HD Radio™ Air Interface Design Description Advanced Application Services Transport

Rev. G

August 23, 2011

TRADEMARKS

HD Radio™ and the HD, HD Radio, and “Arc” logos are proprietary trademarks of iBiquity Digital Corporation.

“iBiquity”, "iBiquity Digital", and the iBiquity Digital logo are also proprietary trademarks of iBiquity.

All other trademarks, whether claimed or registered, are the exclusive property of their respective owners.

iBiquity Digital Corporation
6711 Columbia Gateway Drive, Suite 500
Columbia, MD 21046
Voice: 443-539-4290
Fax: 443-539-4291
E-mail address:
info@ibiquity.com

Doc. No.: SY_IDD_1019s

Table of Contents

Contents

List of Figures

List of Tables

1 Scope

1.1 System Overview

The iBiquity Digital Corporation HD Radio™ system is designed to permit a smooth evolution from current analog amplitude modulation (AM) and frequency modulation (FM) radio to a fully digital in-band on-channel (IBOC) system. This system delivers digital audio and data services to mobile, portable, and fixed receivers from terrestrial transmitters in the existing medium frequency (MF) and very high frequency (VHF) radio bands. Broadcasters may continue to transmit analog AM and FM simultaneously with the new, higher-quality and more robust digital signals, allowing themselves and their listeners to convert from analog to digital radio while maintaining their current frequency allocations.

1.2 Document Overview

This document describes the Advanced Application Services Transport (AAT). It describes the packet encapsulation of the fixed and opportunistic data and the generation of AAS PDUs. The HD Radio Link Subsystem (HD RLS) provides the packet transport mechanism for the Advanced Application Services Transport (AAT). The HD RLS also provides the packet encapsulation process for the Program Service Data (PSD) and the generation of PSD PDUs which is described in Reference [10]. Specific hardware and software implementation is not described. See References [1] to [24] for more details.

2 References

STATEMENT
Each referenced document that is mentioned in this document shall be listed in the following iBiquity document:

- Reference Documents for the NRSC In-Band/On-Channel Digital Radio Broadcasting Standard Document Number: SY_REF_2690s

3 Abbreviations and Conventions

3.1 Abbreviations and Acronyms

AAS	Advanced Application Services
AAT	Advanced Application Services Transport
AM	Amplitude Modulation
BBM	Block Boundary Marker
CRC	Cyclic Redundancy Check
CCC	Configuration Control Channel
DDL	Data Delimiter
DTPF	Data Transport Packet Format
ECK	Encryption Control Key
FCC	Federal Communications Commission
FCS	Frame Check Sequence
FEC	Forward Error Correction
FM	Frequency Modulation
HDC	HD Codec
HDLC	High-Level Data Link Control
HD RLS	HD Radio Link Subsystem
IBOC	In Band On Channel
IETF	Internet Engineering Task Force
IP	Internet Protocol
LCP	Link Control Protocol
LSB	Least Significant Bit
L1	Layer 1
L2	Layer 2
MF	Medium Frequency
MPS	Main Program Service
SPS	Supplemental Program Service
MPSD	Main Program Service Data
SPSD	Supplemental Program Service Data
OFDM	Orthogonal Frequency Division Multiplexing
PDU	Protocol Data Unit
PPP	Point to Point Protocol
PSD	Program Service Data
RF	Radio Frequency
RFC	Request For Comment
RS	Reed Solomon
UINT	unsigned integer
VHF	Very High Frequency

3.2 Presentation Conventions

Unless otherwise noted, the following conventions apply to this document:

- All vectors are indexed starting with 0.
- The element of a vector with the lowest index is considered to be first.
- In drawings and tables, the leftmost bit is considered to occur first.
- Bit 0 of a byte or word is considered the least significant bit.
- In representations of binary numbers, the least significant bit is on the right.

- When presenting the dimensions of a matrix, the number of rows is given first (e.g., an n x m matrix has n rows and m columns).
- In timing diagrams, earliest time is on the left.

3.3 Mathematical Symbols

3.3.1 Variable Naming Conventions

The variable naming conventions defined below are used throughout this document.

