ATSC Standard:
Scheduler / Studio to Transmitter Link

Doc. A/324:2018
5 January 2018

Advanced Television Systems Committee
1776 K Street, N.W.
Washington, D.C. 20006
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
Candidate Standard approved Updated CS approved	30 September 2016 1 November 2017
Standard approved	5 January 2018

Table of Contents

1. SCOPE 1
1.1 Introduction and Background 1
1.2 Organization 2
2. REFERENCES 2
2.1 Normative References 2
3. DEFINITION OF TERMS 3
3.1 Compliance Notation 3
3.2 Treatment of Syntactic Elements 3
3.2.1 Reserved Elements 3
3.3 Acronyms and Abbreviation 4
3.4 Terms 6
4. SYSTEM OVERVIEW 9
4.1 Features 9
4.2 System Architecture 12
4.2.1 System Manager 13
4.2.2 Broadcast Gateway 14
4.2.3 Studio to Transmitter(s) Dataflow 15
4.2.4 STL Operation 15
4.2.5 SFN Operation 15
4.3 Central Concepts: DSTP 16
4.3.1 IP Multicast 16
4.4 Central Concepts: ALPTP 16
4.4.1 IP Multicast 16
4.5 Central Concepts: STLTP 17
4.5.1 STLTP Payload Data 17
4.5.2 IP Multicast and Tunneling 17
4.5.3 Error Control Coding Scheme 17
4.6 System Time Domains 17
4.7 System Manager Configuration Interface 18
4.8 Real-Time Control Interface 18
4.9 Transmitter Requirements Overview 18
4.9.1 Carrier and Timing Offset for Co-Channel Interference Mitigation 19
5. SCHEDULER DESCRIPTION AND NORMATIVE REQUIREMENTS 19
5.1 Relationship of Broadcast Gateway and Scheduler to the System 19
5.2 Scheduler Functionality 20
5.3 Preamble Construction 20
5.4 Scheduler Management Protocol 23
5.5 Data Source Control Protocol (DSCP) 24
6. DATA SOURCE TRANSPORT PROTOCOL 25
6.1 Overview 25
6.2 RTP/UDP/IP Multicast Considerations 25
6.2.1 Multicast Addressing 26
6.3 DSTP Design 26

6.4 Emergency Alert Wakeup RTP Controls 28

7. ALP TRANSPORT PROTOCOL (ALPTP) 30

7.1 Overview 30

7.2 RTP/UDP/IP Multicast Considerations 31

7.2.1 Address Assignments 31

7.2.2 Port Assignments 31

7.3 ALPTP Design 31

8. STL TRANSPORT PROTOCOL....34

8.1.1 Address Assignments 34

8.1.2 Port Assignments 35

8.2 Preamble Generator 35

8.2.1 Preamble Data Stream Protocol 35

8.3 Timing and Management Generator 37

8.3.1 Timing and Management Data Stream Protocol 37

8.3.2 Bootstrap Emission Timing and Frame Identification 42

8.3.3 PLP Data Stream 43

8.3.4 Baseband Packet Data Stream Protocol 43

8.4 Studio to Transmitter Link (STL) Protocol Overview 44

8.4.1 SMPTE ST 2022-1 Features / Resources 44

8.5 STL Transmission Protocol Design 45

8.5.1 Example SMPTE 2022-1 FEC Encoding Process 46

8.6 RTP Header Field Definitions 48

8.6.1 RTP Encapsulation Example 49

9. TRANSMITTER OPERATION NORMATIVE REQUIREMENTS....52

9.1 Timing Manager 52

9.2 Preamble Parser 53

9.3 Other Requirements 54

9.3.1 Frequency Accuracy 54

9.3.2 Transmitter Timing Offsets 54

9.3.3 Center-Frequency and Network-Timing Offsets for Co-Channel Interference Mitigation 55

ANNEX A : PHYSICAL LAYER CONTROL....57

A.1 PHysical layer resources 57

A.1.1 Bootstrap Signaling 57

A.1.2 L1-Basic Signaling 57

A.1.3 L1-Detail signaling 59

ANNEX B : NETWORK CONFIGURATION EXAMPLES 63

B.1 Example studio Network topologies 63

ANNEX C : SCHEDULER FUNCTIONAL DESCRIPTION....66

C.1 Overview 66

C.1.1 Scheduler Operation 66

C.1.2 Key Concepts of Scheduler Delivery Metadata 67

C.1.3 Handling Boundary Conditions 71

C.1.4 Delivery Order Within and Across Multiple Sessions 72

C.1.5 Timelines and Deadlines 73

C.1.6 Concept and Practice of Analyzed Media Duration 73
C.1.7 Sequence of Required Data and Media Events for Acquisition 75

Index of Figures and Tables

Figure 1.1 In-scope interfaces description. 1
Figure 4.1 High-level overview of system configuration. 10
Figure 4.2 System architecture. 13
Figure 8.1 STL Transmission diagram. 45
Figure 8.2 Example FEC encoding process diagram. 47
Figure 8.3 RTP Encapsulation example diagram. 50
Figure 8.4 Tunneled Packet packing details. 51
Figure 9.1 Example use of carrier and timing offsets for neighboring co-channel stations. 56
Figure B.1.1 Simple ALP encapsulation. 63
Figure B.1.2 Multiple PLP example. 64
Figure B.1.3 Fully redundant routing example. 65
Figure C.1.1 Cascade of real-time functions involved with Scheduler. 66
Figure C.1.2 Example depiction of a high-level Scheduler process flow. 67
Figure C.1.3 Illustration of Earliest and Latest Time with a block interleaver. 68
Figure C.1.4 Illustration of Earliest and Latest Time options relative to Media Segment play. 69
Figure C.1.5 Example of MDE and associated Earliest and Latest Times. 70
Figure C.1.6 Relationship of an MDE in LCT packets to IP encapsulation. 71
Figure C.1.7 Media is sent at early boundary; e.g., Period start. 71
Figure C.1.8 Example of ordered delivery across multiple sessions. 72
Figure C.1.9 Explicit delivery order on PLP with multiple sessions. 73
Figure C.1.10 Analyzed Media Duration. 74
Figure C.1.11 Order of events to optimize linear service acquisition w/wo an APP. 77

Table 5.1 Preamble Parameters and Their Sources 22
Table 5.2 Real Time Control Field Definitions 24
Table 6.1 RTP Header Field Definitions for Data Source Transport Protocol 27
Table 6.2 DSTP RTP Packet **payload_type** Encoding 27
Table 6.3 Timestamp Field Definitions for Data Source Transport 28
Table 6.4 RTP **payload_type** Wakeup Control Field Definition 29
Table 6.5 Example Wakeup Bit Controls 29
Table 7.1 RTP Header Field Definitions for ALP Encapsulation 32
Table 7.2 ALP RTP Packet **payload_type** Encoding 33
Table 7.3 Timestamp Field Definitions for ALP Encapsulation 34
Table 8.1 Preamble Payload 36
Table 8.2 RTP Header Timestamp Field Definitions 36
Table 8.3 Timing and Management Stream Packet Payload 39
Table 8.4: RTP Header Field Definitions for STLTP 48

ATSC Standard: Scheduler / Studio to Transmitter Link

1. SCOPE

This standard specifies the protocol on the Studio-to-Transmitter Link (STL) from studio-side infrastructure to a Single Frequency Network (SFN) of Transmitters. It defines delivery protocols for ALP Transport. The document also defines possible interfaces among the studio infrastructure, for example the interconnection of the ATSC 3.0 Link Layer Protocol (ALP) and a Broadcast Gateway. This document specifies certain constraints on the scheduling of content and signaling on the Physical Layer. The described scheduling process enables Preamble generation and emission time management. It specifies certain aspects of Transmitter behavior and certain parameters of Transmitter operation.

Figure 1.1 depicts an example configuration and connection of these entities. This document provides no specification of broadband delivery of ATSC 3.0 media content.

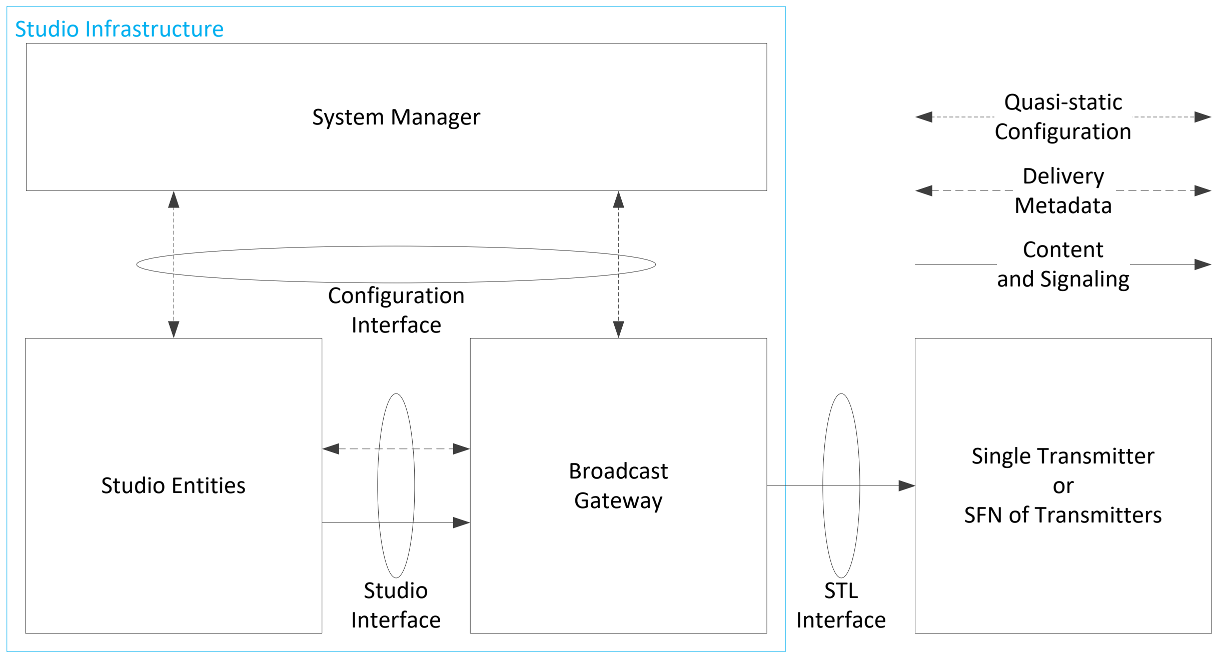

Figure 1.1 In-scope interfaces description.

1.1 Introduction and Background

The ATSC 3.0 system comprises a number of layers that must be connected to one another to construct a complete implementation. Two of the layers that must be interconnected are the Transport Layer and the Physical Layer. In addition, the Physical Layer is designed to be implemented partially at the studio or Data Source and partially at one or more Transmitters. To enable the necessary interoperation of the layers and system segments, appropriate protocols are necessary so that equipment from multiple suppliers can be assembled into a working system. This document defines four protocols, the ATSC Link-Layer Protocol Transport Protocol (ALPTP), the Studio-to-Transmitter Link Transport Protocol (STLTP), the Data Source Transport Protocol (DSTP) and the Data Source Control Protocol (DSCP), for carriage of data through specific portions of the system, as well as a number of operational characteristics of the STL and

Transmitter(s). Also defined are a Scheduler to manage operation of the Physical Layer subsystems and two protocols used by the Scheduler (1) to receive high-level configuration instructions from a System Manager and (2) to provide real-time bit-rate control information to Data Sources sending content through the Transport Layer for emission by the Physical Layer.

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.
- Section 4 – Provides a system overview
- Section 5 – Defines the Scheduler of Physical Layer resources
- Section 6 – Defines the ALP Transfer Protocol (ALPTP)
- Section 7 – Defines the Studio to Transmitter Link Transfer Protocol (STLTP)
- Section 8 – Describes Transmitter operation
- Annex A – Describes the parameters used for Physical Layer control
- Annex B – Provides Network Configuration examples

2. REFERENCES

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

2.1 Normative References

The following documents, in whole or in part, as referenced in this document, contain specific provisions that are to be followed strictly in order to implement a provision of this Standard.

[1] IEEE: "Use of the International Systems of Units (SI): The Modern Metric System," Doc. SI 10, Institute of Electrical and Electronics Engineers, New York, N.Y.

[2] ATSC: "ATSC Standard: System Discovery and Signaling," Doc. A/321:2016, Advanced Television Systems Committee, Washington, D.C., 23 March 2016.

[3] ATSC: "ATSC Standard: Physical Layer Protocol," Doc. A/322:2017, Advanced Television Systems Committee, Washington, D.C., 6 June 2017.

[4] ATSC: "ATSC Standard: Signaling, Delivery, Synchronization, and Error Protection," Doc. A/331:2017, Advanced Television Systems Committee, Washington, D.C., 6 December 2017.

[5] ATSC: "ATSC Standard: Link Layer Protocol," Doc. A/330:2016, Advanced Television Systems Committee, Washington, D.C., 19 September 2016.

[6] IETF: "RTP protocol," RFC 3550, Internet Engineering Task Force.

[7] IETF: "RTP Profile for Audio and Video Conferences with Minimal Control," RFC 3551, Internet Engineering Task Force.

[8] SMPTE: "Forward Error Correction for Real-Time Video/Audio Transport Over IP Networks," Doc. SMPTE 2022-1-2007, Society of Motion Picture and Television Engineers, White Plains, NY, 2007.

[9] IETF: "Generic FEC Payload Format," RFC 5109 (which supersedes IETF RFC 2733), Internet Engineering Task Force.

[10] SMPTE: "Broadcast Exchange Format (BXF) – Protocol," Doc. SMPTE 2021-2:2012, Society of Motion Picture and Television Engineers, White Plains, NY, 2012.

[11] SMPTE: "Media Device Control Protocol (MDCP)," Doc. SMPTE 2071-2:2014, Society of Motion Picture and Television Engineers, White Plains, NY, 2014.

[12] ITU-T, "V.41 Data Communication Over the Telephone Network, Code-Independent Error-Control System," 1993 (or later, if available).

[13] IEEE: "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," Doc. 1588, Institute of Electrical and Electronics Engineers, New York, NY, approved 27 March 2008.

[14] SMPTE: "SMPTE Profile for Use of IEEE-1588 Precision Time Protocol in Professional Broadcast Applications," Doc. SMPTE ST-2059-2, 2015, Society of Motion Picture and Television Engineers, White Plains, NY, 2015.

[15] IETF: "Network Time Protocol Version 4: Protocol and Algorithms Specification," RFC 5905, D. Mills, J. Martin, J. Burbank, W. Kasch, Internet Engineering Task Force, June 2010.

[16] W3C Date and Time Formats, Misha Wolf, Charles Wicksteed, August 27, 1998.

[17] "International Atomic Time," International Bureau of Weights and Measures. Retrieved 22 February 2013. https://www.bipm.org/en/bipm-services/timescales/tai.html

[18] IETF: "RObust Header Compression (ROHC)," RFC 3095, Internet Engineering Task Force.

[19] IETF: "Source-Specific Multicast (SSM)," RFC 3569, Internet Engineering Task Force.

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 [1] shall be used. Where an abbreviation is not covered by IEEE practice or industry practice differs from IEEE practice, the abbreviation in question is described in Section 3.3 of this document.

3.1 Compliance Notation

This section defines compliance terms used in this document:

shall – This word indicates specific provisions that are to be followed strictly (no deviation is permitted).

shall not – This phrase indicates specific provisions that are absolutely prohibited.

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.

3.2 Treatment of Syntactic Elements

This document contains symbolic references to syntactic elements used in its various 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).

3.2.1 Reserved Elements

One or more reserved bits, symbols, fields, or ranges of values (i.e., elements) may be present in this document. These elements are used primarily to enable adding new values to a syntactical structure without altering its syntax or causing a problem with backwards compatibility, but they also can be used for other reasons.

The ATSC default value for reserved bits is '1'. There are no default values for other types of reserved elements. Use of reserved elements, except as defined in other ATSC Standards or by another industry standards-setting body, is not permitted. See individual element semantics for mandatory settings and any additional use constraints. As currently-reserved elements may be assigned values and meanings in future versions of this Standard, receiving devices built to this version are expected to ignore all values appearing in currently-reserved elements to avoid possible future failure to function as intended.

3.3 Acronyms and Abbreviation

The following acronyms and abbreviations are used within this document.

A/V	Audio/Video
AEA	Advanced Emergency Alert
AEAT	Advanced Emergency Alert Table
ALP	ATSC 3.0 Link-Layer Protocol
ALPTP	ALP Transport Protocol
ATSC	Advanced Television Systems Committee
BBP	Baseband Packet
BCH	Bose, Chaudhuri, Hocquenghem
BMFF	Base Media File Format
BPPS	Baseband Packetizer Packet Set
BRET	Bootstrap Reference Emission Time
BS	Bootstrap
BSID	Broadcast Stream ID
bslbf	bit stream, left-most bit first
BSR	Baseband Sampling Rate
BXF	Broadcast eXchange Format
CAP	Common Alerting Protocol
CRC	Cyclic Redundancy Check
CSRC	Contributing Source Identifier
CTI	Convolutional Time Interleaver
D/A	Digital to Analog
DASH	Dynamic Adaptive Streaming over HTTP
dB	deciBel
DDE	Data Delivery Event
Demux	Demultiplexer
DSCP	Data Source Control Protocol
DSS	Data Source Signaling
DSTP	Data Source Transport Protocol
EA	Emergency Alert
ECC	Error Correction Coding
FEC	Forward Error Correction

FFT	Fast Fourier Transform
FI	Frequency Interleaver
FIFO	First In First Out
Gbps	Gigabits per second
GI	Guard Interval
GNSS	Global Navigation Satellite System
GoP	Group of Pictures
GPS	Global Positioning System
HTI	Hybrid Time Interleaver
IDR	Instantaneous Decoding Refresh
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IFFT	Inverse Fast Fourier Transform
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IS	Initialization Segment
ISO	International Organization for Standardization
LCT	Layered Coding Transport
LDM	Layered-Division Multiplexing
LDPC	Low Density Parity Check
LLS	Low-Level Signaling
LMT	Layer Mapping Table
LSB	Least Significant Bit
MDCoIP	Media Device Control over IP
MDE	Media Delivery Event
MHz	Megahertz
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
MMT	MPEG Media Transport
MMTP	MPEG Media Transport Protocol
MPD	Media Presentation Description
MPEG	Moving Picture Experts Group
MSB	Most Significant Bit
MTU	Maximum Transfer Unit
Mux	Multiplexer
NAL	Network Adaption Layer
NoC	Number of Carriers
NRT	Non Real Time
OFDM	Orthogonal Frequency Division Multiplexing
PAPR	Peak to Average Power Ratio
PHY	Physical Layer
PLP	Physical Layer Pipe

PT	Payload Type
PTP	Precision Time Protocol
QP	Quality Point
RAP	Random Access Point
RDT	ROHC-U Description Table
RF	Radio Frequency
RFC	Request For Comments
ROHC-U	Robust Header Compression UDP
ROUTE	Real-time Object delivery over Unidirectional Transport
RTP	Real-time Transport Protocol
SAP	Segment Access Point
SBS	Subframe Boundary Symbol
SCT	Server Current Time
SFN	Single Frequency Network
SISO	Single Input Single Output
SLS	Service Layer Signaling
SLT	Service List Table
SMPTE	Society of Motion Picture and Television Engineers
SSM	Source-Specific Multicast
SSRC	Synchronization Source Identifier
STL	Studio–to-Transmitter Link
STLTP	Studio–to-Transmitter Link Tunneling Protocol
T-RAP	Transport-Random Access Point
TAI	Time Atomic International
tcimsbf	two’s complement integer, msb first
TxID	Transmitter Identification
UDP	User Datagram Protocol
uimsbf	unsigned integer, most significant bit first

3.4 Terms

The following terms are used within this document.

a-millisecond – A time interval approximately equal to one millisecond derived from a binary count of nanoseconds and actually equaling 2^{20} nanoseconds, which represents 1,048,576 nanoseconds (i.e., having a Period of 1.048576 milliseconds).

Advanced Emergency Alert – Provides an emergency notification mechanism in ATSC 3.0 that is capable of forwarding a broad range of emergency data. See [4].

Analyzed Media Duration – The shortest Period between times at which data segment boundaries in all data Streams on the inputs of a Scheduler align. Data segment boundaries are indicated to the Scheduler by markers carried in the headers of the structures transporting data to the Scheduler.

Baseband Packet – A set of payload bits that form the input to an FEC encoding process. There is one Baseband Packet per FEC Frame.

Baseband Packetizer – Functional block that creates Baseband Packets.

Bootstrap – A defined sequence of symbols that introduces each Physical Layer frame and provides a universal entry point into a digital Transmission signal. Each Bootstrap carries a value that serves as an indicator of the format of an immediately following Preamble symbol.

Bootstrap Reference Emission Time – A time value indicating the instant at which the leading edge of the first symbol of a Bootstrap is to be emitted from the transmitting antenna(s), absent any timing offsets of individual Transmitter(s) in a Network.

Broadcast Gateway – A Broadcast Gateway converts source file objects, for example media, system information, and other opaque files, into SFN baseband description for distribution to Transmitters.

Center Frequency – The point in the spectrum of a Physical Layer signal at which equal numbers of carriers are positioned both higher and lower in the spectrum.

Data Consumer – A device that receives a formatted stream of data and further processes and/or distributes it.

Data Producer – A device or process that generates a formatted stream of data.

Data Source – An origination point for data to be transmitted as content by the Physical Layer.

Data Source Signaling – Information sent from a Data Source to downstream functions to identify specific characteristics of the content of certain packets within the data Stream to provide interlayer communications while avoiding layer violations in the system. Typically, Data Source Signaling information is carried in RTP headers in locations defined in this standard.

Earliest Time – A time value that accompanies data sent as input to a Broadcast Gateway to indicate the first instant, as determined using TAI, at which the first byte of related data, including all wrappers and encapsulating protocols, may start emission on the Physical Layer.

Emission Wakeup Field – The two Wakeup Bits in a Bootstrap when treated together as a field.

Exciter – An element of a Transmitter comprising data processing and signal processing functions including at least framing, error correction coding, waveform generation, modulation, and up-conversion to the output RF channel frequency.

Latest Time – A time value that accompanies data sent as input to a Broadcast Gateway to indicate the last instant, as determined using TAI, by which the last byte of related data, including all wrappers and encapsulating protocols, must complete emission on the Physical Layer.

Media Segment – A portion of a data Stream that is treated as a unit by a Scheduler for purposes of analysis and Transmission.

Multiplex – A group of services that are transmitted together over a Network.

Network – A group of Transmitters delivering the same Multiplex.

Packet Set – A group of packets carrying segments of a large data structure that has been segmented for the purpose of carriage across a transport connection that is not configured to carry the large data structure.

Packetizer – A process that treats a collection of data (e.g., a portion of a data Stream) by breaking it into segments and wrapping the segments in a header structure, thereby creating packets for Transmission / delivery.

Period – A duration of time.

Physical Layer – A functional protocol that defines the framing, resource allocation, and waveforms of signals emitted for delivery of data and content to receivers.

Preamble – The portion of a Physical Layer frame that carries L1 signaling data (see [3]) for the frame.

Preamble Generator – A function within a Broadcast Gateway that accepts instructions from a Scheduler, creates and formats Preamble Packets according to those instructions, and releases the Preamble Packets in the form of a Preamble Stream that can be multiplexed with other data Streams for delivery to the Transmitter(s) under control of the Broadcast Gateway.

Preamble Packet – A packet of data that provides a complete set of information necessary for Transmission following the Bootstrap of a Physical Layer frame to instruct receivers regarding the necessary receiver data processing to permit recovery of the data contained within the frame. The packet also serves to instruct Transmitters with respect to the configuration of the Physical Layer frame that is to be emitted so that the Exciter data processing for the frame can be properly configured.

Preamble Parser – A function within an Exciter that receives Preamble Streams, extracts from them Preamble Payload data, and stores the Preamble Payload data until the time for its emission as part of a broadcast signal. The Preamble Parser then makes the Preamble Payload data available to the Exciter control system for use in configuring the data- and waveform-processing of the Exciter, and outputs individual Preamble Packets at the correct times for their inclusion in the emission of the Physical Layer frames the configurations of which they define.

Preamble Payload – The L1 data carried in a Physical Layer frame to define the structure of the frame and to specify the modulation, coding, and other parameters used in delivery of the frame.

Preamble Stream – A data Stream carrying Preamble Packets, comparable to the data Streams carrying data for PLPs and for Timing and Management functionality, that can be multiplexed with other Streams and delivered as a combined data Stream that is tunneled through the STLTP.

reserved – Set aside for future use by a Standard.

Scheduler – 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.

Single Frequency Network – Multiple Transmitters in proximity to one another radiating the same waveform and sharing a frequency.

STL Interface – The STL Interface is the origin point for the Studio-to-Transmitter Link Tunneling Protocol (STLTP).

Stream – A sequential set of packets carrying data from a Data Source to one or more Data Consumers.

Studio Interface – The Studio Interface is the termination point for the ALP Transport Protocol (ALPTP) and/or the Data Source Transport Protocol (DSTP).

System Manager – The System Manager is a conceptual subsystem outside the Transport and Physical Layers and responsible for coordinating the functions of at least those two layers and the static and quasi-static configurations of various system aspects, for example definition of PLPs or assignment of IP addresses and port numbers to Services. The System Manager does not manage real-time traffic directly.

Timing and Management Data – A collection of data sent from a Timing and Management Generator in a Broadcast Gateway to the Transmitter(s) under control of the Broadcast Gateway for purposes of controlling the emission times of Physical Layer frames, establishing various Transmitter configurations, and setting various Transmitter parameters, independent of the configurations of the frames and waveforms of the emitted signal itself.

Timing and Management Generator – A function within a Broadcast Gateway that accepts instructions from a Scheduler, creates and formats Timing and Management Data according to those instructions, and releases the Timing and Management Data in the form of a Timing and Management Stream that can be multiplexed with other data Streams for delivery to the Transmitter(s) under control of the Broadcast Gateway.

Timing and Management Stream – A data Stream carrying Timing and Management Data, comparable to the data Streams carrying data for PLPs and for Preambles, that can be multiplexed with other Streams and delivered as a combined data Stream that is tunneled through the STLTP.

Transmission – The signal that is emitted by a Transmitter and the synchronized signal that is emitted by all Transmitters in a Network.

Transmitter – An individual emitter at a specific geographic location on a specific frequency.

Transport Layer – A functional protocol that defines the formatting of data that will be delivered to receivers after its recovery from the formatting required for its physical delivery by the Physical Layer. It provides logical communication between application processes running on different hosts within a layered architecture of protocols and other network components.

Tunnel Packet – A packet that carries in its payload the contents of one or more multiplexed packet Streams, including the corresponding headers and any other structural elements, i.e., a packet in the “outer” layer of a packet Tunneling system.

Tunneled Data Stream – A Stream of multiplexed Tunneled Packets carried within an STLTP encapsulation for ECC processing.

Tunneled Packet – A packet within a multiplexed group of packet Streams carried in a Tunnel Packet, i.e., a packet in the “inner” layer of a packet Tunneling system.

Tunneling – A process by which a group of parallel and independent packet Streams are carried within a single packet Stream so that they can be processed, transported, and otherwise treated as a single Stream entity.

Tuple – The combination of Internet Protocol (IP) addresses and port numbers.

Wakeup Alert – The occurrence of an AEA message requesting the setting of a non-zero value for the Emission Wakeup Field.

Wakeup Bits – The bits in Bootstrap Symbols 1 and 2 that are used to indicate to receivers that there is high-priority emergency information included in the broadcast. See [4].

4. SYSTEM OVERVIEW

4.1 Features

The STL subsystem exists between the Transport Layer, which creates ATSC 3.0 Link-Layer Protocol (ALP) packets, and the Physical Layer, which formats Streams of ALP packets for Transmission in particular Physical Layer Pipes (PLPs) in an emission configuration specified continuously in detail by a Scheduler. Documents [4] and [5] define the ALP and other Transport Layer protocols. Documents [2] and [3] define the Physical Layer protocols. Figure 4.1 shows a high-level overview of the system configuration with applicable document numbers for the adjoining subsystems not defined herein. As depicted in the figure, data to be transmitted enters a Broadcast Gateway using either ALPTP (defined herein) or DSTP (defined herein) in the lower left of the diagram. Other inputs to the Broadcast Gateway are instructions from a System Manager. A Scheduler internal to the Broadcast Gateway controls the pre-processing functions that occur before delivery of the data and various control information to the Transmitter(s). Delivery of the combined data and instructions to the Transmitter(s) occurs in the lower right of the figure using STLTP (defined herein) with optional ECC applied. At the Transmitter across the top of the figure, the data and instructions from the studio are separated, buffered, and used to control the Transmitter as well as to construct the waveform to be emitted to receivers.

