified in Table 10, and shall be immediately followed by a sequence of zero or more contiguous `AuxDataBlock` triplets as specified in Table 11.

Table 10 – AuxDataBlockTransferHeader Pack

Item Name	Type	Len	UL	Meaning
`AuxDataBlockTransferHeader`	Pack Key	16	06.0E.2B.34.02.7F.01.01.0C.03.01.01.00.00.00.00	AuxDataBlockTransferHeader Pack Key
`Length`	BER Length	5	n/a	Pack length
`Edit Unit Range Start Index`	UInt32	4	06.0E.2B.34.01.01.01.0E.02.40.01.01.00.00.00.00	Index of the first Edit Unit of the timeline interval covered by the transfer.
`Edit Unit Range Count`	UInt32	4	06.0E.2B.34.01.01.01.0E.02.40.01.02.00.00.00.00	Number of Edit Units of the timeline interval covered by the transfer.

The `Length` field shall be a long-form BER value encoded with a fixed length of 5 bytes.

The `Edit Unit Range Start Index` item of `AuxDataBlockTransferHeader` shall be equal to the `start` request parameter.

The `Edit Unit Range Count` item of `AuxDataBlockTransferHeader` shall be equal to the number of edit units beyond `Edit Unit Range Start Index` that the response covers. `Edit Unit Range Count` shall not be larger than the `count` request parameter.

The actual number of Data Items returned is determined by the number of `AuxDataBlock` triplets returned.

EXAMPLE: Assuming that a Processor requests Data Items in the timeline range [0, 100), the Server can, for instance, respond with 50 `AuxDataBlock` triplets and indicate that it considered on the first 75 Edit Units following `Edit Unit Range Start Index` by setting `Edit Unit Range Count` to 75. The Processor can then set `Edit Unit Range Start Index` to 75 in its next request.

Other triplets may follow the last `AuxDataBlock` triplets but none shall be `AuxDataBlock` or `AuxDataBlockTransferHeader` triplets, and all shall be ignored.

Table 11 – AuxDataBlock Pack

Item Name	Type	Len	UL	Meaning
AuxDataBlock	Pack Key	16	06.0E.2B.34.02.7F.01.01.0C.03.01.02.00.00.00.00	AuxDataBlock Pack Key
Length	BER Length	5	n/a	Pack length
Edit Unit Index	UInt32	4	06.0E.2B.34.01.01.01.0E.02.40.01.03.00.00.00.00	Index of the timeline Edit Unit
Edit Unit Edit Rate	Rational	8	06.0E.2B.34.01.01.01.0E.02.40.01.04.00.00.00.00	Edit rate of the timeline Edit Unit
Source Data Essence Coding UL	UL	16	06.0E.2B.34.01.01.01.0E.02.40.01.05.00.00.00.00	Data Essence Coding UL of the source Aux Data Track File
Source Data Item Length	UInt64	8	06.0E.2B.34.01.01.01.0E.02.40.01.06.00.00.00.00	Length in bytes of the Source Data Item element
Source Data Item	Byte Array	var	06.0E.2B.34.01.01.01.0E.02.40.01.07.00.00.00.00	Data Item of the source Aux Data Track File
Source Cryptographic Context Length	UInt64	8	06.0E.2B.34.01.01.01.0E.02.40.01.08.00.00.00.00	Length in bytes of the Cryptographic Context Set
Source Cryptographic Context	Byte Array	var	06.0E.2B.34.01.01.01.0E.02.40.01.09.00.00.00.00	Cryptographic Context Set, if any, associated with the Elements of the Data Item contained in Source Data Item element

The Length field shall be a long-form BER value encoded with a fixed length of 5 bytes.

Note: The size of the Length field effectively limits the maximum length of the Source Data Item field.

Each AuxDataBlock triplet shall correspond to a single Data Item on the timeline, and the returned AuxDataBlock shall be ordered in the order of the underlying Data Items on the timeline.

Note 1: More than one AuxDataBlock triplet can be associated with an Edit Unit if the Edit Unit contains more than one Data Item or multiple Aux Data Track Files have identical Data Essence Coding UL values.

The Source Data Item shall contain the zero or more KLV triplets of the Data Item of the Aux Data Track File at the Edit Unit Index of the timeline.

The Edit Unit Edit Rate item shall be the edit rate of the edit unit on the timeline.

If the accept parameter:

- contains the value encrypted, then the KLV triplets shall be encrypted according to SMPTE ST 429-6 and the Source Cryptographic Context item shall contain the Cryptographic Context Set associated with the KLV triplets;
- otherwise, the KLV triplets shall be plaintext and Source Cryptographic Context item shall contain 0 bytes.

Note 2: This specification supports only a single Cryptographic Context Set per Track File.

6.6.3 Moved Permanently

The server may respond with status "301 Moved Permanently". If so, the protocol, network address and port number for future requests with the given coding_UL value shall be as follows:

- The protocol shall be HTTP if the Location header scheme component is "http" or HTTP over TLS 1.0 if the Location header scheme component is "https".
- The network address and port number shall be the host and port component, respectively, of the Location header. If the port component is absent, the port number shall be the default port associated with the protocol.

6.6.4 Bad Request

If the query component of the URI does not conform to Section 6.5.2, the server shall respond with status code "400 Bad Request".

6.6.5 Internal Server Error

If the server is not capable of fulfilling the request due to an internal error, the server shall respond with status code "500 Internal Server Error".

If the response contains an entity containing an explanation of the error situation, as specified in RFC 2616, the content-type of the entity should be "text/plain" as specified in RFC 2046.