Category	**Definition**	**Examples**
Lower and upper case letters	Indicates scalar quantities	i, j, J, g_{11}
Underlined lower and upper case letters	Indicates vectors	$\underline{u}$, $\underline{V}$
Double underlined lower and upper case letters	Indicates two-dimensional matrices	$\underline{\underline{u}}$, $\underline{\underline{V}}$
[i]	Indicates the i^{th} element of a vector, where i is a non-negative integer	$\underline{u}$[0], $\underline{V}$[1]
[ ]	Indicates the contents of a vector	$\underline{v}$ = [0, 10, 6, 4]
[i] [j]	Indicates the element of a two-dimensional matrix in the i^{th} row and j^{th} column, where i and j are non-negative integers	$\underline{\underline{u}}$[i][j], $\underline{\underline{V}}$[i][j]
$\begin{bmatrix} \quad & \quad \\ & \end{bmatrix}$	Indicates the contents of a matrix	$\underline{\underline{m}} = \begin{bmatrix} 0 & 3 & 1 \\ 2 & 7 & 5 \end{bmatrix}$
n …m	Indicates all the integers from n to m, inclusive	3 …6 = 3, 4, 5, 6
n:m	Indicates bit positions n through m of a binary sequence or vector	Given a binary vector i = [0, 1, 1, 0, 1, 1, 0, 0], $i_{2:5}$ = [1, 0, 1, 1]

3.3.2 Arithmetic Operators

The arithmetic operators defined below are used throughout this document.

Category	**Definition**	**Examples**
·	Indicates a multiplication operation	3·4 = 12
INT( )	Indicates the integer portion of a real number	INT(5/3) = 1 INT(-1.8) = -1
a MOD b	Indicates a modulo operation	33 MOD 16 = 1
⊕	Indicates modulo-2 binary addition	$1 \oplus 1 = 0$
|	Indicates the concatenation of two vectors	$\underline{A}$ = [$\underline{B}$ | $\underline{C}$] The resulting vector $\underline{A}$ consists of the elements of $\underline{B}$ followed by the elements of $\underline{C}$.
j	Indicates the square-root of -1	j = $\sqrt{-1}$
Re( )	Indicates the real component of a complex quantity	If x = (3 + j4), Re(x) = 3
Im( )	Indicates the imaginary component of a complex quantity	If x = (3 + j4), Im(x) = 4
$\log_{10}$	Indicates the base-10 logarithm	$\log_{10}(100) = 2$

4 Overview

4.1 Introduction

The Advanced Applications Services Transport (AAT) is used in the transport of fixed and opportunistic data in the HD Radio system. Figure 4-1 shows the interface of the AAT with the rest of the system. Various Advanced Application Services (AAS) use the Service Interfaces to interact with the HD Radio system. During broadcast, the AAT receives AAS Data from the Service Interfaces and then encodes and encapsulates this data to generate AAS PDUs. The AAS PDUs are then sent to Layer 2 for further processing. The AAS PDUs are sent over different bearer channels which carry fixed data or opportunistic data packets. The opportunistic data bandwidth depends on the audio content transmitted. See Subsection 4.2.1 for details on the bearer channels. Thus, the AAT receives the opportunistic bandwidth status from the Audio Transport allowing the inclusion of opportunistic data.