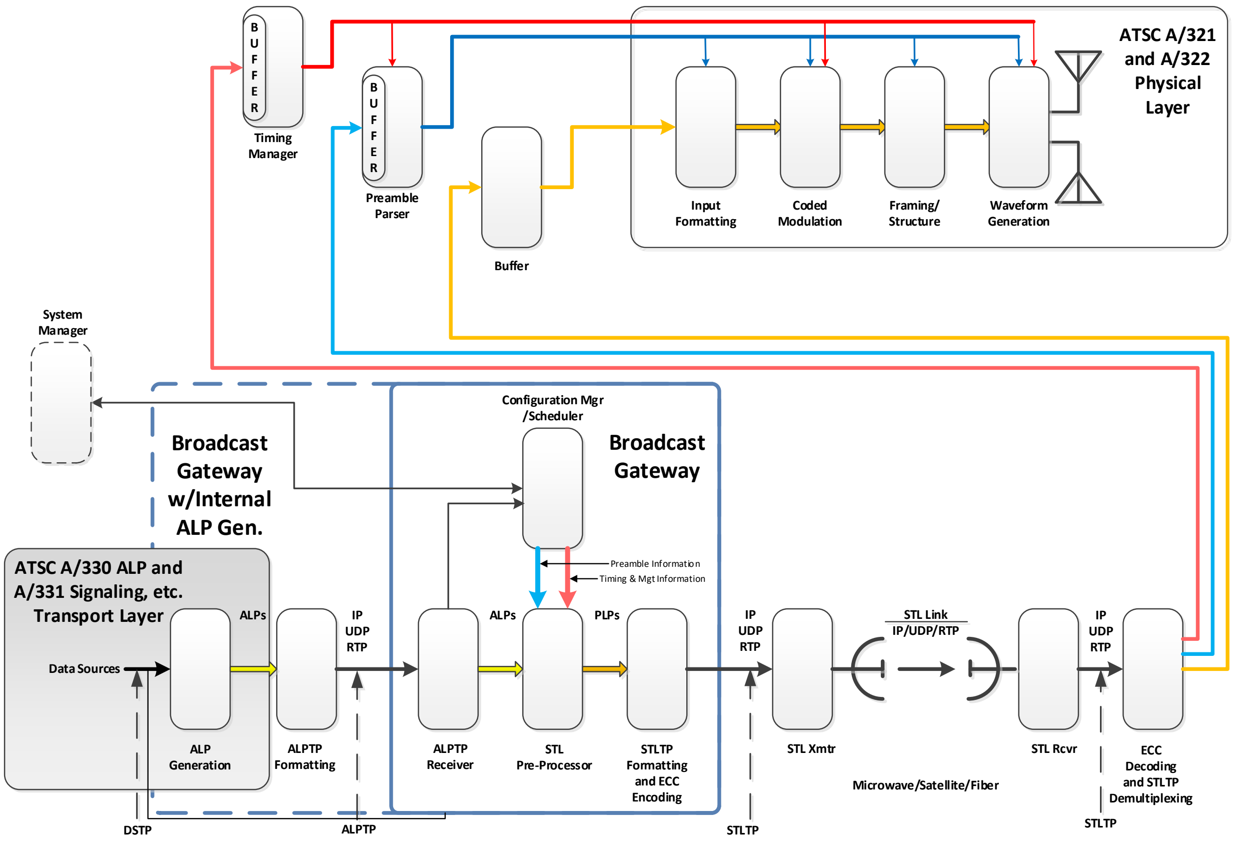

Figure 4.1 High-level overview of system configuration.

There is a one-to-one correspondence between individual Streams of ALP packets and individual PLPs. To prepare ALP packets for Transmission, in the Broadcast Gateway, the ALP packets are encapsulated in Baseband Packets (BBPs), which have defined sizes that are determined by a parameter ($K_{payload}$) related to the specific characteristics of the particular PLP(s) in which they will be carried. The sizes of the BBPs in a given Stream are set to ensure that the assigned capacity of the related PLP in a frame is filled by the BBPs derived from an associated ALP packet Stream. ALP packets either are segmented or are concatenated so that they fill the allocated space in the BBPs carrying them as completely as possible without overflowing the available space.

To manage the flow of data through the system, several buffers are required to hold data for purposes of time alignment of data emission. Buffering also is required in certain instances to enable information to be obtained from a data Stream and used to control particular functionality of the system before the corresponding data is processed further. Two specific instances of such buffering exist in the system. The first buffer inserts at least one Physical Layer frame of delay in the STL Pre-Processor to enable sending Preamble information for a given Physical Layer frame to Transmitters prior to arrival of the data to fill that Physical Layer frame. The second buffer accommodates a delay of up to one second to enable synchronization of frame emission timing in the Physical Layer when the delivery delay to each of the Transmitters in a Network is different.

Maintaining the one-to-one correspondence between particular ALP packet Streams and their assigned PLPs through the system requires a method for identifying both the ALP and PLP data Streams. Since the ATSC 3.0 system works in an Internet Protocol (IP) environment, IP tools and resources are used for Stream identification purposes. Specifically, RTP/UDP/IP multicast stacks are used in both of the ALPTP and STLTP structures, with specific UDP port numbers assigned to particular PLP identifiers and used in both protocols. Thus, for example, an ALP packet Stream designated to be carried in PLP 07 will be carried in an ALPTP Stream with a UDP port value ending in 07, and the Baseband Packet Stream derived from that ALP Stream and to be carried in PLP 07 will be carried within an STLTP Stream with a UDP port value also ending in 07. When the emission operates in Single-PLP mode, all of the data to be transmitted will be carried in only a single ALP Stream, and all of that data will be transmitted with the same level of robustness. When multiple PLP Streams are used, each can have a different tradeoff of data rate versus robustness, and data Streams can be assigned to appropriate combinations by the System Manager. If the ALP Stream(s) is (are) created within the same equipment that provides the Broadcast Gateway functionality, use of ALPTP may not be necessary.

Figure 4.1 shows a single path carrying the ALP packet Stream(s) and then the PLP packet Stream(s) on its (their) way(s) from the ALP generator(s) to the Transmitter(s). In reality, if there are multiple Streams at any point in the system, the processing for each of the ALP packet Streams and/or Baseband Packet (BBP) Streams is applied separately to each of the Streams destined for a different PLP. This separation of processes is diagrammed using parallel paths throughout the remainder of this document, starting with the detailed examination of Figure 4.2.

To manage all of the characteristics of the emission and to coordinate all of the elements of the Physical Layer subsystem with respect to their parameter settings and times of operation, a Scheduler function is included in the Broadcast Gateway. The Scheduler manages the operation of a buffer for each ALP Stream, controls the generation of BBPs destined for each PLP, and creates the signaling data transmitted in the Preamble as well as signaling data that controls creation of Bootstrap signals by the Transmitter(s) and the timing of their emission. To perform its functions, the Scheduler communicates with a System Manager to receive instructions and with the source(s) of the ALP packets both to receive necessary information and to control the rate(s) of their data delivery.

One form of data relationship that the Scheduler must establish is signaling, in the Preamble of any given Physical Layer frame, the presence of Low-Level Signaling (LLS) data in specific PLPs within that frame. To enable the Scheduler to meet that requirement, upstream ALP generators in turn are required to signal to the Scheduler the presence of LLS data in specific ALP packets. When ALPTP is used (i.e., when ALP generators and the Scheduler are in separate equipment units), such signaling takes place in the RTP header that is part of the RTP/UDP/IP multicast stack that comprises ALPTP. This method avoids the layer violation that would occur if the Scheduler had to determine the presence of LLS by inspecting the content of the ALP packets and also covers cases in which the content of the ALP packets is not IP packets.

One of the principal functions of the Scheduler is to generate Preamble data for the Transmitter(s) that it controls. Conceptually, as shown in Figure 4.2, the Preamble generation function is assigned to a Preamble Generator, which is part of the Broadcast Gateway. The Preamble Generator outputs the data that is to be transmitted to receivers to allow their configurations to match the processes and parameters that will be used in Transmission. As the Transmitter(s) process the Preamble data for emission to receivers, the Preamble data also will be used by the Transmitter(s) to set up the Input Formatting, Coded Modulation, Framing/Structure, and Waveform Generation so that the emitted waveform will match what receivers will be instructed by the Preamble to receive. The exact format for the Preamble data is specified in [3].

Similarly, the Scheduler must control the generation and emission of Bootstrap waveforms by the Transmitter(s). To accomplish this, a data structure, similar to the Preamble, is defined in this document to carry Timing and Management data to the Transmitters. Conceptually, as shown in Figure 4.2, a Timing and Management Generator is included in the Broadcast Gateway and provides the function under control of the Scheduler.

BBP data is carried across the STL in an RTP/UDP/IP multicast Stream for each PLP. These Streams are multiplexed into a single RTP/UDP/IP multicast Stream for each broadcast emission to enable reliable delivery to the Transmitter(s) of correctly identified and ordered BBPs. Conceptually, the BBP data Streams, as well as the Preamble Stream and the Timing and Management Stream, are encapsulated as an inner Stream carried through the outer Stream formed by the STLTP. Both inner and outer Streams use IP multicast. The inner Stream provides addressing of BBP Streams to their respective PLPs through use of UDP port numbers. The outer protocol, STLTP, provides maintenance of packet order through use of RTP header packet sequence numbering. The STLTP also enables use of (SMPTE 2022-1) ECC to maintain reliability of Stream delivery under conditions of imperfectly reliable STL Networks.

At the Transmitter(s), an input buffer is used for each PLP to hold BBP data until it is needed for Transmission. There also are FIFO buffers for the Preamble Stream and the Timing and Management Stream. The Preamble Stream processing includes a Preamble Parser that collects all of the configuration information for the next and possibly several upcoming Physical Layer frames to use in configuring the Transmitter data processing for those frames.

Preamble data is scheduled to arrive at the Transmitter input at least one full Physical Layer frame Period prior to the first byte of the associated payload to provide time for the Transmitter data processing to be configured properly. Preamble data also can be sent multiple times in advance to enable acquisition of the data with improved reliability. The same considerations also are applicable to the Timing and Management Data; i.e., it is scheduled to arrive at the Transmitter input at least one Physical Layer frame Period prior to the first byte of the associated payload (+processing delay) it describes, and it can be sent multiple times to enable improved reliability of its acquisition.

4.2 System Architecture

The Studio to Transmitter Link (STL) interface is typically located between the Baseband Packetizer and the Forward Error Correction (FEC) block. There only needs to be one Scheduler and one Baseband Packetizer per RF emission. Multiplexing of multiple Services among stations sharing one RF emission can be accommodated on the input side of the Scheduler.

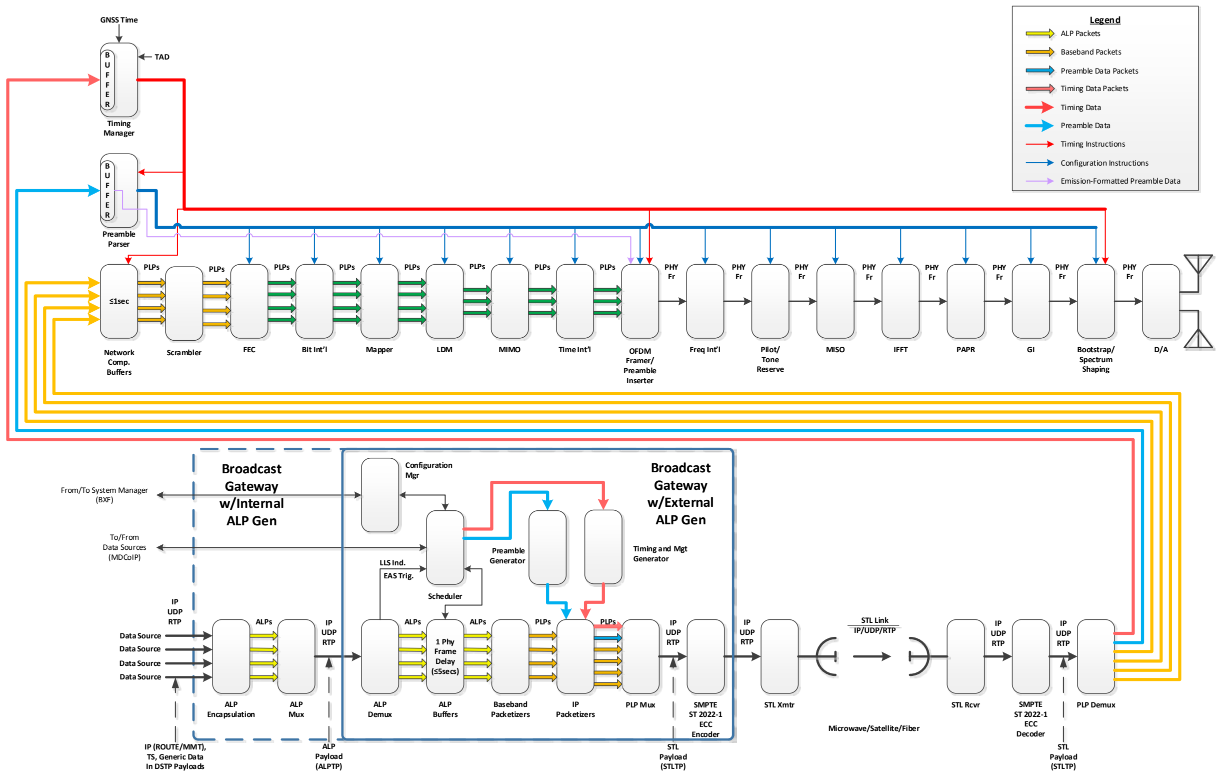

Figure 4.2 System architecture.

Figure 4.2 shows a possible system architecture; other configurations are possible. When considering system configurations, data rates and interfaces between functional blocks must be taken into account in developing practical implementations.

4.2.1 System Manager

Configuration aspects of the overall system are controlled by a single entity called a System Manager, which is represented in Figure 4.2 only as a connection to the Configuration Manager in the Broadcast Gateway. A System Manager can be anything from a web-page based setup screen with manual data entry to a fully automated system; its scope is control of an overall facility or system. The System Manager provides high-level configuration parameters for numerous system functions. The System Manager controls static or quasi-static configurations of the Transmission chain. It controls the Physical Layer configuration with respect to how many PLPs operate and the configurations of the individual PLPs, the Services supplied on those PLPs, and the delivery sessions that support the Services that run in the PLPs. A System Manager can establish a pre-determined schedule for system re-configuration, sending instructions to various system devices and subsystems for delayed execution at specified times. All configuration parameters sent by a System Manager to a Configuration Manager are derived from the Preamble parameter set described in [3] Section 9 and listed in Table 5.1 herein. In response, the Configuration Manager provides feedback of Physical Layer capabilities or structure details of selected Physical Layer frame types. Instructions to the Configuration Manager for these changes in configuration must be sufficiently in advance of the time of emission change. The required minimum advance notice of configuration change time depends on the total reconfiguration latency of the combination of the Configuration Manager and the Scheduler. Further descriptions of Scheduler functionality appear in Sections 4.2.3 and 5.2.

Data Sources feeding the Broadcast Gateway are identified to it by the System Manager using IP addresses and port numbers, with one IP address and port combination (i.e., one Tuple) associated with each Data Source or ALP Stream. These Data Sources also can have separate control IP addresses, in which case the IP addresses of the control connections are indicated to the Broadcast Gateway by the System Manager – at most one per ALP Stream. The operation of the Data Source control interface and the messages that cross it are described in overview in Section 4.8 and in detail in Section 5.5. For each input Data Source or ALP Stream, the System Manager assigns a specific PLP number, which is related to the UDP port address used, as described in Section 7.2.2.

4.2.2 Broadcast Gateway

A Broadcast Gateway is an equipment item that incorporates a number of the conceptual functions necessary in processing data prior to its delivery over an STL to one or more Transmitters. Broadcast Gateways can be implemented in two primary ways, differing primarily in the location of one processing function and the type of interface used to deliver data Streams for Transmission by the Physical Layer subsystem. Both configurations of a Broadcast Gateway are shown in Figure 4.2. In the Broadcast Gateway outlined with a solid-line, an external function, upstream in the system, generates ATSC 3.0 Link-layer Protocol (ALP) packets [5] by encapsulating the data that is to be transmitted in specific PLPs. With external generation of ALP packets, those packets are delivered to the Broadcast Gateway using an ALP Transport Protocol (ALPTP) defined herein, and metadata associated with particular ALP packets and necessary for their Transmission is carried with the packets in their ALPTP headers. In the Broadcast Gateway outlined by the combination of the solid and dashed lines, a function within the Broadcast Gateway performs the ALP packet generation, and a slightly different Data Source Transport Protocol (DSTP) is used to carry both content data and related metadata, the latter of which again is carried in the headers of the DSTP packets.

4.2.2.1 Configuration Manager

The conceptual Configuration Manager within a Broadcast Gateway provides a configuration resource to the System Manager that can accept general waveform structure requirements and from them develop a complete description of the frame structure and waveform that is to be emitted by the Physical Layer. It can inform the System Manager of the capabilities of the system and can provide trial frame structure and waveform details to the System Manager for approval. It also can accept from the System Manager an operational schedule based on use of predefined configurations that the Configuration Manager and the System Manager both have accepted and stored for later use with a naming convention for reference to the predefined configurations. The Configuration Manager provides instructions to the Scheduler, at appropriate times, with respect to the detailed frame structure and waveform that are to be adopted starting at a particular instant and continuing until further instructions are given.

4.2.2.2 Scheduler

The Scheduler within a Broadcast Gateway receives an input of configuration instructions from the Configuration Manager and determines both the frame structure and timing and the waveform details for the next and subsequent Physical Layer frames. The Scheduler must look ahead in time in developing the details of a sequence of Physical Layer frames since instructions about the structures and emission timing of future frames must be sent to Transmitters in advance of arrival of the content data to be transmitted in them. The Scheduler also manages a demultiplexer and a buffer that process data arriving in the form of ALP packets or DSTP packets and any associated metadata in the packet headers. DSTP-delivered data will be encapsulated into ALP packets before reaching the Scheduler, and its associated metadata will be passed to the Scheduler in much the same way as metadata arriving in ALPTP packet headers. The metadata arriving with the content data is needed by the Scheduler for making a number of decisions. Included are decisions to send Wake-Up bits in the emitted Bootstrap of a Physical Layer frame and to signal the presence of various forms of table data (e.g., LLS) in a frame and in a particular PLP within that frame. The Scheduler also controls the Baseband Packetizer blocks, causing them to create Baseband Packets of one of two available sizes according to the characteristics of the specific Physical Layer Pipes (PLPs) into which their Baseband Packets will be sent.

Adjuncts to the Scheduler are a Preamble Generator and a Timing and Management Generator. The Scheduler determines the data to be sent to the Transmitter(s) for inclusion in the transmitted Preamble, and it similarly determines the data to be sent to the Transmitter(s) for control of their operation. The Preamble Generator uses instructions from the Scheduler to form and format a packet Stream to be sent in parallel with the PLP packet Streams to the Transmitter(s) for use by the Transmitters in configuring their operation and for emission to receivers to announce the signal configuration to be received and demodulated. The Timing and Management Generator similarly uses instructions from the Scheduler to form and format a packet Stream to be sent in parallel with the PLP packet Streams to the Transmitter(s) for control of their operations. Unlike the Preamble Stream, the Timing and Management Stream is not broadcast but rather is consumed by the Transmitter(s) where it is used to control the timing of its (their) emissions and other functions either as a group or individually by Transmitter.

4.2.3 Studio to Transmitter(s) Dataflow

The studio to Transmitter link (STL) may operate on any of fiber, satellite or microwave links. STL redundancy is possible, but is out of scope for this document. Internet Protocol (IP) is supported on all link types. Because of its exposure on any of the delivery means to potential security breaches, it is important that the data traversing the STL be protected by a security mechanism. Work is under way on such a system within ATSC.

4.2.4 STL Operation

Broadcasters have a need to send studio-generated data to their Transmitters. Usually those Transmitters are not co-located at the studio. An STL Interface from the studio to the Transmitter(s) is needed. Requirements for such an interface include:

1) Support for Real-Time Protocol / User Datagram Protocol / Internet Protocol (RTP/UDP/IP) IPv4 and addressing
2) Encapsulation of data for the link
3) Providing a synchronization method to TAI for both data and control
4) Providing signaling of the Transmitter time synchronization for data and control
5) Having measurable maximum latency to allow the emission times to be correct
6) Allowing for redundancy of the delivery channel

4.2.5 SFN Operation

Certain specifications in this standard enable Single Frequency Network (SFN) operations, and the protocol for data carriage from the studio to the Transmitter(s) has the capability to support delivery of certain control data individually to each Transmitter in an SFN. Specifically included are the following specifications and capabilities:

1) Inherent synchronization of the data processing functions of multiple Transmitters fed from the same Broadcast Gateway
2) Inherent time alignment of the emissions of multiple Transmitters fed from the same Broadcast Gateway
3) Specification of the carrier frequency tolerance of Transmitters in an SFN
4) Addressing each Transmitter individually and delivering specific data to each
5) Providing offsets from the Bootstrap Reference Emission Time individually to each Transmitter in a Network to facilitate Network service shaping

Timing considerations are satisfied by the Timing and Management Protocol described in Section 8.3.1, and carrier frequency accuracy is specified in Section 9.3.1.

4.3 Central Concepts: DSTP

The Data Source Transport Protocol (DSTP) provides a solution for transferring content data, associated metadata, and signaling through a typical IP Network from Data Sources to a Broadcast Gateway as shown in Figure 4.2. While Data Source outputs are expected to be conducive to carriage on normal IP Networks, they do not provide the resources for carriage of metadata and signaling related to the content that must be communicated to Broadcast Gateways. Also, Data Source packets must be kept in order, and packet loss cannot be tolerated. To address the needs beyond normal UDP/IP functionality, an RTP header is added to each Data Source packet to carry required metadata and signaling that must be communicated from Data Sources to Broadcast Gateways.

4.3.1 IP Multicast

All packets delivered over the DSTP link use an RTP/UDP/IP multicast protocol. Real-time Transport Protocol (RTP) protocol is used with its headers as redefined in Section 6.2. RTP also provides capabilities for ordering packets and determining if any packets have been lost.

4.4 Central Concepts: ALPTP

The ATSC 3.0 Link-layer Protocol Transport Protocol (ALPTP) provides a solution for transferring ATSC 3.0 Link-layer Protocol (ALP) packets [5] through a typical IP Network between two separated devices as shown in Figure 4.2. ALP packets are relatively simple constructs having only a minimal header plus packet length information sufficient just for an emission link layer. These headers are not sufficient for transferring ALP packets through a typical IP Network; consequently, ALPTP is defined to provide necessary additional networking information. Moreover, ALP packets must be kept in sequence for each PLP, and packet loss cannot be tolerated. There is a single ALP Stream associated with each PLP, so a unique IP port is defined for each ALPTP packet Stream to associate it with the PLP to which the data that it carries will be directed.

4.4.1 IP Multicast

All packets delivered over the ALPTP link use an RTP/UDP/IP multicast protocol. Real-time Transport Protocol (RTP) is used with its headers as redefined in Section 7.3. Segmentation and reassembly of large ALP packets and concatenation of small ALP packets for transport on the ALPTP link is performed with RTP. RTP also provides capabilities for ordering packets as well as framing, that is, identifying where each ALP packet starts within the RTP/UDP/IP multicast payload Stream.

4.5 Central Concepts: STLTP

The Studio to Transmitter Link Transport Protocol (STLTP) provides a solution for transferring a potentially large number of parallel Baseband Packet (BBP) data Streams carrying content for a like number of PLPs plus Preamble data and Transmitter control signaling through a typical IP Network from a Broadcast Gateway output to one or more Transmitters using Tunneling, as shown in Figure 4.2. Each Tunneled BBP data Stream feeding a PLP derives from a corresponding ALP Stream, and standardized UDP port numbers are used to establish the associations. Error Correction Coding (ECC) is specified to maintain reliable delivery of STL data over the links between sites, but the ECC method selected requires use of uniform size packets in its data processing. To accommodate this requirement, the multiple PLP, Preamble, and control Streams are multiplexed together on a packet-by-packet basis and then tunneled through a single resulting RTP/UDP/IP multicast Tunnel Packet data Stream. The RTP headers in the Tunnel Packet data Stream provide packet sequencing to help with packet reordering and to identify packet loss. In addition to the multiplexed Tunnel Packet Stream, one or two additional Streams of ECC overhead data having related UDP port numbers can be generated by the ECC subsystem and are carried in parallel with the Tunnel Packet stream from the Broadcast Gateway to the Transmitter(s).

4.5.1 STLTP Payload Data

The STLTP used to traverse STL IP links carries three distinct types of data:

1) Baseband Packet Streams, the data from each of which populates an individual PLP,
2) Preamble data packets derived from the scheduling process outcome and used both to populate emitted Preambles and to control Exciter configurations, and
3) Timing and Management Data packets used to control Transmitter synchronization, to control Bootstrap emission timing, to provide Physical Layer frame identification, and for similar Transmitter management tasks, e.g., TxID control.

4.5.2 IP Multicast and Tunneling

All “inner” layer and “outer” layer packets delivered over the STL using the Tunneling process depend upon an RTP/UDP/IP multicast IPv4 protocol stack. All Tunneled packets have defined port numbers within a single IP address.

Real-time Transport Protocol (RTP) protocol is used with redefined headers, as described in Section 8.4.1. Segmentation and reassembly of large Tunneled payload packets and concatenation of small Tunneled payload packets within the Tunnel Packets is performed using RTP features. A segment sequence number within an “outer” RTP header indicating position of a segment within a larger source packet supports segmentation and reassembly, and a value in an “outer” RTP header indicating the offset of the first “inner” packet segment within the payload of the associated “outer” packet supports concatenation.

4.5.3 Error Control Coding Scheme

Error Correction Coding (ECC) is applied to the STLTP “outer” layer, which improves reliability of delivery of the complete package of STL data to each Transmitter. SMPTE 2022-1 [8] defines the STL ECC and is intended for data rates up to around 1 Gbps, making it appropriate for the STL.

4.6 System Time Domains

As can be seen in both Figure 4.1 and Figure 4.2, the Broadcast Gateway sits astride the boundary between studio and Transmitters, which also is the boundary between the Transport Layer (ALPs) and the Physical Layer (BBPs). Due to the nature of the data formatting in each of those layers, they operate in different time domains. The Transport Layer and layers above it operate in the UTC [16] domain, matching the time on conventional wall clocks, accounting for leap seconds, and often expressing time in NTP [15]. The Physical Layer, on the other hand, operates in the International Atomic Time (TAI) [17] domain, which has second ticks at the same times as UTC but does not account for leap seconds. This allows the Physical Layer to maintain a constant flow of data, including at leap second insertion instants, without interruption of the Transmission process. Reconciling the differences between the two time domains is a task that falls to the Broadcast Gateway for resolution.

Indication of the value of TAI time on the STL Interface (STLTP) is based on the Precision Time Protocol [13] using 32 bits to represent seconds and 32 bits to represent nsecs.

ATSC Physical Layer Time, which is indicated by the 32 LSBs of TAI seconds msecs, µsecs, and nsecs, is delivered to receivers in the Preamble [3]. The 16 MSBs required to deliver a complete representation of the 48 bits of TAI seconds are carried in the System Time fragment [4].

4.7 System Manager Configuration Interface

As described in Section 4.2.1, a System Manager is a conceptual entity that coordinates and controls all broadcaster facilities necessary to produce a specific desired station output configuration and emission. The configuration interface between the System Manager and the Configuration Manager in a Broadcast Gateway transfers information between the two subsystems to enable the system management process with respect to the Scheduler and its associated functions. The System Manager configuration interface comprises a normal TCP/IP connection between the devices with control functionality structured according to the SMPTE Professional Media Over Managed IP Networks (ST 2110) standards suite [11] and carrying the information described in Section 5.4 herein. The messages are used to negotiate a detailed emission waveform configuration between the System and Configuration Managers and to permit the System Manager to provide instructions to the Configuration Manager with respect to emission requirements and schedules.

4.8 Real-Time Control Interface

As mentioned in Section 4.2.1, the System Manager directs the Scheduler through the Configuration Manager to control data delivery from various Data Sources by providing an address and port number combination (a Tuple) for each Data Source feeding data to the Broadcast Gateway for delivery through a PLP. The Scheduler communicates with the Data Source(s) at the address(es) and port number(s) assigned by the System Manager using a Real Time Control Interface that employs SMPTE Professional Media Over Managed IP Networks (ST 2110) methods [11] and carrying the information described in Section 5.5 herein. Messages that can be communicated between the Scheduler and the various Data Sources are defined in Section 5.5. The Real-Time Control Interface provides such functions as discovery of the capabilities of a Data Source, setting a target bit rate for a Data Source, and managing the speed of data delivery from a Data Source in real time, i.e., updating rates of data delivery as necessary to maintain control of buffer fullness throughout the Physical Layer system.

4.9 Transmitter Requirements Overview

Operation of the ATSC 3.0 Physical Layer system depends on standards-compliant Transmitters meeting certain requirements. Among those requirements are: inclusion of Timing Manager functionality, inclusion of Preamble Parser functionality, conforming to certain buffer models and conditions, maintaining a specified frequency accuracy, supporting emission timing offsets from the Bootstrap Reference Emission Time, and providing for adoption of carrier frequency and emission time offsets as instructed by system Schedulers. These specifications and requirements are detailed in Section 9.

4.9.1 Carrier and Timing Offset for Co-Channel Interference Mitigation

When multiple, geographically-neighboring stations are operated on the same RF channel using identical, or even relatively similar, frame and waveform configurations, interference can be caused between the co-channel stations. The interference arises when the pilot locations and characteristics are sufficiently similar in the multiple signals. Such interference is particularly troublesome when the various stations involved operate with highly robust emission characteristics having very low signal-to-noise ratios. The mechanism that leads to such interference is addition of pilot energy at each FFT carrier in a channel at which pilots are positioned. Addition of the pilot energy at each pilot frequency will lead to incorrect channel estimation by receivers because the channel estimation will not be valid for either/any of the received signals in the overlap areas of the Transmitters' signals. The same effect will result from both single-Transmitter and SFN operation by the neighboring stations.

