ATSC Recommended Practice: Techniques for Signaling, Delivery, and Synchronization

Doc. A/351:2022-03
31 March 2022

Advanced Television Systems Committee
1300 I Street, N.W., Suite 400E
Washington, D.C. 20005
202-872-9160

Implementers with feedback, comments, or potential bug reports relating to this document may contact ATSC at https://www.atsc.org/feedback/.

Revision History

Version	Date
A/351:2019 approved Amendment No. 1 approved Amendment No. 2 approved	28 August 2019 20 January 2020 15 February 2021
A/351:2021 published. This is a rollup of Amendments No. 1 and No. 2	15 February 2021
A/351:2022-03 published (references to ATSC documents updated)	31 March 2022

Table of Contents

Index of Figures and Tables

ATSC Recommended Practice: Techniques for Signaling, Delivery, and Synchronization

1. SCOPE

This document provides recommended practices for the ATSC 3.0 Signaling, Delivery, Synchronization, and Error Protection standard as specified in A/331 [1]. The document contains recommendations for broadcasters on the usage of the ROUTE and MMTP protocols and their associated technical capabilities in support of different Service delivery scenarios. In addition, transmission-related guidelines are provided on a variety of other functions and mechanisms as defined in A/331 including Service and audio language signaling, advanced emergency information, Staggercast, and the mapping between Service delivery and lower layer transport.

1.1 Introduction and Background

The ATSC 3.0 Signaling, Delivery, Synchronization, and Error Protection standard [1] specifies a diverse set of IP-based content delivery and Service signaling functionalities. The recommended practices in this document are intended to assist broadcasters in the selection and configuration of A/331-compliant emission side equipment concerning media transport, signaling capabilities and other technical features, to fulfill a variety of use cases and associated Service requirements.

1.2 Organization

This document is organized as follows:

- Section 1 – Outlines the scope of this document and provides a general introduction.
- Section 2 – Lists references and applicable documents.
- Section 3 – Provides definitions of terms, acronyms, and abbreviations for this document.
- Sections 4 and 5 – Recommended Practice sections.

2. REFERENCES

All referenced documents are subject to revision. Users of this Recommended Practice are cautioned that newer editions might or might not be compatible.

2.1 Informative References

The following documents contain information that may be helpful in applying this Recommended Practice.

[1] ATSC: “ATSC Standard: Signaling, Delivery, Synchronization and Error Protection,” Doc. A/331:2022-03, Advanced Television Systems Committee, Washington, DC, 31 March 2022.

[2] IEEE: “Use of the International Systems of Units (SI): The Modern Metric System,” Doc. SI 10-2002, Institute of Electrical and Electronics Engineers, New York, NY.

[3] ATSC: “ATSC Standard: Scheduler / Studio to Transmitter Link,” Doc. A/324:2022-03, Advanced Television Systems Committee, Washington, D.C., 31 March 2022.

[4] ATSC: “ATSC Standard: Physical Layer Protocol,” Doc. A/322:2022-03, Advanced Television Systems Committee, Washington, DC, 31 March 2022.

[5] ISO/IEC: “Information technology – Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats,” Doc. ISO/IEC 23009-1:2014,

2nd Edition, International Organization for Standardization/International Electrotechnical Commission, Geneva Switzerland.

[6] ATSC: "ATSC Standard: Audio Common Elements," Doc. A/342 Part 1:2022-03, Advanced Television Systems Committee, Washington, DC, 31 March 2022.

[7] ATSC: "ATSC Standard: AC-4 System," Doc. A/342 Part 2:2022-03, Advanced Television Systems Committee, Washington, DC, 31 March 2022.

[8] ATSC: "ATSC Standard: MPEG-H System," Doc. A/342 Part 3:2022-03, Advanced Television Systems Committee, Washington, DC, 31 March 2022.

[9] ATSC: "ATSC Standard: A/321, System Discovery and Signaling," Doc. A/321:2022-03, Advanced Television Systems Committee, Washington, DC, 31 March 2022.

[10] ATSC: "ATSC Standard: Interactive Content," Doc. A/344:2022-03, Advanced Television Systems Committee, Washington, DC, 31 March 2022.

[11] ETSI: "Universal Mobile Telecommunications Systems (UMTS); LTE; Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs (3GPP TS 26.346 version 13.3.0 Release 13)," Doc. ETSI TS 126 346 v13.3.0 (2016-01), European Telecommunications Standards Institute, January 2016. http://www.etsi.org/deliver/etsi_ts/126300_126399/126346/13.03.00_60/ts_126346v130300p.pdf

[12] ISO/IEC: "Information Technology – High efficiency coding and media delivery in heterogeneous environments – Part 13: MMT implementation guidelines," Doc. TR 23008 13:2015(E), International Organization for Standardization/International Electrotechnical Commission, Geneva Switzerland.

[13] IETF: RFC 5052, "Forward Error Correction (FEC) Building Block," Internet Engineering Task Force, Reston, VA, August 2007. http://tools.ietf.org/html/rfc5052

[14] IETF: RFC 5651, "Layered Coding Transport (LCT) Building Block," Internet Engineering Task Force, Reston, VA, October 2009. http://tools.ietf.org/html/rfc5651

[15] IETF: RFC 5775, "Asynchronous Layered Coding (ALC) Protocol Instantiation," Internet Engineering Task Force, Reston, VA, April 2010. http://tools.ietf.org/html/rfc5775

[16] IETF: RFC 6330, "RaptorQ Forward Error Correction Scheme for Object Delivery," Internet Engineering Task Force, Reston, VA, August 2011. http://tools.ietf.org/html/rfc6330

[17] IETF: RFC 6363, "Forward Error Correction (FEC) Framework," Internet Engineering Task Force, Reston, VA, October 2011. http://tools.ietf.org/html/rfc6363

[18] IETF: RFC 6726, "FLUTE - File Delivery over Unidirectional Transport," Internet Engineering Task Force, Reston, VA, November 2012. http://tools.ietf.org/html/rfc6726

[19] IETF: RFC 7231, "Hypertext Transfer Protocol -- HTTP/1.1," Internet Engineering Task Force, Reston, VA, June 2014. http://tools.ietf.org/html/rfc7231

[20] DASH IF: "Guidelines for Implementation: DASH-IF Interoperability Points for ATSC 3.0, Version 1.1," DASH Interoperability Forum, June 12, 2018. https://dashif.org/wp-content/uploads/2018/06/DASH-IF-IOP-for-ATSC3-0-v1.1.pdf

[21] ISO/IEC: "Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 1: MPEG media transport (MMT)," Doc. ISO/IEC 23008-1:2017(E), International Organization for Standardization/International Electrotechnical Commission, Geneva Switzerland.

[22] ISO/IEC: "Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 3: 3D audio," Doc. 23008-3:2015, with Amendment 2:2016 and Amendment 3:2017. International Organization for Standardization/International Electrotechnical Commission, Geneva Switzerland.

[23] IETF: RFC 4151, "The 'tag' URI Scheme," Internet Engineering Task Force, Reston, VA, October 2005. https://tools.ietf.org/html/rfc4151

[24] EIDR: "Introduction to EIDR Video Services Registry," The Entertainment ID Registry Association, v0.3, 2016/11/18. http://eidr.org/documents/Introduction_to_the_EIDR_Video_Services_Registry.pdf

[25] EIDR: "EIDR and the DOI Proxy," The Entertainment ID Registry Association, 2015-04-24. http://eidr.org/documents/EIDR_and_the_DOI_Proxy.pdf

[26] ATSC: "ATSC Standard: A/52, Digital Audio Compression (AC-3, E-AC-3)," Advanced Television Systems Committee, Washington, DC, 25 January 2018.

3. DEFINITION OF TERMS

With respect to definition of terms, abbreviations, and units, the practice of the Institute of Electrical and Electronics Engineers (IEEE) as outlined in the Institute's published standards [2] shall be used. Where an abbreviation is not covered by IEEE practice or industry practice differs from IEEE practice, the abbreviation in question will be described in Section 3.3 of this document.

3.1 Compliance Notation

This section defines compliance terms for use by this document:

should – This word indicates that a certain course of action is preferred but not necessarily required.

should not – This phrase means a certain possibility or course of action is undesirable but not prohibited.

As an additional aid to readers, critical recommendations in this document are noted by the graphic ☑. When the section header is checked, the entire section is deemed critical.

3.2 Treatment of Syntactic Elements

This document contains symbolic references to syntactic elements used in the audio, video, and transport coding subsystems. These references are typographically distinguished by the use of a different font (e.g., restricted), may contain the underscore character (e.g., sequence_end_code) and may consist of character strings that are not English words (e.g., dynrng).

3.3 Acronyms and Abbreviations

The following acronyms and abbreviations are used within this document.

3GPP 3rd Generation Partnership Program
AEA Advanced Emergency InformAtion
AEAT AEA Table
ALC Asynchronous Layered Coding
AL-FEC Application Layer Forward Error Correction
ALP ATSC 3.0 Link-layer Protocol
AMP Application Media Player
APD Associated Procedures Description
ATSC Advanced Television Systems Committee
AU Access Unit
A/V Audio/Visual
BA Broadcaster Application
BBP BaseBand Packet
BICM Bit-Interleaved and Coded Modulation
BMFF Base Media File Format
BSID Broadcast Stream ID
CBR Constant Bit Rate
CTI Convolutional Time Interleaver
DASH Dynamic Adaptive Streaming over HTTP
dB decibel
DNS Domain Name System
DOI Digital Object Identifier
DRM Digital Rights Management
DWD Distribution Window Description
EAS Emergency Alert System
EFDT Extended File Delivery Table
EIDR Entertainment Industry Data Registry
ESG Electronic Service Guide
ETSI European Telecommunications Standards Institute
EXT_FTI (LCT) Header Extension for FEC Object Transmission Information
FCC Federal Communications Commission
FDM Frequency Division Multiplexing
FDT File Delivery TableFEC Forward Error Correction
FFT Fast Fourier Transform
FLUTE File Delivery over Unidirectional Transport
GI Guard Interval
HELD HTML Entry pages Location Description
HRBM Hypothetical Receiver Buffer Model
HTI Hybrid Time Interleaver
HTML Hypertext Markup Language

HTTP	Hypertext Transfer Protocol
ID	IDentifier
IEC	International Electrotechnical Commission
IP	Internet Protocol
IS	Initialization Segment
ISO	International Standards Organization
kb	kilobit or kilobits
kHz	kiloHertz
LCT	Layered Coding Transport
LDM	Layered Division Multiplexing
LDPC	Low Density Parity Check
LLS	Low Level Signaling
LMT	Link Mapping Table
MA3	MMT ATSC3
MBMS	Multimedia Broadcast/Multicast Service
MDE	Media Delivery Event
MFU	Media Fragment Unit
MHz	MegaHertz
MIME	Multipurpose Internet Mail Extensions
MMT	MPEG Media Transport
MMTP	MPEG Media Transport Protocol
MPD	Media Presentation Description
MPEG	Motion Pictures Experts Group
MPU	Media Processing Unit
msec	millisecond or milliseconds
NGA	Next Generation Audio
NID	Namespace ID
NRT	Non-Real Time
nsec	nanosecond or nanoseconds
NTP	Network Time Protocol
OFDM	Orthogonal Frequency Division Multiplexing
OTA	Over The Air
OTI	Object Transmission Information
OTT	Over The Top
PBS	Public Broadcasting Service
PHY	PHYsical layer
PLP	Physical Layer Pipe
PTP	Precision Time Protocol
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency

RAP	Random Access Point
RFC	Request for Comments
RMP	Receiver Media Player
ROUTE	Real-Time Object Delivery over Unidirectional Transport
RRT	Rating Region Table
RSAT	Regional Service Availability Table
SAP	Secondary Audio Program
SAP	Stream Access Point
SBN	Source Block Number
SCT	Sender Current Time
SLS	Service Layer Signaling
SLT	Service List Table
SNR	Signal to Noise Ratio
STL	Studio-to-Transmitter Link
S-TSID	Service-based Transport Session Instance Description
TDM	Time Division Multiplexing
TOI	Transport Object Identifier
TS	Technical Specification
TSI	Transport Session Identifier
T-STD	(MPEG-2 Transport Stream) System Target Decoder
TV	TeleVision
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
USBD	User Service Bundle Description
UTC	Universal Coordinated Time
VBR	Variable Bit Rate
VDS	Video Description Services
VoD	Video on Demand
XML	eXtensible Markup Language

3.4 Terms

The following terms are used within this document.

Analyzed Media Duration – As defined in A/324 [3], Analyzed Media Duration is the shortest Period between times at which data segment boundaries in all data Streams on the inputs of a Scheduler align.

Bootstrap – Set of symbols as defined in A/321 [9].

Complete Main (CM) – The CM type of audio Service contains a complete audio program (which typically includes dialog, music, silence, and effects). The CM Service contains any number of channels. Audio in multiple languages is provided by supplying multiple CM Services, each in a different language. See A/52 [26].

Extended FDT – File description entries for the one or more delivery objects carried in a source flow. The Extended FDT contains the descriptions of the affiliated delivery objects, including i) nominal FLUTE FDT parameters as defined in RFC 6726 [18], ii) certain extensions to the FLUTE FDT as defined by 3GPP for MBMS in [11], and iii) ATSC-defined FDT parameters as specified in A/331 [1].

FEC Super-Object – The concatenation of one or more FEC Transport Objects in order to bundle those FEC Transport Objects for FEC protection. See A/331 [1].

FEC Transport Object – The concatenation of a delivery object, padding octets and size information in order to form an N-symbol-sized chunk of data, where $N \geq 1$. See A/331 [1].

HTTP File Repair – HTTP transactions between the receiver and a network repair server, conducted over the broadband channel, which enables the receiver to recover partially delivered object(s).

LLS (Low Level Signaling) – Signaling information which supports rapid channel scans and bootstrapping of Service acquisition by the receiver.

MDE (Media Delivery Event) – A Media Delivery Event (MDE) is the arrival of a collection of bytes that is meaningful to the upper layers of the stack (for example the media player and decoder(s)). MDE data blocks have delivery deadlines. The grouping of bytes that is a RAP is a “Delivery” in ROUTE and the arrival of these bytes is an “Event” at an upper layer.

Media Segment – A DASH Segment that complies with media format in use and enables playback when combined with zero or more preceding segments, and an Initialization Segment (if any). See DASH-IF-IOP-for-ATSC3-0 [20].

Scheduler – As defined in A/324 [3], the Scheduler is a Studio side-function that allocates physical capacity to data Streams based on instructions from the System Manager combined with the capabilities of the specific system.

Service – A collection of media components presented to the user in aggregate; components can be of multiple media types; a Service can be either continuous or intermittent; a Service can be Real Time or Non-Real Time; Real Time Service can consist of a sequence of TV programs.

SLS (Service Layer Signaling) – Signaling which provides information for discovery and acquisition of ATSC 3.0 Services and their content components.

Staggercast – As defined in A/331[1], Staggercast is a method for supporting more robust audio reception by the delivery of a redundant version of a main audio component, possibly coded with lower quality (lower bitrate, number of channels, etc.), and with a significant delay ahead of the audio with which it is associated. Receivers that support the Staggercast feature can switch to the Staggercast stream should main audio become unavailable. The delivery delay between Staggercast audio and main audio is chosen to be high enough to provide robustness due to sufficient time diversity between the two.

4. RECOMMENDED PRACTICE TOPICS

4.1 Supported Service Combinations of Physical Layer and Media Codec(s)

In the Physical Layer Specification [4] there is support for Layered Division Multiplexing (LDM). The physical layer provides Core Physical Layer Pipe(s) (PLP(s)) and Enhanced PLP(s). These capabilities may be applied in various manners with the media streams to create a set(s) of streaming media Services. Some examples of possible Services are depicted in **Figure 4.1**, which is comprised of a. through g. sub-figures.

Descriptions of the Service depicted by each sub-figure follow.

a) Shows N languages of audio presentation carried in one or more Core PLPs. These may be organized as N Complete Main audio streams delivered in per language ISO BMFF container streams. This is less efficient than the Complete Main audio presentation carried in one ISO BMFF container stream plus N-1 languages of dialog carried in their respective ISO BMFF container streams. Use of N dialog container streams plus a stream carrying the music and effects is also possible.
b) The separate dialog ISO BMFF file streams may be carried in Enhanced PLPs, rather than in Core PLP(s).
c) The separate 2^{nd}- N^{th} dialog ISO BMFF file streams may be delivered via broadband.
d) Shows N video components carried in one or more Core PLPs. (Multiple video components may be desired for picture-in-picture or other scenarios.) One video component is signaled as the primary video component.
e) A single video component of a presentation may be carried in a single common ISO BMFF container stream with up to two layers of optional spatial scalability.
f) The base layer of a spatially scaled video may be sent in a Core PLP as an ISO BMFF container stream. The related enhancement layer can be sent in a separate ISO BMFF container stream in the same PLP or a separate Enhanced PLP.
g) Video components, e.g., a non-primary video component or an enhancement layer of a spatially scaled video, may be delivered via broadband.