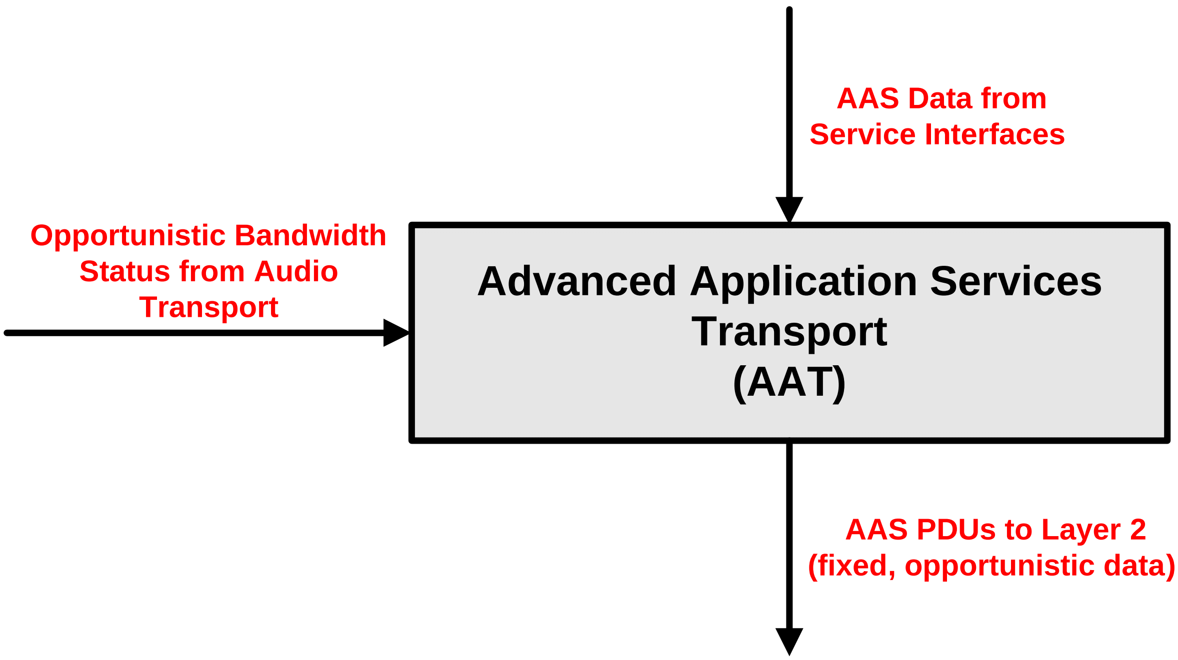

Figure 4-1: AAS Transport Interface

4.2 Packet Transport Mechanism

The packet transport capability is provided by HD RLS which is iBiquity's implementation of the packet transport mechanism for the AAT. It performs the framing and encapsulation of the data packets and generates the AAS PDUs consisting of the fixed and opportunistic data. In addition, the HD RLS also serves as the packet transport mechanism for the PSD Transport and the generation of PSD PDUs which are interleaved along with the audio content ([10]). Figure 4-2 shows the HD RLS packet transport mechanism and the processing of various Data Services which are eventually output to Layer 2 on the different bearer channels. The output of Layer 2 is carried by the respective Layer 1 logical channels (References [1] and [2]) to the Waveform/Transmission layer.

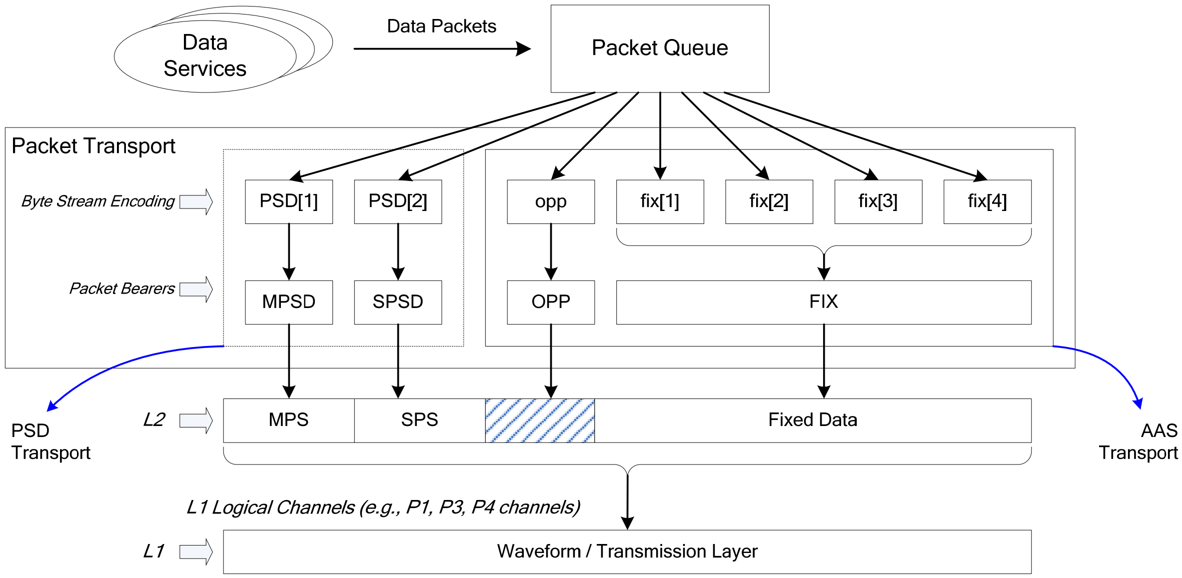

Figure 4-2: HD RLS Packet Transport Mechanism

At the transmit side, data packets from various Data Services are queued for transmission. Packets to be transmitted are encoded in streams that are transmitted over one or more bearer channels which are then packed into a Layer 2 PDU for broadcast through the HD Radio system (See Subsection 7.1). The actual transmission of a packet may be affected by numerous factors, including bandwidth allocation for the bearer channel and packet length. Therefore, packets may be transmitted over multiple L1 frames.