To counter the sort of interference described, the carrier frequencies of neighboring stations can be offset by a small amount to avoid pilot super-positioning, which would otherwise result in pilot level addition. In addition, super-positioning of Bootstrap Reference Emission Times must be avoided to prevent failure of Bootstrap recovery from the individual neighboring stations in regions of signal overlap. Section 9.3.3 herein describes methods for avoidance of super-positioning of both pilots and Bootstraps by neighboring stations, thereby extending their interference-free service areas when they operate with low C/N threshold configurations.

5. SCHEDULER DESCRIPTION AND NORMATIVE REQUIREMENTS

Schedulers are the central subsystems in the Physical Layer that enable movement of data from Data Sources, translation of data formatting from Transport Layer forms to Physical Layer requirements, configuration of the Transmitter(s) and the waveforms it/(they) emit, buffering of data to stage its delivery, timing of emission of Physical Layer frames, and all the other functions necessary to implement the ATSC 3.0 Physical Layer as embodied in [2] and [3]. A Scheduler accepts inputs from a System Manager that give it general instructions as to how the emission is to be configured and scheduled, and it outputs data and instructions to the Transmitter(s), controlling precisely how the emissions actually are configured.

5.1 Relationship of Broadcast Gateway and Scheduler to the System

Figure 4.2 shows a conceptual block diagram of a Broadcast Gateway and its associated interfaces. A configuration interface allows provisioning of quasi-static aspects of Physical Layer configuration, such as PLP definitions. A Data Source Interface has two ports, one of which delivers content to the Broadcast Gateway along with Data Source Signaling that carries inter-layer information and the other of which is used for real-time control exchanges between the Broadcast Gateway and one or more Data Sources. The required message interchanges on the control interface are defined in this document. Output from the Broadcast Gateway and its Scheduler are via an STL Interface that communicates using STLTP, which carries a complete description of a Physical Layer instance on a frame-by-frame basis to one or more Transmitters.

5.2 Scheduler Functionality

The functional assets of a Scheduler are defined by data size(s), time(s), and levels of robustness on the Physical Layer. The Physical Layer can deliver defined quantities of data at certain discrete times with particular amounts of resilience to channel impairments. There are a large number of parameters, many having a wide range of settings, under control of the Scheduler, and a major part of its function is to make intelligent choices among settings to most efficiently accomplish the objectives communicated to it by the System Manager.

A Broadcast Gateway receives the following inputs and information:

1) Data for transmission, either in the form of ALP-encapsulated content data (via ALPTP) or in the form of content data Streams (via DSTP) that must be ALP-encapsulated within the Broadcast Gateway;
2) Inter-layer signaling sent from the Transport Layer to the Physical Layer in the form of Data Source Signaling to identify certain types of content requiring specific actions to be taken by the Scheduler to properly signal or treat the content in the broadcast emission;
3) Configuration and scheduling instructions from the System Manager; and
4) Capability and status information from the Data Source that feeds each of the ALP/PLP data Streams.

From these inputs, a Scheduler shall create an efficient allocation of the Physical Layer resources conforming to the scheduling and configuration instructions that it receives from its associated System Manager. The Scheduler shall manage buffer fullness throughout the Broadcast Gateway and Transmitter chain, based on current sizes of subframes and frames, maximum delay of the STL delivery Network, available STL channel bandwidth, and requirements for robustness and security across the STL. The Scheduler shall determine Bootstrap Reference Emission Times, create timed control instructions for all Transmitters (collectively and individually), and pass timing and management control messages to the Timing and Management Generator so that it can create Timing and Management Data Packets to be sent to the Transmitter(s). As part of managing robustness and security across the STL, the Scheduler shall instruct the Preamble Generator and the Timing and Management Generator how many repetitions of their respective packets shall be sent to the Transmitter(s) and at what times relative to delivery of the contents of the Physical Layer frames with which they are associated.

The Scheduler shall define all of the Physical Layer parameters, such as frame lengths, subframe sizes, PLP configurations, and modulation and coding type and value selections. It shall provide this information, along with all other details to be communicated to both the Transmitter(s) and off-air receivers, to the Preamble Generator for formation into Preamble Packets. The Scheduler shall assign frame identifiers to all Baseband Packets using the Bootstrap Reference Emission Times of the frames in which the Baseband Packets are to be emitted.

5.3 Preamble Construction

The Preamble defined in [3] is used to control the configurations of both Transmitter(s) and off-air receivers so that the signals emitted match the decoding processes in receivers, leading to successful data recovery. The Scheduler shall construct a single Preamble Packet for each Physical Layer frame, for this purpose, which packet shall be sent to the Transmitter(s), used by them for their own configurations, and transmitted as part of the Physical Layer frame with which the Preamble is associated. Multiple copies of the Preamble Packet for a particular Physical Layer frame may be sent to the Transmitter(s) over a period of time to improve robustness of the

Preamble data delivery to the Transmitter(s), through use of majority logic or similar techniques, as described in Section 8.3.1.

A Preamble Packet consists of two fundamental categories of data: quasi-static and dynamic, with the former changing only at infrequent intervals and the latter changing potentially on every frame. The quasi-static information comes from the System Manager, while the dynamic information is developed within the Scheduler as part of its functionality. Both types of information end up in the Preamble. The various Preamble parameters are shown in Table 5.1, and their sources in either the System Manager or within the Scheduler processes are indicated by check marks in the two columns on the right side of the table.

Table 5.1 Preamble Parameters and Their Sources

	Parameters	**Instructions from System Manager**	**Scheduler Generated**
Per Frame Data	Channel bandwidth	✓	
	Sampling Frequency	✓	
	Bootstrap Major and Minor version values	✓	
	Emergency Alert Wakeup		✓
	Preamble Structure indicator	✓	
	Minimum time to next Frame		✓
	Frame length	✓	✓
	Number of subframes		✓
	Number of symbols in the Preamble		✓
	Frame alignment	✓	
	Frame PAPR	✓	
	CRC values (L1B)		✓
	CRC values (L1D)		✓
	Preamble NoC	✓	
	FEC Mode for L1-Detail	✓	
	Additional parity for next frame		✓
	BSIDs involved in channel bonding	✓	
	MIMO scattered pilot encoding method	✓	
	Return Channel flag	✓	
	LLS flag		✓
	Time info flag		✓
Per Subframe Data	Subframe size	✓	
	Subframe PLP count	✓	
	Subframe MIMO/MISO/SISO	✓	
	Subframe FFT size	✓	
	Subframe NoC	✓	
	Subframe GI	✓	
	Subframe scattered pilot pattern	✓	
	Subframe scattered pilot boost	✓	
	Subframe boundary symbol flags	✓	
	Subframe FI mode	✓	
Per PLP Data	PLP ID	✓	
	PLP type		✓
	PLP size		✓
	PLP start position		✓
	PLP number of subslices		✓
	PLP subslice interval		✓
	PLP LLS flag		✓
	PLP scrambler type	✓	
	PLP FEC mode	✓	
	PLP position of first complete FEC Block		✓
	PLP LDPC code rate	✓	
	PLP modulation	✓	
	PLP time interleaver mode	✓	

	Parameters	Instructions from System Manager	Scheduler Generated
	PLP CTI depth	✓	
	PLP CTI start row		✓
	PLP CTI position of first complete FEC block		✓
	PLP HTI inter-subframe interleaving flag	✓	
	PLP HTI number of TI blocks or subframes		✓
	PLP HTI max interleaving FEC blocks per interleaving frame	✓	
	PLP HTI number of FEC block in the current interleaving frame		✓
	PLP HTI cell interleaver flag	✓	
	PLP LDM layer	✓	
	PLP LDM injection level	✓	
	PLP BSIDs involved in channel bonding	✓	
	PLP Center frequency of channels involved in bonding	✓	

5.4 Scheduler Management Protocol

The interface between the Scheduler and System Management functions shall use SMPTE Broadcast eXchange Format (BXF) protocol as described in [10].

The parameters using this protocol are those listed in Table 5.1 under the "Parameters" column with a check mark in the Instructions from System Manager column. These parameters are quasi-static in nature and do not routinely change between Physical Layer frames. Scheduler configurations and constraints are set with these parameters, which are allowed to change quasi-statically. An emission schedule for a set of parameters can be similar to a program schedule, in which they can change over the course of a day. Detailed description of these parameters is provided in Annex A.

In addition to providing the quasi-static Preamble parameters, the System Manager shall identify for the Scheduler:

1) The Source and Destination multicast IP addresses of each Data Source associated with each ALP Stream. Only one Data Source shall be selected for a given ALP Stream at a time.
2) The Destination multicast IP address associated with each ALP Stream, where the port number used indicates the particular ALP / PLP pair.
3) The Control IP address for each Data Source providing data for each ALP Stream. Redundant Data Sources for any given ALP Stream are possible, but only one can be selected for use at a time.

Messages between a System Manager and a Scheduler shall include:

- Capabilities inquiry to inform the System Manager of the assets and capabilities of the Scheduler.
- Configuration design inquiry to provide a detailed Physical Layer frame design from a Scheduler to a System Manager to meet a set of requirements from the System Manager.
- Current configuration inquiry to report to the System Manager the current state of operation of the Scheduler and any presets available for use.
- System Setup instructions to provide to the Scheduler the fundamental configuration settings for operation in the particular system.

- Scheduled configuration instructions to instruct the Scheduler when to place into operation one or a number of preset configurations stored in the Scheduler and available for use.

Scheduler feedback to the System Manager shall indicate the Physical Layer capabilities and the structure details for each requested frame type. Scheduler to System Manager messages include:

- Capabilities response
- Configuration design response (acknowledgement of preset instructions)
- Current configuration response
- System setup acknowledgement
- Scheduled configuration acknowledgement

5.5 Data Source Control Protocol (DSCP)

The interface between the Scheduler and the one or multiple Data Sources that provide content data for emission shall permit the Scheduler to control operation of the Data Sources so that they deliver the amount of data needed at the times at which it is needed to utilize the capacity of the various PLPs efficiently.

The Scheduler shall use the Data Source Control Protocol (DSCP) to direct Data Sources to provide data in specific target bit rate ranges and to throttle those Data Sources to efficiently fill PLP capacities. Via the DSCP, real-time control messages are sent to Data Source IP control addresses as specified by the System Manager. Data Sources can be multiplexers, encoders, servers, and the like.

Messages between Schedulers and Data Sources can include:

- Capabilities inquiries and responses
- Target bit rate range instructions from a Scheduler to a Data Source
- Throttling instructions from a Scheduler to a Data Source to make fine adjustments of data delivery

The control protocol for communication between the Scheduler and Data Sources shall use at least the fields shown in Table 5.2. The Scheduler configures the Physical Layer on a per PLP basis, therefore the DSCP has a link to a Data Source for each PLP. The DSCP can accommodate seamless changes that do not affect service.

Table 5.2 Real Time Control Field Definitions

Syntax	No. of Bits	Format
rt_control () {		
bit_rate_capacity	18	uimsbf
bit_rate_granularity	10	uimsbf
bit_rate_target	18	uimsbf
bit_rate_max	18	uimsbf
}		

bit_rate_capacity field shall indicate to the Scheduler the Data Source bit rate capacity (i.e. maximum bit rates) in units of kbps.

bit_rate_granularity field shall indicate to the Scheduler the Data Source step size in bit rate in units of kbps. Rate of change in the PLP can be from a few kbps to tens of Mbps depending on input coding rates and Physical Layer frame configurations. The fastest rate of change depends on the shortest frame length, which is 50msec in time-aligned frame mode.

bit_rate_target field shall indicate the desired bit rate for Scheduler capability in handling PLP payloads. Range of targets shall be between 1 kbps and 157 Mbps with units of kbps.

bit_rate_max field shall indicate the maximum bit rate value not to be exceeded into the Scheduler in units of kbps.

6. DATA SOURCE TRANSPORT PROTOCOL

The Data Source delivery structure uses the Data Source Transport Protocol (DSTP), which applies between Data Sources and an ALP encapsulation function. There can be multiple Data Sources for each PLP, each carried on a separate packet Stream.

All Data Source packets shall be carried with RTP/UDP/IP multicast IPv4 protocol using the RTP header modifications described in Section 6.2.1. Since Data Source packets originate as UDP/IP, the Data Source Transport Protocol adds an RTP header to each packet with no modification to the underlying UDP/IP packet structure or payload.

Identification of Data Source packet Streams is required to tie together packets destined for specific PLPs and the corresponding ALP Streams. The Data Source packets are constructed with destination address and port numbers as described in A/331 [4]. For example, Low Level Signaling (LLS) packets are sent to a known multicast destination address 224.0.23.60 with destination port number 4937. Similarly, other Data Source packets have addressing intended to be processed by receivers after their transmission. To avoid complications of changing multicast addresses and other mapping issues, each Data Source packet Stream destination address and port number shall remain as destined for receivers. IP source address and port numbers shall be used to bind Data Source packet Streams with specific PLPs as described in Section 6.2.

The system diagram in Figure 4.2 provides the functional context of the Data Source Transport Protocol. It is possible that the actual implementation may be much more complicated, requiring multiple sources of Data Source packets. Such sources could be for redundancy or could be external to the studio.

6.1 Overview

Data Source packets are distributed over UDP/IP multicast using a modified RTP [6] protocol. UDP/IP multicast is used successfully in many industries to deliver redundant video Streams over copper and fiber physical layers. Routing and redundancy can be accomplished using IGMPv3 Source-Specific multicast (SSM) (RFC 3569) [19]. Examples of various network configuration topologies are described in the Network Configuration Examples found in Annex B.

Since UDP does not guarantee packet order or, in severe cases, delivery, RTP shall be used to maintain and allow reconstruction of the Data Source packet order as well as to detect loss in the network. The RTP field details are defined in Section 6.2.1 below.

6.2 RTP/UDP/IP Multicast Considerations

In addition to the context described above, it is recommended that RTP/UDP/IP multicast with SSM be used when moving multicast traffic between Data Producers and Data Consumers. Here a ‘Data Producer’ is defined as a device or process generating an IP multicast, in this case, a Data Source, and a ‘Data Consumer’ is a device or process receiving the IP multicast, in this case, the ALP encapsulation function. Since RTP defines a ‘sequence number’ field, the various packets can be recovered and correctly ordered within each Data Consumer. This allows packets to be reordered, duplicate packets to be ignored, and packet loss to be detected, possibly avoiding the problems sometimes encountered by UDP datagram delivery.

In redundant systems, the Data Consumer must be aware of the several sources of contribution multicast Streams. For each contribution Stream, an IGMPv3 SSM 'join' is issued by the Data Consumer for a particular multicast destination IP address, destination IP port number and IP source address. The combination of addresses and port numbers is referenced as a 'Tuple'. The router only sends UDP packets matching the specified Tuple requested in the join to the Data Consumer's physical IP connection. If the Data Consumer detects loss or problems with the particular packet Stream, it can release that Stream and join another using the same IP destination address and port with a different IP source address. The Producers, in this case, operate independently and with no knowledge of the redundant topology.

6.2.1 Multicast Addressing

The DSTP stream shall be broadcast using a multicast address in the range 239.0.0.0 through 239.255.255.255, which is within the Organization Local Scope range defined by IETF. The destination addresses and ports used shall be identical to those destined for the receiver environment. Source addresses and ports shall be used to map Data Source Streams into individual ALP and, thus, PLP Streams.

6.3 DSTP Design

The Data Source packet Streams transport the various signaling, audio, video, captioning and data that make up an ATSC 3.0 broadcast [4]. As indicated in the introduction to Section 4.1, the Data Source protocol creates packets with the destination addresses and ports intended to be seen by receivers. The Broadcast Gateway differentiates various data sources by using IP source address and source port. An RTP header is prepended to each packet to provide ordering and timing information required to route the Data Source Streams. RTP also allows extensions if other information is required, for example, for ECC or encryption. Placing delivery metadata in RTP allows the Data Source packet payloads to remain opaque to the transmission system. The ALP encapsulation function removes the RTP header before placing the UDP/IP packets in the ALP Stream.

The DSTP design redefines three fields of the standard RTP header. The **timestamp** and **synchronization source identifier (SSRC)** fields have been redefined to carry specific scheduling information for the Data Source packet contained within the RTP payload. The two fields define a time window, and minimum and maximum times, respectively, when the contained Data Source packet is to be emitted. The timestamp format is defined in Table 6.3.

The meaning of the **payload type** field has been redefined specifically for this application. Normally, there are payload types specified for particular types of content being streamed via RTP. This specification explicitly defines other signaling that is ultimately used by the Broadcast Gateway. The **payload type** encoding is defined in Table 6.1.

Note that Low-Level Signaling (LLS) must be indicated in various data structures of the Physical Layer. Additional signaling is not required when LLS is sent from Data Sources since the multicast IP destination address and port are always 224.0.23.60 and 4937, respectively. Different LLS Streams may be identified by using separate IP source addresses and IP source ports.

Table 6.1 provides the syntax of the RTP header for Data Source delivery.

Table 6.1 RTP Header Field Definitions for Data Source Transport Protocol

Syntax	No. of Bits	Format
RTP_Fixed_Header() {		
version (V)	2	'10'
padding (P)	1	bslbf
extension (X)	1	bslbf
CSRC_count (CC)	4	'0000'
marker (M)	1	'0'
payload_type (PT)	7	uimsbf
sequence_number	16	uimsbf
timestamp_min()	32	Table 7.3
timestamp_max()	32	Table 7.3
}		

Please refer to RFC 3550 [6] for base definitions of the fields described in Table 6.1.

version − Indicates the version number of the RTP protocol. '10' is currently defined by [6].

padding −Indicates, when set to '1', that padding is included in the payload. No padding is included when the value is '0'. Padding shall be supported as specified in [6].

extension −Indicates, when set to '1', that an RTP extension follows the header. When set to '0', no header extension is included. No extensions are required by this specification; however, use of the standard extension mechanism shall be allowed. In other words, processors of DSTP must not preclude use of extensions.

CSRC_count − Indicates the number of additional contributing source identifiers (CSRCs). No additional CSRCs are involved in this application; therefore, this field shall be set to '0000'.

marker − A bit indicating a payload start. This bit shall be set to '0' since packet fragmentation is not performed at this layer.

payload_type –The **payload_type** encoding defined here spans **payload_type** values from 80 (0x50) through 95 (0x5f), which all are in the undefined range documented in [7]. The value of the **payload_type** for DSTP shall be as defined in Table 6.2.

Table 6.2 DSTP RTP Packet **payload_type** Encoding

Syntax	No. of Bits	Format
payload_type() {		
prefix	3	'101'
reserved	2	'11'
wakeup_control()	2	Section 6.4
}		

wakeup_control() – A 2-bit field that communicates information needed to control the emission wakeup field. See Section 6.4 for a definition of this field. Note that the **wakeup_control()** field shall be ignored for packets with addresses and ports other than 224.0.23.60 and 4937, respectively.

sequence_number − As per [6], the **sequence_number** shall increment by one, modulo 2^{16} or 65536, for each packet with the same IP address and port Tuple. The initial **sequence_number** should be randomized, although this is not important in this setting since the stream will not be restarted frequently nor is it expected to be on a public network. Since UDP/IP does not guarantee ordered delivery and may even duplicate packets, the **sequence_number** can be used to reconstitute any stream as produced and to detect loss, allowing a backup stream to be selected.

timestamp_min − Defined by Table 6.3, this value shall specify the earliest time at which the start of the payload may be delivered. A value of '0' shall indicate that the packet may be delivered with best effort.

timestamp_max − Defined in Table 6.3, this value shall indicate the latest time at which the start of the payload may be delivered. A value of '0' shall denote that the packet may be delivered with best effort. Note that this field replaces the SSRC_ID field specified in RFC 3550 [6].

Table 6.3 Timestamp Field Definitions for Data Source Transport

Syntax	No. of Bits	Format
timestamp () {		
seconds	16	uimsbf
fraction	16	uimsbf
}		

timestamp fields shall be formatted according to the short-form of NTP specified in RFC 5905 [15].

seconds shall carry a value equal to the 16 least significant bits (LSBs) of the seconds portion of the UTC time value of the targeted Bootstrap Reference Emission Time.

fraction shall carry a 16-bit fractional seconds value of the UTC time of the targeted Bootstrap Reference Emission Time—allowing a resolution of approximately 15 microseconds.

The timestamp fields within the RTP header allow a Data Source system to specify schedule constraints on delivery of packets. Presumably, this scheduling information would be communicated somehow from upstream systems.

Please note that NTP is based on UTC and thus is adjusted for leap seconds. There is no adjustment for leap seconds in emission timing, which is based purely on TAI seconds and fractional seconds. See Section 8.3.2 for more information on Bootstrap emission timing.

6.4 Emergency Alert Wakeup RTP Controls

Emergency Alert Wakeup RTP fields in DSTP provide signaling information that indicates to the Scheduler how to manage the **ea_wake_up_1** and **ea_wake_up_2** bits in the Bootstrap. In the text that follows, these two bits are treated together as the "Emission Wakeup Field". There are two flags provided in the **wakeup_control()** field. The first, **wakeup_active**, indicates that a currently-active Advanced Emergency Alert (AEA) message has the @wakeup attribute set to "true". The flag shall reflect this condition on the DSTP RTP headers of all LLS packets generated by a given source, not just on packets containing an Advanced Emergency Alert Table (AEAT). In contrast, the

AEAT_wakeup_alert flag shall reflect specific values within the current packet and apply only to packets containing an AEAT.

Table 6.4 RTP **payload_type** Wakeup Control Field Definition

Syntax	No. of Bits	Format
wakeup_control() {		
wakeup_active	1	bslbf
AEAT_wakeup_alert	1	bslbf
}		

wakeup_active – A 1-bit flag that, when set to '1', shall indicate that the source providing the LLS data is requesting that the Emission Wakeup Field be non-zero. When set to '0', this flag shall indicate that the source providing the LLS data is requesting that the Emission Wakeup Field be set to a value of zero if no other source is requesting a non-zero value.

AEAT_wakeup_alert – A 1-bit flag that, when set to '1', shall indicate that the packet contains a new or updated AEAT that carries a new Wakeup Alert. When set to '0', it shall indicate that the packet does not contain an AEAT with a new Wakeup Alert. This bit shall only be '1' when the **LLS_table_version** of the AEAT LLS_table() has been incremented and the AEAT contains an AEA element with the @wakeup attribute newly set to 'true'. In other words, if a new AEA is added with @wakeup set to 'true' or a previous AEA has been updated with @wakeup set to 'true', then the **LLS_table_version** will be incremented and the **AEAT_wakeup_alert** shall be set to '1'.

The **wakeup_active** field controls whether the Emission Wakeup Field is 'off' (zero) or 'on' (non-zero). **AEAT_wakeup_alert** controls when the Emission Wakeup Field toggles through the 'on' (non-zero) states, i.e., 1, 2, and 3. (See Table 6.5 for an example of **wakeup_active** and **AEAT_wakeup_alert** usage.) It is expected that LLS tables, such as the SLT or SystemTime tables, will be updated relatively frequently, allowing timely control. There is no restriction, however, on providing extra LLS signaling to accommodate updating the Emission Wakeup Field. For example, the Data Source could emit an SLT specifically to return the Emission Wakeup Field value to zero.

Table 6.5 Example Wakeup Bit Controls

Time	LLS Table	Wakeup State	wakeup_active	AEAT_wakeup_alert	Emission Wakeup Field
t0	SLT	Off	0	0	00
t1	SystemTime	Off	0	0	00
t2	New AEA @wakeup=true	On	1	1	01
t3	SLT	On	1	0	01
t4	Same AEA	On	1	0	01
t5	SystemTime	On	1	0	01
t6	Updated AEA @wakeup=true	On	1	1	10
t7	SLT	On	1	0	10
t8	SLT	Off	0	0	00
t9	New AEA @wakeup=false	Off	0	0	00

Note that this example is for a single source of LLS. If there is more than one source of LLS, the Scheduler is responsible for combining those controls and setting the Emission Wakeup Field appropriately.

7. ALP TRANSPORT PROTOCOL (ALPTP)

The ALP delivery structure uses the ALP Transport Protocol (ALPTP), which applies between ALP sources and the Broadcast Gateway input when the ALP encapsulation function is external to the Broadcast Gateway. There are different Data Sources for each PLP, and ALP packets carry a single source data Stream for each individual PLP. Any source multiplexing is upstream of ALP encapsulation and header compression functions.

All ALP packets shall be carried with RTP/UDP/IP multicast IPv4 protocol using the RTP header modifications described in Section 7.3.

Identification of ALP packet Streams is required to tie together packets of specific ALP Streams and to provide identification of all packets in each ALP Stream. Port numbers shall be used to differentiate ALP packets and to associate them with their respective ALP Streams.

The system diagram in Figure 4.2 provides the functional context of the ALP Transport Protocol. It is possible that the actual implementation may be much more complicated, requiring multiple sources of ALP packets. Such sources could be for redundancy or could be external to the studio.

For example, consider a Transmitter-sharing agreement in which there are two Physical Layer Pipes (PLPs) defined, with each of those PLPs being dedicated to one of two separate stations. The individual studios could produce separate ALP packet Streams that are then delivered for injection into the shared Transmitter. If the studios were separated geographically, some sort of robust connection would be required to deliver the ALP packet Stream to the Scheduler / framing subsystem.

The functionality of the ALP standard [5] is sufficient for an emission link-layer protocol but should not be burdened with additional functionality to support intra- and inter-studio routing and distribution.

7.1 Overview

ALP packets are distributed over UDP/IP multicast using a modified RTP [6] protocol. UDP/IP multicast is used successfully in many industries to deliver redundant video Streams over copper and fiber Physical Layers. Routing and redundancy can be accomplished easily using IGMPv3 Source-Specific Multicast (SSM) (RFC 3569) [19]. Examples of various Network configuration topologies are described in the Network Configuration Examples found in Annex B.

A multicast IP address is reserved along with a set of ports to map particular ALP packet Streams to specific Physical Layer Pipes (PLPs) (see Section 7.2 below). By dedicating an address and ports to specific PLPs, distribution, routing, and recovery of data Streams intended for Transmission on specific PLPs is made easier.

Since UDP does not guarantee packet order or, in severe cases, delivery, RTP shall be used to maintain and allow reconstruction of the ALP packet order as well as to detect loss in the Network. Because ALP packets can exceed the Maximum Transfer Unit (MTU) size of most IP installations, typically 1500 bytes, RTP framing features are used to allow systems to reconstruct ALP packets easily upon reception. The RTP field details are defined in Section 7.3 below.

7.2 RTP/UDP/IP Multicast Considerations

In addition to the context described above, it is recommended that RTP/UDP/IP multicast with SSM be used when moving multicast traffic between Data Producers and Data Consumers. Since RTP defines a 'sequence number' field, the various packets can be recovered and correctly ordered within each Data Consumer. This allows packets to be reordered, duplicate packets to be ignored, and packet loss to be detected, possibly avoiding the problems sometimes encountered by UDP datagram delivery. RTP has framing information allowing relatively easy reconstruction of the enclosed ALP packet Stream.

In redundant systems, the Data Consumer must be aware of the several sources of contribution multicast Streams. For each contribution Stream, an IGMPv3 SSM 'join' is issued by the Data Consumer for a particular multicast IP destination address, IP destination port number and source IP address. The combination of addresses and port numbers is referenced as a 'Tuple'. The router only sends UDP packets matching the specified Tuple to the Data Consumer's physical IP connection. If the Data Consumer detects loss or problems with the particular packet Stream, it can release that Stream and join another using the same destination address and port with a different source address. The Producers, in this case, operate independently and with no knowledge of the redundant topology.

7.2.1 Address Assignments

IPv4 packet format and addressing shall be used exclusively on the ALPTP link. The ALPTP Stream shall be broadcast using a multicast address in the range 239.0.0.0 through 239.255.255.255 which is within the Organization Local Scope range defined by IETF.

7.2.2 Port Assignments

Each multicast destination address has usable port numbers of 1 through 65535. IP port numbers 30000 through 30063 shall be used corresponding to PLP 0 through 63, respectively. All packets have defined port numbers within a single IP address. Packet types are described in Section 7.3.

7.3 ALPTP Design

ALP describes the link-layer encoding for emitting various types of packets of a broadcast [5]. As indicated in the introduction to Section 7 above, the ALP protocol should not be burdened with routing requirements within the broadcast facility since this data is irrelevant to broadcast receivers. RTP is used to provide ordering, timing and framing information required to route ALP Streams. RTP also allows extensions if other information is required, for example, for ECC or encryption. Placing delivery metadata in RTP allows the ALP data payloads to remain opaque to the Transmission system.

The ALPTP design redefines one field of the standard RTP header and defines specific usage for three others. The **marker (M)** bit is defined by the RTP standard, RFC 3550 [6], to indicate significant events such as frame boundaries. The marker bit in ALPTP shall indicate that an ALP packet starts at the first byte following the RTP header fields. This is because the typical UDP/IP MTU size is 1500 bytes and ALP allows much larger packets. The **marker (M)** bit as well as the **sequence number** field of the RTP header can be used to resynchronize with the ALP headers and to determine the ordering of packets.

The **timestamp** and **synchronization source identifier (SSRC)** fields have been redefined to carry specific scheduling information for the ALP packets contained within the RTP payload. The two fields define a time window, and minimum and maximum times, respectively, when the contained ALP packet(s) is (are) to be emitted. The timestamp format is defined in Table 7.3.