This description of potential combinations is by no means comprehensive, but rather illustrative of various Service delivery options.

Figure 4.1 Supportable physical layer and codec Service delivery.

4.2 Interaction of the Physical Layer and the ROUTE/DASH Stack in a Receiver

An example implementation of DASH is shown in **Figure 4.2**. This figure is patterned after "Figure 1 — Example system for DASH formats" in the ISO/IEC DASH spec [5].

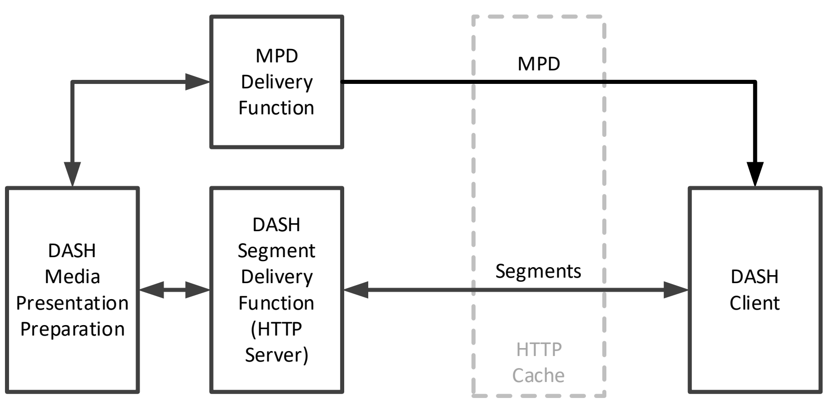


Figure 4.2 MPEG-DASH System Architecture.

An abstracted behavior of the ATSC 3.0 physical layer and ROUTE protocol stack from the perspective of HTTP cache on the device side of the ATSC 3.0 network is shown in **Figure 4.3**.

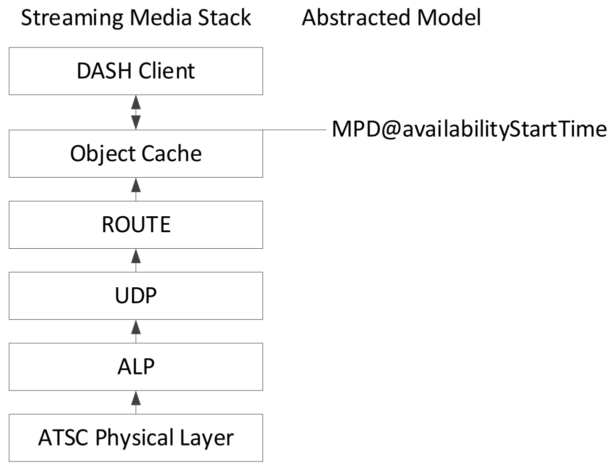


Figure 4.3 Receiver side cache availability start time model.

The last byte of a to-be-delivered Media Segment will be sent at such a time that the complete Media Segment can be fetched at `MPD@availabilityStartTime` from the device-side HTTP cache. The Media Segment has to be sent one theoretical physical layer delivery delay prior to `MPD@availabilityStartTime`.

If the MDE mode is being supported, the Sender Current Time (SCT) field in the `EXT_TIME` LCT extension header of the ROUTE packet should correspond to the time of the first byte of the to-be-delivered Segment. The conditionals in these statements allow for operational schemes where the related system Random Access Point (RAP) is sent earlier or later than one Segment duration after the most recent previous delivery deadline. The deadline for delivery advances at one Segment per Segment; however, this does not ensure that the span of the actual delivery of a given Segment is one Segment duration.

The Scheduler ensures that the delivery of the complete Segment will occur before a specific deadline, a corresponding `MPD@availabilityStartTime`.

Figure 4.4 illustrates the relationship between `MPD@availabilityStartTime`, the Scheduler deadline, and receiver start-up, and is discussed in some detail in the paragraphs below.

As discussed in Annex C of [3], there is a certain data organization to a system RAP and trailing media that provides for a channel change. The delivery of said metadata (signaling) is not described here. In considering this figure assume that the required system RAP and media are made available to the Scheduler at or before the required time.

This discussion concerns how required behavior of the Scheduler enables receiver start-up and media playback. As shown in black text and arrows above there is an `MPD@availabilityStartTime` and two related Scheduler deadlines. The earliest possible start time for the receiver corresponds to the earliest Scheduler deadline which is a fixed duration physical layer delay before `MPD@availabilityStartTime`. The receiver may start to play the whole or partial Segments in the object cache and be assured that the remaining media frames will be delivered before the object cache is emptied. This is illustrated as "MDE Delay" whose value should be set such that each Media Segment will complete delivery prior to the completion of playback of those same said Segments. The duration of the 'MDE Start Up Delay' illustrated in blue text and arrows above is delivered to the receiver on a Segment by Segment basis by the difference between the Sender Current Time (SCT) value and the `EXT_ROUTE_PRESENTATION_TIME` value in the LCT extension header. The request of Segment prior to `MPD@availabilityStartTime` should cause the HTTP interface to stream the Segment.

A receiver with the theoretically minimum physical layer delay can start and play whole Media Segments according to `MPD@availabilityStartTime`. All receivers that start up at the time `MPD@availabilityStartTime` + `@suggestedPresentationDelay` should achieve synchronous playback across all those receivers.

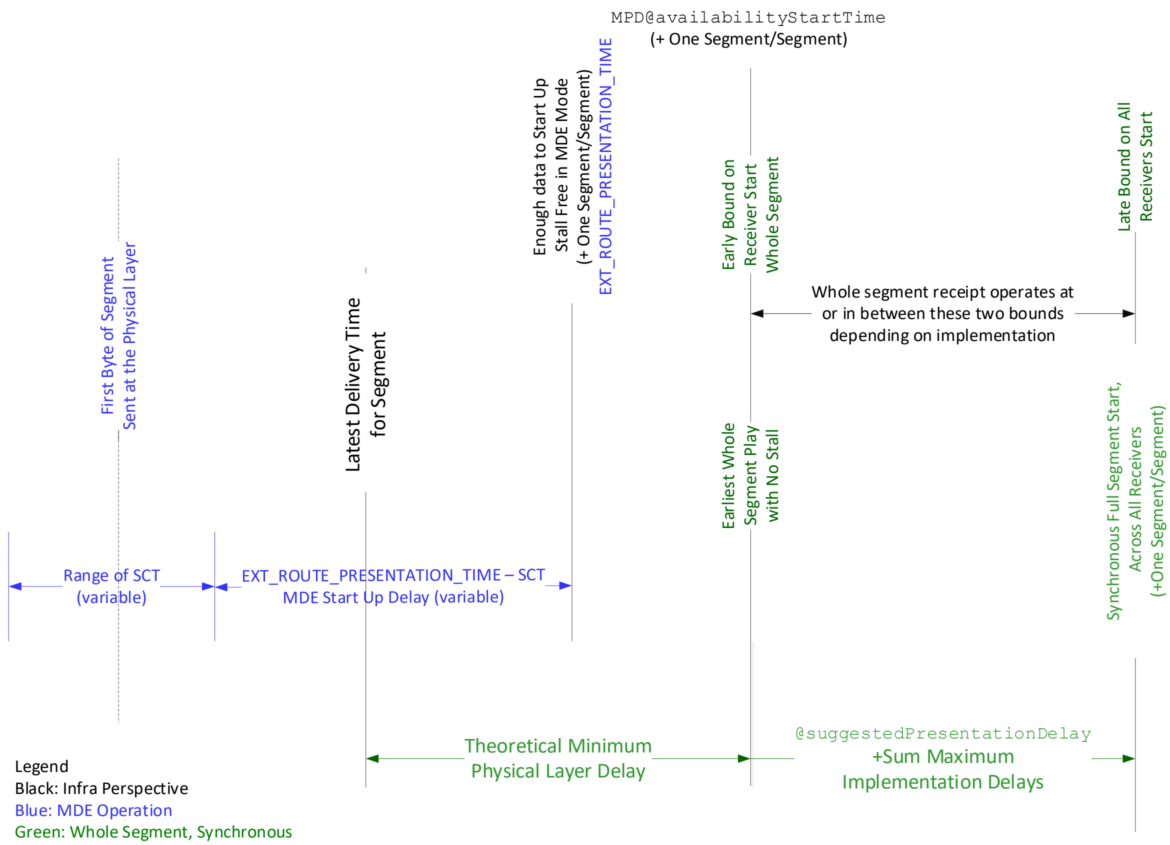


Figure 4.4 Playback time for device-referenced availability start time.

4.2.1 Impact of Physical Layer Configuration on Receiver Stack Delay

The minimum absolute delay encountered by a desired Segment, at the physical layer, depends on the configuration of the physical layer. A conceptual model for physical layer delay is shown in **Figure 4.5**.

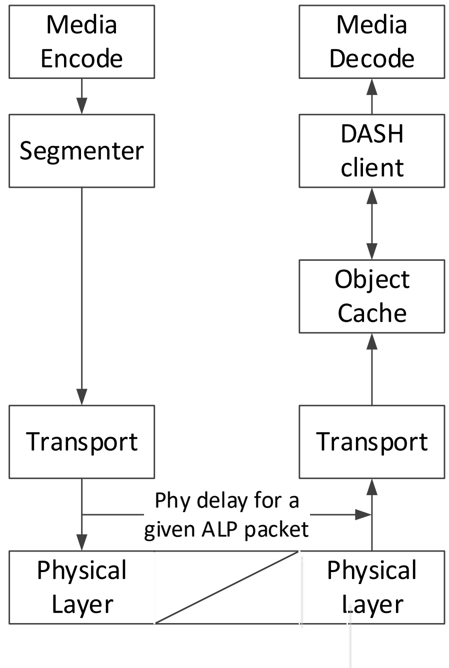


Figure 4.5 Physical layer delay conceptual model.

Network latency, transmitter delay and all factors in a broadcaster gateway are factored out in this analysis. Time relationships stay the same to ALP packets and the latencies in a studio are not necessary for this physical layer delay calculation of ALP packets.

Calculation of the latency is best described with a series of small 'golden nugget' type of equations that can be used for any purpose, including latency. Delay across the physical layer can be broken into two parts, symbol duration and packet delay. Symbol duration is calculated as:

$$symbol\ duration = \frac{FFT\ Size + GI\ samples}{0.384 \times (N + 16)}$$

where N = bsr_coefficient in A/321 [9] Section 6.1.1.1. (N = 2 for 6 MHz channel)

For extra information, physical layer OFDM frame duration is calculated as:

$$frame\ duration = Bootstrap + symbol\ duration \times Number\ of\ symbols$$

Each physical layer frame is demarked with bootstrap symbols. There are a variety of symbol types within a physical layer frame: preamble, subframe boundary or data. Each of those OFDM symbols (preamble, subframe boundary or data) contains a certain number of data cells. A data cell is one point in the modulation constellation (that contains a set amount of incoming bits). Those data cells carry a certain number of bits (as determined by the modulation order) of the ALP packet information. An example frame format that might help illustrate the concept is provided in **Figure 4.6**.

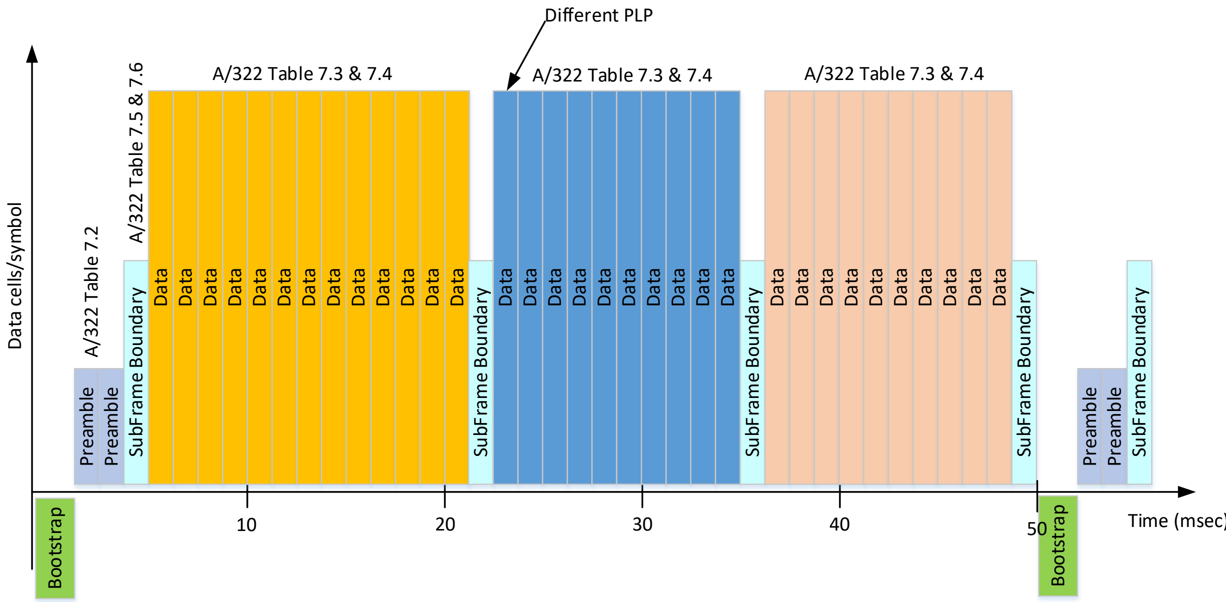


Figure 4.6 Example PHY OFDM frame structure.

A 4-symbol Bootstrap has 2.000msec duration (assuming sampling frequency is 384kHz × 16 with bsr_coefficient = 0) and zero data cells. The other OFDM symbol durations are calculated as in the above equation. Data cell capacity for preamble symbols is given in Table 7.2 of A/322 [4] in Section 7.2.6.2. Data cell capacity for subframe boundary symbols is given in Tables 7.5 and 7.6 of A/322 [4] in Section 7.2.6.4. Data cell capacity for data symbols is given in Tables 7.3 and 7.4 of A/322 [4] in Section 7.2.6.3.

Each data cell represents the amplitude value for each carrier in the FFT. FEC Frame creation, symbol formation, and frame multiplexing type of functions do not add latency. There is one dominant function that deliberately adds latency, time interleaving. Time interleaving aids robust reception of broadcast signals, meaning data cell values are distributed among the FFT carriers with a time depth. There are two modes of time interleaving, convolutional and hybrid convolutional / block interleaving. Convolutional time interleaving depth is calculated as:

$$CTI_{depth} = N_{rows} \times (N_{rows} - 1)$$

Where $N_{rows} = \{512,724,887,1024\}$ and if QPSK modulation is chosen, further values of $N_{\mathrm{rows}} = \{1254, 1448\}$ can be used.

Time interleaver depth of these N_{rows} values are calculated as:

$$Time\ depth = \frac{Time\ interleaver\ cell\ depth}{data\ cells/_{symbol}\ Table\ lookup} \times symbol\ duration$$

For example, choosing 64QAM modulation allows a choice of 1024 rows in the convolutional time interleaving resulting in 1,047,552 cell depth distribution. Using 8K FFT with number of carriers = 6913 and scattered pilot patter of SP3_2, there are 5711 data cells / data symbol. In addition, a guard interval choice of 1024 samples results in a symbol duration of 1.333 msec. Therefore, equating time spread from time depth,

$$Time\ spread = \frac{1{,}047{,}552\ cells}{5711\ cells/symbol} \times \frac{1.33\ \text{msec}}{symbol} \cong 244.56\ \text{msec}$$

This is an approximation because only data symbols were used, but preamble and subframe boundary symbols may also be populated with data cells.