4.2.1 Bearer Channels

The HD Radio system provides multiple channels for carrying data streams (References [1] and [2]). These channels are categorized based on the type of data they carry and are referred to as bearer channels. Each of these bearer channels is used to transport data packets in one or more encoded byte streams.

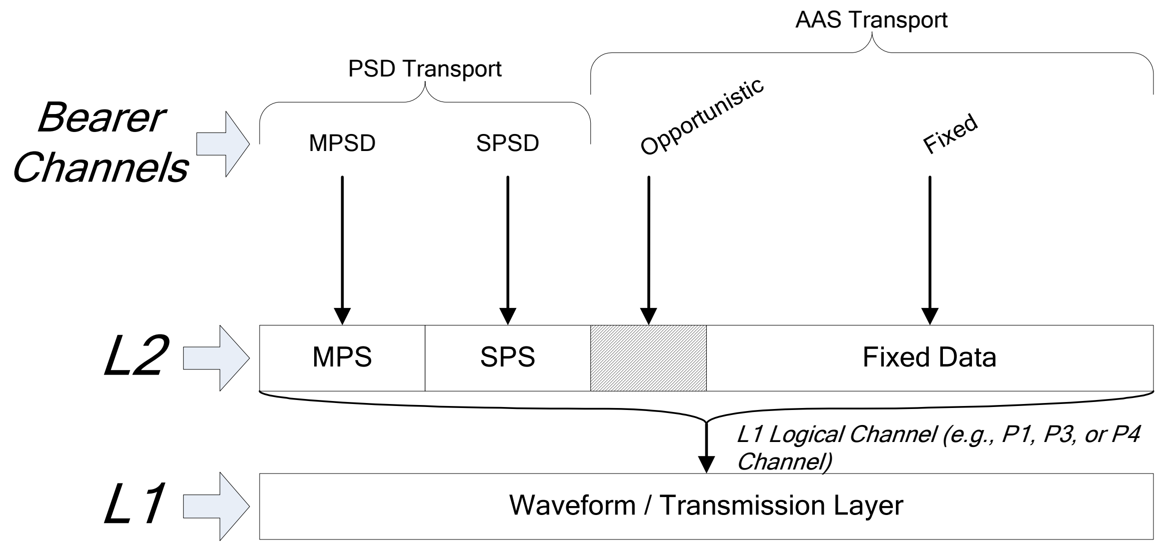

Figure 4-3: Bearer Channels

The different bearer channels are as follows:

- **PSD** – A PSD bearer channel is created from bytes allocated within the Audio Transport frames that carry digital audio for Main Program Services (MPS) and Supplemental Program Services (SPS). The Audio Transport obtains the encapsulated PSD byte streams, if present, and multiplexes them with the encoded audio packets. The PSD byte streams can be either associated with the Main Program Service Data (MPSD) or with the Supplemental Program Service Data (SPSD). The HD RLS mechanism within the PSD Transport encapsulates the PSD as PSD PDUs which are then multiplexed along with the audio program. The Audio Transport provides the mechanisms for inserting PSD at the transmitter and extracting it at the receiver.
- **Opportunistic** – If the instantaneous audio content requires less than its regularly allocated portion of the L2 PDU, the instantaneously available capacity is used to create an opportunistic bearer channel. The opportunistic bytes, if they exist, are always located in the L2 PDU before the fixed data. If there is no fixed data allocation, opportunistic bytes are located at the end of the L2 PDU.
- **Fixed** – A fixed data bearer channel uses a dedicated portion of the L2 PDU which has been allocated for data services. The fixed data is always located at the end of the L2 PDU and maintains a constant size over long periods (that is, many PDUs).

The AAS Transport uses the HD RLS mechanism to encapsulate the fixed and opportunistic data (respectively, from these bearer channels) as AAS PDUs which are sent to Layer 2.

Section 7 provides detailed descriptions of how data is sent in each type of bearer channel.

4.2.2 Packet Structure

The data packets in the HD Radio system are structured as variable length datagrams, as briefly shown in Figure 4-4. Section 5 contains a detailed description of data packet structure.