Finally, the meaning of the **payload_type** field has been defined specifically for this application. Normally, there are payload types specified for particular types of content being streamed via RTP. This specification extends the dynamic payload types to explicitly define types of ALP content being carried. Some Low-Level Signaling (LLS) must be indicated in various data structures of the Physical Layer. The **payload_type** identifies these specific ALPTP packets so that they may be handled appropriately. The **payload_type** encoding is defined in Table 7.2.

Table 7.1 provides the syntax of the RTP header for ALP delivery.

Table 7.1 RTP Header Field Definitions for ALP Encapsulation

Syntax	No. of Bits	Format
RTP_Fixed_Header() {		
version (V)	2	'10'
padding (P)	1	bslbf
extension (X)	1	bslbf
CSRC_count (CC)	4	'0000'
marker (M)	1	bslbf
payload_type (PT)	7	uimsbf
sequence_number	16	uimsbf
timestamp_min()	32	Table 7.3
timestamp_max()	32	Table 7.3
}		

Please refer to RFC 3550 [6] for base definitions of the fields described in Table 7.1.

version − The version number of the RTP protocol. ‘10’ is currently defined by [6].

padding − Indicates, when set to ‘1’, that padding is included in the payload. No padding is included when the value is ‘0’. Padding shall be supported as specified in [6].

extension − Indicates, when set to ‘1’, that an RTP extension follows the header. When set to ‘0’, no header extension is included. No extensions are required by this specification; however, use of the standard extension mechanism shall be allowed. In other words, processors of ALPTP shall not preclude use of extensions.

CSRC_count − Indicates the number of additional contributing source identifiers (CSRCs). No additional CSRCs are involved in this application; therefore this field shall be set to ‘0000’.

marker − A bit indicating a payload start. This bit shall be set to ‘1’ to indicate that an ALP Packet starts at the first byte after the RTP header and any extensions; it otherwise shall be set to ‘0’. A Data Consumer can synchronize with the packet Stream by looking for the first packet containing a **marker** bit set to ‘1’. If the **marker** bit is not set to ‘1’, then the packet is a continuation of the previous packet payload as long as the **sequence_number** is monotonically increasing modulo 2^{16} or 65535.

payload_type –The **payload_type** encoding defined here spans **payload_type** values from 80 (0x50) through 95 (0x5f), which all are in the undefined range documented in [7]. Table 7.2 shall define additional payload types for ALP Transmission.

Table 7.2 ALP RTP Packet **payload_type** Encoding

Syntax	No. of Bits	Format
payload_type() {		
prefix	3	'101'
alp_table_flag	1	bslbf
if (**alp_table_flag** == '0') {		
reserved	2	'11'
table_type	1	bslbf
}		
else {		
lls_present_flag	1	bslbf
if (**lls_present_flag** == '0') {		
reserved	2	'11'
}		
else {		
wakeup_control()	2	Sec. 6.4
}		
}		
}		

alp_table_flag – A 1-bit flag that, when set to ‘0’, shall indicate that the remaining bits describe a particular ALP metadata structure, as opposed to the contents of an ALP data packet. When set to ‘1’, indicates that the remaining bits describe packets other than ALP signaling. Note that ALP metadata structures are carried in RTP packets separate from RTP packets carrying ALP data packets.

table_type – A 1-bit flag that shall indicate whether the RTP packet contains an LMT ‘0’ or an RDT ‘1’. Thus an entire **payload_value** of 80 (0x50) corresponds to an LMT while 81 (0x51) indicates that an RDT is contained in the RTP packet.

lls_present_flag – A 1-bit flag that shall indicate that the ALP packet payload contains at least one instance of a Low-Level Signaling table in compliance with [4]. A value of ‘1’ shall indicate that the packet contains a Low-Level Signaling table. A value of ‘0’ shall indicate that no LLS is present in the ALP payload. Thus, an entire **payload_type** value of 89 (0x59) indicates that LLS is present in the RTP packet. A **payload_type** value of 88 (0x58) indicates all other ALP payloads that do not contain ALP tables or LLS.

wakeup_control() – A 2-bit field that communicates information needed to control the Emission Wakeup Field. See Section 6.4 for a definition of this field. This field directly reflects the values provided as input to the ALP generator from the Data Source providing LLS (see Section 6.4 for a description of the data source signaling of the wakeup_control() field).

sequence_number – As per [6], the **sequence_number** shall increment by one, modulo 2^{16} or 65536, for each packet from the same source. The initial **sequence_number** should be randomized, although this is not important in this setting since the Stream will not be restarted frequently nor is it expected to be on a public Network. Since UDP/IP does not guarantee ordered delivery and may even duplicate packets, the **sequence_number** can be used to reconstitute the Stream as produced and to detect loss, allowing a backup Stream to be selected.

timestamp_min − Defined by Table 7.3, this value shall specify the Earliest Time at which the start of the payload may be delivered. A value of '0' shall indicate that the packet may be delivered with best effort. Timestamp times are valid only when the **marker** bit is set to '1', indicating that the time is associated with the beginning of the ALP packet. The **timestamp_min** value shall be ignored when the **marker** bit is not '1'.

timestamp_max − Defined in Table 7.3, this value shall indicate the Latest Time at which the start of the payload may be delivered. A value of '0' shall denote that the packet may be delivered with best effort. Timestamp times are valid only when the **marker** bit is set to '1', indicating that the time is associated with the beginning of the ALP packet. The **timestamp_max** field shall be ignored when the **marker** bit is not '1'. Note that this field replaces the SSRC_ID field specified in RFC 3550 [6].

Table 7.3 Timestamp Field Definitions for ALP Encapsulation

Syntax	No. of Bits	Format
timestamp () {		
seconds	16	uimsbf
fraction	16	uimsbf
}		

timestamp fields shall be formatted according to the short-form of NTP specified in RFC 5905 [15].

seconds shall carry a value equal to the 16 least significant bits (LSBs) of the seconds portion of the UTC time value of the targeted Bootstrap Reference Emission Time.

fraction shall carry a 16-bit fractional seconds value of the UTC time of the targeted Bootstrap Reference Emission Time–allowing a resolution of approximately 15 microseconds.

The timestamp fields within the RTP header allow an ALP encapsulation system to specify schedule constraints on delivery of the contained packets. Presumably, this scheduling information would be communicated somehow from upstream systems.

Please note that NTP is based on UTC and thus is adjusted for leap seconds. There is no adjustment for leap seconds in emission timing, which is based purely on TAI seconds and fractional seconds. See Section 8.3.2 for more information on Bootstrap emission timing.

8. STL TRANSPORT PROTOCOL

The STL Transport Protocol (STLTP) shall consist of an outer "Tunneling" streamed layer and an inner layer consisting of a number of "tunneled" Streams. Each of the Streams shall carry sequences of packets comprising an RTP/UDP/IP multicast header plus payload data. All packets delivered over the STL shall use an RTP/UDP/IP multicast IPv4 protocol.

8.1.1 Address Assignments

IPv4 packet format and addressing shall be used exclusively on the STL. The multicast (destination) address range is 224.0.0.0 – 239.255.255.255. Of that range, 239.0.0.0 – 239.255.255.255 are for private addresses and shall be used for both inner tunneled and outer Tunneling packets.

8.1.2 Port Assignments

Each multicast destination address has 1 – 65535 usable port numbers. Values of 30000 – 30065 inclusive shall be used for this standard. Ports 30000 through 30063 shall be used to identify inner tunneled Streams having destinations of PLPs 0 through 63, respectively; port 30064 shall be used to identify the inner tunneled Stream carrying the Preamble data; and port 30065 shall be used to identify the inner tunneled Stream carrying the Timing and Management Data.

8.2 Preamble Generator

Preamble information is constructed in the Preamble Generator according to instructions from the Scheduler. It is output by the Preamble Generator in the form of RTP/UDP/IP multicast packets, similar to those used to carry Baseband Packets (BBPs) in PLP Streams, so that they form a Stream that can be multiplexed together with the PLP Streams in the STLTP. The Preamble Stream carries a description of the configuration of the Transmitter processing functions and the resulting emitted waveform that is identical to the Preamble data structure that is included in an ATSC 3.0 transmitted waveform.

To set up Transmitter configurations, the Preamble must be sent from the Scheduler to arrive at the Transmitter(s) at least 1 Physical Layer frame in advance of the start of construction by the Transmitter(s) of the frame that the Preamble describes.

To enable receivers to decode transmitted data, the same Preamble data used to configure the Transmitter(s) for a particular frame shall be carried in the emitted preamble waveform immediately following the Bootstrap of that frame.

8.2.1 Preamble Data Stream Protocol

The Preamble data shall be delivered in an RTP/UDP/IP multicast Stream conforming to RFC 3550 [6] with the constraints defined below. The maximum Preamble data structure size can exceed the typical 1500-byte MTU, so a mechanism is defined herein to allow segmentation of the Preamble data across multiple RTP/UDP/IP packets. Note that such segmentation is only required to conform with typical MTU sizes of 1500 bytes. If the local Network allows larger multicast packets, this segmentation may not be needed.

The payload data for each Preamble Stream RTP/UDP/IP packet shall be a fragment of the Preamble Payload data structure described in Table 8.1. To provide validation that the L1_Basic_signaling and L1_Detail_signaling structures are delivered correctly over the STL, a 16-bit cyclic redundancy check is provided. The CRC is applied to the combined **length** field, L1_Basic_signaling and L1_Detail_signaling, and appended as the last 16 bits of the payload data. The resultant Stream of Preamble Payload packets shall have destination address 239.0.51.48 and destination port 30064.

Table 8.1 Preamble Payload

Syntax	No. of Bits	Format
Preamble_Payload () {		
length	16	uimsbf
L1_Basic_signaling ()	200	Table 9.2 of [3]
L1_Detail_signaling ()	var	Table 9.12 of [3]
crc16	16	uimsbf
}		

The following paragraphs describe the fields shown in Table 8.1.

The **length** field shall contain the number of bytes in the Preamble Payload data structure following the **length** field excluding the **crc16** bytes. The **length** field allows Data Consumers to avoid knowing the detailed syntax of Preamble data structures and still to reconstruct the overall Preamble Payload data structure.

The **crc16** field shall be the value resulting from application of a 16-bit cyclic redundancy check, as defined in [12], applied to the combined **length**, L1_Basic_signaling() and L1_Detail_signaling() structures in the Preamble Payload immediately preceding the **crc16** field.

The Preamble Generator shall form the necessary Preamble Payload data, as detailed in Table 8.1, from the Scheduler configuration and calculated information. Once the data structure has been populated, it shall be partitioned, if necessary, into multiple RTP/UDP/IP packets, each conforming, with the necessary headers, to the local Network MTU size. This process results in creation of a Preamble Payload Packet Set that typically consists of multiple packets of the same size followed by a smaller remainder packet. Constructing the packets in the way described in the preceding sentence, however, is not normative.

The RTP header fields of the Preamble Payload Packet Set shall be as described below, configured with the **marker (M)** bit of the packet containing the beginning of a Preamble Payload data structure set to one '1'. The **marker (M)** bits of remaining packets shall be set to zero '0'. This allows the Transmission system on the Data Consumer end of the STL to reconstruct the Preamble Payload data structure after any resequencing takes place. The timestamps of the packets of a given Preamble Payload Packet Set shall have the same values. The timestamp values are derived from a subset of the Bootstrap_Timing_Data, providing a mechanism to uniquely associate each of the Preamble Payload packets with a specific Physical Layer frame. The format of the timestamp field is described in Table 8.2.

Table 8.2 RTP Header Timestamp Field Definitions

Syntax	No. of Bits	Format
timestamp () {		
seconds_pre	22	uimsbf
a-milliseconds_pre	10	uimsbf

seconds_pre shall carry a value equal to the 22 least significant bits (LSBs) of the **seconds** field of the Bootstrap_Timing_Data(), described in Table 8.3.

a-milliseconds_pre shall carry a 10-bit value identical to the value contained in the 3rd through 12th MSBs of the **nanoseconds** field of the Bootstrap_Timing_Data(), described in Table 8.3. Note that the **a-millisecond_pre** value is used in the RTP Header Timestamp only as an identifier of the Bootstrap Reference Emission Time of the Frame in which its contents belong; consequently, the somewhat longer Period of an **a-millisecond_pre** relative to precisely one millisecond is immaterial for this use.

On receipt, the Preamble Payload Packet Set shall be correctly ordered and extracted into the Preamble Payload data structure as described in Table 8.1. The Data Consumer can accumulate RTP packets until it has received all of the bytes defined by the length field in the first packet. If a packet is missed, as determined by a missing sequence number, or if a packet with the **marker (M)** bit set to '1' is received prematurely, indicating the start of the next Preamble Payload Packet Set, then one or more packets have been lost and the entire Preamble Payload data set has been lost. Any accumulated data shall be discarded.

The RTP header fields shall follow the syntax defined in RFC 3550 [6], with the following additional constraints:

The **Padding (P)** bit shall conform to the RFC 3550 [6] specification.

The **Extension (X)** bit shall be set to zero '0', indicating the header contains no extension.

The **CSRC Count (CC)** shall be set to zero '0', as no CSRC fields are necessary.

The **marker (M)** bit shall be set to one '1' to indicate that the first byte of the payload is the start of the Preamble Payload data. A zero '0' value shall indicate that the payload is a continuation of the Preamble Payload data from the previous packet.

The **Payload Type (PT)** shall be set to 77 (0x4d), indicating the Preamble Payload type.

The **Sequence Number** shall conform to the RFC 3550 [6] specification.

The **Timestamp** shall be defined as in Table 8.2. The timestamp shall be set to the same value for all of the Preamble Payload Packet Set.

The **Synchronization Source (SSRC) Identifier** shall be set to zero '0'. There shall be no other sources of Preamble Payload data carried within an STLTP Stream. Any redundant sources can be managed using IGMP Source-Specific Multicast (SSM) mechanisms.

8.3 Timing and Management Generator

Timing and Management information is constructed in the Timing and Management Generator according to instructions from the Scheduler. It is output by the Timing and Management Generator in the form of RTP/UDP/IP multicast packets, similar to those used to carry Baseband Packets (BBPs) in PLP Streams, so that they form a Stream that can be multiplexed together with the PLP Streams in the STLTP. These Timing and Management packets are not emitted. The resulting Timing and Management Stream carries a set of instructions for controlling the emission of Physical Layer frames comprising a Bootstrap, Preamble, and payload data. Configurations of Bootstraps and certain other components of the Physical Layer frames are carried in the Timing and Management Stream. Also included in the Timing and Management Stream are the emission time of each Bootstrap and, hence, the start of each Physical Layer frame, the offset times of each Transmitter in an SFN from the Bootstrap Reference Emission Times for the Network, and other information used to control the Transmitter(s).

To set up Transmitter configurations, the Timing and Management Data for a Physical Layer frame must be sent from the Scheduler to arrive at the Transmitter(s) at least 1 Physical Layer frame in advance of the start of construction by the Transmitter(s) of the Physical Layer frame that the Timing and Management Data references.

8.3.1 Timing and Management Data Stream Protocol

The Timing and Management Data shall be delivered in an RTP/UDP/IP multicast Stream conforming to RFC 3550 [6] with the constraints defined below. The maximum Timing and

Management data structure size may exceed the typical 1500-byte MTU, so a mechanism is defined herein to allow segmentation of the Timing and Management data across multiple RTP/UDP/IP packets. Note that such segmentation is required only to conform with typical MTU sizes of 1500 bytes. If the local network allows larger multicast packets, this segmentation may not be needed.

The payload data for each Timing and Management Stream RTP/UDP/IP packet shall be a fragment of the TMP() data structure described in Table 8.3. To provide validation that the TMP() structure is delivered correctly over the STL, a 16-bit cyclic redundancy check is provided as part of the TMP() data. The resultant stream of TMP() packets shall have IP destination address 239.0.51.48 and destination port 30065.

The Timing and Management Generator shall form the necessary TMP() data, as detailed in Table 8.3, from the Scheduler configuration and calculated information. Once the data structure has been populated, it shall be partitioned, if necessary, into multiple RTP/UDP/IP packets, each conforming, with the necessary headers, to the local network MTU size. This process results in creation of a TMP Packet Set that typically consists of multiple packets of the same size followed by a smaller remainder packet. Constructing the packets in the way described in the previous sentence, however, is not normative.

The RTP header fields of the TMP Packet Set shall be as described below, configured with the **marker (M)** bit of the packet containing the beginning of a TMP() data structure set to one '1'. The **marker (M)** bits of remaining packets shall be set to zero '0'. This allows the transmission system on the consumer end of the STL to reconstruct the TMP() data structure after any ordering correction takes place. The timestamps of the packets of a given TMP Packet Set shall have the same values. The timestamp values are derived from a subset of the Bootstrap_Timing_Data, providing a mechanism to uniquely associate each of the TMP packets with a specific Physical Layer frame.

The RTP header fields shall follow the syntax defined in RFC 3550 [6] with the following additional constraints:

The **Padding (P)** bit shall be set to zero '0', indicating no padding is present in the Timing and Management Data packet.

The **Extension (X)** bit shall be set to zero '0', indicating the header contains no extension.

The **CSRC Count (CC)** shall be set to zero '0', as no CSRC fields are necessary.

The **marker (M)** bit shall be set to one '1' to indicate that the first byte of the payload is the start of the TMP data. A zero '0' value shall indicate that the payload is a continuation of the TMP data from the previous packet.

The **Payload Type (PT)** shall be set to 76 (0x4c) indicating the Timing and Management Data payload type.

The **Sequence Number** shall conform to the RFC 3550 [6] specification.

The **Timestamp** shall be defined as in Table 8.2.

The **Synchronization Source (SSRC) Identifier** shall be set to zero '0'. There should be no other sources of Timing and Management Data carried by the STLTP. Any redundant sources can be managed using IGMP Source-Specific Multicast (SSM) mechanisms.

Table 8.3 Timing and Management Stream Packet Payload

Syntax	No. of Bits	Format
Timing & Management_Packet (TMP) () {		
Structure_Data () {		
length	16	uimsbf
version_major	4	uimsbf
version_minor	4	uimsbf
maj_log_rep_cnt_pre	4	uimsbf
maj_log_rep_cnt_tim	4	uimsbf
bootstrap_major	4	uimsbf
bootstrap_minor	4	uimsbf
min_time_to_next	5	uimsbf
system_bandwidth	2	uimsbf
bsr_coefficient	7	uimsbf
preamble_structure	8	uimsbf
ea_wakeup	2	bslbf
num_emission_tim	6	uimsbf
num_xmtrs_in_group	6	uimsbf
xmtr_group_num	7	uimsbf
maj_log_override	3	bslbf
num_miso_filt_codes	2	bslbf
tx_carrier_offset	2	tcimsbf
reserved	6	for (i=0; i<6; i++) ‘1’
}		
Bootstrap_Timing_Data () {		
for (i=0; i<=num_emission_tim; i++)		
seconds	32	uimsbf
nanoseconds	32	uimsbf
}		
}		
Per_Transmitter_Data () {		
for (i=0; i<=num_xmtrs_in_group; i++) {		
xmtr_id	13	uimsbf
tx_time_offset	16	tcimsbf
txid_injection_lvl	4	uimsbf
miso_filt_code_index	2	bslbf
reserved	29	for (i=0; i<29; i++) ‘1’
}		
}		
Packet_Release_Time () {		
pkt_rls_seconds	4	uimsbf
pkt_rls_a-milliseconds	10	uimsbf
reserved	2	‘11’
}		
Error_Check_Data () {		
crc16	16	uimsbf
}		
}		

length shall indicate the number of bytes in the Timing and Management Data packet following the RTP/UDP/IP header structure. Up to 65,535 bytes can be indicated.

version_major, in conjunction with **version_minor**, shall indicate the version of the protocol used to construct the Timing and Management Data packet. Increments in the value of **version_major** are intended to indicate changes in the structure that are not fully compatible with lower-ordered **version_major** values. Timing and Management packets constructed according to this version of this standard shall have the value of **version_major** set to 0.

version_minor, in conjunction with **version_major**, shall indicate the version of the protocol used to construct the Timing and Management Data packet. Increments in the value of **version_minor** are intended to indicate changes in the structure that are fully backward compatible with lower-ordered **version_minor** values. Timing and Management packets constructed according to this version of this standard shall have the value of **version_minor** set to 0.

maj_log_rep_cnt_pre shall indicate the number of repetitions of Preamble data in the Preamble Stream at UDP port 30064 prior to emission of the Preamble. Permitted values are 1, 3, 5, 7, and 9. Note that values for **L1B_lls_flag** and **L1D_plp_lls_flag** may be correct only in the final copy of the Preamble data sent to Transmitters prior to emission. Consequently, majority logic error correction can be applied reliably to all portions of the Preamble Stream data except the two flag values noted. See Section 9.2 for details of placement of the repeated data.

maj_log_rep_cnt_tim shall indicate the number of repetitions of Timing and Management Data in the Timing and Management Stream at UDP port 30065 prior to emission of the next Bootstrap. Permitted values are 1, 3, 5, 7, and 9. Note that values for the **ea_wakeup** bits may be correct only in the final copy of the Timing and Management Data sent to Transmitters prior to emission. Consequently, majority logic error correction can be applied reliably to all portions of the Timing and Management Stream data except the **ea_wakeup** values noted. See Section 9.1 for details of placement of the repeated data.

bootstrap_major shall indicate the value of the **bootstrap_major_version** of the Bootstrap symbols that introduce the Physical Layer frame identified by the Bootstrap_Timing_Data(), which value shall indicate the root of the Zadoff-Chu sequence of the Bootstrap symbols, as specified in [2].

bootstrap_minor shall indicate the value of the **bootstrap_minor_version** of the Bootstrap symbols that introduce the Physical Layer frame identified by the Bootstrap_Timing_Data(), which value shall indicate the seed for the PN sequence of the Bootstrap symbols, as defined in [2].

min_time_to_next shall be the enumerated value indicating the minimum time until the next frame of the same type as defined in [2].

system_bandwidth shall be the enumerated value indicating the bandwidth of the RF Transmission channel as defined in [2].

bsr_coefficient shall be the binary value associated with the baseband sampling frequency as defined in [2].

preamble_structure shall be the enumerated value indicating the Preamble configuration as defined in [3].

num_emission_tim shall indicate the number of sequential Bootstrap Reference Emission Times that are contained within the Bootstrap_Timing_Data() 'for' loop. Up to 64 values may be indicated. The values shall range from 0 thru 63, and shall be expressed as the number of values carried in the packet minus 1. At least the next Bootstrap Reference Emission Time shall be carried, and it shall be carried in index 0 of the 'for' loop.

ea_wakeup shall signal the states of the two ea wakeup bits to be included in the Bootstrap signal at the start of the next Physical Layer frame to be emitted.

num_xmtrs_in_group shall indicate the number of Transmitters minus one to which data is addressed in the Per_Transmitter_Data() 'for' loop (e.g., 1 to 64 Transmitters are indexed 0 to 63). The value can be less than the total number of Transmitters in the Network, in which case data addressed to groups of Transmitters shall be sequenced in order across multiple Timing and Management Data packets.

xmtr_group_num shall indicate the ordinal number of a group of Transmitters to which information in the Per_Transmitter_Data() loop is addressed. The value of the field may range from 0 through 127. Only a single value of **xmtr_group_num** shall apply to a given Timing and Management Stream data packet. Information for individual Transmitters shall be organized in groups identified by values of **xmtr_group_num** starting at 0 and incrementing by 1 from one Timing and Management Stream data packet to the next, until the highest-numbered group is reached, at which point the value shall start again at 0 in the following such packet.

Bootstrap_Timing_Data() shall contain a list of the Bootstrap Reference Emission Times of the next and, optionally, successive future frames, the list having a total number of entries equaling the value of **num_emission_tim**. The values of the Bootstrap Reference Emission Times shall increase monotonically from the first entry in the list to the last.

maj_log_override shall indicate that all previous instances of Timing and Management Data and Preamble data for the next and following Physical Layer frames shall be ignored and that the information in the current Timing and Management packet and a subsequent Preamble data packet shall be used to configure the next Physical Layer frame. The non-override condition shall be indicated by a value of '000' in this field. An override condition shall be indicated by a value of '111' in this field. Other values are reserved.

num_miso_filt_codes shall indicate the number of different MISO filter codes in use within an SFN, as represented by the variable 'M' in Annex J of A/322 [3].

tx_carrier_offset shall indicate the carrier offset of the Transmitter(s) in the frequency domain. The carrier offset shall be expressed in units of a positive or negative integer number of carriers, and it shall be a two's complement signed integer binary number having a range from –1 to +1 decimal, representing from –1 to +1 OFDM carriers. The carrier offset value shall be equal to the product of the value of **tx_carrier_offset** and the carrier frequency spacing in Hz of an 8K FFT for the value of **bsr_coefficient** indicated in the Structure_Data() for the same frame. Carrier frequency spacing (in Hz) of an 8K FFT equals BSR (in Hz) divided by 8192. For example, in a system operating with a 6 MHz channel bandwidth and a BSR of 6.192 Mega-samples/second, **bsr_coefficient**=2, the carrier frequency spacing of 8K carriers is 843.75 Hz, and the carrier frequency offset will be -843.75 Hz, 0 Hz, and +843.75 Hz for values of **tx_carrier_offset** of -1, 0, and +1, respectively. **tx_carrier_offset** = -2 shall be reserved for future use.

The **tx_carrier_offset** value also is used by the Scheduler to set the Bootstrap Reference Emission Time as described in Section 9.3.3.2 below.

seconds shall carry a value equal to the 32 least significant bits (LSBs) of the seconds portion of the TAI time value of the associated Bootstrap Reference Emission Time, as expressed using the Precision Time Protocol (PTP) defined in [13] and [14].

nanoseconds shall carry a value equal to the nanoseconds portion of the TAI time value of the associated Bootstrap Reference Emission Time. It shall be expressed as a 32-bit binary value having a range from 0 through 999,999,999 decimal.

Per_Transmitter_Data() shall contain information addressed individually to one or a group of Transmitters, with the number of Transmitters for which data is included in the loop equaling the value in **num_xmtrs_in_group.**

xmit_id shall indicate the address of the Transmitter to which the following values are being sent and shall correspond to the seed value used by the TxID code sequence generator of that Transmitter. The value of the address shall be an unsigned integer binary number having a range of possible values from 0 through 8191 decimal.

tx_time_offset shall indicate the emission time offset of the Transmitter to which it is addressed relative to the Bootstrap Reference Emission Times of all frames. The Transmitter time offset shall be expressed in units of positive or negative integer steps of 100 ns and shall be a two's complement signed integer binary number having a range from –32,768 through +32,767 decimal, representing time offsets from –3,276.8 through +3,276.7 microseconds.

txid_injection_lvl shall indicate the Injection Level of the TxID signal below the average power of the Preamble symbols emitted by the Transmitter to which its value is addressed. The Injection Level shall indicate the value in dB listed in [3] Table N.3.1 for the TxID Injection Level Code included in the **txid_injection_lvl** field (or Off for code value 0000).

miso_filt_code_index shall indicate the specific MISO filter code assigned to the individual Transmitter, as represented by the variable 'h' in A/322 Annex J [3]. The same value of MISO filter code index shall apply to a particular Transmitter regardless of whether 64-coefficient or 256-coefficient filters are in use.

pkt_rls_seconds shall be the seconds portion of the time of release from the Broadcast Gateway of the specific Timing and Management packet in which the value is found. Its value shall be expressed as 4 bits representing the 4 LSBs of the seconds value of the TAI time when the first bit of the IP header of the Timing and Management packets is released from the Broadcast Gateway.

pkt_rls_a-milliseconds shall be the milliseconds portion of the time of release from the Broadcast Gateway of the specific Timing and Management packet in which the value is found. Its value shall be expressed as 10 bits representing the 3rd through 12th MSBs of the nanoseconds value of the TAI time when the first bit of the IP header of the Timing and Management packets is released from the Broadcast Gateway. Its range will be from 0 to 953 (decimal) as a consequence of the Period of an **a-millisecond** being slightly longer than precisely a millisecond. (See the definition of an **a-millisecond** in Section 3.4.

crc16 shall be the value resulting from application of a 16-bit cyclic redundancy check, as defined in [12], applied to all fields in the Timing and Management Packet payload from the **length** field through the field (and any reserved bits) immediately preceding the **crc16** field.

8.3.2 Bootstrap Emission Timing and Frame Identification

Bootstrap Reference Emission Time shall be used for Physical Layer frame identification. In the RTP headers of the STLTP, the timestamp fields are used to carry number patterns matching portions of a complete Bootstrap Reference Emission Time. The portions matched have been selected for unique identification of the Physical Layer frames with which the STLTP packets are associated. The portions of the Bootstrap Reference Emission Time used for identification are the 22 LSBs of the seconds value and the 3rd through 12th MSBs of the nanoseconds value, the latter of which represent approximately millisecond time increments. The selected portions of the Bootstrap Reference Emission Time are sufficient to ensure no ambiguity in the association of STLTP packets with Physical Layer frames.

8.3.2.1 Baseband Packet Delivery

The Baseband Packetizer functionality accepts ALP packets from the Transport Layer as described in [5]. All ALP packets are converted to Baseband Packets as described in [3]. This complete description of the baseband data to be radiated is described in the associated Preamble. The Baseband Packetizer then encapsulates Baseband Packets into RTP/UDP/IP packets per Scheduler instructions and destination PLPs.