Hybrid time interleaving depth is more complicated, and care must be taken so that the total number of memory elements does not exceed 2^{19}cells, but the equation is:

$$HTI_{depth} = \left(\frac{N_{iu}+1}{2}\right) \times N_{cells} \times \left\lceil \frac{N_{FEC\_TI\_MAX}}{N_{iu}} \right\rceil$$

Where:

$N_{iu} = Number\ of\ Interleaving\ Units$

$N_{cells} = \frac{\text{FEC}\ codelength}{\eta_{mod}}$; where $\eta_{mod} = log_2(modulation\ order)$

$N_{FEC\_TI\_MAX} = Maximum\ number\ of\ FEC\ Blocks$

For example, selecting a FEC code length of 64800 bits, 64QAM modulation, $N_{FEC\_TI\_MAX} = 37$, $N_{iu} = 4$ results in:

$$HTI_{depth} = \frac{4+1}{2} * \times \frac{64800\ bits}{6\ bits} \times \left\lceil \frac{37}{4} \right\rceil$$

$$HTI_{depth} = 2.5 \times 10800 \times 10 = 270{,}000\ cells$$

Using the same choices as in the convolutional interleaver case above for symbol duration and data cell / symbol section, the time spread results in:

$$Time\ spread = \frac{270{,}000\ cells}{5711\ cells/symbol} \times \frac{1.333\ \text{msec}}{symbol} = 63.035\ \text{msec}$$

Frequency interleaving operation occurs within one OFDM symbol duration, along with the frame construction of PLPs. BICM operation adds bits to incoming baseband packet payload bits ($k_{payload}$), interleaves them, applies LDPC coding, etc. These operations are all on bits in which processing time is less than the elementary period $T = \frac{1}{0.384 \times (N+16) MHz}$, which is ≈145 nsec for N = 2 (6 MHz channel). Therefore,

$$Packet\ Delay_{ALP} = \frac{Time\ Interleaver\ cell\ depth}{data\ cells/_{symbol}\ Table\ lookup} \times \frac{FFT\ size + GI\ samples}{0.384 \times (N+16)\ \text{MHz}}$$

Upon reception, the BICM operation delay will depend on the number of iterations in LDPC decoding in addition to this ALP packet delay, but that additional delay depends on implementation and received SNR quality (lower SNR generally results in more LDPC iterations).

Packets from the data source at the transmitter-side of the system emanate from the network layer (i.e., are IP packets). Transport layer ALP packets add a header to those IP packets and baseband packets again add another header to ALP packets. These headers are addition of bits and do not contribute to latency. There is one function in the Studio to Transmitter Link (STL) that concerns time at the broadcaster studio, which is the Scheduler. There is one physical layer OFDM frame buffer at the studio for the Scheduler to correctly assemble packets and direct how to configure the physical layer with the incoming IP packets. However, the time value delivered to exciters is set such that this frame delay is not included in physical layer latency as time relationships stay the same to ALP packets. Therefore, calculated packet delays from a transmitter-side data source (ROUTE or MMT) is calculated as:

$$Packet\ Delay_{IP} = \frac{Time\ Interleaver\ cell\ depth}{data\ cells\!\left/_{symbol}\right. Table\ lookup} \times \frac{FFT\ size + GI\ samples}{0.384 \times (N + 16)\ \mathrm{MHz}}$$

> Note: The de-interleaver delay is not considered in the construction of the MDE related delay, because MDE delivery inherently adapts to delay introduced by both the physical layer and the protocol stack.

4.3 Example Multiplex Constructions

ATSC 3.0 can carry streaming media and Non-Real-Time (NRT) based Services in a variety of manners. This section describes several alternatives and provides motivation as to why such multiplex constructions may exist. Example reasons are: saves mobile device battery power, provides faster channel change, simpler to implement, etc. These various multiplexing schemes may impose restrictions or require the use of defined features, which are pointed out on a per use case basis. These examples are not comprehensive but are illustrative of certain aspects that should be considered.

The Scheduler as defined in A/324 [3] organizes and maps source essence and metadata into ATSC 3.0 physical layer frames. To accomplish media scheduling, there must be an Analyzed Media Duration. This Analyzed Media Duration is specified, for example, as a few second(s) duration in media time. The Scheduler's task is to map available media, signaling, and NRT data onto ATSC 3.0 physical layer frame(s). The number of frames mapped is related to the Service delivery structure. In this section "Service" can mean an application-based, streaming media-based, or an AEAT-based Service.

These examples discuss the broadcast component of ATSC 3.0 Services. Any of these scheduling schemes can be used with hybrid Service delivery. The binding of broadband-delivered Service components to broadcast-delivered Service components happens above the stream layer, as defined in the MPD via Media Segment labeling.

All PHY scheduling schemes depicted use Time Division Multiplexing (TDM). This is a convenient means to draw the multiplex. The ATSC 3.0 physical layer can allow a mixture of Frequency Division Multiplexing (FDM) and TDM, but this is not depicted.

4.3.1 Single PLP Service Delivery One PHY Frame per Analyzed Media Duration

Single PLP delivery is the simplest scheduling scheme for ATSC 3.0. As shown in **Figure 4.7**, the scheme transports all Services in a single PLP with each Service allocated Constant Bit Rate (CBR). There is a constant physical layer frame size with the duration equal to the longest delivered Media Segment(s). The duration of all delivered Media Segments could be common in value. The order of delivery of the various per-Service Segments within a physical layer frame is unknown to the receiver at the higher layers, i.e., above the physical layer, the receiver must receive at least an entire physical layer frame to start media playback. While simple, this scheme has a longer channel change duration than can be possible with additional physical layer frames. This scheme consumes a total of one receiver PLP. The receiver radio is always on, so there is high receiver power consumption.

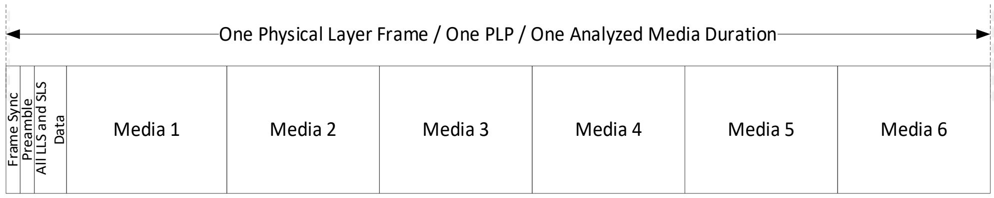


Figure 4.7 Single PLP delivery, one PHY frame / analyzed media duration.

4.3.2 Single PLP Multiplex with Multiple PHY Frames per Analyzed Media Duration

This delivery scheme is shown in **Figure 4.8**. This is a reorganization of the **Figure 4.7** multiplex to provide faster channel change. Each of the six physical layer frames as shown in **Figure 4.8** contains a complete Service RAP for its corresponding Service. The receiver power consumption is the same as in **Figure 4.7**. The receiver power consumption could be substantially decreased by allocating each physical layer frame to a different PLP for each Service, although this option is not illustrated.

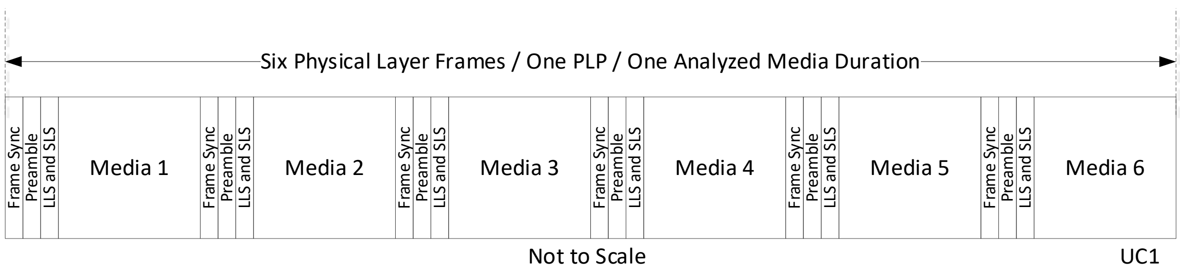


Figure 4.8 Single PLP delivery, six PHY layer frames / analyzed media duration.

4.3.3 Multiple PLP Statistical Multiplexing

It is well known that Variable Bit Rate (VBR) allocation across multiple Services achieves better Service capacity than CBR per Service. This sort of scheme is shown in **Figure 4.9**, whereby the audio and subtitle streams of six different Services are carried on a separate PLP from the PLP carrying the VBR-encoded video streams of the same six Services. While more Service capacity efficient than the schemes as shown in **Figure 4.7** or **Figure 4.8**, the channel change time is the same as that depicted in **Figure 4.7**, while the receiver power is high. This is not a particularly PLP resource efficient scheme. Unless there is at least a 1 dB difference in the robustness requirements of the individual Service components, they might be best run as a single PLP.

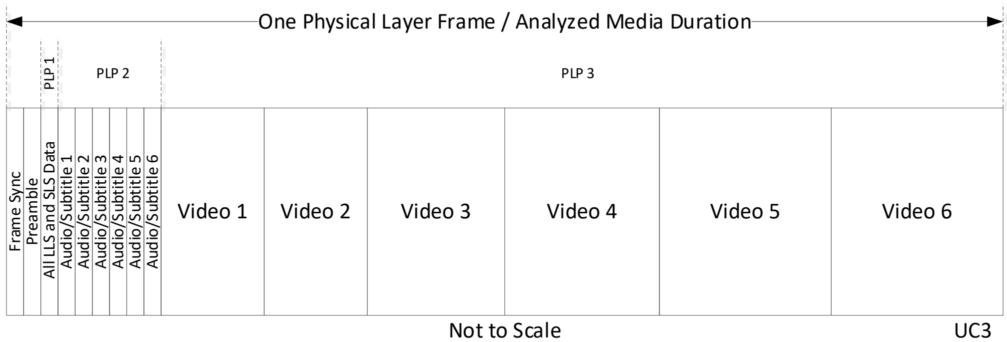


Figure 4.9 Three PLP stat mux, one PHY frame / analyzed media duration.

4.3.4 Multiple PLP Stat Multiplex with NRT

This statistical multiplex use case adds a dedicated PLP for NRT content carriage. The NRT PLP in this case is a shared resource i.e. all the NRT traffic for this instance of ATSC 3.0 can be delivered via this dedicated NRT PLP. Audio and signaling each have a dedicated PLP. **Figure 4.10** is an inefficient utilization of a PLP and provides no benefit relative to **Figure 4.11**, which merges signaling and audio carriage in a single PLP.

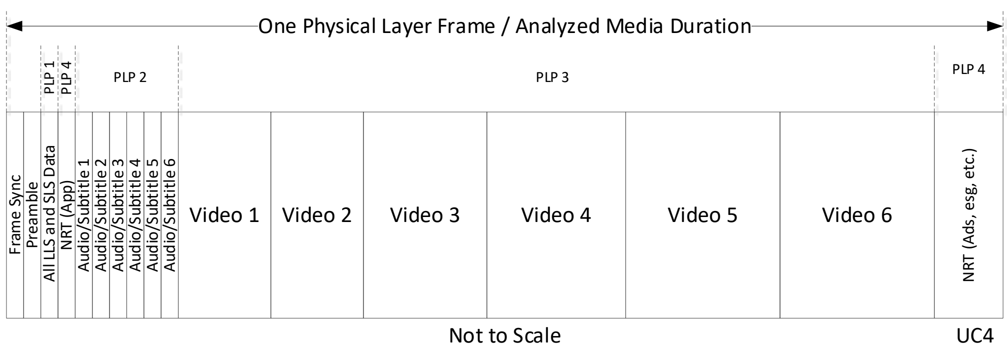


Figure 4.10 Four PLP stat mux one PHY frame / analyzed media duration.

4.3.5 Multiple PLP Stat Mux with NRT and Layered Video Service

This statistical multiplex use case adds physical layer specific layered delivery of media content via Layered Division Multiplexing (LDM). The Core Layer of LDM delivery utilizes a PLP resource. The Enhanced Layer of the LDM delivery also uses another PLP resource, even if no content for this Service is carried in the corresponding Core Layer PLP. As suggested in the **Figure 4.10** discussion above this configuration which combines delivery of signaling and audio is likely more efficient, because the signaling might not consume an entire Baseband Packet(s) (BBP) that could otherwise be filled with audio. Receiver power consumption is high, similar to **Figure 4.7**.

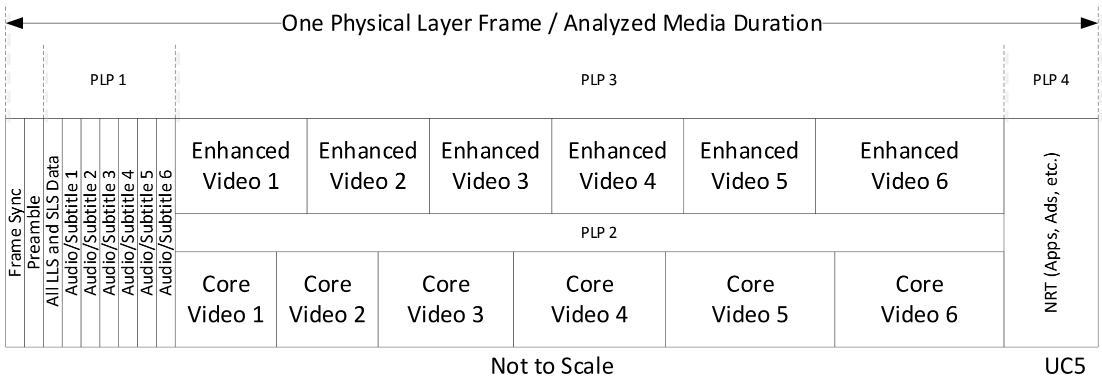


Figure 4.11 Stat mux, layered video, one PHY frame / analyzed media duration.

4.4 Advanced Multiplex Constructions

The examples provided in Section 4.3 above have the Analyzed Media Duration and the Media Segment size that are bound together by the Scheduler. An N second Media Segment has a related physical layer frame or pattern of frames that repeat on an N second cadence. The media required for N seconds of playback is delivered in N seconds on the physical layer. This is a straightforward approach to scheduling, but this is not a requirement for ATSC 3.0.

The Scheduler can have an Analyzed Media Duration that is longer than the delivered Media Segments and resulting physical layer frame(s). This sort of scheme might be used to increase efficiency or provide for multiple Media Segment durations on a single instance of ATSC 3.0. The more media time that is managed by the Scheduler in an Analyzed Media Duration, the more efficient the stat mux can be; more time results in better decorrelation among the Media Segments. There is further discussion of Analyzed Media Duration in A/324 Annex C [3].

4.5 ROUTE Usage

4.5.1 Introduction

ROUTE is a transport protocol for the broadcast transmission of delivery objects associated with ATSC 3.0 Services. A/331 [1] defines various mechanisms and options for the delivery of ATSC 3.0 Service content and Service Layer Signaling to receivers using the ROUTE protocol, as well as the parameters of ROUTE-related Service signaling. This section is intended to provide guidelines and explanations on the use of those methods in the signaling and transport of real-time and non-real time Services and content using ROUTE, with focus on DASH-formatted streaming Services delivery, given the primary interest in linear TV Services delivery during initial deployment of ATSC 3.0 Services.

4.5.2 Streaming Service Delivery

In streaming Service delivery, the source protocol, defined in terms of a source flow (and more formally, by the `SrcFlow` element in the S-TSID), employs ROUTE packets to send delivery objects. Each delivery object carried in the source flow should correspond to an entire DASH Media Segment or a Subsegment. Use of the repair protocol is optional. For example, a typical linear TV Service is often delivered exclusively using the source protocol. A Video-on-Demand (VoD) Service with less stringent playout delay requirement could be delivered by the source protocol in conjunction with the repair protocol, or by the repair protocol exclusively, if all targeted receivers are expected to be AL-FEC capable. Whereas the source protocol/flow uses the File

Mode or Entity Mode (see Sections 4.5.4.1 and 4.5.4.2, respectively), the repair protocol/flow uses the (Unsigned) Package Mode (see Section 4.5.4.3).

In general, it is expected that streaming media delivery objects are formed into DASH Segments with duration of perhaps one to several seconds to ensure fast start-up and low channel change delay. For even faster start-up, the MDE mode as described below in Section 4.5.9.1, and also in Section 4.2 may be used.

Signaling information that enables receiver acquisition of media content of the streaming Service is provided by the combination of the LLS and SLS metadata fragments. The SLT of the LLS identifies, for each ATSC 3.0 Service, the ROUTE session and subordinate LCT channel in which the SLS fragments of that Service are carried. The S-TSID fragment of SLS identifies the one or more ROUTE sessions(s) and, for each of which, the subordinate LCT channel(s) that carry the media components of the parent Service. A given LCT channel, as identified by its TSI value, could be used to transmit a source flow, a repair flow, or a pair of source and repair flows.

4.5.3 NRT Service and Content Delivery

ROUTE delivery of Non-Real Time (NRT) content, whereby the NRT files are targeted for use either directly by the receiver (for example, by the Receiver Media Player (RMP) as defined in A/344 [10]), or by an Application Media Player (AMP) of a broadcaster application, is expected to strictly use the File Mode. In particular, the file/object metadata as represented by the Extended FDT is expected to be embedded within the S-TSID fragment that is transmitted before the NRT content file(s) described by the Extended FDT.

It is recommended that the broadcaster's decision on implementation of the source flow and/or the repair flow for NRT content delivery be based on the expected AL-FEC capability of targeted receivers. The specific guidelines are summarized in **Table 4.1**.