6.6.6 404 Not Found

If the Data Items are not available encrypted according to SMPTE ST 429-6 (resp. in plaintext), and the accept parameter does not contain the encrypted (resp. plaintext) value, the server shall respond with status code "404 Not Found".

Note: This specification does not specify the conditions under which Data Items are available encrypted or plaintext.

Annex A Using Primary Picture Output Offset and Primary Picture Screen Offset (Informative)

Figure A.1 illustrates the relationship between the `Primary Picture Output Offset` (DS) and `Primary Picture Screen Offset` (MP) fields of the Synchronization Signal.

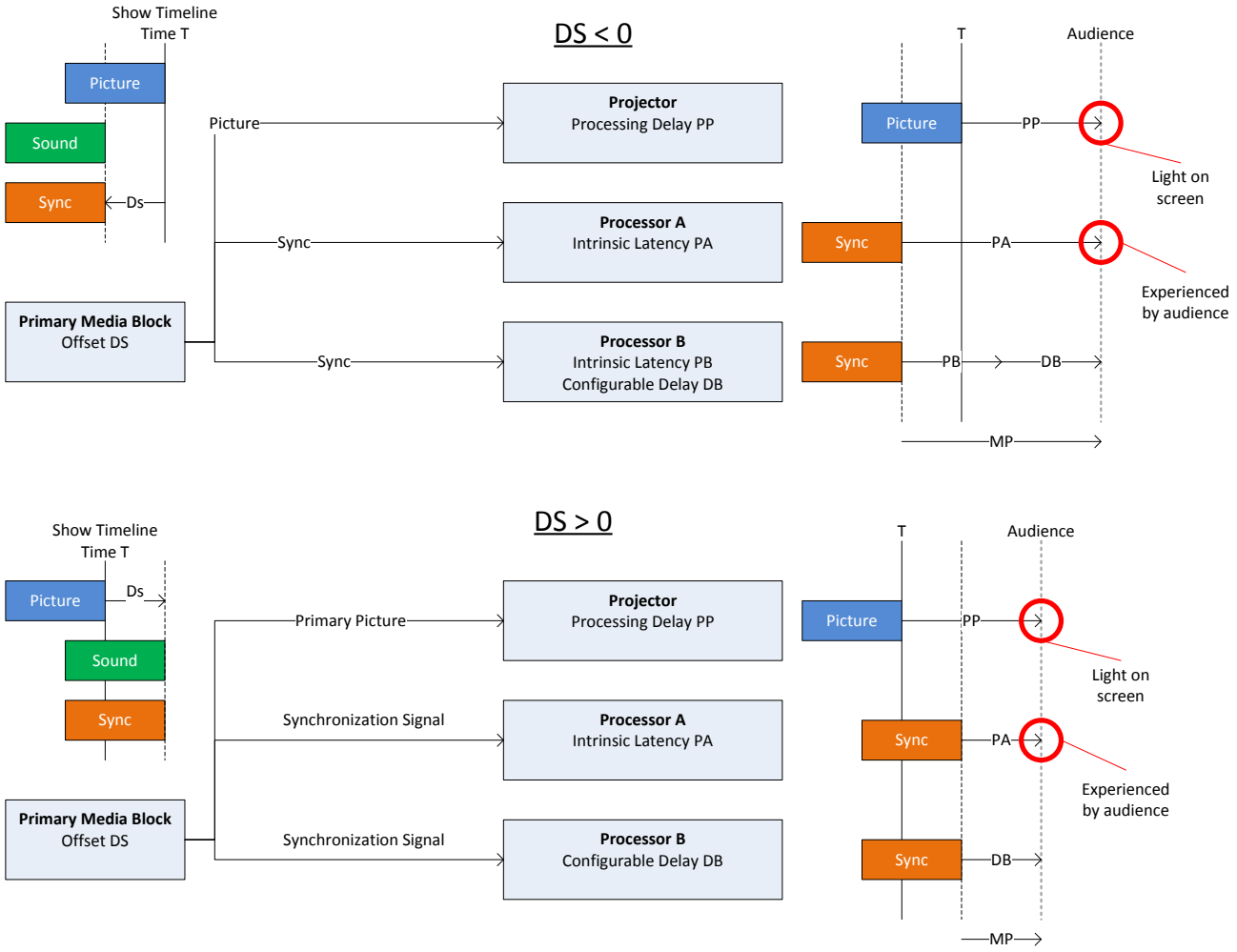

Figure A.1 – Timing Model for Positive and Negative Values of `Primary Picture Output Offset`

The Primary Picture Screen Offset (MP) is typically set at time of installation as the maximum intrinsic latency reported by the Processors receiving the Synchronization Signal. The intrinsic latency PX of a Processor X is the minimal delay between the time a Synchronization Signal Packet is received by the Processor and the time the essence output by the Processor is experienced by the audience. PX includes processing delay within the Processor as well as any propagation delay to the and within the auditorium.

The Primary Picture Output Offset (DS) can be determined by subtracting the largest Processor intrinsic latency (MP) from the Projector processing latency (Pp), i.e. DS = PP - MP.

In order for the essence output by a Processor X to be perceived by the audience simultaneously with the primary picture, the Processor X can insert an additional configurable delay DX to its internal processing such that PX + DX = MP; i.e., DX is equal to the value of the Primary Picture Screen Offset minus its latency.

Alternatively, a Processor wishing to synchronize its operation with the Primary Picture Edit Unit as output from the Media Block, can determine the time of the latter by subtracting DS from the time the corresponding Packet is received.