8.3.3 PLP Data Stream

Each PLP is parallel data, simultaneous in time and inherently packetized into Baseband Packets. The payload must be signaled for correct PLP processing. The port number used to deliver each ALP Stream, as described in Section 5.4, associates the ALP Stream with a particular PLP.

8.3.4 Baseband Packet Data Stream Protocol

The Baseband Packet data shall be delivered in an RTP/UDP/IP multicast Stream conforming to RFC 3550 [6] with the constraints defined below. The Baseband Packet data structure size may exceed the typical 1500 byte MTU, so a mechanism is defined herein to segment the Baseband Packet data across multiple RTP/UDP/IP multicast packets. Note that this is required only to conform with typical MTU sizes of 1500 bytes, should the FEC configuration have large Baseband Packets on the corresponding PLP. If the local Network allows larger multicast packets, this segmentation may not be required.

The payload data for each RTP/UDP/IP multicast packet shall be either a fragment of or the entire Baseband Packet data structure as defined in [3]. The collection of packet(s) representing a single Baseband Packet is referred to hereafter as the Baseband Packetizer Packet Set (BPPS). The resultant packet Stream shall have IP destination address 239.0.51.48 and IP destination ports 30000 through 30063, corresponding to the PLPs numbered from 0 through 63, respectively.

The Baseband Packetizer, in each path from the corresponding ALP Stream to the target PLP Stream, will form each Baseband Packet from ALP packets in the incoming ALP data Stream as described above. Once a Baseband Packet is ready for emission, it will be partitioned, if necessary, into multiple RTP/UDP/IP multicast packets each conforming, with the necessary headers, to the local Network MTU size. In summary, the resultant BPPS typically will consist of multiple packets of the same size followed by a smaller remainder packet. Constructing the packets in the way described in the previous sentence is not normative and any size or number of packets is permitted within a BPPS.

The RTP header fields of the BPPS shall be set as described below, with the **marker (M)** bit of the packet containing the beginning of the BPPS set to one ‘1’. The **marker (M)** bit of any remaining packets shall be set to zero ‘0’. This allows the Transmission system on the Data Consumer side of the STL to reconstruct the Baseband Packet after any ordering correction takes place. The timestamps of the packets of the BPPS shall be set to the same value indicating the Bootstrap Reference Emission Time of the Physical Layer frame. In addition to the **marker (M)** bit being set to ‘1’ on the first BPPS packet, the **Synchronization Source (SSRC) Identifier** field of the first BPPS packet shall be set to the overall length of the Baseband Packet data structure in bytes across the entire BPPS for a given PLP.

On receipt at the Data Consumer, the BPPS shall be correctly ordered and extracted into the Baseband Packet data structure. The Data Consumer can accumulate RTP packets until it has received all of the bytes defined by the length field in the first BPPS packet. If a BPPS packet is missed as determined by a missing sequence number or if a packet with the **marker (M)** bit set to ‘1’ is received prematurely, indicating the start of the next BPPS, then one or more packets have been lost and the entire Baseband Packet data structure has been lost. Any accumulated data shall be discarded.

Note that, depending on time interleaver modes, a Baseband Packet at the current timestamp may span into the subsequent Physical Layer frame due to the delaying nature of time interleaving. In such a case, the Baseband Packet at the current timestamp would belong to both the current and the subsequent Physical Layer frames.

The RTP header fields shall follow the syntax defined in RFC 3550 [6] with the following additional constraints:

The **Padding (P)** bit shall conform to the RFC 3550 [6] specification.

The **Extension (X)** bit shall be set to zero ‘0’ indicating the header contains no extension.

The **CSRC Count (CC)** shall be set to zero ‘0’ as no CSRC fields are necessary.

The **marker (M)** bit shall be set to one ‘1’ to indicate that the first byte of the payload is the start of the Baseband Packet data structure. A zero ‘0’ value shall indicate that the payload is a continuation of the Baseband Packet data structure from the previous BPPS packet.

The **Payload Type (PT)** shall be set to 78 (0x4e) indicating the Baseband Packet payload type.

The **Sequence Number** shall conform to the RFC 3550 [6] specification.

The **Timestamp** shall be defined as in Table 8.2. The timestamp shall be set to the same value for all packets of a given BPPS.

When the **marker (M)** bit is zero ‘0’, the **Synchronization Source (SSRC) Identifier** shall be set to zero ‘0’. When the **marker (M)** bit is set to one ‘1’, indicating the first packet of the BPPS, the **SSRC** field will contain the total length of the Baseband Packet data structure in bytes. This allows the Data Consumer to know how much data is to be delivered within the payloads of the BPPS.

8.4 Studio to Transmitter Link (STL) Protocol Overview

The STL Transport Protocol (STLTP) shall be an RTP/UDP/IP multicast Stream based on the SMPTE ST 2022-1 [8] standard. SMPTE ST 2022-1, in turn, extends the RTP [6] protocol with Generic FEC Payload Format [8]. RTCP and other control Streams are explicitly excluded by SMPTE ST 2022-1 ([8] § 7.1, ¶ 2).

To use SMPTE ST 2022-1, all STL communications shall be coalesced into a single IP Stream with a fixed packet size. This is known as ‘Tunneling’. Both single and two-dimensional FEC schemes are supported. Note that, per ST 2022-1, column and row FEC packets shall be sent on separate ports to the same multicast address as the primary data Stream. No IP header compression shall be performed. The RTP **marker (M)** bit combined with an offset field allows Data Consumers to determine where IP headers begin within a given tunnel payload. The packet Streams destined for the STL shall be collected into fixed-size RTP payloads, XOR FEC calculations shall be performed, if enabled, and the resulting packets shall be sent to the STL according to [8].

8.4.1 SMPTE ST 2022-1 Features / Resources

The STLTP shall be a derivative of SMPTE ST 2022-1 [8]. SMPTE ST 2022-1 [8] is defined to allow encapsulated MPEG-2 transport Streams to be delivered over RTP/UDP/IP in a robust manner applying various levels of FEC as desired. The sections above describe a mechanism of encapsulating various forms of IP packets instead of MPEG-2 transport Stream packets within an ST 2022-1 Stream to take advantage of the robustness provided by that protocol.

In addition to the constraints defined by [8], Section 8.6 below redefines the SSRC_ID field. This field, along with the **marker (M)** bit, provides a means of framing IP packets within the STLTP payloads for easy reconstruction on the Data Consumer side. All other aspects of ST 2022-1 remain identical to the capabilities described in [8] including support for any FEC constraints detailed in Section 8 of that standard.

Note that latency will be longer with larger packet lengths.

8.5 STL Transmission Protocol Design

Figure 8.1 provides a detailed diagram of the portion of the broadcast Physical Layer chain described by this section. Refer to Figure 4.2 for a complete diagram of the system architecture.

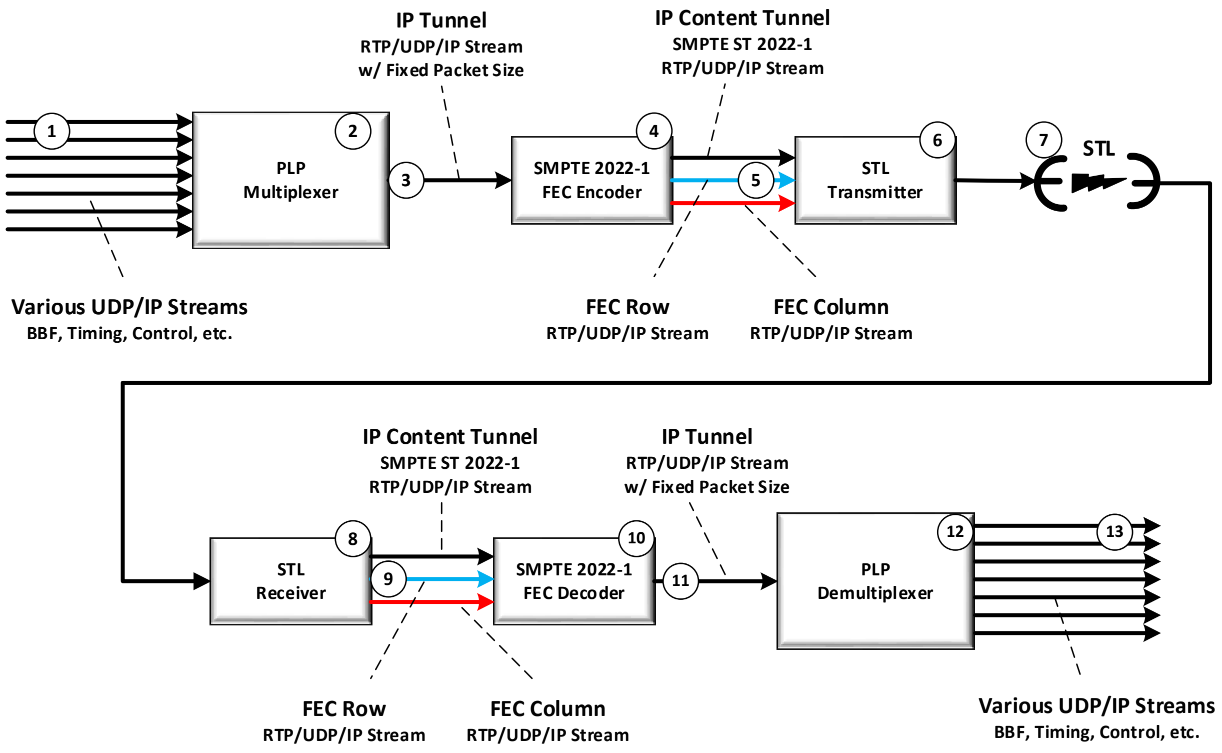

Figure 8.1 STL Transmission diagram.

The following paragraphs describe each of the call-outs, (1) through (13), in Figure 8.1. Items (1) through (5) are further detailed in the FEC Encoding Process Section 8.5.1 below.

1) The multiple paths represent the RTP/UDP/IP Streams that are generated for each PLP as described in Sections 8.2, 8.3, and 8.3.4. These are referred to as the Tunneled Packet Streams.
2) The PLP Mux is configured to accept packets from multiple RTP/UDP/IP multicast Streams to be tunneled.
3) The Tunneled Packets are grouped into fixed-size payloads to accommodate the SMPTE ST 2022-1 FEC process [8]. The STLTP RTP fields are defined to allow the Tunneled Packet Streams to be easily recovered and forwarded (refer to Section 8.6 below). The fixed packet size of the ST 2022-1 packets is not defined by this standard and is assumed to be configurable. It is expected that the packet size is within the typical 1500 MTU byte range limit. Note that the larger the packets the longer the latency when performing FEC

calculations. The resultant fixed-size RTP packets with the Tunneled Packet payloads are referred to as the Tunnel Packets.

4) The process described in [8] buffers multiple fixed-sized payloads and performs XOR operations on groups of Tunnel Packets producing additional FEC packets. The level of error correction can be configured including the number of columns ('L') and the number of rows ('D'). Level A and Level B configurations, described in [8], shall be supported with a preference for use of the 2D scheme (Level B).
5) The tunnel Stream and up to two FEC packet Streams are sent to the STL transmitter using up to three RTP/UDP/IP multicasts on three separate ports. If the tunnel Stream is sent to the STL transmitter on IP address 239.X.Y.Z, with port N, then the first FEC Stream is sent on the same address 239.X.Y.Z, with port N+2, and the second FEC Stream is also sent on the same address 239.X.Y.Z, with port N+4. This is in accordance with SMPTE ST 2022-1 ([8] § 8.1, ¶ 6).
6) The three Streams are processed by an IP-capable STL Transmitter where they are
7) sent via the STL to
8) the STL Data Consumer where they are extracted into
9) up to three RTP/UDP/IP Streams; i.e. the Tunnel Packet Stream and up to two FEC Streams.
10) These Streams are processed by the SMPTE ST 2022-1 decoder. Note that the decoder detects missing packets from the RTP sequence numbers on the tunnel Stream and reconstitutes them from the FEC Stream(s).
11) The resultant Tunnel Packet Stream is an in-order collection of fixed-size packets identical to the input Stream described in step (3) above.
12) The PLP Demux extracts each packet Stream from the payloads of the Tunnel Packets. The Tunnel Packet RTP headers contain information allowing the packet Streams to be reframed.
13) The reconstituted IP packets then are forwarded to the remaining stages of the Transmission system.

8.5.1 Example SMPTE 2022-1 FEC Encoding Process

Figure 8.2 provides a detailed example diagram of how the various BPPS, Preamble, and Timing and Management packet Streams are encoded into STLTP Tunnel Packets and ST 2022-1 [8] FEC packets. This exemplifies some of the steps shown in Figure 8.1. The process shown in Figure 8.2 implements steps 2, 3, and 4 from Figure 8.1. This example shows a combination of the PLP Multiplexer function and the FEC Encoder function operating on a shared memory area. A similar decode process occurs where these packets are gathered into memory, FEC is used to reconstitute missing packets, and the Tunneled Packets will be extracted directly from the in-memory buffers.

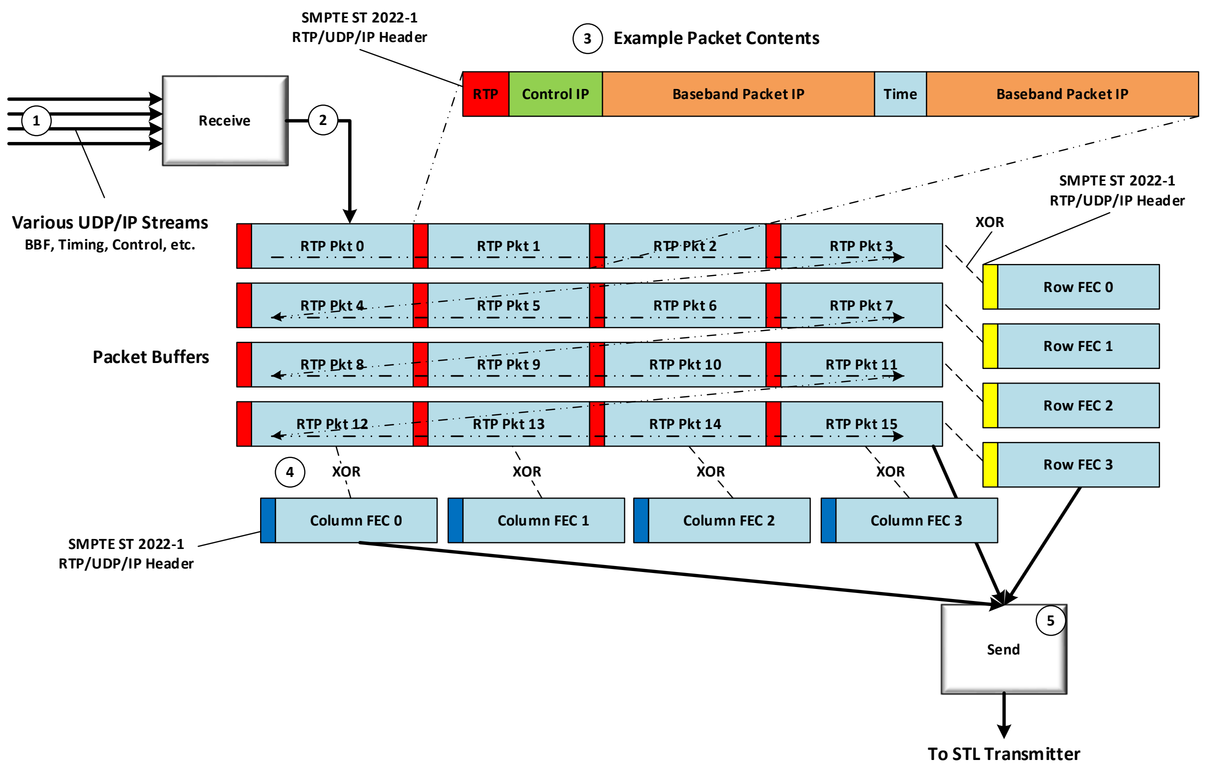

Figure 8.2 Example FEC encoding process diagram.

The following paragraphs describe each of the call-outs from Figure 8.2.

1) The multiple paths represent the RTP/UDP/IP Streams that are generated for each PLP as described in Sections 8.2, 8.3, and 8.3.4. These are referred to as the Tunneled Packet Streams.
2) As packets to be tunneled are received from the input interface, they are placed in memory as they are received. A conceptual memory buffer is preconfigured to contain as many Tunnel Packets as required to meet the FEC configuration of packet columns ('L') and rows ('D') with the Tunnel Packet payloads and placeholder RTP headers. The appropriate marker bit and offset fields will be updated in these headers as the Tunneled Packets are saved into the buffers.
3) The example packet contents diagram a conceptual model of how Tunneled Packets are placed into individual Tunnel Packet buffers. Tunneled Packets are placed into the buffer sequentially as received. If a Tunneled Packet exceeds the length of the current Tunnel Packet buffer, the portion of the Tunneled Packet that fits will be placed into the buffer and the remainder placed into the next buffer. The offset field of the next Tunnel Packet buffer RTP header will be set to indicate the byte of the next Tunneled Packet start within the Tunnel Packet payload and the marker bit will be set to indicate that a payload (i.e. Tunneled Packet) start is present within the particular buffer. An example of this packing process is described in detail in Section 8.6.1 below.
4) When a column or row is completed, the FEC function (XOR per [8]) is performed creating an FEC packet for the column or row, respectively. The headers and content of these packets are described in [8].

5) The FEC packets along with the content Tunnel Packets are sent to the STL Transmitter per the guidelines defined in [8]. There are a number of packet interleaving options defined in [8] that can be configured based on the number of columns and rows defined and the system latency desired.

8.6 RTP Header Field Definitions

The STLTP RTP/UDP/IP multicast Tunnel Packets shall contain an RTP header compliant with the header fields specified in [8], Section 7.1, with the changes described below. In addition to the constraints defined by [8], this document changes the definition of one of the standard fields to enable internal payload framing. The **marker (M)** bit along with the repurposed **SSRC_ID** field, now defined as the **packet_offset**, shall be used to identify the start of the first IP packet within the payload of the particular Tunnel Packet. This mechanism is described in more detail below.

Table 8.4 defines the fields of the STLTP RTP header. Paragraphs following the table describe each of the fields.

Table 8.4: RTP Header Field Definitions for STLTP

Syntax	No. of Bits	Format
RTP_Fixed_Header() {		
version (V)	2	'10'
padding (P)	1	bslbf
extension (X)	1	'0'
CSRC_count (CC)	4	'0000'
marker (M)	1	bslbf
payload_type (PT)	7	'1100001'
sequence_number	16	uimsbf
timestamp	32	Table 7.2
packet_offset	32	uimsbf
}		

Please refer to [6] for base definitions of the fields described in Table 8.4 and refer to [8] Section 7.1 for constraints to those base definitions.

version − The version number of the RTP protocol. '10' is currently defined by [6].

padding − Indicates whether padding is included in the payload. A value of '0' indicates no padding is included, while a value of '1' indicates that padding is present. No padding is required; however it may be necessary to add padding if the aggregate incoming bit rate falls below a certain threshold. It is currently assumed that the overall broadcast Stream will be essentially constant since it must synchronize with a constant broadcast bit rate. Under RTP [6] rules, the last byte of the payload is used to indicate the number of padding bytes in the payload.

extension − Indicates that an RTP extension follows the header. No extensions are currently defined, so this field shall be set to '0'.

CSRC_count − Provides the count of additional contributing source identifier fields. ST 2022-1 [8] requires this field to be set to '0'.

marker − A bit indicating a payload start occurs within the Tunnel Packet payload. When the **marker** bit is set to '1', the first byte of the first IP packet starting within the payload shall be indicated by the value of the **packet_offset** field. Subsequent IP packets can be found by using the length fields within each Tunneled Packet. Typically, bytes preceeding the first Tunneled Packet start will be a continuation of the last Tunneled Packet of the previous Tunnel Packet.

payload_type − As payload types are defined in [7], values from 97 through 127 are allocated for dynamic assignment. A value of '97' shall be used for Tunnel Packets.

sequence_number − As per [6], the **sequence_number** shall increment by one, modulo 2^{16} or 65536, for each packet of an RTP Stream. The initial **sequence_number** should be randomized, although doing so is not important in this setting since the Stream will be not be restarted frequently and the Tunnel Packets are not expected to be placed on a public Network. Since UDP/IP does not guarantee ordered delivery and even may duplicate packets, the RTP **sequence_number** can be used to reorder the Stream to its original sequence and to detect missing packets. In this context, '*Stream*' refers to a set of packets destined for the same IP address and port. SMPTE ST 2022-1 [8] relies on the **sequence_number** to determine where content packets are placed in sequence to allow replacement of missing packets. Note that the **sequence_number** applies to each individual Stream. The FEC packets shall have different sequence numbers from the Tunnel Packets, each Stream of which shall be separately sequenced by its own RTP **sequence_number** value.

timestamp − The **timestamp** value shall indicate the Bootstrap Reference Emission Time of the Physical Layer frame in which the content in the associated packet is intended to be emitted.

packet_offset − This field redefines the RTP SSRC_ID field [6]. Under [8], this field can be assigned any value and is to be ignored by the receiver. In this application, if the **marker** bit equals '1', this field shall define the payload offset of the first Tunneled Packet. The field shall indicate the number of bytes *after the RTP header* before the first IP packet starts within the tunneled payload. If the Tunneled Packet starts immediately after the header, then **packet_offset** shall be '0'. Figure 8.3 provides a graphical example of how Tunneled Packets are encapsulated within the fixed-size SMPTE ST 2022-1 RTP Tunnel Packets.

8.6.1 RTP Encapsulation Example

Following is an example that shows how encapsulation is done using RTP to create a complete Tunneled Data Stream for SMPTE 2022-1 ECC processing. It examines the case of a Single-PLP emission. (Examination of a Multiple-PLP STLTP Stream follows Figure 8.4.) It is broken into two diagrams that show the overall structure of an STLTP Stream (in Figure 8.3) and the details of the packing process, including the locations of the several headers of the different layered protocols within the Stream (in Figure 8.4).

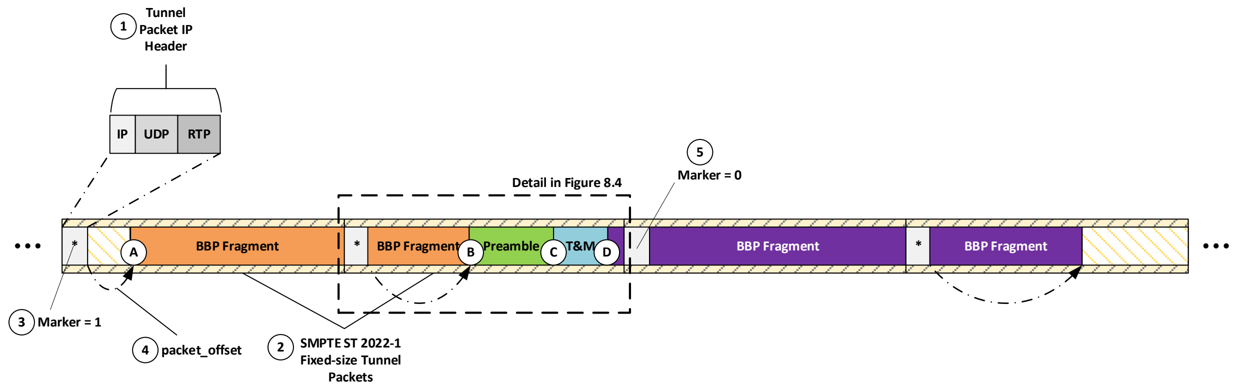

Figure 8.3 RTP Encapsulation example diagram.

The following paragraphs describe each of the call-outs in Figure 8.3. Note that this is an example of a potential implementation to illustrate how the fixed-size Tunnel Packets are constructed according to this specification. Actual implementations could differ substantially; however, the resulting packet Stream, however, would have the same semantic organization.

1) A standard RTP/UDP/IP header precedes each of the STLTP fixed-size Tunnel Packets that contain the content to be protected by the ST 2022-1 FEC process. To optimize the encapsulation process, a set of Tunnel Packets is allocated within a memory buffer, RTP headers (1) are initialized by the Data Producer for all of the Tunnel Packets within the buffer and only specific fields need to be modified as Tunneled Packets are added. Fields such as length, IP address, and port numbers need not be changed once the buffers have been initialized.
2) The example in Figure 8.3 is presented from the decoding standpoint, showing how the Tunnel Packets would be perceived on the Data Consumer side of the STL at the Transmitter(s), assuming that the Data Consumer started with an arbitrary packet in an ongoing Stream of Tunnel Packets. Each Tunnel Packet in the Stream (2) is of a fixed size and contains a header and payload that may or may not contain the start of an IP content packet.
3) If the **marker** bit (3) of a Tunnel Packet is set to ‘1’, signified in the diagram by an asterisk in the RTP header box, then **packet_offset** (4) will be used to indicate the byte where the first Tunneled Packet starts within the payload of the Tunnel Packet.
4) The **packet_offset** is the number of bytes after the RTP header where the first byte of the first tunneled IP header resides within the Tunnel Packet payload (A). This field is used only if the **marker** bit is set to ‘1’. Note that when more than one packet starts in the Tunnel Packet payload, only the first is indicated by the **packet_offset**. This is shown in Figure 8.3 at Tunneled Packets (B) and (C). The start of Tunneled Packet (B) is indicated by the **packet_offset** as being immediately after the end of the remainder of Tunneled Packet (A). The start of Tunneled Packet (C) can be determined using the length of Tunneled Packet (B). Of course, the start of packet (B) also could be determined by using the total length of Tunneled Packet (A), which started in the previous 2022-1 fixed-size Tunnel Packet. Robust decoders likely would use both the length of the previous packet and the **packet_offset** to verify the encapsulation process and that the payload packets were correctly formed. This mechanism of framing the encapsulated IP Streams allows Data Consumers

to recover lost connections quickly since only one or two fixed-size Tunnel Packets are required to begin extracting the Tunneled Packets.

5) The Tunneled Packet, the start of which is labeled (D) in the figure, shows a large packet spanning three ST 2022-1 fixed-size Tunnel Packets. In this case, the middle 2022-1 Tunnel Packet does not have the **marker** bit set to ‘1’ and the **packet_offset** field would be ignored by the Data Consumer.

Figure 8.4 shows the details of the encapsulated Tunnel Packet outlined by the dashed line in Figure 8.3. In the top row of Figure 8.4 are packets from three separate Tunneled Packet Streams. In the center row of Figure 8.4 are the details of the construction of the Tunnel Packet that contains the Tunneled Packets or segments of Tunneled Packets from the top row. As shown by the bottom row, the middle row is a more detailed version of the Tunnel Packet structure shown within the dashed line in Figure 8.3.

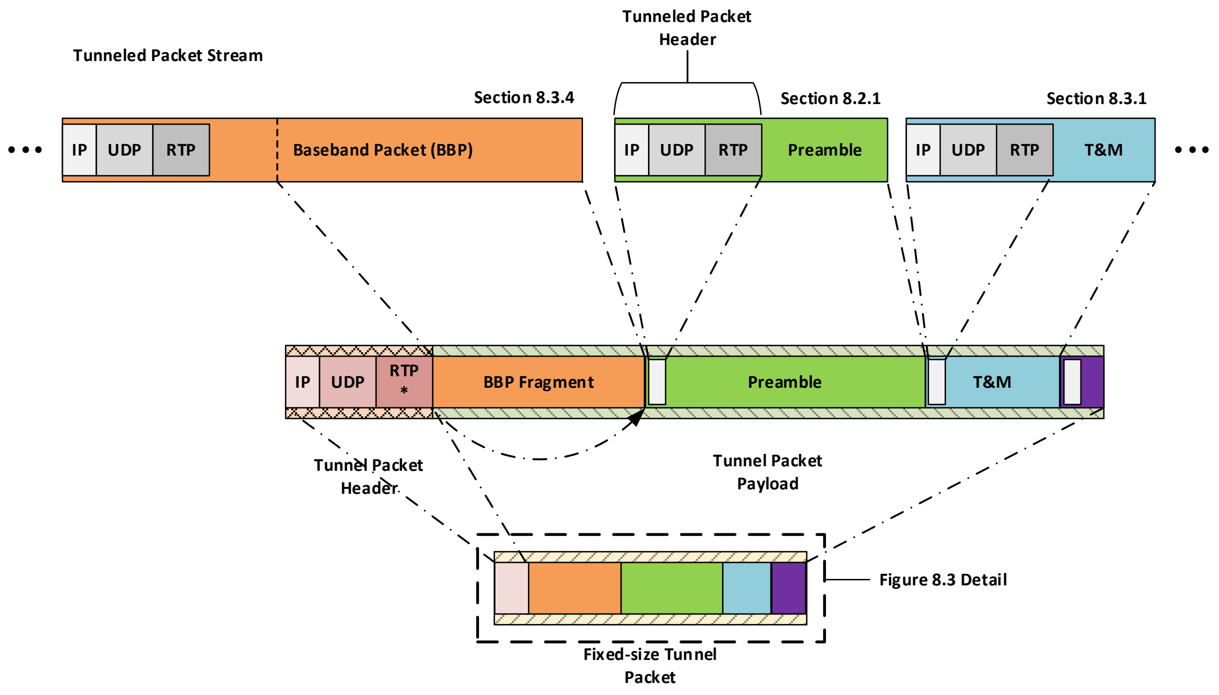

Figure 8.4 Tunneled Packet packing details.