Table 4.1 Recommended Use of Source and/or Repair Protocol for NRT Content Delivery as Function of Expected Receiver Capability to Support AL-FEC

Expected AL-FEC Capability of Receivers	Source Protocol/Flow	Repair Protocol/Flow
Receivers are strictly AL-FEC incapable	YES	NO
Mix of AL-FEC capable and incapable receivers	YES	YES
Receivers are strictly AL-FEC capable	NO	YES

Sending of only the repair protocol/flow without the source protocol flow, when targeted receivers are all expected to be AL-FEC capable, can emulate FLUTE operation.

FEC Object Transmission Information (FEC OTI) can be sent in one of two ways:

1) If the Extended FDT Instance is carried in the S-TSID, then the FEC OTI parameters are defined according to RFC 6330 [16] and are carried in the `RepairFlow@fecOTI` attribute.
2) If the Extended FDT Instance is sent as a separate delivery object (with TOI=0) in the same ROUTE session and LCT channel that carries the delivery object described by the Extended FDT Instance, it is recommended that the ROUTE sender embed the FEC OTI parameters in the Extended FDT Instance in an equivalent manner to that defined in RFC 6726 [18], i.e., placing of FEC OTI parameters in the FDT Instance, as opposed to transmitting those parameters in the ALC-defined LCT Header Extension `EXT_FTI` as specified in RFC 5775 [15]. The reason is to incur lower transmission overhead, since the ALC method requires repetitive sending of the FEC OTI parameters, typically in every ROUTE packet.

Similar to streaming Services delivery, signaling information that enables receiver acquisition of NRT content is provided by the combination of the LLS and SLS metadata fragments. Note that in the case of NRT delivery, the Distribution Window Description (DWD) fragment is mandatory SLS information for providing the broadcast delivery schedule of the NRT content of concern.

4.5.4 Delivery Modes

4.5.4.1 File Mode

In the File Mode of transporting delivery objects, the file/object metadata as represented by the Extended FDT could be either embedded within, or referenced as a separate delivery object by, the S-TSID fragment of the SLS. If it is possible to use either method, the former is generally preferable for reliability reasons. Extended FDT recovery via NRT delivery of Service signaling separately from media content is typically more reliable as compared to inband delivery of EFDT with the media content, since AL-FEC for such recovery, if necessary, is more reliable in the transmission of NRT content files as compared to the transmission of EFDT Instances. The reason being the typically much larger size of an NRT content file as compared to that of an EFDT Instance, and correspondingly, greater time diversity achievable for more robust AL-FEC recovery of the former over the latter file type.

In addition, use of Entity Mode for NRT content delivery may also be used in place of the File Mode when the latter is configured for sending of the Extended FDT as a separate delivery object from the delivery object it describes – see additional information in Section 4.5.4.2.

4.5.4.2 Entity Mode

The Entity Mode should be used for the transport of a DASH-formatted streaming Service when it is not possible for the broadcast system to embed all necessary delivery object metadata, in the form of an Extended FDT, within the S-TSID fragment. For example, the values of certain parameters might not be known or definable in a way to enable their derivation at the receiver, in advance of the Service delivery via broadcast. Furthermore, the broadcaster can rely on the greater reliability of recovery of such dynamic object metadata achievable by sending the metadata along with the delivery object as a single, compound object via the Entity Mode, as opposed to sending the metadata as a separate delivery object in the same LCT channel that carries the media content described by that dynamic metadata. It is expected that `@fileTemplate` in the Extended FDT will continue to be sent as static metadata to allow derivation of the `@Content-Location` from the TOI.

As indicated in A/331 [1], A.3.3.3, the object metadata will be conveyed by one or more entity headers which correspond to one or more of the representation header fields, payload header fields and response header fields as defined in Sections 3.1, 3.3 and 7, respectively, of RFC 7231 [19]. Those entity headers may appear either before or after the delivery object in ROUTE transmission. Operation in the latter case is similar to HTTP chunked transfer coding whereby metadata can be sent as trailer fields at the end of the message, i.e., after the message body.

As mentioned in Section 4.5.4.1, for NRT content delivery, use of the Entity Mode might be preferred over File Mode operation where the Extended FDT is sent as a separate delivery object (with TOI=0) in the same LCT channel as the delivery object described by the Extended FDT. The reason is the more reliable recovery of the Extended FDT since it is transported together with the entity payload as a compound object much larger in size than Extended FDT itself. Therefore, and similar to the sending the Extended FDT in the S-TSID, a larger delivery object enables greater time diversity to be achieved in the application of the AL-FEC code, which leads to more robust AL-FEC recovery. A caveat for choosing the Entity Mode over File Mode for NRT file delivery is that AL-FEC is employed in the expectation that its use is beneficial for the associated reception environment (e.g., by mobile devices).

4.5.4.3 Package Mode

As described in A/331 [1], the use of Package Mode allows the bundling of multiple delivery objects in a single multipart MIME structure for ROUTE delivery. Two Package Modes are defined: Unsigned Package Mode and Signed Package Mode. When the constituent delivery objects of the bundle are either SLS fragments, or application-related files of an HTML entry package (as described by the HELD), Signed Package Mode delivery must be used.

It should be noted that in the ROUTE repair protocol, multiple delivery objects, possibly originating from different source flows, are reformatted as FEC Transport Objects that are in turn combined to form a FEC Super-Object. Such combining of FEC Transport Objects into a FEC Super-Object has functional similarity to the bundling of delivery objects in Package Mode operation, from the perspective of creating a larger-sized compound object which allows more reliable reception when AL-FEC is used.

4.5.5 Extended FDT Usage

The Extended FDT as defined in A/331 [1] comprises the FLUTE FDT specified in RFC 6726 [18], extensions defined by 3GPP and specified in ETSI TS 126 346 [11], and the ATSC-defined extensions in A/331. As previously described, the Extended FDT could be either embedded in the S-TSID to describe the delivery objects carried by the associated source flow, or sent as a unique delivery object with TOI=0 on the same LCT channel which carries the delivery object described by the Extended FDT. Among the FDT extension parameters defined by ATSC and 3GPP, some additional explanations are provided on the following ones.

4.5.5.1 ATSC-defined FDT Extensions

`@maxExpiresDelta` – This parameter, when present, is intended for use by the receiver to determine the expiration time of an Extended FDT Instance (as given by the sum of the value of this attribute and the wall clock time at the receiver when the receiver acquires the first ROUTE packet carrying data of the delivery object described by this Extended FDT Instance). When this attribute is present, the derived expiration of the Extended FDT Instance will take precedence over the value given by **`FDT-Instance`**`@Expires`. In addition, note that according to A/331 [1], at the same time of Extended FDT Instance expiration as derived from `@maxExpiresDelta`, the ROUTE sender should cease transmission of data for the corresponding delivery object. The reason is that since the receiver is expected to ignore any additional incoming data for the delivery object whose Extended FDT Instance has expired, transmitting such data would be simply a waste of transmitter and RF resources.

`@maxTransportSize` – This parameter indicates the maximum size of any of the delivery objects corresponding to DASH Media Segments described by the parent Extended FDT Instance. It should be present in the Extended FDT Instance represented by the **`EFDT`** element in the S-TSID metadata fragment that corresponds to the source flow whose delivery objects are described by this Extended FDT Instance. It should be used solely in conjunction with the delivery of DASH-formatted media content whereby the **`EFDT.FDT-Instance.File`** element should not be present for any of the Media Segments described by this Extended FDT Instance, and as a consequence, the **`File`**`@Transfer-Length` attribute is not available to describe the size of any of the Media Segments. `@maxTransportSize` might be used by the receiver to allocate the necessary buffer space for the recovery of the entire set of delivery objects described by this Extended FDT Instance.

The `@appContextIdList` attribute, in conjunction with `@filterCodes`, is mainly intended to be included in the Extended FDT Instance for the delivery of application-related files, for example Broadcaster Application entry pages and associated multimedia files of application enhancements for linear TV Services. As described in A/331 [1] under the HELD and DWD metadata fragments, these parameters identify application resources intended to be stored in corresponding Application Context Caches in the receiver, for subsequent retrieval by the Broadcaster Application via the Receiver Web Server.

`@fileTemplate` – It should be emphasized that this FDT extension enables a compact Extended FDT Instance to be embedded in the S-TSID for the description of DASH Segments carried in source flows. Specifically, substitution by the TOI value, present in the header of ROUTE packets carrying Media Segments as delivery objects, for the `$TOI$` pattern in the URI string conveyed by this attribute allows receiver derivation of the `Content-Location` attribute, for the delivery object described by this Extended FDT Instance. In doing so, it avoids the need for the Extended FDT Instance, under the **`FDT-Instance`** element, to include any **`File`** child elements for the Media Segments of the DASH-formatted media stream delivered via ROUTE, with the exception of a single **`File`** instance associated with the Initialization Segment.

4.5.5.2 3GPP-defined FDT Extensions

A/331 [1] specifies by reference to ETSI TS 126 346 [11] a set of FDT extensions defined by 3GPP for MBMS. One or more of these parameters are intended for inclusion in the Extended FDT to support receiver operation of HTTP File Repair, over the broadband network, to recover lost data during broadcast reception of file content. See Section 8.3.3 of A/331 [1] on the definitions of the following 3GPP-defined FDT extensions, and Section 4.5.7 in this document for more information on HTTP File Repair. The list of these parameters is as follows:

- **`Base-URL-1`**,
- **`Base-URL-2`**,
- **`Alternate-Content-Location-1`**,
- **`Alternate-Content-Location-1@Availability-Time`**,
- **`Alternate-Content-Location-1.Alternate-Content-Location`**,
- **`Alternate-Content-Location-2`**,
- **`Alternate-Content-Location-2@Availability-Time`**,
- **`Alternate-Content-Location-2.Alternate-Content-Location`**.

If the broadcaster intends to support file repair, the broadcaster should deploy at least one HTTP server as the (primary) file repair server whose location is given by the URI expressed by **`Alternate-Content-Location-1.Alternate-Content-Location`**. In the event that the value of this element is a relative URI, the **`Base-URL-1`** element must be included in providing a Base URI for resolving the relative reference. If the broadcaster wishes to additionally deploy a back-up file repair server, its location should be given by **`Alternate-Content-Location-2.Alternate-Content-Location`**, and similar to the previous example, include a Base URI via **`Base-URL-1`** should **`Alternate-Content-Location-2.Alternate-Content-Location`** represent a relative URI.

4.5.6 AL-FEC Usage

4.5.6.1 General

As previously described in Sections 4.5.2 and 4.5.3, whether or not a broadcaster chooses to employ AL-FEC as a means to enhance the reliability of Service/content reception and associated quality of experience for the end user is dependent on the type of Service (e.g., Linear TV, VoD or NRT-related) as well as the expected AL-FEC capability of targeted receivers. Regarding the Service type, as explained in those sections, a linear TV Service, with stringent requirement on very-low playout delay, can be delivered without AL-FEC. On the other hand, a VoD Service, with less stringent playout delay requirement could be delivered by the (null AL-FEC-based) source protocol in conjunction with AL-FEC-based repair protocol. Delivery of an NRT Service/content item with typically lax playout requirement can also employ AL-FEC. With regards to receiver capability to process AL-FEC, the broadcaster implementations should generally abide by the guidelines as shown in **Table 4.1**.

4.5.6.2 Source Protocol

The source protocol, used for source flow delivery, does not employ a "real" (i.e. non-null) AL-FEC scheme or code. For example, A/331 [1] in A.3.5.1 and A.3.8 indicates that source delivery is considered a special case of the use of the Compact No-Code Scheme associated with FEC Encoding ID = 0 in which the encoding symbol size is exactly one byte, and the FEC Payload ID field conveys a 32-bit `start_offset` value. The latter corresponds to the byte number, of an N-byte delivery object, represented by the first byte of the payload portion of the corresponding ROUTE packet in which the entire or fractional part of the delivery object is carried. Such nomenclature is used because the stated principle of ROUTE source and repair protocol operation is based on FECFRAME mechanisms as defined in RFC 6363 [17].

The source flow (delivered using the source protocol) must always be transmitted when the broadcaster expects the targeted receivers to be AL-FEC incapable, or that only a portion of those receivers are AL-FEC capable. Furthermore, the source flow must be transmitted for a linear TV Service, and is recommended to be transmitted for a VoD Service, due to the requirement for relatively low start-up delay for those Services (e.g., a few seconds for linear TV and perhaps 10-20 sec for VoD). For these types of Services, the repair protocol/flow is optional to employ since the incurred latency of AL-FEC decoding may be deemed to be too high.

On the other hand, for the delivery of NRT Services or NRT content associated with data Services (for example, the ESG Service, the EAS or the DRM Data Service), and if the broadcaster expects all of the targeted receivers to be AL-FEC capable (e.g., a Service specifically targeting mobile ATSC 3.0 receiver devices), the source protocol may be optional to use – i.e., only the repair protocol is used, as previously alluded to in Section 4.5.3, and further discussed in the next section.

The usage of ALC (RFC 5775 [15]) and LCT (RFC 5651 [14]) with regards to existing ALC and LCT headers and LCT header extensions is described in A.3.4 and A.3.6 of A/331 [1]. The construction of ROUTE packets which carry delivery objects of the source flow is described in A.3.5. Basic ROUTE sender and receiver operations are described in A.3.8 and A.3.9, respectively.

It should be noted that the following ATSC-defined extensions to the FLUTE FDT, **`FDT-Instance`**`@maxExpiresDelta` and **`FDT-Instance`**`@maxTransportSize` are designed for ROUTE delivery of DASH streaming content. In such scenario, the latest permitted (i.e., "expiry") wall clock times for transmission of any given delivery object, or Media Segment, in a streamed sequence will be expressed by the sum of the `@maxExpiresDelta` value and the expected wall clock time of arrival of the first ROUTE packet carrying data of that delivery object at the receiver.

4.5.6.3 Repair Protocol

The repair protocol, used for repair flow delivery, employs the RaptorQ AL-FEC scheme as defined in RFC 6330 [16]. It is optional to use in ROUTE. As previously described in Sections 4.5.3, 4.5.6.1 and 4.5.6.2, a broadcaster's decision on whether or not to use the repair protocol depends on the type of Service (e.g., fixed or mobile reception, associated start-up latency requirements) and the expected AL-FEC capability of targeted receivers.

4.5.6.3.1 (AL-)FEC Transport and Super-Object Construction

In **Figure 4.12**, a copy of Figure A.4.1 in A/331 [1], is useful in depicting the generation of ROUTE packets delivered using the repair protocol.

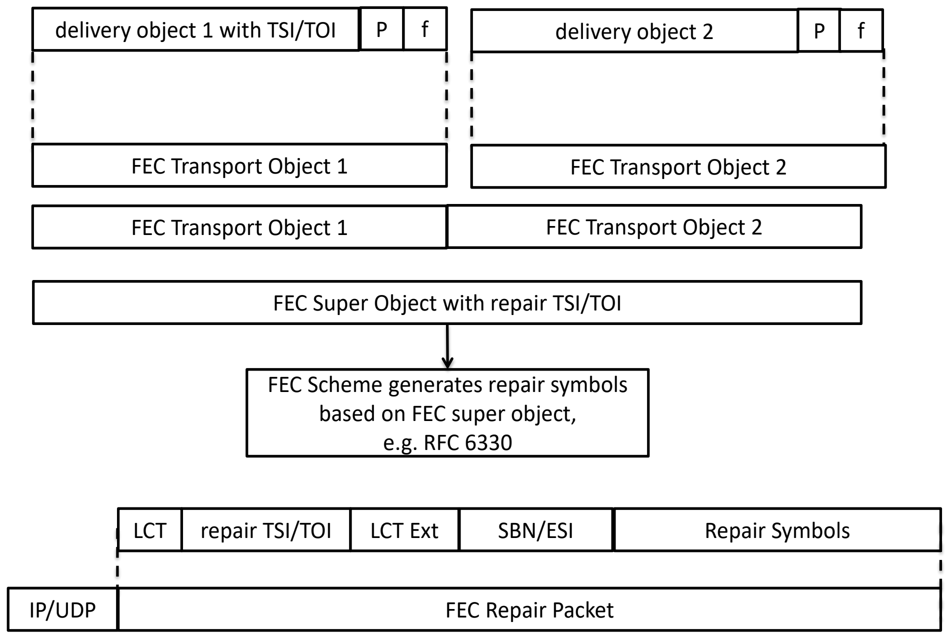


Figure 4.12 AL-FEC packet generation.

The process makes use of the delivery objects carried on the source flow, in the following sequential order:

1) Formation of a FEC Transport Object from a FEC object, the latter of which in this example is identical to a delivery object with associated TSI and TOI. This is shown in **Figure 4.13** where the FEC object is appended by ***P***- padding octets, followed by a 4-octet field ***f*** whose value (F) denotes in octets the size of the FEC object. The size of the subsequent FEC Transport Object in whole symbols is given by $\boldsymbol{S}$=ceil[(F+4)/Y], where the ceil[] function rounds the fractional number (F+4)/Y to the next highest integer value, and Y is the FEC repair symbol size in octets.

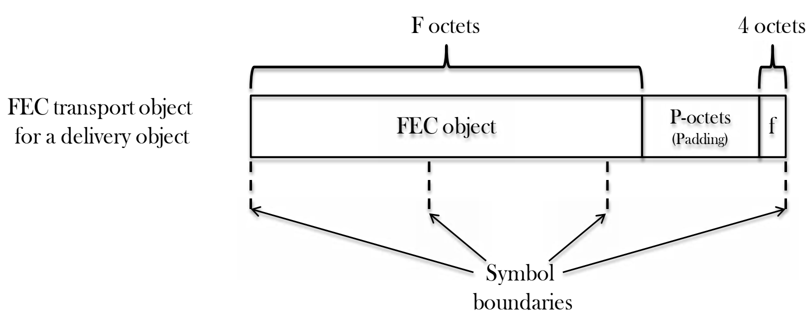


Figure 4.13 FEC Transport Object formation.

2) Concatenation of one or more FEC Transport Objects to form a FEC Super-Object, identified by its TOI along with the TSI of the LCT channel delivering the corresponding repair data for the FEC Super-Object. The constituent FEC Transport Objects of a FEC Super-Object could originate from the same source flow (e.g., carrying a video stream) or from different source flows (e.g., one carrying a video stream and another carrying an audio stream). The purpose of aggregating multiple FEC Transport Objects into a larger FEC Super-Object is to increase the size of a FEC-protected transport object to obtain greater time diversity in the transmission of the resulting repair symbols on the repair flow, therefore enhancing the robustness of FEC decoding at the receiver.
3) Forwarding of the FEC Super-Object to the RaptorQ (RFC 6330 [16]) FEC encoder (referred to in **Figure 4.12** as the "FEC Scheme"), which in turn produces repair symbols as the payload of ROUTE repair packets. Those ROUTE packets carrying the repair symbols of a given [FEC Super-Object]$_j$ will contain the same TOI_j in its LCT header. A subsequent ROUTE repair packet with TOI_k is indication that repair symbols of a different [FEC Super-Object]$_k$ are carried in that packet.

The repair packets are broadcast using the repair protocol. As indicated in A.4.2.4 of A/331 [1], the repair protocol is based on ALC and LCT as defined in RFC 5775 [15] and RFC 5651 [14], respectively. The TSI field in the LCT packet header identifies the repair flow in which the repair packet is delivered, and the first bit of the Protocol Specific Indication (`PSI` bit `X`), the Source Packet Indicator (`SPI`), is set to '0' to indicate a repair packet. The AL-FEC Scheme is as defined in RFC 6330 [16], and whereby only repair packets are transmitted.

4.5.6.3.2 AL-FEC Information Provided to Receivers

According to A/331 [1], the following AL-FEC related information needs to be communicated to the receiver, via a combination of the contents of the **`RepairFlow`** element in the S-TSID, and parameters conveyed in the LCT header and header extensions:

- The FEC configuration consisting of
  - FEC Object Transmission Information (OTI) per RFC 5052 [13].
  - Additional FEC information as indicated in Table A.4.1 of A/331 [1].
- The total number of FEC objects included in the FEC Super-Object, N.
- For each FEC Transport Object,
  - TSI and TOI of the delivery object used to generate the FEC object associated with the FEC Transport Object,
  - Start octet within the delivery object of the associated FEC object, if applicable, and
  - The size in symbols of the FEC Transport Object, S.

4.5.7 HTTP File Repair

The HTTP-based File Repair procedure, as described in Section 8.3 of A/331 [1], enables the receiver to acquire missing data in the broadcast reception of delivery objects. Such loss of broadcast data reception could be due to the receiver not being AL-FEC capable to process the repair flow associated with the source flow, or even if it is AL-FEC capable and ROUTE repair flow was sent along with the source flow, excessive reception errors occurred which resulted in the inability to recover the entire delivery object.

As previously discussed in Sections 4.5.2 and 4.5.3, the choice by the broadcaster to employ AL-FEC in ROUTE, i.e., use of the repair protocol/flow, typically depends on the type of Service and the broadcaster's expectation on AL-FEC capability of the targeted receiver population. For example, as shown in **Table 4.1**, the repair protocol/flow might only be used if at least a portion of targeted receivers are AL-FEC capable. Under the assumption that either a portion or the entirety of targeted receivers are AL-FEC capable, the following methods for transmission of the source and/or repair flows are recommended.

4.5.7.1 Transmission of both Source and Repair Flows

Figure 4.14 illustrates the proposed methodology for the sending of Service content via source and repair data to receivers of which the broadcaster expects some to be AL-FEC capable and others are not.

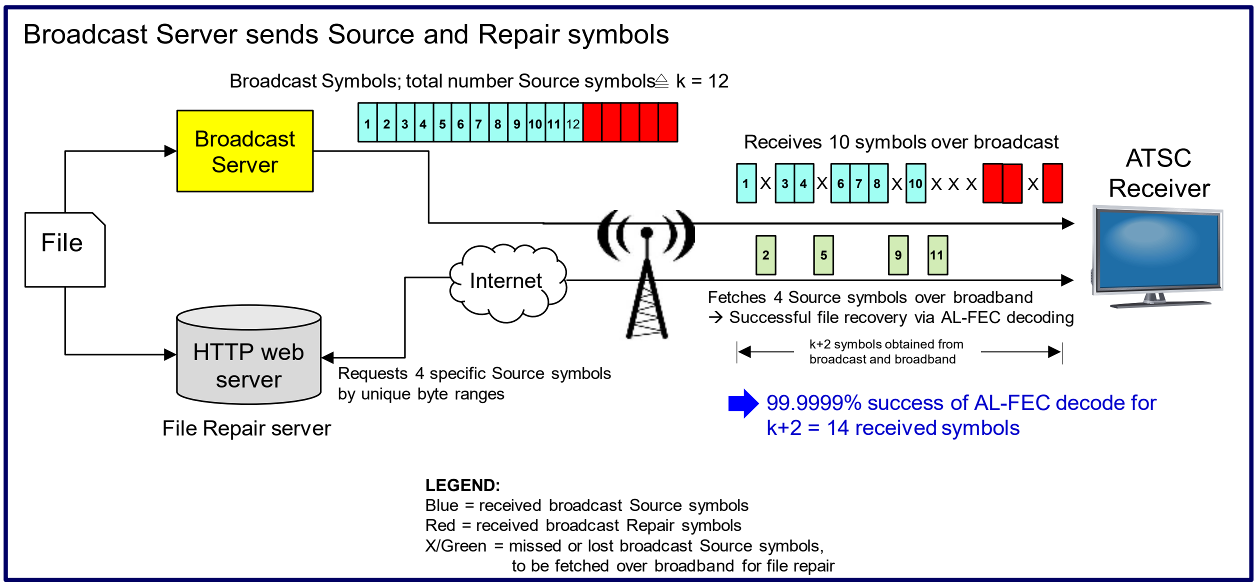


Figure 4.14 Transmission of source followed by repair data.

A receiver which is not AL-FEC capable can determine whether the entire delivery object/file is successfully received. If not, it can request the missing data via one or more byte-range requests to the file repair server implemented by a standard HTTP/web server. A receiver which is AL-FEC capable would similarly first determine whether the delivery object/file is successfully recovered from the source flow. If not, the receiver would additionally acquire repair data sent on the repair flow. In the event that the combination of source and repair data received over broadcast is insufficient to recover the original file, the receiver can request the missing data via one or more byte-range requests to the file repair server.

4.5.7.2 Transmission of Repair Flow Only

Figure 4.15 illustrates the proposed methodology for the sending of Service content via strictly the repair flow to receivers for which the broadcaster expects all receivers to be AL-FEC capable.

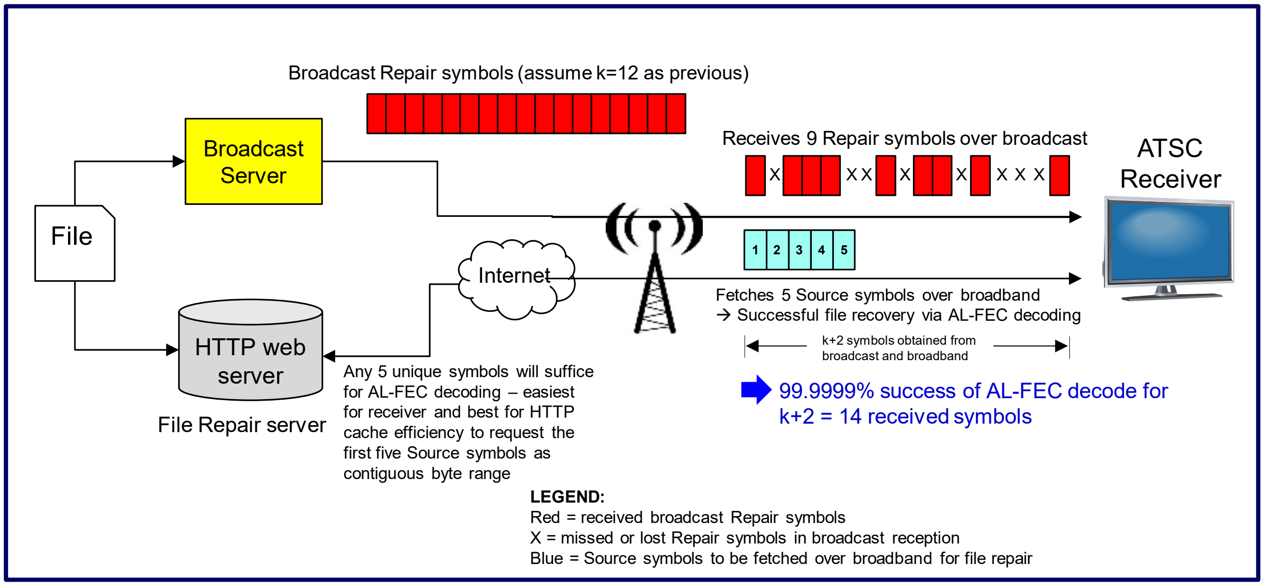


Figure 4.15 Transmission of strictly repair data.

An AL-FEC capable receiver first determines whether the delivery object/file can be successfully recovered from the data received on the repair flow. If not, the receiver can compute the number of additional FEC symbols required to ensure full file recovery and translate that information to a contiguous byte range from which it can request from the file repair server, starting from the beginning of the source file.

4.5.8 Service Signaling

Service signaling associated with ROUTE delivery of Services comprises LLS and SLS. The functionality of Service signaling should be mostly evident from the detailed descriptions of the parameters in the metadata fragments or tables of the LLS and SLS in A/331 [1]. LLS information is delivered directly over UDP/IP, and among its functions, the SLT identifies the ROUTE session in which the SLS data is delivered. Such "bootstrapping" of SLS discovery is described in Section 4.5.8.1. The mandatory and optional SLS fragments associated with ROUTE-based Service delivery are described in Section 4.5.8.2.

4.5.8.1 LLS

The primary aspects of the LLS from the ROUTE perspective is the SLT and announcement of ROUTE as the delivery protocol for the SLS information associated with the Service delivered by the ROUTE protocol. This is indicated by the value of the `SLT.Service.BroadcastSvcSignaling@slsProtocol` attribute, which must be set to "1". The identity of the ROUTE session in which the SLS fragments are delivered is given by the triplet of attributes [`@slsDestinationIpAddress`, `@slsDesytinationUdpPort`, `@slsSourceIpAddress`] of the `SLT.Service.BroadcastSvcSignaling` element. The LCT channel of the ROUTE session delivering SLS fragments must have its TSI value set to "0". Note that the content components of the Service described by its SLS information could be delivered on the same LCT channel delivering the SLS, or on different LCT channel(s) – i.e., with TSI ≠ "0".

4.5.8.2 SLS

The SLS fragments as shown in **Table 4.2** are mandatory to be transmitted via ROUTE by the broadcaster to enable reception of ATSC 3.0 Services, depending on the type/nature of the Service. Other types of SLS fragments may be additionally transmitted, as indicated in the "Note" column.

Table 4.2 Required and Optional SLS Fragments Depending on the Service Type

Service Type	Mandatory SLS Fragments	Optional SLS Fragments
Linear A/V or audio-only Service	USBD, S-TSID, MPD	APD, RSAT
Linear A/V or audio-only Service with app-based feature(s)	USBD, S-TSID, MPD, HELD and DWD	"
App-based Service	USBD, S-TSID, HELD	"
ESG Service	"	APD
EAS Service	"	APD
DRM Data Service	"	APD

4.5.9 Fast Start-up and Channel Change Mechanisms

4.5.9.1 MDE delivery

Media Delivery Event (MDE) mode enables chunked delivery of media content to the DASH client in support of earlier start-up of playback than whole Segment based playback. A fundamental principle of MDE mode is that delivery of a Segment to the DASH client can be completed during its playback of already-received media. **Figure 4.16** illustrates the assumed model of ATSC baseband delivery i.e. baseband packets in to baseband packets out with a knowable delay.

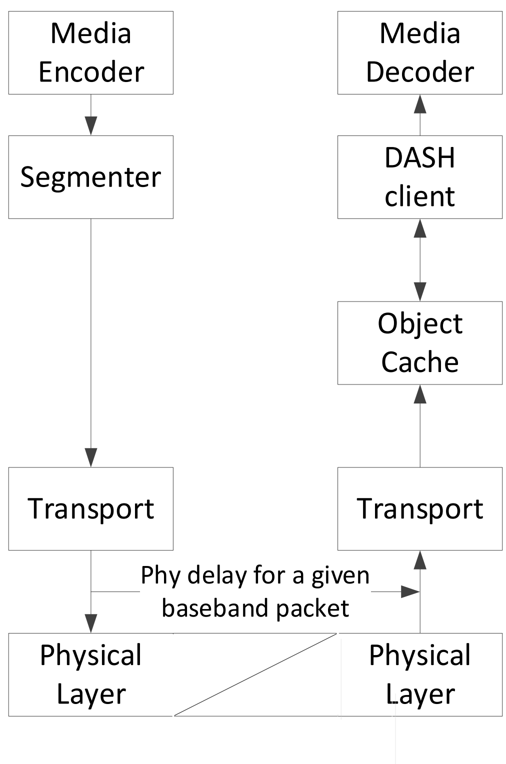


Figure 4.16 Baseband delivery model.

A key feature that enables MDE delivery is an implicit ROUTE (MDE Mode) timer function. Associated with the Object Cache as shown in **Figure 4.16** is a timer that defines the hold time required between a) the receipt of the first packet containing the start of a Media Segment associated with a SAP, and b) the release of the first byte range of that sent Media Segment to the DASH client upon a Segment request. The timing relationship of MDE Mode media delivery is depicted in **Figure 4.17**. LCT packets are marked with the Sender Current Time (SCT) and EXT_ROUTE_PRESENTATION_TIME, when the packet contains or immediately follows a Stream Access Point (SAP). SAP in this context complies with the ISO/IEC 23009-1 [5] definition: i.e., a position in a Representation from which playback is possible by making sole use of the media stream data from that position onward.

The receipt by a DASH client of the corresponding MPD and IS for this media stream is required before playback can start. The MPD is available from the SLS, while delivery of the IS immediately precedes the delivery of the Media Segment and should be made available to the DASH client from the ROUTE cache (Object Cache in the above diagram) well before trailing Media Segment data can be accessed by the DASH client. Doing so enables the DASH client to obtain the necessary initializing data before it tries to play the media data.