Starting from the top row of Figure 8.4, each of the three Tunneled Packet Streams has its own procession of packets, complete with IP/UDP/RTP header structures. Either complete packets or segments of packets from the Tunneled Packet Streams are multiplexed together on the second row to form the payload of a Tunnel Packet, and another header is prepended to the payload to form a complete new packet. As shown in the bottom row, that packet is the one in dashed lines in Figure 8.3.

Looking at the middle row of Figure 8.4 in more detail, it can be seen that there are four IP/UDP/RTP header sequences. The first, on the left, is the header sequence for the Tunnel Packet. The second header sequence is that of the complete Preamble Packet, shown in green, that is fully encapsulated within the Tunnel Packet. The third header sequence is that of the complete Timing and Management packet, shown in cyan, that is fully encapsulated within the Tunnel Packet. The fourth header sequence leads into an initial segment of a Baseband Packet, shown in purple. Also included in the packet described in the middle row is a segment of a Baseband Packet, shown in orange, that does not have its own header because it is a continuation from an earlier Tunnel Packet that contained the header sequence for that Tunneled Packet.

In the sequence of payload packets and segments described within the payload of the Tunnel Packet, there is a pointer from the Tunnel Packet RTP header to the start of the first bit of the header sequence (i.e., the IP header) that introduces the Preamble Packet. Since the header of the Preamble Packet is the first packet to appear within the Tunnel Packet, the Tunnel Packet RTP header pointer points there. Since the RTP header pointer did not point to the first byte following itself, the implication is that the first data segment within the Tunnel Packet payload is a segment of a Tunneled Packet that was not completed in an earlier Tunnel Packet. A Data Consumer of the STLTP Stream, upon examination of the Tunnel Packet RTP header, can determine both that the first data within the Tunnel Packet payload is a segment of a previously incomplete packet and where the next Tunneled Packet begins within the Tunnel Packet. By examining the length value contained within the header sequence of that next Tunneled Packet, the Data Consumer can locate the next header within the Tunnel Packet payload. This process can continue until the Data Consumer finds a Tunneled Packet header that indicates its length to be longer than the space remaining available in the Tunnel Packet, which implies that there will be at least a segment of the final Tunneled Packet contained within the next Tunnel Packet having the same port address (i.e., going to the same PLP) as the current Tunnel Packet.

9. TRANSMITTER OPERATION NORMATIVE REQUIREMENTS

There are a number of requirements placed on Transmitter(s) that facilitate their use of the data arriving over the STL from a Broadcast Gateway and that enable operations of Transmitters in SFNs. The requirements relate to timing of emissions, buffer management, frequency control, offsetting of carrier frequencies for interference mitigation, and the like.

9.1 Timing Manager

A Timing Manager is required functionality in a Transmitter to permit use of the Timing and Management Data sent to Transmitters by a Broadcast Gateway. The Timing Manager shall receive, buffer, and process Timing and Management Data arriving in the form described in Section 8.3 from the STL PLP Demux, as shown in Figure 4.2 (heavy, light-orange line). Timing and Management Data must arrive at a Transmitter in advance of the time for its application. This data shall control the overall operation of the Transmitter and the times at which the numerous processes in the Transmitter are executed. The Timing and Management Data that the Transmitter receives is not intended for emission but rather for control purposes only.

The Timing Manager also shall receive time information directly from a highly accurate source such as the Global Navigation Satellite System (GNSS) or communicated through a data Stream capable of precise time information transfer such as the Precision Time Protocol (PTP). The time information shall be used to accurately control the emission time of the leading edge of the first Bootstrap symbol of each Physical Layer frame. The same timing information used for control of emission timing also shall be capable of use for derivation of a precise frequency reference for control of the Transmitter emission frequency.

The Timing and Management Data shall be stored in a buffer to permit Transmission of Bootstrap Reference Emission Times by the Scheduler to the Transmitter(s) well in advance of those emission times. The size of the buffer should be determined from the maximum advance of delivery time of the Bootstrap Reference Emission Time information permitted by the specification of the Timing and Management Data Stream protocol, as described in Section 8.3.1.

The primary data carried by the Timing and Management Stream is the Bootstrap Reference Emission Time at which each Physical Layer frame is to be emitted. Multiple instances of the Timing and Management Data for a given frame may be sent to Transmitter(s) by the Scheduler. Those instances will be identified as to the Physical Layer frame to which they apply by having a matching time value indicating the Bootstrap Reference Emission Time of the frame. The multiple instances can be held in a buffer and used by the Timing Manager to improve reliability of the Timing and Management Data through application of majority logic or other processing of the redundant copies of the data.

The number of copies of the Timing and Management Data that are sent to the Transmitter(s) for each Physical Layer frame by the Scheduler is indicated in the Timing and Management Data Stream itself, as specified in Table 8.3. Similarly, the number of copies of the Preamble data that are sent to the Transmitter(s) for each Physical Layer frame by the Scheduler also is indicated in the Timing and Management Data Stream. The number of Preamble copies sent by the Scheduler shall be communicated by the Timing Manager to the Preamble Parser so that it can utilize similar processing of the redundant data for purposes of improving reliability of the data.

Other functions of the Timing Manager are:

- Control of the configuration of Bootstrap symbols, waveforms, and the data carried
- Control of the emission times of Bootstrap symbols
- Compensation of the Transmitter-to-antenna emission delay (TAD)
- Control of buffer delays in the Transmitter data processing path
  - Including compensation for STL Network delivery delays in SFNs
- Control of Preamble data insertion following Bootstrap emission
- Parsing of Timing and Management Data for individual Transmitters in SFNs
- Control of Transmitter emission time offsets in SFN operation
- Control of TxID insertion
  - Including insertion level and off

9.2 Preamble Parser

A Preamble Parser is required functionality in a Transmitter to permit use of Preamble data from a Scheduler both to set up the modulation, coding, and other processes in the Exciter and to communicate to receivers the configuration of the signals emitted by the Transmitter. The Preamble Parser shall receive from the STL PLP Demux Preamble data in the form described in Section 8, as shown in Figure 4.2 (heavy, light-blue line) and shall buffer that data. Preamble data will arrive at a Transmitter in advance of the time for its emission. The Preamble data shall be used in two ways: (1) to set up the Transmitter data and signal processing, as controlled by the Scheduler through the delivered Preamble data, and (2) to deliver to the Preamble Inserter block the Preamble data Stream at the correct time for its emission in the transmitted signal.

To achieve the two required uses of the Preamble data, the Preamble Parser shall provide two primary functions. The first function shall be a buffer that holds the Preamble data Stream until it is needed for emission. The output of the buffer shall release the Preamble data Stream to the Preamble Inserter (thin, light-violet line in Figure 4.2) at the appropriate time, as determined by the Timing Manager, so that emission of the Preamble data occurs immediately following emission of the Bootstrap. Both the **length** data at the start and the **crc16** value at the end of the Preamble Payload data packet shall not be emitted by the Transmitter and can be deleted after any use by the Transmitter for error detection. The second function shall be parsing of the data in the Preamble into separate instructions for each of the data- and signal-processing stages of the Transmitter and delivery of those instructions to the processing stages at the appropriate times, as determined by the Timing Manager (thick, dark-blue line in Figure 4.2).

Advance arrival of the Preamble data at the Preamble Parser is for the purpose of providing time for the Exciter to set up its processing stages to the configuration specified by the Scheduler for the waveform, in advance of processing the next Physical Layer frame. During that setup Period, the Preamble data Stream itself is held in the buffer until it is needed for emission in the appropriate frame.

Multiple instances of the Preamble data for a given frame may be sent to Transmitter(s) by the Scheduler. Those instances will be identified as to the Physical Layer frame to which they belong with a time value matching the Bootstrap Reference Emission Time of the frame. The multiple instances can be used by the Preamble Parser to improve reliability of the Preamble data through application of majority logic or other processing of the redundant copies of the data. The number of copies of the Preamble that are sent by the Scheduler to the Transmitter(s) is indicated in the Timing and Management Data Stream that goes to the Timing Manager. The Timing Manager is expected to communicate that information to the Preamble Parser (thin, red line in Figure 4.2) when it is intended to process the redundant data for improved data reliability.

9.3 Other Requirements

To enable use of Transmitters in Single Frequency Networks (SFNs) and to minimize interference between stations, certain additional requirements not covered elsewhere in this standard are placed on Transmitters. Those requirements relate to frequency accuracy, timing offsets for network management, and co-channel interference minimization through application of frequency and timing offsets.

9.3.1 Frequency Accuracy

The Center Frequency of the transmitted signal shall be either the frequency of the middle carrier in the signal's spectrum when an odd number of carriers is in use or shall be the frequency half-way between the two middle carriers when an even number of carriers is in use. To enable both SFN operation and the co-channel interference mitigation methods discussed in Section 9.3.3, the Transmitter Center Frequency shall be maintained at the nominal Center Frequency ±0.5Hz, with a long-term-averaged error of zero. Use of GNSS as a time base and frequency reference can enable such precision.

9.3.2 Transmitter Timing Offsets

To cause Transmitters in a Network to emit their waveforms at times needed for Network shaping, offsets may be applied to emission time values individually at each Transmitter. A Transmitter timing reference with high precision, low jitter, and zero long-term drift can be derived from a single reference and may be delivered to each site or derived at each site. Each Transmitter must be locked to a source equal to the GNSS Time interval. It can be GPS time or TAI, but it must be independent of leap seconds.

A frequency reference with high precision, low jitter and zero long-term drift can be derived from the timing reference. GPS-like sources can be available simultaneously at multiple locations at the studio and Transmitter to enable such an accurate reference.

9.3.3 Center-Frequency and Network-Timing Offsets for Co-Channel Interference Mitigation

The ATSC 3.0 Physical Layer protocol supports low-SNR operation, which can extend coverage areas beyond those served by individual broadcast stations using earlier digital transmission systems. When there are two or more neighboring stations operating with low-SNR configurations on the same RF channel while transmitting different data, reception failure can occur due to channel estimation mismatches in strong co-channel interference environments. For example, when two co-channel stations use the same pilot pattern, FFT size, and guard interval value, after undergoing different propagation channel distortions, the frequency responses of the two arriving signals will be different. In areas where the pilot patterns of the two signals from the co-channel stations overlap, linear addition of the powers of the individual pilots will occur at receiving locations, resulting in creation of a channel model that does not represent the frequency response of either received signal. Receivers at those locations will attempt to correct the channel impairments of the resultant, combined "channel" represented by the summed pilots and consequently will receive neither of the desired signals arriving at the receiver input.

Similarly, when Bootstrap signals from two neighboring co-channel stations are overlapped in the time dimension, receivers may not detect whether such overlap is due to multipath distortion of a single station or to interference from a co-channel station. To enable receivers to separate such interfering signals without requiring design changes, carrier and timing offsets, as described in the following sub-sections, shall be applied to Transmitters so that ATSC 3.0 receivers can successfully receive low-SNR signals in locations where strong, co-channel interference exists. The offsets must be assigned to triads of adjacent stations in such a way that each station in a triad has a different offset value. Given that most stations will belong to more than one triad, offsets must be organized on a large-area basis, considering all of the stations on each channel across an entire region.

9.3.3.1 Center Frequency Offset

Pilot pattern overlap always occurs when neighboring co-channel stations use the same FFT sizes, guard intervals, and pilot patterns. Even when neighboring co-channel stations use some different Physical Layer parameters (FFT sizes, guard intervals, or pilot patterns), there may be partial pilot overlap, depending on the choices of pilot patterns. Such pilot pattern overlap (both full and partial overlap) causes the sorts of channel estimation errors described above, which result in reception failure or performance degradation. To avoid overlapping pilot patterns, offset of Transmitter Center Frequencies shall be used. A Center Frequency offset parameter, having values of –1, 0, and +1 OFDM carriers, is carried in the **tx_carrier_offset** field of the Timing and Management Data. The actual offset frequency value to be applied to each Transmitter shall be the carrier frequency spacing in an 8K FFT for the channel bandwidth in use. The offset frequency value in Hz is defined in the semantics for **tx_carrier_offset** found following Table 8.3. The baseband sampling rate is indicated by the **bsr_coefficient** field of the Timing and Management Data. When signals from two or more neighboring, co-channel stations are receivable in an area, different carrier offset values shall be assigned to each of the co-channel stations. For a group of Transmitters that emits the same waveform (i.e., an SFN), the same carrier offset value shall be applied.

9.3.3.2 Network Timing Offset

Bootstrap signal overlap can occur when neighboring co-channel stations use the same frame size or integer multiples of a particular frame size, which is most likely to occur when frame sizes related to 1-second periods are used. To avoid such time-domain overlaps of Bootstraps from multiple stations, Bootstrap Reference Emission Time offsets of the station waveforms shall be applied by the respective Schedulers. Bootstrap Reference Emission Time offsets shall be applied when multiples or submultiples of Physical Layer frame lengths occur at integer multiples of 1-second periods. Directions of the Bootstrap Reference Emission Time offsets shall be the same as indicated by the **tx_carrier_offset** values sent by Schedulers to their Transmitters in the Timing and Management Data. Positive values of **tx_carrier_offset** shall indicate delays, and negative values of **tx_carrier_offset** shall indicate advances, in the Bootstrap Reference Emission Times sent by Schedulers to their Transmitters. When the assigned offset for a station is zero, its Bootstrap Reference Emission Time shall be 0 ± 1 msec of a Second Tick of TAI time. When the assigned offset for a station is non-zero, its Bootstrap Reference Emission Time offset shall be ≥ ± (Bootstrap Length + 1 msec) up to ± (Bootstrap Length + 10 msec) of a Second Tick of TAI time, with the sign of '±' determined by the sign of **tx_carrier_offset.** Bootstrap symbols are 0.5 msec in length, and the minimum number of Bootstrap symbols is 4, making Bootstrap Length 2 msec or greater, depending upon whether the Bootstrap is extended per [2].

An example of using Center-Frequency and Network-Timing offsets among neighboring co-channel stations is shown in Figure 9.1.

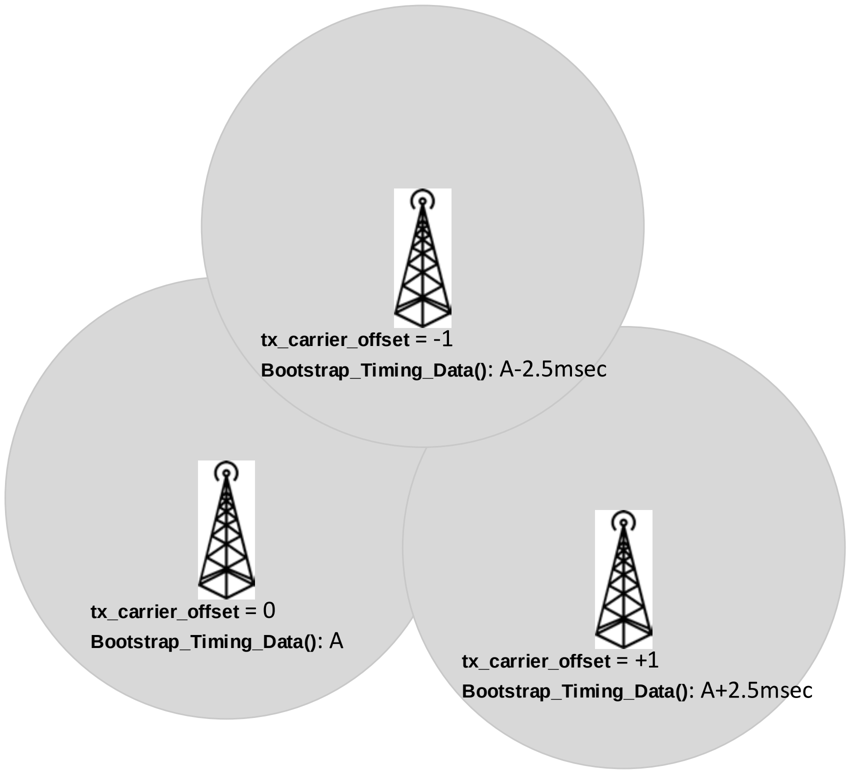
tx_carrier_offset = -1
Bootstrap_Timing_Data(): A-2.5msec
tx_carrier_offset = 0
Bootstrap_Timing_Data(): A
tx_carrier_offset = +1
Bootstrap_Timing_Data(): A+2.5msec

Figure 9.1 Example use of carrier and timing offsets for neighboring co-channel stations.

Annex A: Physical Layer Control

A.1 PHYSICAL LAYER RESOURCES

System Manager pre-determined schedule consists of at least the control parameters listed in Section 9 of [3]. The control inputs to the Physical Layer are listed in the following sub-sections. Dependency of some parameters may be based on others from a formula. Those will be identified when possible.

A.1.1 Bootstrap Signaling

bootstrap_major_version – This parameter is defined by [2] and constrained by [3]. Intent is to signal a structure change that is not backwards compatible with existing version(s) of the Bootstrap.

bootstrap_minor_version – This parameter is defined by [2] and constrained by [3]. Intent is to signal a structure change that is backwards compatible with existing version(s) of the Bootstrap.

ea_wake_up – This parameter is defined by [2] to signal an emergency alert.

min_time_to_next – This parameter is defined by [2] to signal when the next similar Bootstrap type (a Bootstrap that matches the same major and minor version number as the current Bootstrap) will be available in an emission.

system_bandwidth – This parameter is defined by [2] to signal the RF channel bandwidth of the post-Bootstrap portion of the current Physical Layer frame.

bsr_coefficient – This parameter is defined by [2] and constrained by Table 9.1 in [3] with broadcaster intended baseband sample rate.

preamble_structure – This parameter is defined by [2] and constrained by Table H.1.1 in [3] with settings for FFT size, guard interval, L1-Basic FEC Mode and Preamble scattered pilot patterns for Preamble symbols chosen by broadcaster. Preamble symbols should be at least as robust as the most robust payload settings.

A.1.2 L1-Basic Signaling

L1B_version – This parameter is defined and constrained by [3]. It is minor version signaling of the Preamble L1-Basic signaling structure.

L1B_mimo_scattered_pilot_encoding – This parameter is defined by [3] and signals the MIMO pilot encoding scheme used by any MIMO subframes.

L1B_lls_flag – This parameter is defined by [3] to signal if Low Level Signaling (LLS) is available in the current frame. LLS is defined by [4] and is the starting point for finding which services are available on a given broadcast channel.

L1B_time_info_flag – This parameter is defined by [3] to signal the presence of timing information in the current frame.

L1B_return_channel_flag – This parameter is defined by [3] to signal the presence of a dedicated return channel.

L1B_papr_reduction – This parameter is defined by [3] to signal the technique to reduce the peak to average power ratio within the current frame.

L1B_frame_length_mode – This parameter is defined by [3] to signal that the current frame is time-aligned with excess sample distribution to the guard intervals of data payload OFDM symbols or that the current frame is symbol-aligned with no excess sample distribution.

L1B_frame_length – This parameter is defined by [3] to signal the time Period measured from the beginning of the first sample of the Bootstrap associated with the current frame to the end of the final sample associated with the current frame. Frame length has many considerations like the longest segment length as defined in [4], or time interleaving depth that broadcasters must choose depending on desired content and robustness. Sizes are constrained by [3] to be between 50msec and 5000msec in 5msec increments.

L1B_excess_samples_per_symbol – This parameter is defined by [3] to signal the additional number of excess samples included in the guard interval of each non-Preamble OFDM symbol of the post-Bootstrap portion of the current frame when time-aligned frames are used.

L1B_time_offset – This parameter is defined by [3] to signal the number of sample periods between the nearest preceding or coincident millisecond boundary and the leading edge of the frame.

L1B_additional_samples – This parameter is defined by [3] to signal the number of additional samples added at the end of a frame to facilitate sampling clock alignment.

L1B_num_subframes – This parameter is defined by [3] to signal the number of subframes present within the current frame.

L1B_preamble_num_symbols – This parameter is defined by [3] to signal the total number of OFDM symbols contained within the Preamble, not including the first Preamble symbol.

L1B_preamble_reduced_carriers – This parameter is defined by [3] to signal the number of control units of carriers by which the maximum number of carriers for the FFT size used for the Preamble is reduced. It applies to all Preamble symbols except the first one of the current frame which uses the maximum possible carrier reduction.

L1B_L1_Detail_content_tag – This parameter is defined by [3] and signals new information is available in L1-Detail.

L1B_L1_Detail_size_bytes – This parameter is defined by [3] to signal the size (in bytes) of the L1-Detail information.

L1B_L1_Detail_fec_type – This parameter is defined by [3] to signal the FEC type for L1-Detail information protection. It is a combination of 16K LDPC code length with variety of non-uniform constellations and code rates.

L1B_L1_Detail_additional_parity_mode – This parameter is defined by [3] to signal the Additional Parity Mode which gives the ratio (*K*) of the number of additional parity bits for the next frame’s L1-Detail that are carried in the current frame to half of the number of coded bits for the next frame’s L1-Detail signaling.

L1B_L1_Detail_total_cells – This parameter is defined by [3] to signal the total size (specified in data cells) of the combined coded and modulated L1-Detail signaling for the current frame and the modulated additional parity bits for L1-Detail signaling of the next frame.

L1B_first_sub_mimo – This parameter is defined by [3] to signal whether MIMO is used for the first subframe of the current frame.

L1B_first_sub_miso – This parameter is defined by [3] to signal whether MISO is used for the first subframe of the current frame.

L1B_first_sub_fft_size – This parameter is defined by [3] to signal the FFT size associated with the first subframe of the current frame.

L1B_first_sub_reduced_carriers – This parameter is defined by [3] to signal the number of control units of carriers by which the maximum number of carriers for the FFT size used for the first subframe of the current frame is reduced.

L1B_first_sub_guard_interval – This parameter is defined by [3] to signal the guard interval length used for the OFDM symbols of the first subframe of the current subframe.

L1B_first_sub_num_ofdm_symbols – This parameter is defined by [3] to signal a value equal to one less than the total number of data payload OFDM symbols, including any subframe boundary symbol(s), present within the first subframe of the current frame.

L1B_first_sub_scattered_pilot_pattern – This parameter is defined by [3] to signal the scattered pilot pattern used for the first subframe of the current subframe, whether it is SISO or MIMO.

L1B_first_sub_scattered_pilot_boost – This parameter is defined by [3] to signal the amplitude of the scattered pilots used for the first subframe of the current frame.

L1B_first_sub_sbs_first – This parameter is defined by [3] to signal whether or not the first symbol of the first subframe of the current frame is a subframe boundary symbol.

L1B_first_sub_sbs_last – This parameter is defined by [3] to signal whether or not the last symbol of the first subframe of the current frame is a subframe boundary symbol.

A.1.3 L1-Detail signaling

L1D_version – This parameter is defined by [3]. It signals the minor version of the Preamble L1-Detail structure for the current frame.

L1D_num_rf – This parameter is defined by [3] to signal the number of RF channels involved in channel bonding of the current ATSC 3.0 system, not including the present RF channel.

L1D_rf_id – This parameter is defined by [3] to index the implicit IDs of the other RF channels involved in channel bonding.

L1D_bonded_bsid – This parameter is defined by [3] to signal the Broadcast Stream ID of a separate RF channel that is channel bonded with the current RF channel.

L1D_time_sec – This parameter is defined by [3] to signal the seconds component of the time information.

L1D_time_msec – This parameter is defined by [3] to signal the milliseconds component of the time information.

L1D_time_usec – This parameter is defined by [3] to signal the microseconds component of the time information.

L1D_time_nsec – This parameter is defined by [3] to signal the nanoseconds component of the time information.

L1D_mimo – This parameter is defined by [3] to signal whether MIMO is used for the given subframe.

L1D_miso – This parameter is defined by [3] to signal whether MISO is used for the given subframe.

L1D_fft_size – This parameter is defined by [3] to signal the FFT size associated with the given subframe.

L1D_reduced_carriers – This parameter is defined by [3] to signal the number of control units of carriers by which the maximum number of carriers for the FFT size used for the given subframe is reduced.

L1D_guard_interval – This parameter is defined by [3] to signal the guard interval length used for the OFDM symbols of the given subframe.

L1D_num_ofdm_symbols – This parameter is defined by [3] to signal the value equal to one less than the total number of data payload OFDM symbols, including any subframe-boundary symbol(s), present within the given subframe.

L1D_scattered_pilot_pattern – This parameter is defined by [3] to signal the scattered pilot pattern used for the given subframe, whether it is SISO or MIMO.

L1D_scattered_pilot_boost – This parameter is defined by [3] to signal the amplitude of the scattered pilots used for the given subframe.

L1D_sbs_first – This parameter is defined by [3] to signal whether or not the first symbol of the given subframe is a subframe boundary symbol.

L1D_sbs_last – This parameter is defined by [3] to signal whether or not the last symbol of the given subframe is a subframe boundary symbol.

L1D_subframe_multiplex – This parameter is defined by [3] to signal whether the given subframe is time-division multiplexed / concatenated in time with adjacent subframes.

L1D_frequency_interleaver – This parameter is defined by [3] to signal whether the frequency interleaver is enabled or bypassed for the given subframe.

L1D_sbs_null_cells – This parameter is defined by [3] to signal the number of null cells in the subframe boundary symbols of the current subframe.

L1D_num_plp – This parameter is defined by [3] to signal a value equal to one less than the total number of PLPs used within the given subframe.

L1D_plp_id – This parameter is defined by [3] to signal a value equal to the ID of the given PLP, with a range from 0 to 63, inclusive.

L1D_plp_lls_flag – This parameter is defined by [3] to signal whether the given PLP contains the Low Level Signaling (LLS) defined by [4].

L1D_plp_layer – This parameter is defined by [3] to signal a value equal to the layer index of the given PLP.

L1D_plp_start – This parameter is defined by [3] to signal a value equal to the index of the data cell that holds the first data cell of the current PLP in the given subframe.

L1D_plp_size – This parameter is defined by [3] to signal a value equal to the number of data cells allocated to the given PLP.

L1D_plp_scrambler_type – This parameter is defined by [3] to signal the choice of scrambler type for the given PLP.

L1D_plp_fec_type – This parameter is defined by [3] to signal the Forward Error Correction (FEC) method used for encoding the given PLP.

L1D_plp_mod – This parameter is defined by [3] to signal the modulation used for the given PLP, whether SISO or MIMO.

L1D_plp_cod – This parameter is defined by [3] to signal the coding rate used for the given PLP.

L1D_plp_TI_mode – This parameter is defined by [3] to signal the time interleaving mode for the given PLP. The Scheduler is responsible for selecting the appropriate time interleaver mode to use for each core PLP configured for an RF channel. An Enhanced PLP follows the time interleaver mode of the Core PLP(s) with which it is layered-division multiplexed.

L1D_plp_fec_block_start – This parameter is defined by [3] to signal the start position of the first FEC Block that begins within the current PLP during the current subframe.

L1D_plp_CTI_fec_block_start – This parameter is defined by [3] to signal the position, after the CTI, of the first cell of the first complete FEC Block, before the CTI, for the current PLP in the current or subsequent subframe.

L1D_plp_num_channel_bonded – This parameter is defined by [3] to signal the number of RF channels, not including the present RF channel, involved in channel bonding of the given PLP.

L1D_plp_channel_bonding_format – This parameter is defined by [3] to signal the channel bonding format for the given PLP, whether plain channel bonding or SNR averaging channel bonding.

L1D_plp_bonded_rf_id – This parameter is defined by [3] to signal the RF id(s) of the RF channel(s) that are used for channel bonding with the given PLP.

L1D_plp_mimo_stream_combining – This parameter is defined by [3] to signal whether the stream combining option of the MIMO precoder is used in the given PLP.

L1D_plp_mimo_IQ_interleaving – This parameter is defined by [3] to signal whether the IQ polarization interleaving option of the MIMO precoder is used in the given PLP.

L1D_plp_mimo_PH – This parameter is defined by [3] to signal whether the phase hopping option of the MIMO precoder is used in the given PLP.

L1D_plp_type – This parameter is defined by [3] to signal whether the given PLP is non-dispersed (i.e. all data cells of the current PLP have contiguous logical addresses and subslicing is not used for the current PLP) or whether the current PLP is dispersed (i.e. not all data cells of the current PLP have contiguous logical addresses and subslicing is used for the current PLP).

If L1D_plp_type has a value of one, the number of subslices and subslice interval are set.

L1D_num_subslices – If the given PLP is dispersed, this parameter is defined by [3] to signal a value equal to the one less than the actual number of subslices used for the given PLP within the given subframe.

L1D_subslice_interval – If the given PLP is dispersed, this parameter is defined by [3] to signal a value equal to the number of sequentially-indexed data cells measured from the beginning of a subslice for a given PLP to the beginning of the next subslice for the same PLP.

L1D_plp_TI_extended_interleaving – This parameter is defined by [3] to signal whether extended interleaving is to be used for the given PLP.

L1D_plp_CTI_depth – This parameter is defined by [3] to signal the number of rows used in the convolutional time interleaver.

L1D_plp_CTI_start_row – This parameter is defined by [3] to signal the position of the interleaver selector at the start of the given PLP within the current subframe.

L1D_plp_HTI_inter_subframe – This parameter is defined by [3] to signal the hybrid time interleaving mode for the given PLP.

L1D_plp_HTI_num_ti_blocks – This parameter is defined by [3] to signal either the number of TI blocks per interleaving frame, N_{TI}, when **L1D_plp_HTI_inter_subframe**=0 or the number of subframes, P_I, over which cells from one TI block are carried when **L1D_plp_HTI_inter_subframe**=1 in the given PLP.

L1D_plp_HTI_num_fec_blocks_max – This parameter is defined by [3] to signal a value one less than the maximum number of FEC blocks per interleaving frame for the given PLP.