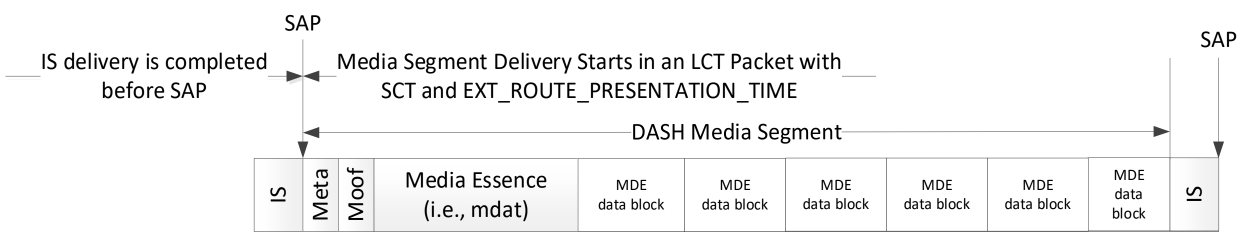
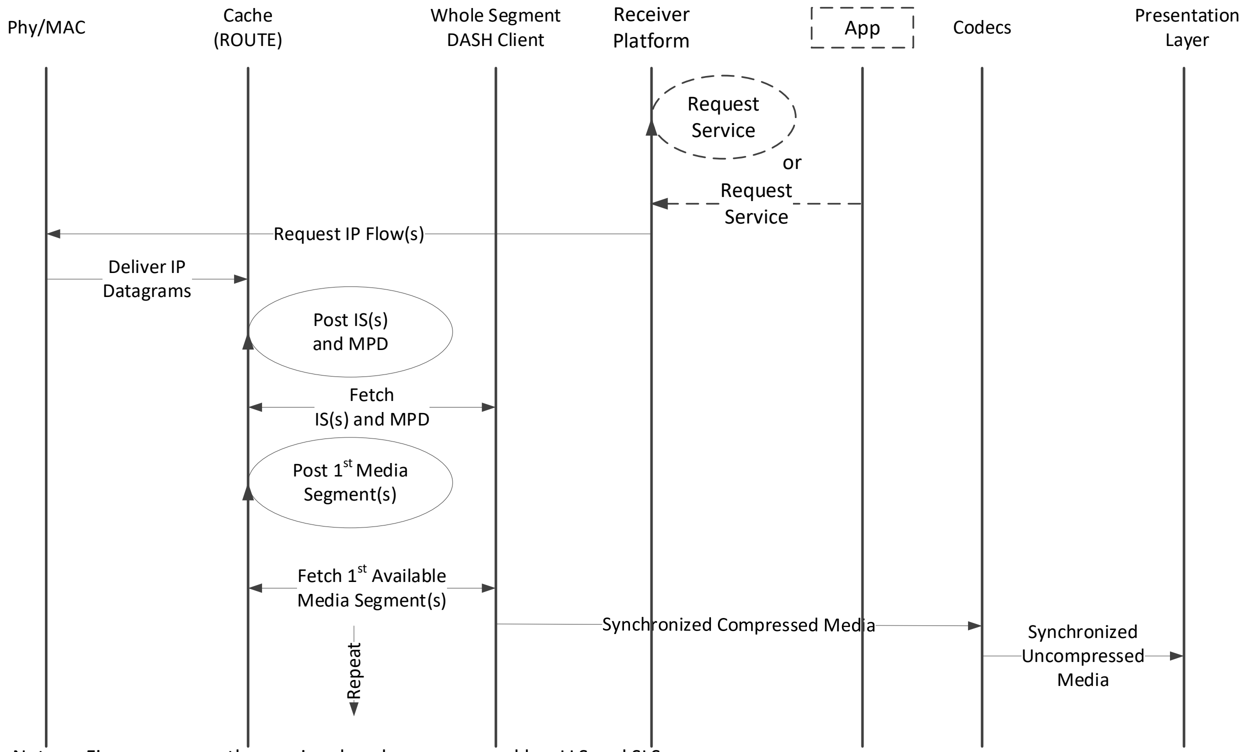


Figure 4.17 Relationship among IS, SAP, and Media Segment.

The difference between the values of EXT_ROUTE_PRESENTATION_TIME and SCT represents the delay time for the start of MDE playback. This delay time is defined by the characteristics of the Scheduler. The Scheduler will operate such that delivery of the final MDE data block of the Media Segment to the DASH client is guaranteed to occur prior to the time that the entire Segment is provided to the DASH client in normal, whole Segment based playback mode. The setting of this delay in the ROUTE receiver may be accomplished via trial-and-error, i.e., increasing the ROUTE timer delay value until no stalls are observed upon start of playback. Alternatively, the Scheduler may operate according to a hard constraint that ensures, for example, (N – A) frames of an N-media frame Segment will have been sent to the receiver and are available in the ROUTE cache for access by the DASH client, prior to the transmission by the ROUTE sender of the remaining A media frames, such that the cache will never under-run. The outcome of the trial-and-error method for setting the ROUTE timer will converge to the same requirement on Scheduler operation.

Figure 4.18 depicts a call flow for DASH client with whole Segment based playback. The DASH client requests the next desired Segment *at or after* its availability start time. The DASH client plays media upon receipt of the whole segment.



Notes: Figure assumes the receiver has done a scan and has LLS and SLS
Broadcaster App may or may not be running

Figure 4.18 Whole segment call flow on receiver.

Figure 4.19 depicts a call flow for DASH client with MDE mode support. The DASH client requests the next desired Segment *before* its availability start time. The ROUTE receiver, on behalf of the cache, responds with Accepted (202). The ROUTE receiver delivers the Segment as byte ranges commencing with the expiration of the ROUTE timer. The DASH client plays media from that time onward on a Segment duration by Segment duration basis.

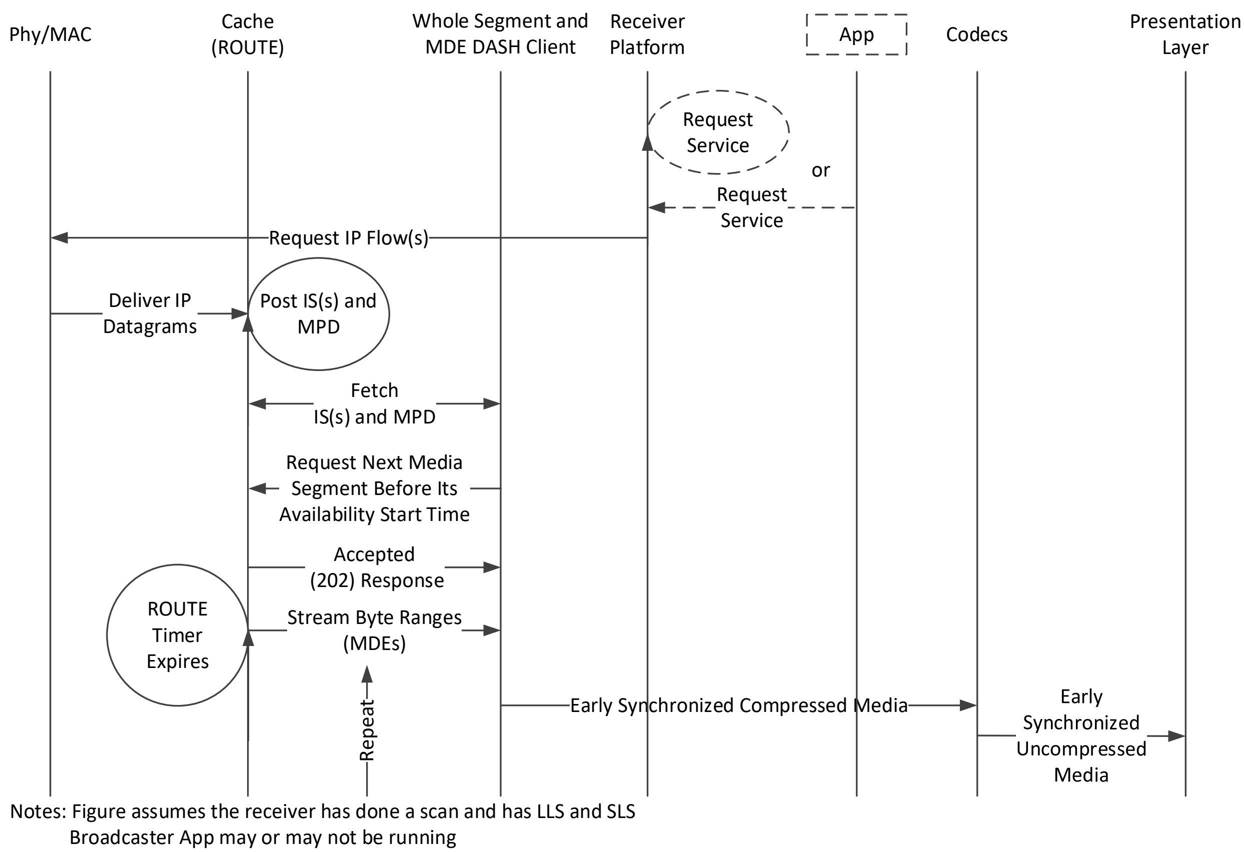


Figure 4.19 MDE delivery call flow.

The MDE-capable DASH client functions in an identical manner as to the whole Segment based client, when fetching whole Segments. ROUTE caches that do not support MDE mode will return a 400 Bad Request error message to an early request by an MDE-capable DASH client.

4.5.9.2 MPD-less Playback

MPD-less media playback is a feature that enables low-latency play-out of a DASH streaming Service without acquisition by the receiver of an MPD, and correspondingly, allows bypassing of the DASH client for initiating and controlling media presentation. Alternatively, MPD-less playback can be combined with normal MPD-based playback for which initial playout is performed without the MPD, and later on, when the MPD becomes available, it can be used to enable selection among richer content offerings and alternatives for presentation to the user. The main benefit of MPD-less playback is to reduce or eliminate altogether system overhead associated with broadcast MPD delivery as part of the SLS. In the alternative use of MPD-less playback as mentioned above, whereby a combination of MPD-less followed by MPD-full playback is employed, it allows fast Service start-up or channel change without the typical need for the MPD to be sent in every system RAP as well as in every broadcast media component associated with the DASH Media Presentation.

The basic operational principle of MPD-less media playback vs. regular MPD-enabled media playback is illustrated in **Figure 4.20**

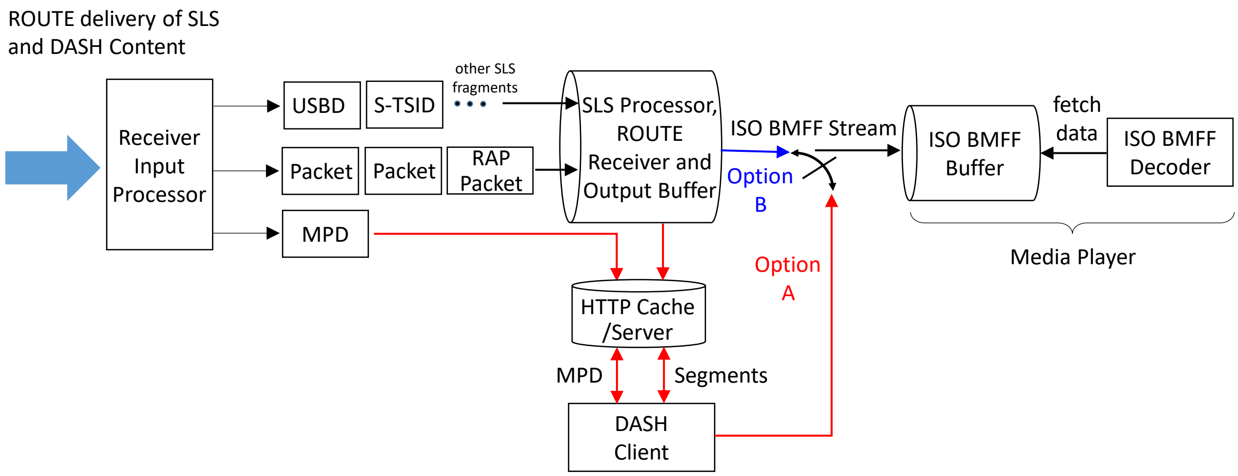


Figure 4.20 MPD-less vs. MPD-based playback.

As shown in the above diagram and by the colored arrows/text, there are two options for delivering the DASH-based ISO BMFF media streams to the media player in the receiver, and these options can be employed either independently or in a combined fashion. Option A is the typical or normal implementation of MPD and DASH Segments delivery to a DASH client which controls the acquisition of the desired DASH Representation(s) according to the Segment availability timeline, with playout based on the media presentation timeline. Option B depicts MPD-less playback operation whereby SLS fragments minus the MPD are sent over ROUTE and delivered to the ROUTE receiver (which in the above reference model is assumed to include SLS processing functionality). Essential additional information is included in the S-TSID to enable the ROUTE receiver to select the DASH Representation to be forwarded to the media player for subsequent playout, without the MPD and involvement of the DASH client. That additional S-TSID information can be considered a minimal but crucial subset of MPD information that permits bootstrap of media playout. Option B can be combined with Option A such that after the initial playback has begun, after subsequent receiver acquisition and forwarding of the MPD to the DASH client, standard DASH client-controlled playback can ensue.

The additional information carried in the S-TSID are provided in the **MediaInfo** element in S-TSID, namely the **S-TSID.RS.LS.SrcFlow.ContentInfo.MediaInfo** element. A snippet of the tabular representation of this element in the S-TSID, as copied from A/331 [1] and highlighting the **MediaInfo** data structure, is shown in **Table 4.3**

Table 4.3 S-TSID Metadata in Support of MPD-less Playback

Element or Attribute Name	Use	Data Type	Description
SrcFlow		srcFlowType	Source flow carried in the LCT channel.
@rt	0..1	boolean	Indication of whether the source flow conveys real-time content.
@minBuffSize	0..1	unsignedInt	The minimum number of kilobytes required in the receiver transport buffer for the LCT channel.
EFDT	0..1		The extended FDT instance.
FDT-Instance	1	fdt: FDT-InstanceType	A FLUTE FDT per RFC 6726 **Error! Reference source not found.** with ATSC- and 3GPP-defined extensions.
ContentInfo	0..1		Additional information that can be mapped to the application Service that is carried in this transport session (i.e., LCT channel).
<choice>			
MediaInfo	0..1		DASH Representation
@startUp	0..1	boolean	A Boolean flag, used for default “MPD-less startup” operation, that provides indication on whether the DASH resource carried by this LCT channel should be delivered to the media rendering application for decoding and rendering.
@lang	0..1	lang	The audio language of the DASH resource delivered by this LCT channel.
@contentType	0..1	contentType	The media type of the DASH resource delivered by this LCT channel.
@repId	1	StringNoWhitespace	Representation ID of the DASH Representation delivered by this LCT channel.
ContentRating	0..N		Content rating information associated with the program to which the media content of this source flow pertains.
@schemeIdUri	0..1	anyURI	Content advisory rating scheme associated with the program to which the media content of this source flow pertains.
@value	1	string	Content advisory rating value associated with the program to which the media content of this source flow pertains.
AEAMedia	0..1		Container of identifiers of AEA messages.
AEAId	1..N	string	Identifier of an AEA message to which the AEA media files carried in this LCT channel are associated.
Payload	0..N		Information on the payload of ROUTE packets carrying the objects of the source flow.
@codePoint	1	unsignedByte	A numerical representation of the combination of values specified for the attributes of this **Payload** element.
@formatId	1	unsignedByte	The payload format of the delivery object.
@frag	0..1	unsignedByte	Indication of how the payload of ROUTE packets carrying the objects of the source flow are fragmented for delivery.
@order	0..1	boolean	Indication of how the payload of ROUTE packets carrying the objects, or a portion thereof, of the source flow as DASH Segments are delivered relative to the order of their generation by the DASH encoder.
@srcFecPayloadId	0..1	unsignedByte	The implied meaning and representation of the ROUTE packet header “FEC Payload ID” in the packets carried by this source flow.

As shown in the table, the parameters under `MediaInfo` enable selection and filtering logic in the ROUTE receiver to determine, for example, whether the DASH resource carried by the parent LCT channel and source flow is appropriate to be directly forwarded to the media player for decoding and playout, as well as the media type, the Representation identifier, audio language, and content rating associated with the program to which the media content of this DASH resource belongs. For example, MPD-less playback of a DASH resource for which the `MediaInfo@startUp` attribute is set to 'true' may be prohibited for rendering if its associated content rating exceeds the maximum permissible value set by the user.

4.6 MMT Usage

4.6.1 Introduction

MMTP can be used for delivering broadcast Services. For hybrid delivery, MMTP signaling can point to additional content, such as that delivered by broadband, that is provided outside of the MMT protocol.

The following describes various aspects of MMTP delivery and includes various recommendations.

4.6.2 Low-Delay MPU Streaming

The MMT protocol is designed to operate in a very low latency mode. The MPU mode of the MMT protocol utilizes three types of delivery units: MPU metadata, fragment metadata and MFU (actual media data). Examining the MMTP payload header, the receiver can identify which type of delivery unit is being used.

Similar to an RTP stream, an MPU stream can be sent progressively as soon as the media data becomes available as described in Sections 5.3 and 5.5 of ISO/IEC 23008-13 [12]. MPU metadata can be generated and delivered before media encoding is started. Then the encoded media data is encapsulated into MMTP packets as available. Fragment metadata can be constructed progressively and completed when all media data for a movie fragment is available.

4.6.3 Buffer Model and Synchronization

With MMT, decoding can start as soon as the client receives a packet containing the beginning portion of an intra-coded video frame. In order to enable such packet-based operation without causing buffer overflow and underflow, it is essential to use a precise buffer model, like the MPEG-2 T-STD model, which allows the sender to model decoder buffer status at each packet boundary taking into account packet-level multiplexing of media data.