L1D_plp_HTI_num_fec_blocks – This parameter is defined by [3] to signal a value one less than the number of FEC blocks contained in the current interleaving frame for the given PLP.

L1D_plp_HTI_cell_interleaver – This parameter is defined by [3] to signal whether the cell interleaver is used in the given PLP.

L1D_plp_ldm_injection_level – This parameter is defined by [3] to signal the Enhanced PLP injection level relative to the Core PLP.

L1D_bsid – This parameter is defined by [3] to signal the Broadcast Stream ID of the current RF channel.

Annex B: Network Configuration Examples

B.1 EXAMPLE STUDIO NETWORK TOPOLOGIES

The following discussion provides various Network topology examples to clarify how systems may be deployed to achieve various capabilities and robustness. It is informative only. Each deployment will be different and may change over time as new capabilities are added.

For environments with a single ALP Stream Producer and Data Consumer, simply connecting the two units with a cross-over cable would suffice. The Network connections would have to be on the same subnet but would require no further setup. Typically, systems are connected using hubs or switches as is depicted in Figure B.1.1. Note that Figure B.1.1 also shows using RTP/UDP/IP multicast to move various signaling, NRT data, audio and video components into a conceptual multiplexer. The multiplexer would remove the RTP headers prior to the ALP encapsulation stage, resulting in {4} in Figure B.1.1, since broadcast receivers do not support RTP as part of the emission standard. Using RTP allows the multiplexer to manage the order and timing of the contribution source Streams which are UDP/IP multicast as defined in [4]. The Data Sources may also supply timing information in the RTP Streams to enable reconstruction of packet spacing within the ALP Stream.

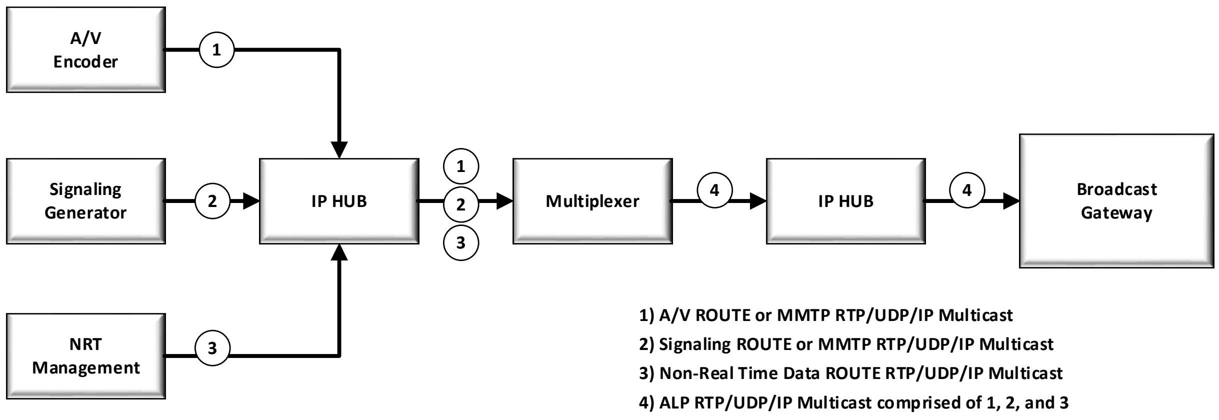
A/V Encoder
Signaling Generator
NRT Management
IP HUB
Multiplexer
IP HUB
Broadcast Gateway
1
2
3
4
1) A/V ROUTE or MMTP RTP/UDP/IP Multicast
2) Signaling ROUTE or MMTP RTP/UDP/IP Multicast
3) Non-Real Time Data ROUTE RTP/UDP/IP Multicast
4) ALP RTP/UDP/IP Multicast comprised of 1, 2, and 3

Figure B.1.1 Simple ALP encapsulation.

Figure B.1.2 shows a case in which separate A/V encoding systems produce audio and video for two PLPs. In this example, a signaling system would produce signaling information that would be included in both PLP Streams. Signaling could be included in either Stream individually or in a dedicated PLP in the same manner. The non-real-time (NRT) data generation management system also would produce ROUTE Streams that could be included in either or both PLPs as desired. In Figure B.1.2, NRT data is only included in PLP 1. Note that in this example, all Streams are on separate multicast addresses on the input to the multiplexer. The output of the multiplexer ({5} and {6}) would be on the same multicast address with different ports. Since there is only one source for each Stream in this example, there is no need for IGMPv3 Source Specific Multicasting

(SSM) and all Network connections can be accomplished with relatively inexpensive hubs or switches.

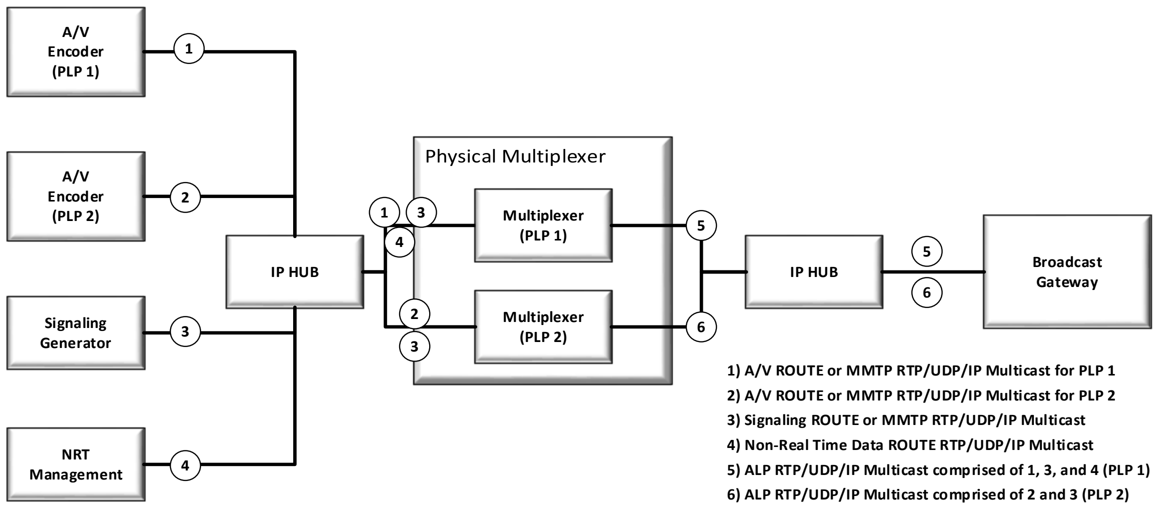
A/V Encoder (PLP 1)
A/V Encoder (PLP 2)
Signaling Generator
NRT Management
IP HUB
Physical Multiplexer
Multiplexer (PLP 1)
Multiplexer (PLP 2)
IP HUB
Broadcast Gateway
1) A/V ROUTE or MMTP RTP/UDP/IP Multicast for PLP 1
2) A/V ROUTE or MMTP RTP/UDP/IP Multicast for PLP 2
3) Signaling ROUTE or MMTP RTP/UDP/IP Multicast
4) Non-Real Time Data ROUTE RTP/UDP/IP Multicast
5) ALP RTP/UDP/IP Multicast comprised of 1, 3, and 4 (PLP 1)
6) ALP RTP/UDP/IP Multicast comprised of 2 and 3 (PLP 2)

Figure B.1.2 Multiple PLP example.

A more typical production situation is diagrammed in Figure B.1.3. This example shows a redundant encoding and signaling environment, redundant multiplexer and redundant output systems feeding into separate STLs (not shown). The multiplexers combine various ROUTE and/or MMTP multicasts into individual ALP Streams, each destined for a unique PLP, removing RTP in the process. A single physical multiplexer likely would create multiple ALP Streams for different PLPs but, functionally, each PLP Stream can be considered separately.

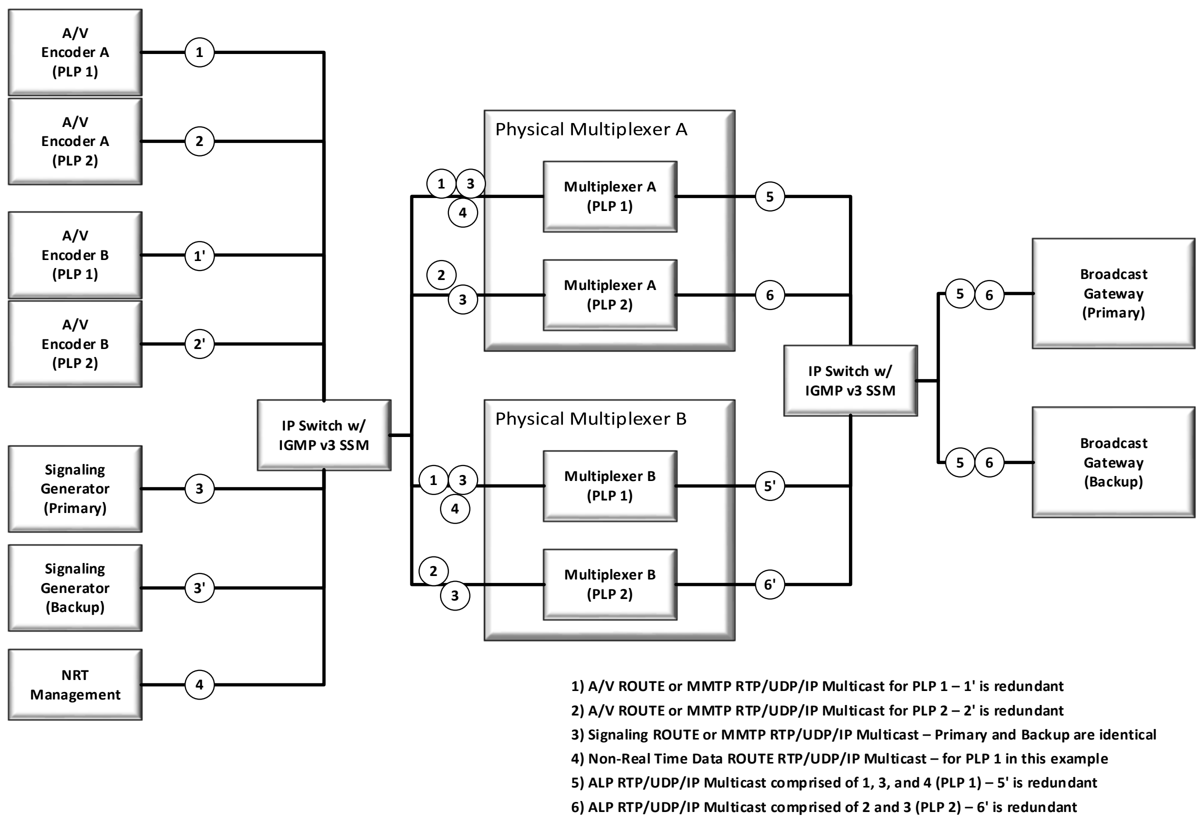
A/V Encoder A (PLP 1)
A/V Encoder A (PLP 2)
A/V Encoder B (PLP 1)
A/V Encoder B (PLP 2)
Signaling Generator (Primary)
Signaling Generator (Backup)
NRT Management
1
2
1'
2'
3
3'
4
IP Switch w/ IGMP v3 SSM
Physical Multiplexer A
Multiplexer A (PLP 1)
Multiplexer A (PLP 2)
Physical Multiplexer B
Multiplexer B (PLP 1)
Multiplexer B (PLP 2)
5
6
5'
6'
IP Switch w/ IGMP v3 SSM
Broadcast Gateway (Primary)
Broadcast Gateway (Backup)
1) A/V ROUTE or MMTP RTP/UDP/IP Multicast for PLP 1 – 1' is redundant
2) A/V ROUTE or MMTP RTP/UDP/IP Multicast for PLP 2 – 2' is redundant
3) Signaling ROUTE or MMTP RTP/UDP/IP Multicast – Primary and Backup are identical
4) Non-Real Time Data ROUTE RTP/UDP/IP Multicast – for PLP 1 in this example
5) ALP RTP/UDP/IP Multicast comprised of 1, 3, and 4 (PLP 1) – 5' is redundant
6) ALP RTP/UDP/IP Multicast comprised of 2 and 3 (PLP 2) – 6' is redundant

Figure B.1.3 Fully redundant routing example.

Note that none of the redundant systems needs to be aware of its counterpart. For example, A/V Encoder A (PLP 1) can be configured identically to A/V Encoder B (PLP 1) except, of course, for its Network address. The emitted multicasts (in Figure B.1.3, the circled numbers {1} and {1'}) are identical in all aspects except that the UDP headers would contain different source addresses. Both Multiplexer A (PLP 1) and Multiplexer B (PLP 1) would be configured with the addresses of the PLP 1 A/V Encoders. Initially, they would both begin ALP encapsulation of A/V Stream {1} into ALP Streams {5} and {5'} by joining the multicast address and port using the address of the A/V Encoder A (PLP 1). If any problems were detected with {1} by either multiplexer, either would then switch to the second A/V Stream, {1'}. This would be accomplished by simply leaving the IGMP multicast group for A/V Encoder A (PLP 1) {1} and 'joining' the A/V Encoder B (PLP 1) multicast {1'}. No other communication is required. Similarly, the {5} or {5'} ALP PLP 1 multicasts would feed both Broadcast Gateway systems based on joins from each. Each Broadcast Gateway would necessarily need to know the source addresses of both Physical Multiplexer A and B so that it could switch between the two sources if the quality of any of the Streams degrades.

Note that a single router or switch with IGMPv3 SSM capability could be used to route all the multicast traffic between each system. There are many options for duplicating routers and even having ring and tree configurations. Be aware that all multicast addresses would need to be unique within the broadcast plant Network to enable a variety of routing options unless Virtual Local Area Networks were used.

Annex C: Scheduler Functional Description

C.1 OVERVIEW

Section 5 above described the high level functions and requirements for the Scheduler and its place within a typical ATSC 3.0 headend. This informative annex provides more detail on the function of a reference Scheduler.

C.1.1 Scheduler Operation

The operation of the Scheduler is constrained by a combination of dynamic, quasi-static, and static parameters. The exact definition of these constraints is left to implementation. This document defines a required function, but not the parameters of that function.

There are two categories of metadata associated with content to be delivered. There is metadata associated with the content which must transit the link as they comprise information needed to successfully decode and render the media. There is another class of metadata which is exclusive to the task of scheduling media in an ATSC 3.0 system. This scheduling metadata is referred to as delivery metadata. This section discusses the functions and application of this delivery metadata. There is some discussion with respect to the order in which metadata and content should be delivered in order to optimize channel change speed.

The cascade of functions involved in the process of scheduling is shown below in Figure C.1.1. This figure contains no requirements as it is merely an example of an architecture. References to time are with respect to Server Current Time (SCT) as discussed in [4].

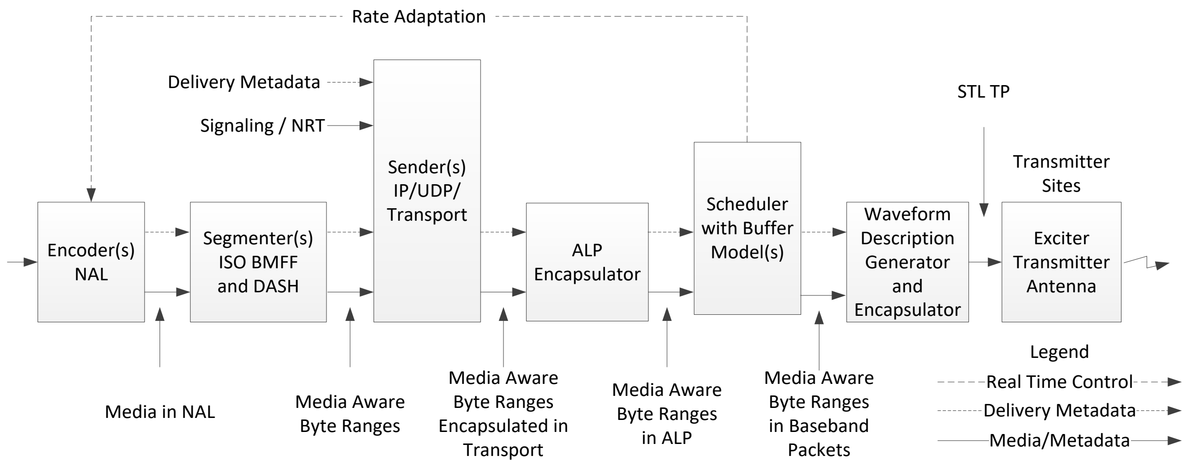
Rate Adaptation
Delivery Metadata
Signaling / NRT
Sender(s) IP/UDP/ Transport
STL TP
Transmitter Sites
Encoder(s) NAL
Segmenter(s) ISO BMFF and DASH
ALP Encapsulator
Scheduler with Buffer Model(s)
Waveform Description Generator and Encapsulator
Exciter Transmitter Antenna
Media in NAL
Media Aware Byte Ranges
Media Aware Byte Ranges Encapsulated in Transport
Media Aware Byte Ranges in ALP
Media Aware Byte Ranges in Baseband Packets
Legend
Real Time Control
Delivery Metadata
Media/Metadata

Figure C.1.1 Cascade of real-time functions involved with Scheduler.

Figure C.1.2 below shows a conceptual depiction of a Scheduler process flow. The central concept is that there are durations of time under construction with known target radiation times. The process generates an endless sequence of Media Segments, which are mapped into Physical Layer frames and transmitted. There are no requirements contained in this figure; it is an example.

The point of the figure is there is a process in time which results in Media Segments being constructed and radiated. The construction of said Media Segments and Physical Layer frames is a series of intermediate processes that each run on a deadline. For the purposes of this figure, there is one set of Media Segments (e.g., audio, video, and captions) per Service per Physical Layer frame. This is not required and may not be optimum. This figure is depicted with a one-to-one correspondence among Media Segments and Physical Layer frames as a convenience to illustrating the process flow in time. The relationship between an Analyzed Media Duration and Physical Layer frames is discussed in some detail in Section C.1.3 below.

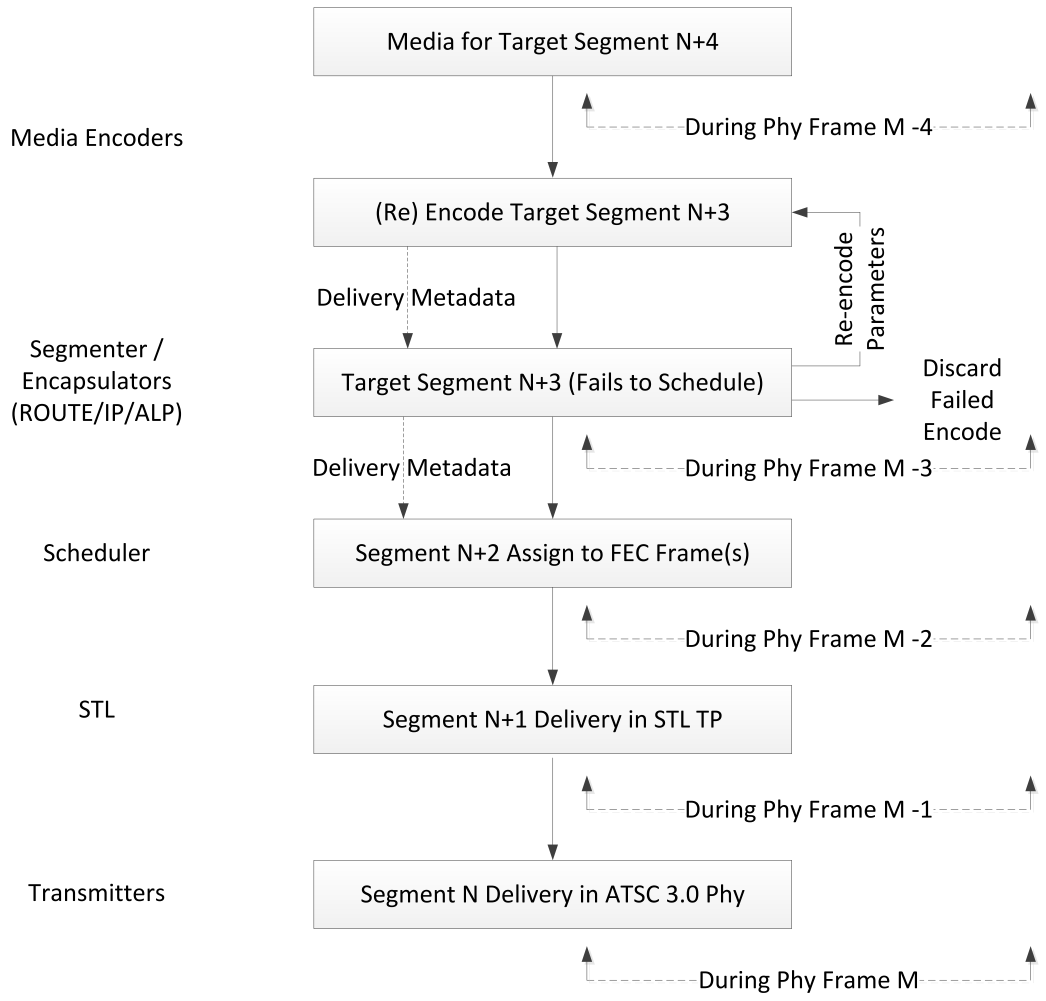
Media for Target Segment N+4
During Phy Frame M -4
Media Encoders
(Re) Encode Target Segment N+3
Re-encode Parameters
Delivery Metadata
Segmenter / Encapsulators (ROUTE/IP/ALP)
Target Segment N+3 (Fails to Schedule)
Discard Failed Encode
Delivery Metadata
During Phy Frame M -3
Scheduler
Segment N+2 Assign to FEC Frame(s)
During Phy Frame M -2
STL
Segment N+1 Delivery in STL TP
During Phy Frame M -1
Transmitters
Segment N Delivery in ATSC 3.0 Phy
During Phy Frame M

Figure C.1.2 Example depiction of a high-level Scheduler process flow.

C.1.2 Key Concepts of Scheduler Delivery Metadata

The concept of earliest delivery at the Physical Layer is illustrated in Figure C.1.3 below. The data contained in this FEC Frame will radiate from the ATSC 3.0 Transmitter after this Earliest Time. As shown, this description is inclusive of all processes comprised in the generation of the transmitted FEC Frame. If the FEC Frame were being discussed in the context of the Sync and Delivery [4] the Physical Layer FEC Frame contains data that is a Data Delivery Event at the receiver.

Latest Time at the Physical Layer is illustrated in Figure C.1.3 below. Conceptually, this is constructed and constrained in a manner similar to Earliest Time above. The data contained in this FEC Frame will radiate from the ATSC 3.0 Transmitter before this Latest Time.

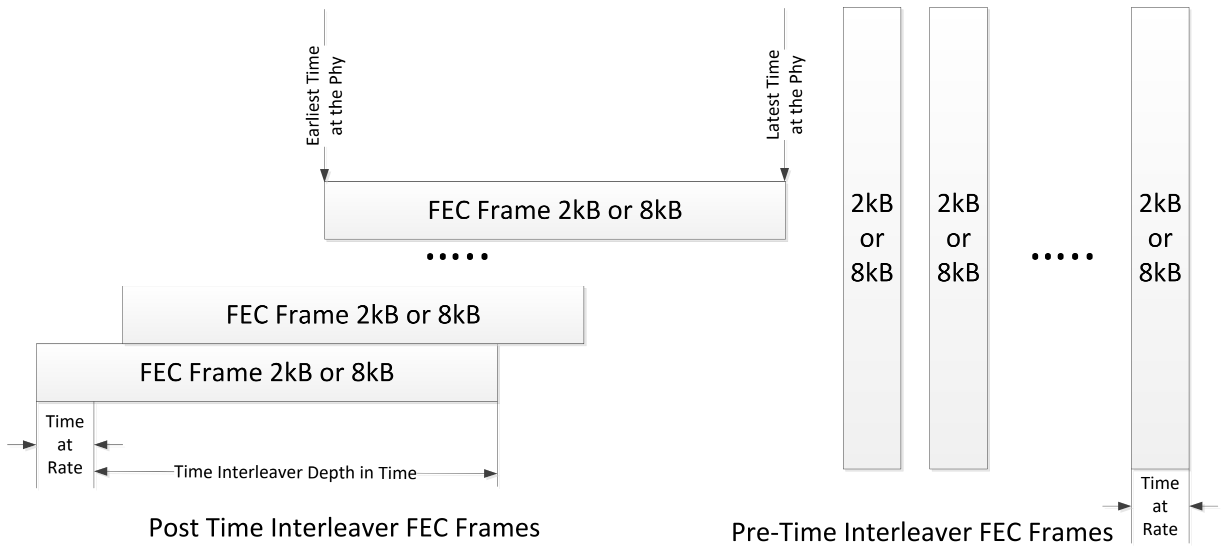

Figure C.1.3 Illustration of Earliest and Latest Time with a block interleaver.

Figure C.1.3 depicts the Earliest and Latest Time of a FEC Frame as realized in the Physical Layer. In practice the Scheduler function should receive a notably wider time range of delivery metadata, such that the Scheduler is not over-constrained.

Figure C.1.4 provides a conceptual view of Earliest and Latest Times in the context of a Media Segment playback duration. Figure C-4 depicts a broader notion of Earliest and Latest Times for multiple use cases.

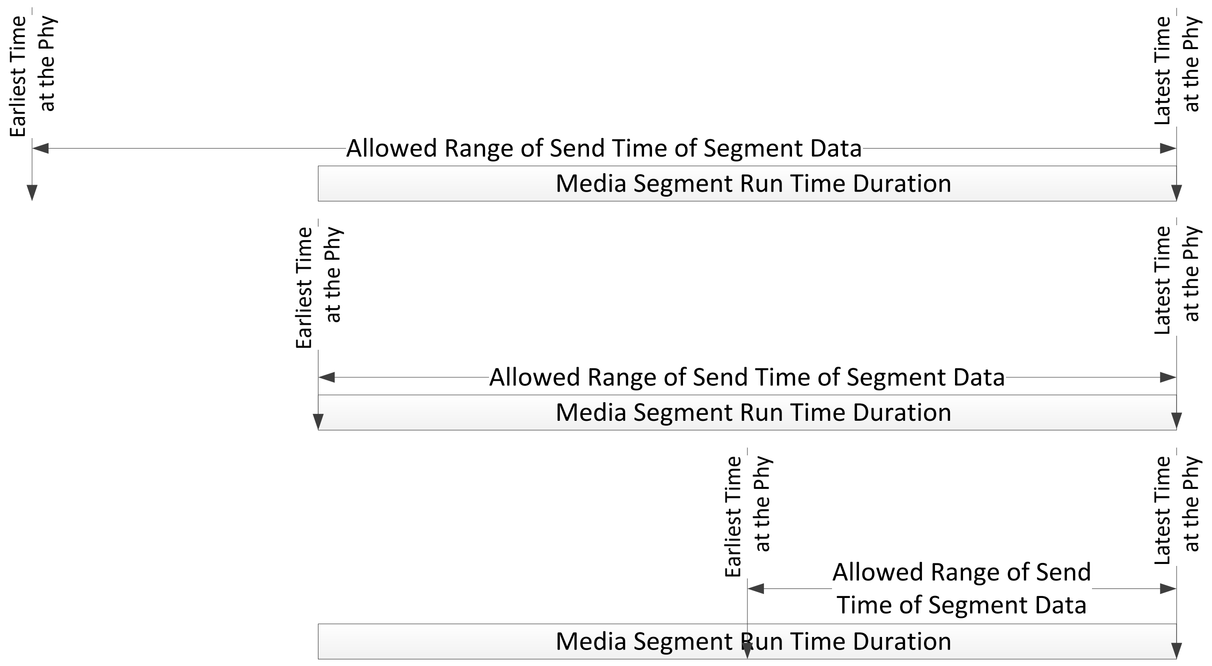
Earliest Time at the Phy
Latest Time at the Phy
Allowed Range of Send Time of Segment Data
Media Segment Run Time Duration
Earliest Time at the Phy
Latest Time at the Phy
Allowed Range of Send Time of Segment Data
Media Segment Run Time Duration
Earliest Time at the Phy
Latest Time at the Phy
Allowed Range of Send Time of Segment Data
Media Segment Run Time Duration

Figure C.1.4 Illustration of Earliest and Latest Time options relative to Media Segment play.

These use cases are intended to illustrate that the concept of Earliest and Latest Times is flexible. These use cases are not requirements. The three use cases depict that the mechanism can describe a variety of implementations. In the top use case, the specification of Earliest Time would allow the Scheduler to send Media Segment data significantly ahead of the actual demand time for the start of playback. The Late Time in this example is bounded by the playback deadline for the Media Segment. The time shown is a duration not the actual delivery or play time. The middle use case depicts a situation where the media playback and media delivery is constrained to be within a Media Segment playback duration. This might be required, for example if there is any switching among PLPs, for example at a DASH Period boundary. The bottom use case illustrates a use case in which the delivery of a Media Segment is accomplished in less than a Media Segment run time duration. This sort of use case can result in faster channel change. These use cases are examples and are not intended to constrain the definition or usage of Earliest and Latest Times or actual implementation.