MMT provides explicit and precise modeling of delivery delay and forwarding of payload data to the decoder buffer similar to the MPEG-2 T-STD model by using a Hypothetical Receiver Buffer Model (HRBM). The HRBM model makes it possible for broadcasters to model the time at which each packet is received by the client and the time at which the payload data of each packet is entered in/removed from the decoder buffer. This enables a broadcaster server to model the behavior of a receiver's buffers to ensure that any processing the receiver performs on the packet stream is within the reception constraints of the receiver.

The HRBM can model the delay until an access unit (AU) of media data is delivered to the decoder buffer. **Figure 4.21** shows the components of delay in the HRBM. A1 corresponds to the time required to receive an AU sent by an MMTP server and depends on physical layer parameters. A2 needs to be set large enough to correct any transmission error. A3 can be adjusted to result in the desired amount of total broadcast delivery delay.

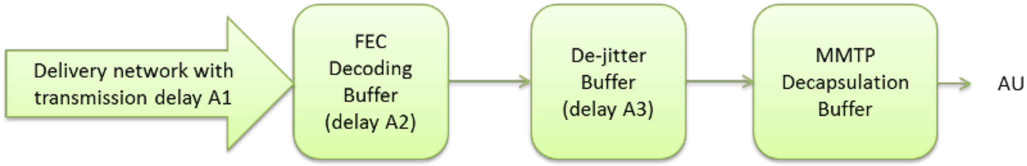


Figure 4.21 Delay in HRBM.

4.6.4 Service Signaling

MPEG Media Transport Protocol (MMTP) is used to deliver content in the form of MPUs as well as the MMT Signaling Components that describe the Streaming Service Components (MPU). This, in addition to the Bootstrap Signaling via the Service List Table (SLT), describes the Service.

The Low Level Signaling (LLS) can include the SLT as one of its instance types. The SLT includes the signaling for one or more Services. Each Service contains zero or one **BroadcastSvcSignaling** instance and each **BroadcastSvcSignaling** instance includes the @slsProtocol attribute, which can indicate MMTP as the protocol of the Service. The @slsDestinationIpAddress and the @slsDestinationUdpPort attributes point to the location of the Service Layer Signaling (SLS).

The entry point of the SLS fragment is the USBD fragment which includes the **UserServiceDescription**, which in turn references the MMT Package Table.

The MMT Package Table contains the list of all Assets and their location information as specified in subclause 10.3.4 of ISO/IEC 23008-1 [21].

Also delivered by the MMTP session signaled in the SLT is the MMT ATSC3 (MA3) message mmt_atsc3_message(), which carries system metadata specific for ATSC 3.0 Services including Service Layer Signaling such as Video Stream Properties Descriptor, Audio Stream Properties Descriptor, Caption Asset Descriptor, etc.

When App-Based features are used for MMTP delivered contents, the MA3 messages may also carry the HELD, Application Event Information, Inband Event Descriptor, and the DWD. For hybrid delivery, the MMTP-specific SLS includes the MPD for broadband components.

4.6.5 Signal Signing

MMT messages, including the mmt_atsc3_message() and messages defined in MMT, are wrapped in the signed_mmt_message() structure. In order to thwart man-in-the-middle attacks, ATSC 3.0 requires that all messages be signed, including all MMT messages. Service providers should ensure that these messages are signed properly with up-to-date credentials in order to avoid being inadvertently identified as insecure by receivers.

4.6.6 Multi-Stream and Scalable Coding

Multiple streams, whether used for separate presentation or for single presentation as in the case of scalable coding, can be delivered in multiple PLPs. The SLS from a single PLP can point to content delivered by other PLPs.

The number of PLPs should not exceed the capabilities of target receivers deployed in the market.

4.6.7 Delivery of Encrypted MPUs

Editor's Note: This section is for further study due to dependence on A/331 [1] adoption of the related description.

4.7 Switching between MMT and ROUTE

In general terms, a switch between transport formats should be restricted to situations where an instance of ATSC 3.0 is being switched from one transport to another on a semi-permanent basis. Operational practices likely dictate that the transition should occur in the early morning, for example 2 AM. It is recommended that the change should occur at a Period boundary (in DASH nomenclature). This would likely be at the top of an hour. The receiver should be expected to redo the scan for the instance of ATSC 3.0 that is transitioning. It is expected that the receiver should re-acquire the current tuned Service upon transition. The "same Service" can be known by a match of the 16-bit unsigned short integer value of **SLT.Service**@serviceId, **SLT.Service**@majorChannelNo and **SLT.Service**@minorChannelNo. Broadcasters should try to align the Service numbers and names across a transition. Services may be added or deleted upon a transition, but the behavior of the receiver may be less predictable.

4.8 Number of PLPs and Recommended Usage

It is desirable to construct a Service with as few PLP resources as possible for a given application. A Service intended for consumer receivers should never require more than four PLP resources in the receiver (see Section 5.1.1 of A/322 [4]). When Layered-Division Multiplexing is used, it is important to understand that delivery of any Service-related content in a physical layer "Enhanced Layer" consumes two PLP resources independent of whether any content in this Service is being delivered in the related "Core Layer."

4.8.1 Efficient Utilization of PLP Resources and Robustness

In Section 4.3 above, various PLP and PHY frame assignment schemes were provided without much discussion of the relative robustness of objects carried in each PLP or across the various assigned PLPs. The relative robustness of delivery varies according to the to-be-delivered object size and Modulation and Coding selection. Since there is one BaseBand Packet (BBP) per physical layer FEC block and the Low Density Parity Code (LDPC) codes are erasure codes, it is reasonable to consider that a given BBP has a finite, less than one probability of reception. There is less than 100% probability that a given BBP is received correctly.

It is easy to assess the relative probability of correct object reception by size of the object in BBPs and probability of correct reception of each BBP. Suppose that the probability of correct BBP reception is 99.99%, then the relative probability of correct reception of various size objects can be easily calculated. Assuming for the purposes of this exercise that the BBP is delivered in a hypothetical 64kb block with a code rate of approximately 1/2, the resulting BBP size is approximately 32 kb.

For example:

A second of signaling might be 32 kb, 1 BBP with probability of correct reception of $(0.9999)^1$,

A second of audio might be 320 kb, 10 BBPs, with probability of correct reception of $(0.9999)^{10}$,

A second of video might be 3200 kb, 100 BBPs, with probability of correct reception of $(0.9999)^{100}$.

It is straightforward to observe that the smaller objects are inherently more robust than the larger ones, when the objects are delivered in a shared PLP or a second PLP of the same robustness.

There is a notion that signaling might require a more robust PLP to assure a high probability of correct reception, for example relative to video. The probability of loss of signaling is a factor of 100 less than a modest video object when they share the same robustness level PLP. While the subjective impression is that audio and video fail at the same threshold Signal to Noise Ratio (SNR), this is because the LDPC coded loss rate is very steep; i.e., several orders of magnitude per dB of SNR, so a factor of 10 in BBP loss rate is not obvious to a casual observer. The fact is that signaling will be substantially more reliable than either audio or video when delivered in the same PLP. There is no need for a robust PLP dedicated to signaling. A separate "robust PLP for signaling" can consume a PLP resource unnecessarily.

There may be certain circumstances in which a broadcaster desires an easily discernable difference in audio vs. video robustness. The broadcaster may then select to deliver audio in a more robust PLP. In this case signaling should be carried with audio, where one second of signaling will be a factor of 10 less likely to be lost than a corresponding second of audio. Audio can be several dB more robust than video, if so desired and the loss of capacity is acceptable.

How much more robust does a PLP need to be for the audio to be discernably more robust? Most users can just detect a 1dB difference between audio and video delivery robustness at the physical layer. All users will easily detect a 3 dB difference in robustness for audio vs. video.

ATSC 3.0 can adjust delivery QoS to the carried Service components. Real time streaming media has a series of repetitive media deadlines to be met, e.g. maximum delay, typical object sizes, etc. Non-Real-Time delivery has a different set of QoS requirements with respect to object sizes, and maximum delay. Services with similar QoS requirements for their media may share a PLP resource, as is shown for every use case except for the one shown in **Figure 4.8**. Due to the nature of NRT delivery, e.g. large objects, can span multiple seconds of delivery time and may have a relatively loose relationship to real time Services. There is a possibility of sharing NRT resources across related Services. It is a good idea to carry NRT object(s) with a scope of `group_id`.

In the absence of AL-FEC, larger objects can require a more robust selection of Modulation and Coding parameters. AL-FEC can be used to adjust for the variable impact of object sizes.

If a Service is intended for delivery to battery-powered devices that receive the ATSC 3.0 physical layer transmission, then the utilization of per-Service PLPs is desirable. For the use case depicted in **Figure 4.10**, there could be a dedicated PLP assigned to each of the six video streams. This would substantially decrease the radio power consumption with no increase in the number of PLPs required for receiver acquisition of media content carried in each Service. Such a scheme uses more total PLPs, but still less than four PLPs in any single Service.

Recommendations for Efficient PLP assignments in Services:

1) Should try to minimize the number of PLPs required for reception of each Service, so that the receiver may have a spare PLP resource, for background reception of other PLPs as the receiver sees fit, for example background NRT reception, or periodic signaling updates if signaling is in more than one PLP.
2) Should not dedicate PLP(s) to signaling, unless the loss of a PLP from the maximum of four PLPs and the capacity loss are acceptable.
3) Should carry signaling in the most robust PLP included as part of the described Service.
4) Should try to share the NRT delivery PLP(s) across multiple Services within a group, so receiver can receive objects for the group-related Services. All Services on an instance of ATSC 3.0 should share a single NRT PLP, when possible. Large objects in the absence of AL-FEC may require a more robust PLP to assure delivery.
5) Should divide, for example, video streams into separate per Service PLPs, if receiver power consumption is a primary concern. Due to the lower bit rate required, per Service PLPs are

less important for audio and signaling. It is more efficient to organize the combination of signaling and audio so as to fully consume an integer number of BBPs.

4.9 ROUTE Session to PLP Mapping

There is no defined use case that requires that a ROUTE session as known by its source address, destination address, and destination port number needs to be split across two different PLP(s). It is recommended that a defined ROUTE session should exist entirely within a single PLP.

4.10 Repetition Rate of Signaling

The repetition rate for all required signaling LMT, LLS (RRT, SLT, System Time, and AEAT) and SLS per Service potentially impacts channel change time and scan time. The repetition rate for SLS (Service Layer Signaling) fragments potentially impacts channel change time. The minimum channel change time results from all the RAP data being delivered in the minimum time for the desired Service and the receiver knowing that it is safe to start playback. “Safe to start playback” for Segment level play means that all required signaling and at least one Segment of each media type included in the Service are available in the device cache. This is a necessary but possibly not sufficient condition. As depicted above in **Figure 4.4**, the use of `@suggestedPresentationDelay` will ensure safe start for all receivers. For “Safe to start” MDE playback the similar requirement is that all the required signaling has been received and enough time has elapsed to allow enough of each type of Media Segment to populate the receiver cache, before the end of the current Media Segment(s) playback.

The maximum duration for one instance of complete signaling delivery should be 5 seconds or less; thus a receiver executing scan or channel change may need to wait up to 10 seconds on each ATSC 3.0 allocation to be assured of complete reception. There is no means to signal that a given delivery of sub-five second signaling is complete. There is no means to assure that all signaling is delivered at any point in time within a given 5 second interval, hence the 10 second maximum wait.

A minimum set of LMT, SLT, RRT and the Service SLS should be sent for each instance of Media Segment(s) that contain SAP for all required media components in the Service of interest. The Service RAP for each available Service need not repeat faster than once per second.

4.11 SLS Package Structure Management and SLS Fragment Versioning

The following recommendation is given regarding the use of multipart MIME encapsulation in ROUTE delivery of SLS fragments of a Service. Such SLS packaging is described in Section 7.1.6.1 of A/331 [1], and its delivery may employ the Unsigned Package Mode or Signed Package Mode, as described in Annex A.3.3.4 and A.3.3.5, respectively of A/331 [1].

In ATSC 3.0, SLS information is only meaningful with respect to currently-valid SLS fragments, which are those contained in an SLS package with the highest version number. The version number is given by the “**`Version`**” field of the re-purposed TOI field of ROUTE packets which carry SLS data, as defined in Annex C of A/331 [1]. “**`Version`**” is incremented by 1 each time any one or more SLS fragments in the package are changed.

Given that “current” is the only meaningful semantics with regards to validity of SLS fragments, it is recommended that broadcasters do not use the `@validFrom` and `@validUntil` attributes of the **`metadataEnvelope.item`** element when SLS packages are delivered via broadcast. If a new version of any SLS fragment is sent in a new instance of an SLS package (as identified by incremented “**`Version`**” value in the TOI field), the intention is for the receiver to ignore the older version of the same fragment obtained from a previous SLS package(s), even if the `@validUntil` attribute of that SLS fragment is present and has not expired. Furthermore, any instance of an SLS package should not contain more than a single version (as denoted by **`metadataEnvelope.item`**`@version`) of a given SLS fragment type, since otherwise it might confuse the receiver as to which instance of that fragment is valid for use.

4.12 Audio Signaling

4.12.1 Introduction

There is a clash between the current ability of most broadcasters to deliver only two AC-3 audio streams, and the FCC regulatory and multi-language marketplace requirements to deliver more than two. This section explains how the Next Generation Audio (NGA) technologies of ATSC 3.0 can satisfy those requirements by using the audio signaling in the ATSC 3.0 Delivery Layer.

4.12.2 Today's Practice

Most terrestrial TV broadcasters today deliver two audio streams and designate one as the primary audio, signaled as a Complete Main Service, with the correct language in the AC-3 and ISO 639 descriptors. In the United States the primary audio stream is usually English language, with audio signaled as English language. The second audio stream is usually signaled as Spanish language, but from time to time may actually be video description audio or Spanish audio or emergency information audio. This continues the analog television tradition of designating the second audio program stream as SAP.

While all broadcasters carry the AC-3 audio descriptor, not all broadcasters correctly populate the values for language and audio service on the second audio stream. Rather, some broadcasters set those values to match the language and service in the ISO 639-2 language descriptor. In this case, the second audio stream is labeled as Spanish language Complete Main (bsmod value '`0`') even when the second audio stream is actually English language and video description audio (bsmod value '`2`').

However, some Spanish language TV stations found that when their main language was Spanish and they signaled it as Spanish in the AC-3 and ISO 639-2 descriptors, they got calls reporting no audio (when there was nothing on the secondary audio) or the wrong audio (getting the secondary audio stream when they wanted the main audio). Consequently, these stations now carry the Spanish language audio but signal it as English and carry English on the second audio stream but signal it as Spanish.

For most broadcasters, the values in the AC-3 audio descriptor are fixed and do not change throughout the day rather than dynamic, even when the type of service in the second audio stream is not fixed and changes from one program to another. Consequently, even when the second audio changes from Spanish to video description audio at a program boundary, those broadcasters continue to signal video description audio as Spanish Complete Main in the AC-3 audio descriptor.

There are a number of stations that switch language frequently or carry simultaneous programs in a variety of languages. For example, in the San Francisco Bay Area there is a station running three Services, all directed to non-English speaking audiences. This station signals its languages in the AC-3 descriptor as "multiple" when the actual language of a program might be Korean, Mandarin, Cantonese, Farsi or Hindi.

4.12.3 ATSC 3.0 Audio

The coding of ATSC 3.0 audio (Next Generation Audio or NGA) is described in A/342. A/342 Part 1 [6] specifies the common framework for ATSC 3.0 Audio. It is intended to be used in conjunction with the specific audio technologies described in subsequent parts of A/342.

In ATSC 3.0, an Audio Presentation is a set of Audio Program Components representing a version of the Audio Program that may be selected by a user for simultaneous decoding. An Audio Presentation is a sub-selection from all available Audio Program Components of one Audio Program. A Presentation can be considered the NGA equivalent of audio Services in predecessor systems, which each utilized complete mixes (e.g., "SAP" or "VDS").

Depending on the coding technology and/or Delivery Layer signaling, "Audio Presentation" may be called Preselection, Presentation or Preset. See A/342 Part 1 [6] Table 4.2. Audio Presentations are combinations of Audio Program Components representing versions of the audio program that may be selected by a user. For example, a complete audio with English dialog, a complete audio with Spanish dialog, a complete audio (English or Spanish) with video description, or a complete audio with alternate dialog may all be selectable Presentations for a Program.

The ATSC 3.0 NGA audio elementary stream contains metadata/signaling that allows selected components to be rendered, such as language. That metadata/signaling should be inserted in the elementary stream at the time that the NGA coding is done.

The AC-4 metadata/signaling is described in A/342 Part 2 [7] Section 5.2.3. The MPEG-H metadata/signaling is discussed in A/342 Part 3 [8] Section 4.2.1 and is defined in ISO/IEC 23008-3 [22].

4.12.4 Delivery Layer Signaling

Audio signaling is included in the Delivery Layer using Service Layer Signaling. See A/331 [1]. Table 6.1 of A/342 Part 1 [6] describes the audio characteristics that are signaled in the A/331 [1] Delivery Layer. For MMTP-Specific Service Layer Signaling, this signaling is included in the Audio Stream Properties Descriptor. See A/331 [1] Section 7.2.3.4. For DASH Service Layer Signaling, this signaling is included in the DASH Media Presentation Description (MPD). See ISO/IEC 23009-1 [5].

The broadcaster should ensure that the audio signaling in the Delivery Layer correctly reflects the audio metadata/signaling in the audio elementary stream. In this way, receivers can combine the audio components in ways that can satisfy FCC regulatory requirements and provide multi-language dialog streams to meet marketplace needs. By using the Delivery Layer signaling, receivers need not process elementary stream metadata/signaling to find and select the needed components.

4.12.5 Video Description Service

Video Description Service is an audio Service carrying narration describing a television program's key visual elements. These descriptions are inserted into natural pauses in the program's dialog. Video description makes TV programming more accessible to individuals who are blind or visually impaired.

The Video Description Service may be provided by sending a collection of "Music and Effects" components, a Dialog component, and an appropriately labeled Video Description component, which are mixed at the receiver. Alternatively, a Video Description Service may be provided as a single component that is a Complete Mix, with the appropriate label identification.

4.12.6 Multi-Language

In today's practice, multi-language support is achieved by sending Complete Mixes with different dialog languages. In the ATSC 3.0 audio system, multi-language support can be achieved through a collection of "Music and Effects" streams combined with multiple dialog language streams that are mixed at the receiver.

The broadcaster should ensure that dialog components are labeled with the correct language.

4.12.7 Default Audio Presentation

One Audio Presentation is required to be signaled as the default (main), and to have all of its Audio Program Components present in the broadcast stream. See A/342 Part 1 [6] Section 6.1.2. The main Audio Presentation is intended to be the default in cases where no other selection guidance (user-originated or otherwise) exists.

4.12.8 Typical Operating Profiles

Annex A of A/342 Part 1 [6] contains examples of Audio Presentation/Presets and the combinations of components that can be combined to make up the Audio Presentations. For example, a Presentation might include four components: Music & Effects, English dialog, Spanish dialog and English video description service. The user might have specified a Preset for rendering that combines Music & Effects with one of the three dialog components.

4.13 ESG Data Delivery When 4 PLPs Are in Use

ESG data is delivered as an NRT data object. The ESG data objects should be included in the NRT PLP, as described above. If the ESG data is per group, then the ESG per group should be delivered in the NRT delivery associated with its group. The ESG data objects should not in general be delivered in a separate PLP from other similarly sized NRT objects.

4.14 Synchronous Playback for DASH

DASH supports synchronous playback across multiple device types and physical layer delivery methods. This is accomplished utilizing a common playback time line based on UTC. This UTC wall clock common time line is established via ATSC 3.0 physical layer time and the System Time LLS fragment for ATSC 3.0 OTA broadcast. Broadband-connected devices establish their UTC wall clock time line via either an NTP or PTP server, when ATSC 3.0 established time is not available. A relative accuracy of a few msec for wall clock is sufficient to synchronize media across multiple devices.

There are two specific DASH parameters associated with synchronous playback. These are `@suggestedPresentationDelay` and `@availabilityTimeOffset`. `@suggestedPresentationDelay` is applicable to all MPDs and its function is illustrated in **Figure 4.22**. `@availabilityTimeOffset` is applicable to broadband MPDs exclusively.

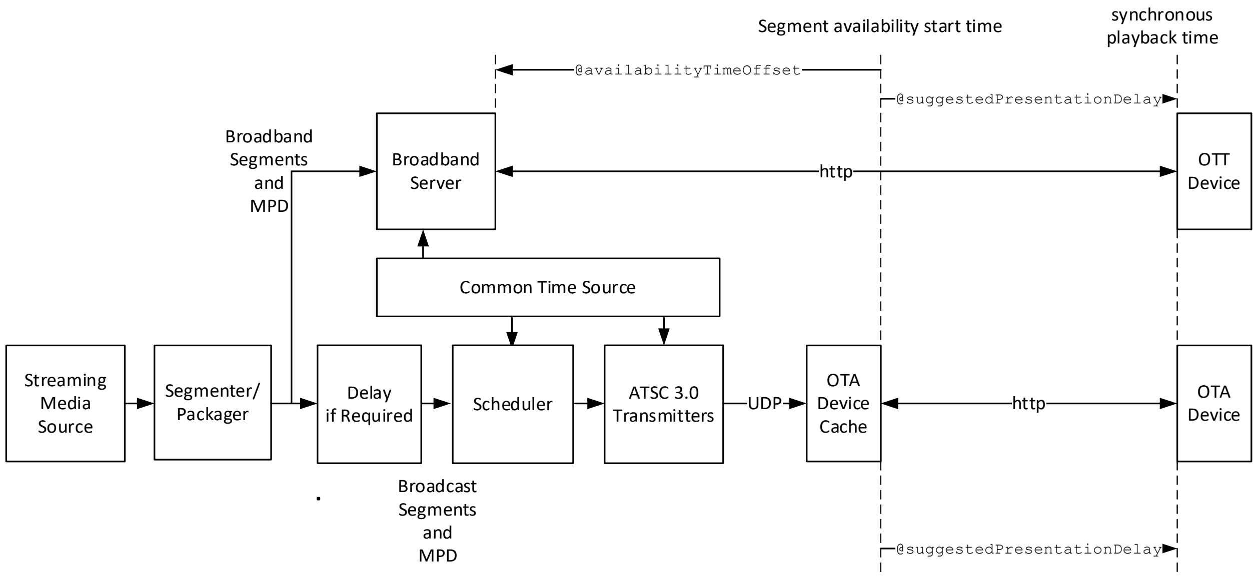


Figure 4.22 DASH attributes and synchronous playback.

The attribute `@availabilityTimeOffset` provides to the device the amount of time before the broadcast Segment availability start time that the broadband device can fetch a Segment.

The setting of `@availabilityTimeOffset` is based on how long broadband delivery takes minus `@suggestedPresentationDelay`. The broadband delivery time can be measured by a variety of means. The actual broadband media delivery delay is known to the origin server. The duration of `@suggestedPresentationDelay` is set to allow at least a full Segment of each media type to be delivered to the OTA object cache in the slowest receivers, prior to start of playback. Ensuring that the OTA object cache is populated for the slowest receiver stack results in delayed playback for all other devices with a faster stack. The presence of `@suggestedPresentationDelay` in an MPD indicates that synchronous playback can be supported. Synchronous playback should be attempted, when it is present, however the receiver may opt out.

4.15 Advanced Emergency Information Usage

4.15.1 Signaling AEA-related files in the AEAT

AEA-related files signaled in the AEAT should conform to file types permitted for BAs according to A/344 [10].

Broadcasters should only signal AEA-related files in the AEAT that may be usefully rendered to a viewer.

The **Media**@mediaDesc attribute in the AEAT is used by a BA to provide context to the viewer for a given media file. Information provided in the **AEAT.AEA.Media**@mediaDesc attribute should provide the necessary context for a viewer to understand what a given media file is. Examples of **Media**@mediaDesc could be “Photo of suspect X related to Y incident” or “Evacuation map of XYZ county as of 1/1/2019” or “Details related to XYZ AMBER Alert”.

4.15.2 Updating and Cancelling AEA Messages

AEA messages of type "cancel" or "update" should reference a previous AEA message that is not expired. Note that sending an AEA message of type "cancel" or "update" that references an AEA message that is expired or that does not exist may produce unexpected behavior in receivers.

The AEAT does not carry new information intended for viewers when the AEA message type is "cancel". Broadcasters wishing to cancel an AEA message **and** provide new information to viewers, such as reasons for cancelation, etc. should either:

- use AEA message type "update"
- use AEA message type "cancel" to kill the previous message and then issue a new AEA message type "alert" to provide the new information to the viewer

4.15.3 Use of AEA Location

When AEA geo-location is provided in more than one format, the locations indicated by each format should match as accurately as possible, given the capabilities of the location format. Note that receivers might process only one of the provided formats. Broadcasters are encouraged to provide location information in more than one format when possible.

4.16 Staggercast

Staggercast is a robustness feature that can be optionally added to audio components consisting of delivery of a redundant version of a main audio component, possibly coded with lower quality (lower bitrate, number of channels, etc.), and with a significant advance ahead of the audio with which it is associated. Receivers that support the Staggercast feature can switch to the Staggercast stream should main audio become unavailable. The delivery delay between Staggercast audio and main audio is chosen to be high enough to provide robustness thanks to sufficient time diversity between the two.

Staggercast audio stream may be delivered either over ROUTE/DASH or MMT and the following signaling should be associated to it.

4.16.1 DASH signaling for Staggercast

To explicitly signal that a Representation is only suitable for Staggercast, a specific scheme with an Essential Property descriptor should be used in the MPD as described in Section 5.4.3.5 of DASH-IF-IOP-for-ATSC3-0 [20].

4.16.2 MMT signaling for Staggercast

A specific ATSC Staggercast Descriptor (see A/331 [1] Table 7.6) should be sent along with the MMT Staggercast audio asset to identify the audio asset as a Staggercast audio asset.

The frequency at which such signaling should be done depends on the delay chosen between the Staggercast audio and the main audio. For optimum performance, an ATSC Staggercast Descriptor should be sent at least once during such delay period. For instance, if there is a 3 seconds delay between Staggercast audio and main audio, an ATSC Staggercast Descriptor should be sent at least every 3 seconds.

The content of the ATSC Staggercast Descriptor is described in A/331 [1] Section 7.2.3.3.

4.17 Year 2036 Wrap-Around

Certain functions in the scheduling and delivery of content are based on the use of clocks expressed as, for instance, numbers of seconds since an epoch. No matter how large the incrementing number used to express time, it eventually will reach its maximum value and have to wrap around, starting again from zero and beginning a new epoch. An example of this is in the Extended File Delivery Table (EFDT) in the ROUTE scheme, which depends on a 32-bit value to express seconds using the Network Time Protocol (NTP) epoch, which began at zero hours, zero minutes, zero seconds on 1 January 1900.

With 32 bits to express the seconds count, it will wrap around to zero after 136 years, 36 days, 6 hours, 28 minutes, and 16 seconds. That is, starting at the NTP epoch, the first wrap-around will occur on February 7, 2036, at 06:28:16 (after accounting for leap days). To avoid confusion, the period starting at the beginning of 1900 is called epoch 0. The one starting in February 2036 is called epoch 1. While it is possible to directly label epochs in which particular times occur, when relatively short periods are involved, it generally is not necessary to do so.

Consider a case in which it is necessary, early on February 7, 2036, to reference a time a couple of days in the future, for instance, to set an expiration time for some data. If a value of 149504 seconds is sent from one device to another, it would represent 1 day, 17 hours, 31 minutes, and 44 seconds into an epoch. Since it would not be logical for a forward-looking time instant to be on January 2, 1900, at 17:31:44 (at the beginning of epoch 0) it would be easy for a device receiving a reference to such a time to recognize that the time must be in epoch 1 and on February 9, 2036, at 00:00:00.

Equipment built to the A/331 standard [1] and dependent on time protocols with constantly incrementing values (such as NTP) should be designed with the fact of eventual time wrap-arounds – no matter how large the number used to express time may be – fully taken into account in their designs.

5. globalServiceID

A/331 [1] defines the attribute `globalServiceID` as xs:anyURI, so the field must be a valid URI. If a URL is used, a valid scheme is required; e.g. "http:"; and if a URN is used, a valid NID is required; e.g. "smpte". Note that "atsc" is not a valid NID.

In order to avoid unintended duplicate Service entries in the Service Announcement (and thus the receiver onscreen guide), not only does the attribute need to be globally unique, but also 1:1 correspondence to a Service. For example, a broadcaster that has a Service that is provided over both broadcast and broadband might want only a single Service Announcement entry, since the receiver can determine which transport is best or provide view options without the viewer seeing multiple "channels" in the onscreen guide. Similarly, the same artifacts might happen if a single Service is broadcast over multiple transmitters. In most cases, the `@globalServiceID` values should be identical. This indicates that the value should be transport agnostic; e.g. not make use of BSID or major/minor channel numbers unless they are the same for identical Services. For a simple, single-station broadcaster, it would work OK.

The `globalServiceID` attribute value should be controlled and established by the broadcaster on a Service by Service basis. Two options for establishing these IDs are discussed below, one using "tag:" URIs and another using EIDR Video Service URLs. Other options are possible but require that individual broadcasters individually ensure global uniqueness and 1:1 with Services.

If an identical Service is sent over both broadband and broadcast, it should have the same `globalServiceID` value.

5.1 Tag URI

In order to address the above, broadcasters should create URIs using the "tag:" URI defined in RFC 4151 [23] (and used in A/331 [1] for XML namespaces). The form of this is:

```
tag:[DNSname],[year]:[uniqueString]
```

where:

`DNSname` is a registered domain name controlled by the broadcaster. This DNS should remain in the control of the broadcaster even if there is no site content.

`year` is any year in which the broadcaster controlled the `DNSname`.

`uniqueString` is a string of symbols that is unique to the broadcaster and used 1:1 to identify a Service. In order to use this core "tag:" syntax for other purposes by the broadcaster, it is recommended that the string begin with "`globalServiceID`", but any discriminator including none will satisfy the requirements in this section.

For example, for the San Diego PBS station, KPBS, it might look like the following.

```
tag:kpbs.org,2019:globalServiceID/1
tag:kpbs.org,2019:globalServiceID/2
tag:kpbs.org,2019:globalServiceID/3
```

where the trailing numbers uniquely identify a Service within the control of KPBS. For example, "1" might refer to the main transmitter minor channel 1, "2" might refer to minor channel 2, and "3" might refer to a broadband Service.

5.2 EIDR Video Service URL

Alternatively, in order to address the requirements of global uniqueness and 1:1 with Services, a URL based on EIDR Video Service IDs can be used. The form of this is:

```
https://doi.org/10.5239/[eidrVideoServiceID]
```

where `eidrVideoServiceID` is the EIDR Video Service ID that corresponds to a Service registered with EIDR.

The EIDR Video Service ID is an identifier in a database curated by EIDR [24]. Video Service IDs begin with "10.5239", as opposed to the more familiar "10.5240" used for an EIDR Content ID.

The ID for each video Service is guaranteed by EIDR to be unique. Each minor channel or Service is assigned a distinct Video Service ID, meeting the requirement that the attribute be 1:1 to the Service. The registry entry identifies the delivery model (e.g., Linear, Internet), and multiple delivery models can be listed for a Service.

For example, KPBS's main Service (the main transmitter's minor channel 1) has Video Service ID:

```
10.5239/8A23-2B0B
```

KPBS's "World", "Create", and "Kids" Services (minor channels 2, 3, and 4, respectively) are:

```
10.5239/C980-1B32
10.5239/5B9E-3BA9
10.5239/5788-A4A2
```

These result in EIDR URLs as follows:

```
https://doi.org/10.5239/8A23-2B0B
https://doi.org/10.5239/C980-1B32
https://doi.org/10.5239/5B9E-3BA9
https://doi.org/10.5239/5788-A4A2
```

Broadcasters should search the EIDR database (`https://ui.eidr.org/search`) to determine if their Services are already registered. Broadcasters can request an addition or modification to the registrations if they are missing or incomplete by submitting a Help Desk ticket to EIDR (support@eidr.org) [24]. EIDR will check for duplicates and request more information about the registration data if necessary. If an organization has an extensive channel list, EIDR can ingest that list for bulk match/registration, which can include third-party channel ID (also known as an Alternate ID) so the organization can use the Video Service registry as a cross-reference as described in Section 2.1 of [24].

5.2.1 Additional Information on EIDR and DOI URLs Using EIDR

In addition to meeting the requirements for uniqueness and a 1:1 correspondence to Services, this scheme has the additional feature that the URL resolves to metadata about the Service referenced by the Video Service ID. Note that URLs used as `globalServiceID’s` are not required to correspond to an actual web page; this is just an advantage of using the EIDR scheme.

The Digital Object Identifier (DOI) Registry (www.doi.org) provides a web-based lookup to access entries in the EIDR registry.

To use the EIDR Video Service ID as a `globalServiceID`, the ID can be prepended with “`https://doi.org/`” to construct a URL that resolves to the Video Service's entry in the EIDR database via the DOI Registration Authority [24]. This lookup is globally accessible and does not incur a fee (see Section 1.2 of [24]).

If they so choose, receivers can access raw (XML) data about the Service by adding “`?locatt=type:Full`” to the end of the URL [25]. If human readability is desired, a query string can be added to the end of the URL to assist; e.g., “`?KPBS-1`”.

– End of Document –