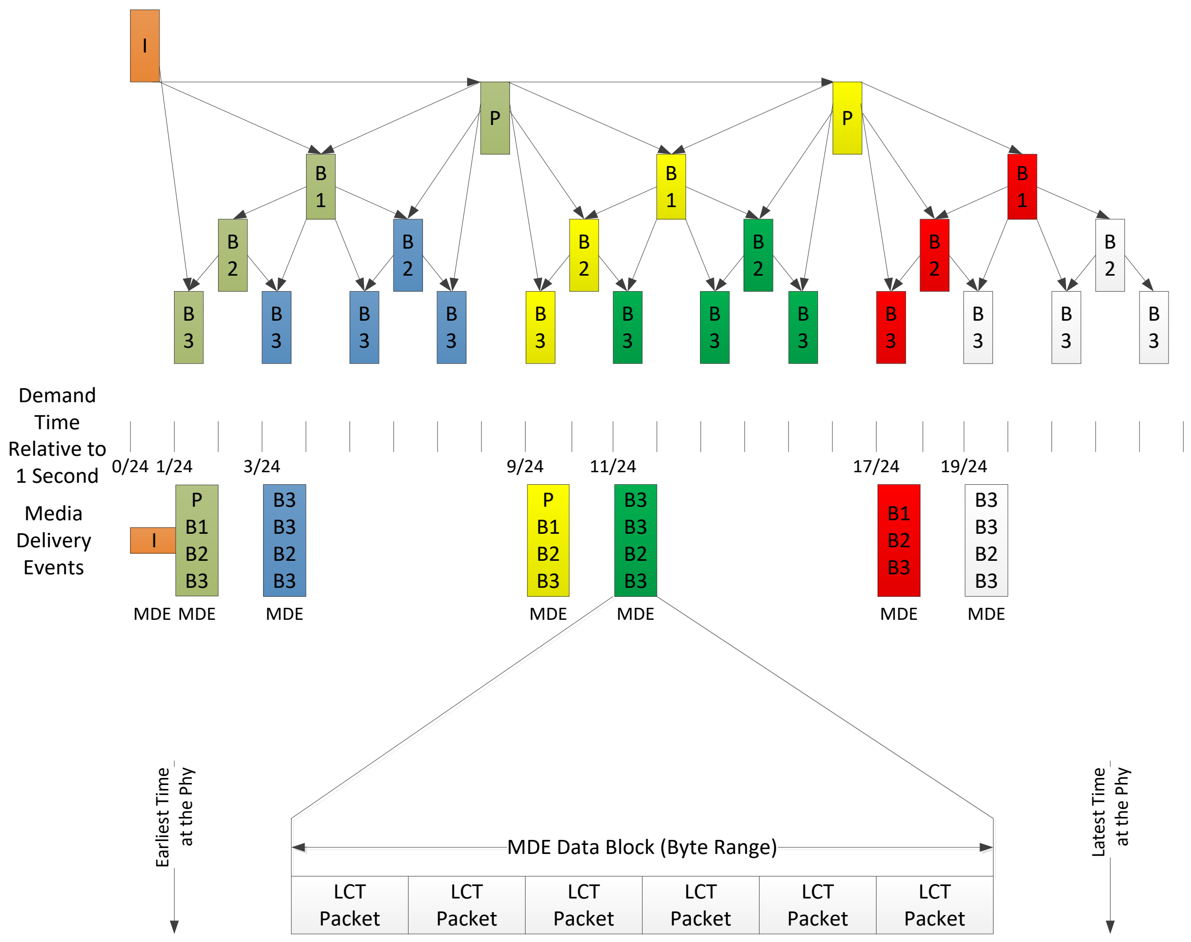

Figure C.1.5 Example of MDE and associated Earliest and Latest Times.

Figure C.1.5 illustrates the relationship of an individual MDE to its associated Earliest and Latest Times. The Latest Time in this case is related to 11/24ths of the media second as compared to a 0 Latest Time for the I frame MDE of the same "GoP". The 0 Latest Time repeats in this example on 1 second boundaries. If in this example, the Physical Layer frame and Media Segment boundaries conform to whole seconds, Earliest Time could be N.0000… and Latest Time N.99999... Given that in-order delivery is enforced for streaming media the functional requirement is within this whole second for this one second duration Media Segment delivery.

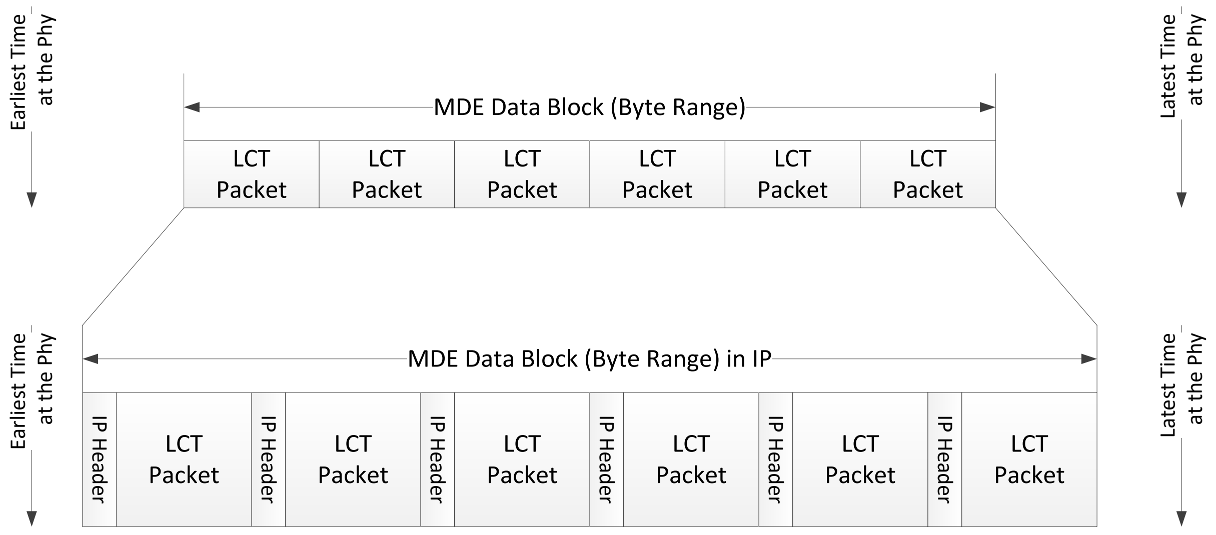

Figure C.1.6 Relationship of an MDE in LCT packets to IP encapsulation.

In Figure C.1.6, an example of an MDE with IP encapsulation is shown. This figure contains no requirements beyond the Earliest and Latest Times being inherited from the MDE. There is no requirement that there is one LCT packet per IP header. There is no requirement that there is more or less than one LCT packet per IP Header. The only constraint is that IP packets containing the MDE are subject to the indicated Earliest and Latest Times of the source MDE data block.

C.1.3 Handling Boundary Conditions

To this point these concepts have been discussed in broad terms. Known metadata about the timeline of the encapsulated and to-be-encapsulated media is expressed down the stack to the Scheduler. This process may be summarized as the encapsulated media inherits the delivery requirements of contained media. There are assorted boundary conditions to consider. This section illustrates a few of these and describes how delivery metadata may be handled.

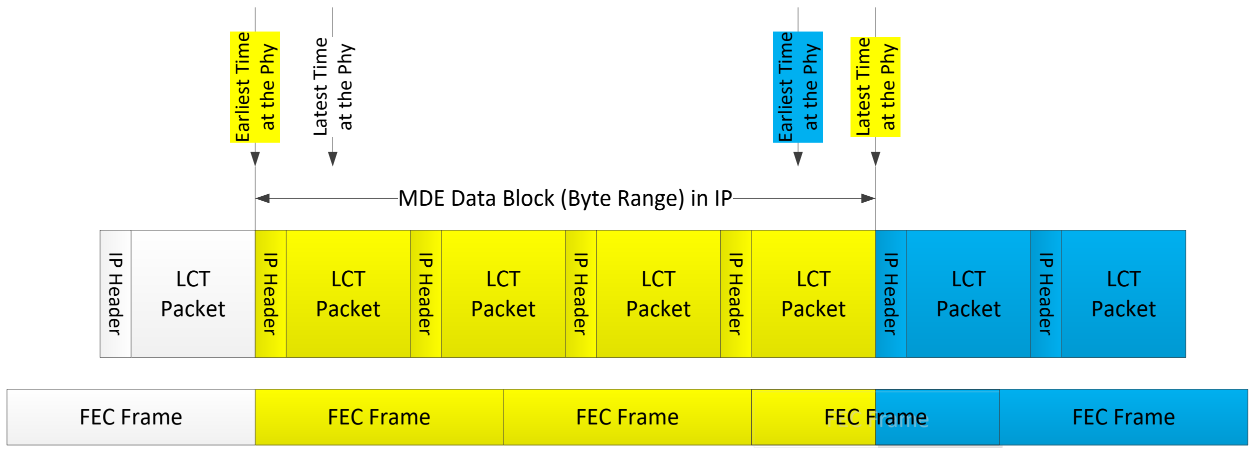

Figure C.1.7 Media is sent at early boundary; e.g., Period start.

Figure C.1.7 depicts a use case for which the leading data of the first FEC Frame of the center MDE has a forward-justified IP packet (and ALP encapsulation) in the first as-assigned FEC Frame. This is caused by the leading LCT packet in for example MDE with RAP having been identified as “Start in new FEC Frame”. This could also occur because the Earliest Time of the center MDE in Figure 5.9 is later than the Latest Time of the leading MDE, which is not shown and is more constraining for the Scheduler. The trailing FEC Frame after the center MDE contains a fragment of the trailing MDE. It should be noted that the sequence of FEC Frames may not be continuous in time and is only shown as such for the convenience of illustrating the mapping of IP packets to FEC Frames. The ALP encapsulation is not shown for convenience, but is required. This sequence of encapsulated MDEs is assumed to be from a single content type with this media content having been declared as in-order delivery.

C.1.4 Delivery Order Within and Across Multiple Sessions

To this point in the discussion, there has been no statement with respect to whether a sequence of IP packets containing LCT packets represents one or more sessions. For the purpose of convenience, it has been assumed that examples have represented a single session, but possibly with multiple content types within the delivered Media Segment files. This section provides an example of a Presentation containing multiple sessions. Perhaps there is one content type per session.

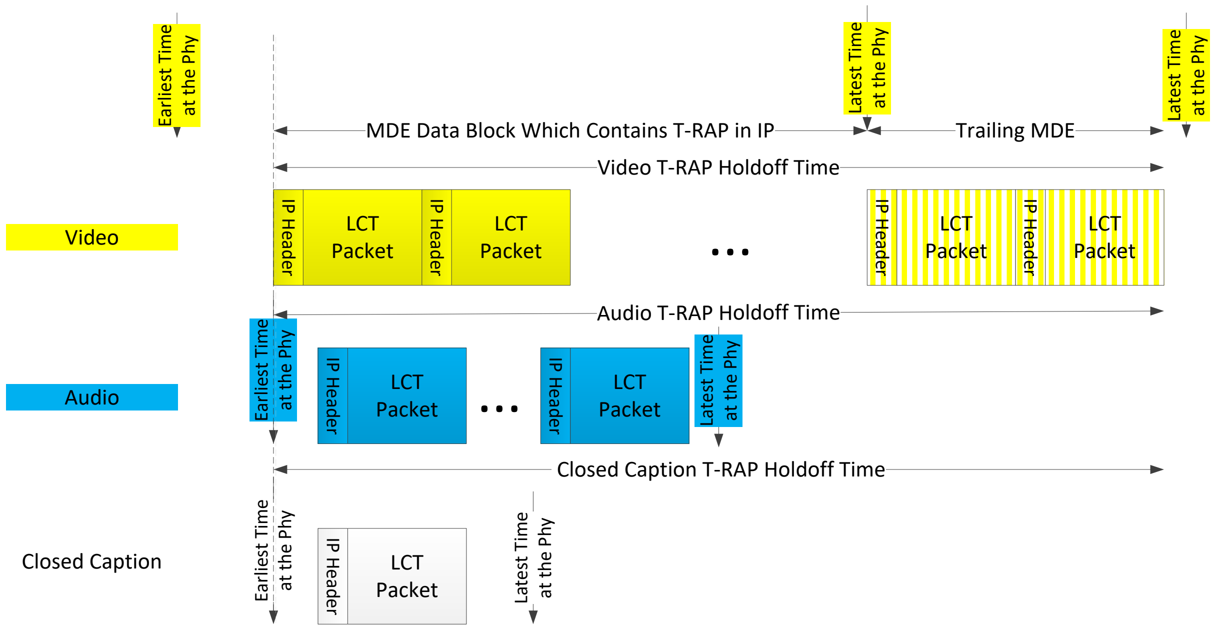

Figure C.1.8 Example of ordered delivery across multiple sessions.

In Figure C.1.8 an example is shown of delivery of multiple sessions across a single PLP or multiple PLPs. In this example the MPD is delivered in the T-RAP of video and the DASH client will see and request byte range delivery ahead of Media Segment availability start time. The video typically has the longest RAP cadence, so it is best to align MPD delivery to the video RAP. A key requirement for MDE is the request of byte range delivery ahead of Media Segment availability start time. The MDE hold-off time due to `EXT_ROUTE_PRESENTATION_TIME - SCT` will expire ahead of Media Segment availability start time and this can be understood via `@availabilityTimeOffset` in the MPD.

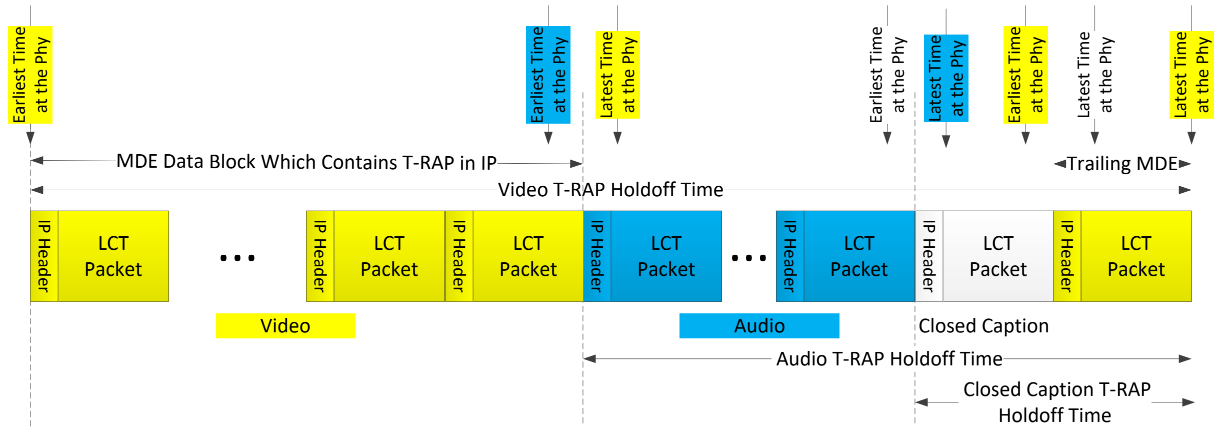

Figure C.1.9 Explicit delivery order on PLP with multiple sessions.

The order of the to-be-delivered IP packets may be preserved in a PLP. This is shown in Figure C.1.9. This is not relevant for a service spanning multiple PLPs. This requires that the sequence of LCT packets in IP or compressed IP is preserved in the emission. None of the timing description changes, but the PLP mapper is configured to observe in-order delivery of the IP Stream as delivered to a specific PLP at the Scheduler.

C.1.5 Timelines and Deadlines

The delivery of media as ISO BMFF files is an essentially endlessly repeating loop. Each Media Segment of media in Live Streaming is expressed in an ISO BMFF container that has an internal time line that starts at 0 and repeats on a, for example, N second interval. A DASH MPD has a defined presentation timeline for the ISO BMFF files and the relationship of the send times at the Physical Layer has a nominally static relationship. The MDE method is adaptive to the stack delay of the receiver i.e. it starts as soon as possible without a stall. In each case, the correct timing must be defined by the segmenter and encapsulator respectively.

If all the video “GoPs” run on the same time cycle i.e. the I frames across Services align in time, most of the benefit of MDE mode will likely accrue to no time guard banding being required for stack delay variability. For Media Segment playback, the MPD must work for the slowest stack. MDE playback starts as soon as after receipt of a T-RAP as allowed. In order to achieve faster start-up the use of staggered Media Segment start times is likely required and non-uniform bandwidth assignment priority vs. time. These are implementation details, which do not change the fundamental mechanisms i.e. each MDE and Media Segment have a time deadline for delivery at the Physical Layer expressed by a Latest Time.

C.1.6 Concept and Practice of Analyzed Media Duration

The discussion up to this point has assumed that media is encoded based on a one-second Media Segment duration; while this is conceptually convenient, it is a bit simplistic as compared to the capabilities of ATSC 3.0. The generalized construct for creating Physical Layer frames is an Analyzed Media Duration. As discussed above the delivery on the Physical Layer must meet a set of requirements driven by media time line(s). The Physical Layer frames defined by the Scheduler do not have to correspond precisely to the Media Segment durations, as long as the time delivery requirements are met. The boundaries of Physical Layer frames may not or do not directly correspond to Media Segments, but they are related. The next paragraphs describe how the general method may be applied.

For each ALP Stream input, there is a Data Source. Video Data Sources can be the most demanding in terms of ISO BMFF Media Segment file size and therefore can drive requirements for ALP buffer size and Analyzed Media Duration. Large encoded Media Segment files may or may not be represented as MDE. An Analyzed Media Duration is a Period of time that is sufficient across all ALP Streams provided to the Scheduler such that an analysis Period is not dependent on other scheduling Periods. See Figure C.1.10 below as an example illustration of IDR positions and how the ALP buffer may go to zero (empty) at the end of each Media Segment. If it does not go empty, it is because a portion of the next Media Segment is already present. This description is valid for Media Segment(s) delivery, irrespective of the ISO BMFF file delivery method for those Media Segments.

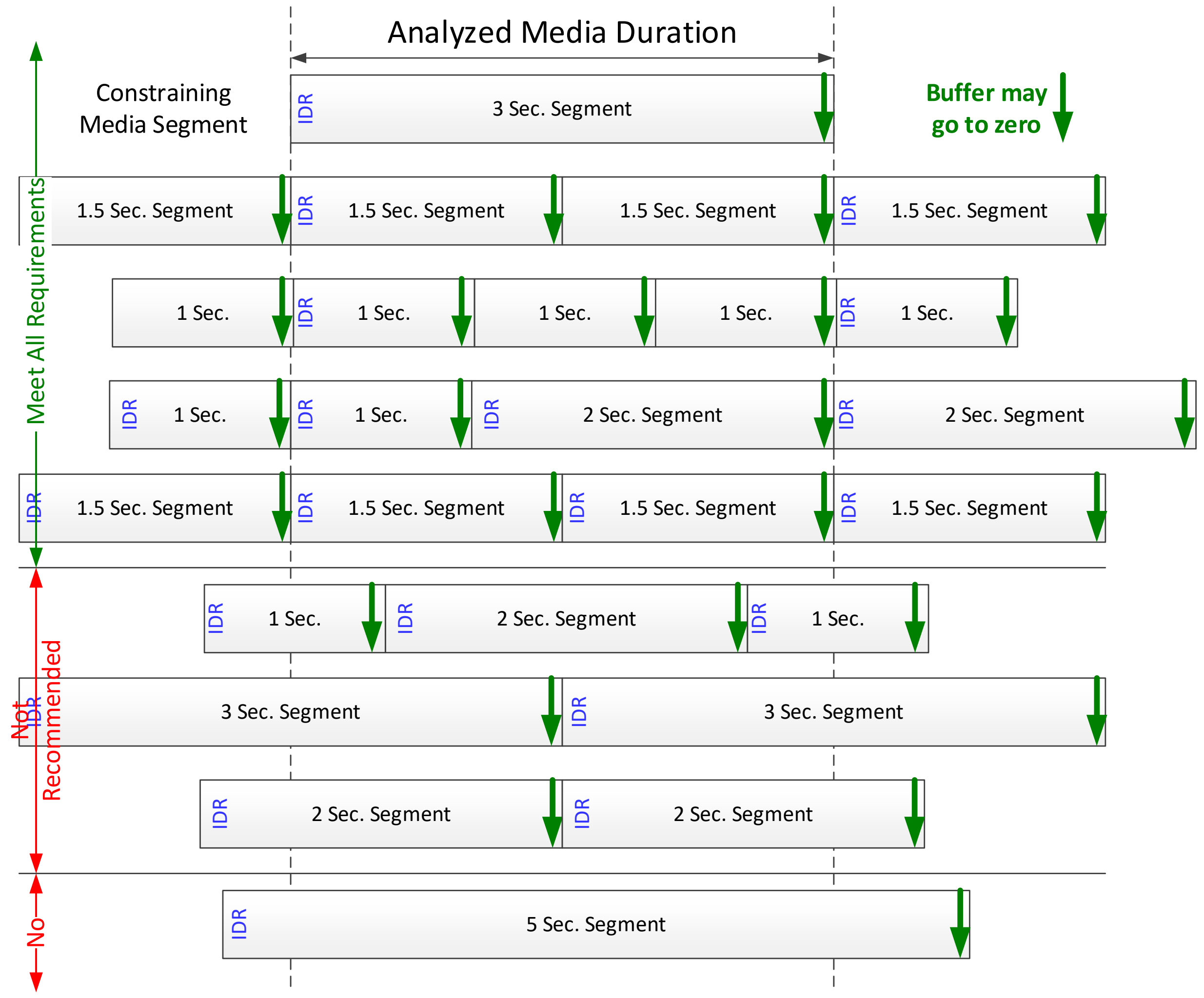

Figure C.1.10 Analyzed Media Duration.

Figure C.1.10 above has three main sections. The top section shows examples of Media Segment lengths providing an optimum ALP buffer length of one Media Segment (longest Media Segment among input ALP Streams). This Analyzed Media Duration starts with a Media Segment containing a SAP/RAP and ends with the ALP buffer going to zero. This is a sufficient set of conditions to allow single-Period scheduling. There is no need or requirement to have more than one Analyzed Media Duration in the ALP buffer. Such a configuration allows combinations of smaller Media Segments to be analyzed as well.

The middle 'Not Recommended' section of the figure shows examples of ALP buffer lengths that span more than one Media Segment. For the Scheduler to work properly, two Analyzed Media Durations need to be stored in this case.

The bottom section shows an example ALP buffer length that is less than the longest Media Segment present. This could require three Media Segment durations to schedule, which is not notionally supported. In this use case, the best approach would be to increase the Analyzed Media Duration and possibly align the Media Segment time boundary and the Analyzed Media Duration. In general terms, the possibility of staggered Media Segment start times may be handled by a longer Analyzed Media Duration and use of two corresponding buffers.

There is an absolute time at which negotiations among Data Sources and the Scheduler must be completed and the data delivered. This time is a Data Delivery Deadline and the Scheduler needs to analyze entire Media Segments to meet this Data Delivery Deadline. Current Media Segments must be sent before next Media Segments are loaded to meet each Media Segment Data Delivery Deadline. The ALP buffer should send all Media Segment contents at the end of every Media Segment, independent of whether there is an IDR frame at the beginning. Every Media Segment may be used to ensure ALP buffer flushing. The time deadline for delivery of the complete Media Segment can result in the buffer going to zero. There is no requirement that the buffer go to zero, as it is possible for the Scheduler to "pull forward" media from a future Analyzed Media Duration. Consider that the earliest delivery time can allow the Scheduler to deliver media in a Physical Layer frame to be delivered ahead of the one that could meet the latest requirement.

On each input ALP Stream, Media Segment durations can vary (for example with layered coding), but they usually have an integer relationship in duration. For example, there can be 0.5-second Media Segments with 2-second enhancement Media Segments, or 1.5-second Media Segments with 3 second enhancement Media Segments, but preferably not 0.5-second Media Segments with 0.75-second enhancement Media Segments. Note that 0.5-second and 0.75-second segments could run in an Analyzed Media Duration of 1.5 seconds or 3 seconds. The constraining duration is the longest duration segment when all segment relationships are integer. The smallest Analyzed Media Duration allows all segment durations present to be present in integer numbers. Smaller Media Segments can combine to form the length of the largest Media Segment as shown in Figure C.1.10.

C.1.7 Sequence of Required Data and Media Events for Acquisition

Figure C.1.11 shows required data and associated events to achieve streaming media service with or without an application. There are two sequences of events. The first grouping is related to the Physical Layer. The Scheduler needs to understand that packets containing, for example, the SLT and SLS need to occur in tight time proximity after the Bootstrap and Preamble. The cyclic temporal location of the Bootstrap and Preamble is likely aligned to media T-RAP timeline, so as to minimize wait states. Multiple staggered media start times and T-RAPs may require that multiple Bootstraps and the associated signaling are required to minimize channel change time. If

ROHC-U [18] header compression is being utilized, then there is a need to synchronize the context refresh to the T-RAP. This should be supported optionally as shown below in Figure C.1.11. The as depicted below first Baseband Packets after the Preamble may be delivered in a dedicated signaling PLP.

The delivery of an application may be incremental, which is to say that each of the utilized or utilizable data objects for the application may not be delivered entirely within the RAP. It is reasonable and good practice to define the RAP delivered portion of the app such that service acquisition and start may occur. Such details are out of scope for this document.

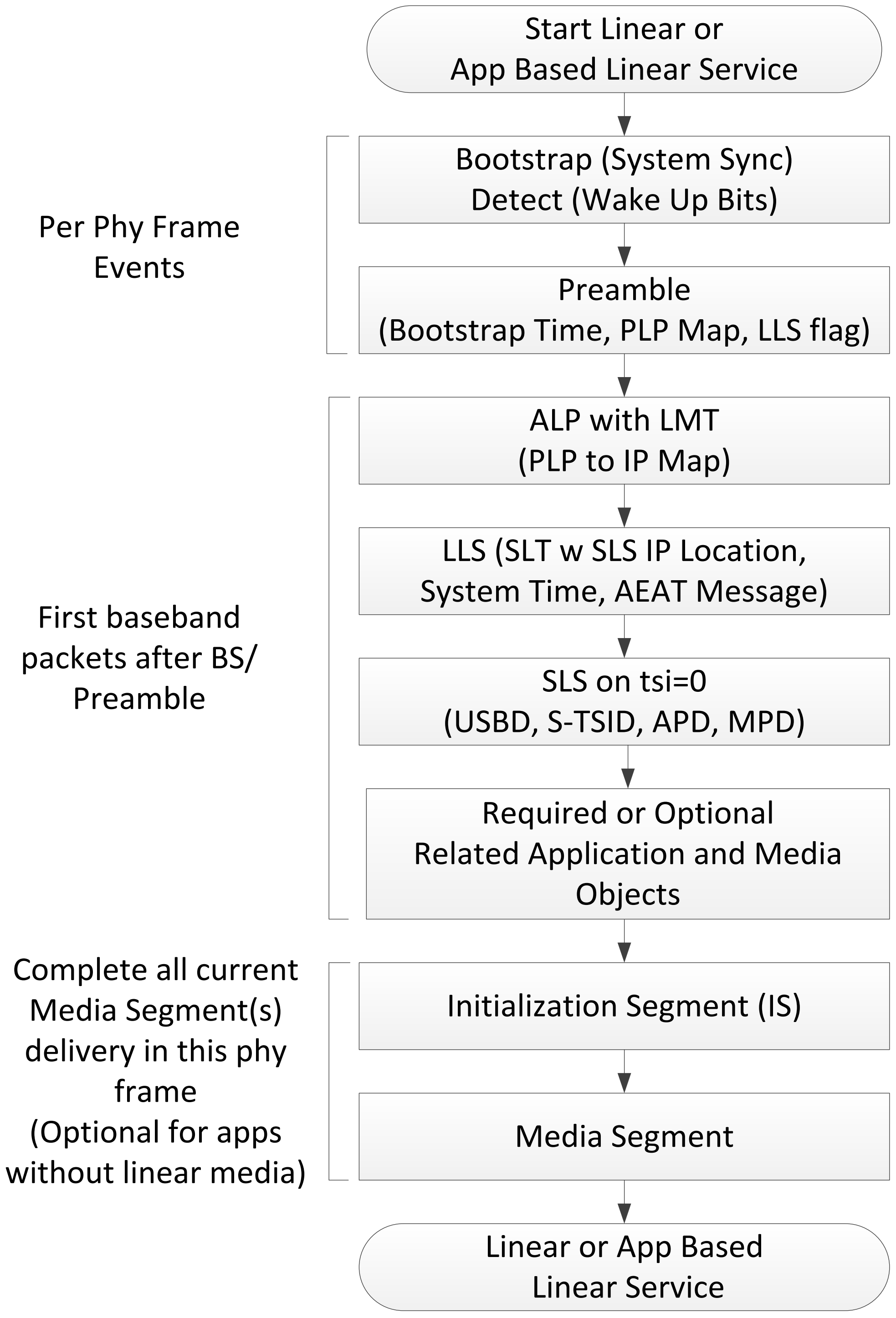
Start Linear or App Based Linear Service
Bootstrap (System Sync) Detect (Wake Up Bits)
Preamble (Bootstrap Time, PLP Map, LLS flag)
Per Phy Frame Events
ALP with LMT (PLP to IP Map)
LLS (SLT w SLS IP Location, System Time, AEAT Message)
SLS on tsi=0 (USBD, S-TSID, APD, MPD)
Required or Optional Related Application and Media Objects
First baseband packets after BS/ Preamble
Initialization Segment (IS)
Media Segment
Complete all current Media Segment(s) delivery in this phy frame (Optional for apps without linear media)
Linear or App Based Linear Service

Figure C.1.11 Order of events to optimize linear service acquisition w/wo an APP.

End of Document