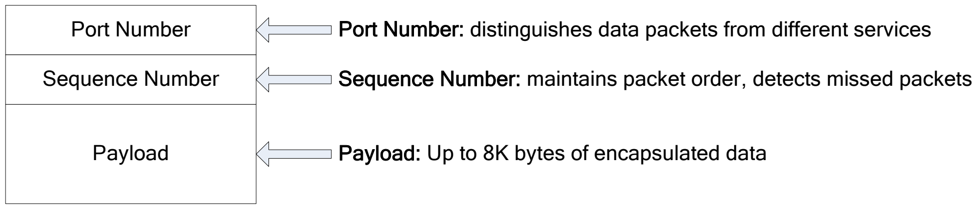

Figure 4-4: Packet Structure

4.2.3 Byte Stream Encoding

Packets are encoded in a serial byte stream before they are carried on the bearer channels. To improve reliability, the encoded byte stream may be protected with an adjustable level of Forward Error Correction (FEC) that consists of a block error code, interleaving, and data blocking, The FEC level can be customized for each bearer channel.

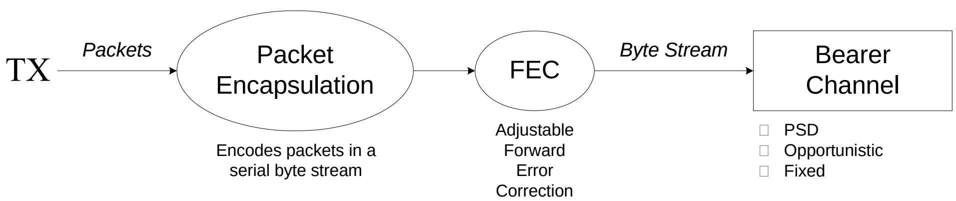

Figure 4-5: Byte Stream Encoding

Section 6 contains a detailed description of packet encapsulation and FEC used to encode packets in byte streams.

5 Data Packet Structure

The AAS Data packets are structured by HD RLS to allow applications at the transmit side to send data to applications at the receiver over the AAT. The fields of the data packet are listed in Table 5-1.

Table 5-1: Data Packet Format

Field	Size in Bytes and Format*
PORT	2 (little-endian format)
SEQ	2 (little-endian format)
Payload[ ]	1 to 8192 (byte format)

* Payload is limited to a maximum size of 8192 bytes after the addition of any escape bytes as defined in Subsection 6.2.5. The PORT and SEQ fields are nominally 2 bytes in length but may be 3 or 4 bytes in length if one or both bytes need to be escaped, respectively.

5.1 Port Number

Port numbers in HD RLS are used to identify data packets with respect to the data services supported by their content. Port numbers indicate to the receiver the application to which a received packet should be directed. The format is little-endian.

While certain port numbers are always associated with specific use, other port numbers may be allocated by HD Radio AAS for specific allocation. The mapping of port numbers to applications is not defined by HD RLS. For example, port number 0x5100 is used for Main Program Service Data (MPSD); ports 0x0000 through 0x03FF are reserved for use by the HD Radio system and are not available to applications. Table 5-2 lists the assignment of the port numbers.

NRSC Supplemental Information provides the most up-to-date information regarding the port number assignments: http://www.ibiquity.com/broadcasters/us_regulatory/nrsc_supplemental_information.

Table 5-2: Port Number Assignment

Port Number	Status
0x0000 to 0x0400	Reserved for System Use
0x0401 to 0x50FF	Available for Specific Applications
0x5100	MPSD
0x5101 to 0x5200	Reserved for System Use
0x5201 to 0x5207	SPSD
0x5208 to 0x52FF	Reserved for System Use
0x5300 to 0x7CFF	Reserved for Future Applications
0x7D00 to 0x7EFF	Invalid – Shall not be used
0x7F00 to 0xFEFF	Reserved for Future Applications
0xFF00 to 0xFFFF	Reserved for System Use

5.2 Sequence Number

At the transmitter, a packet sequence number is maintained for each packet sent to a given port. It is incremented (by 1) for every new packet for that port. The sequence number is incremented independently per port. This allows for packet order to be verified at the receiver and for lost packets to be detected through missing sequence numbers. The sequence number is in little-endian format.

5.3 Packet Payload

The packet payload can be of any size up to 8192 encapsulated (i.e., including escape mechanism for specific characters – see Section 6.2.5) bytes in length.

6 Byte Stream Encoding

6.1 Overview

Each data bearer channel in the HD Radio system transports a stream of bytes. The allocated number of bytes in each L1 frame is constant (fixed data) or variable (PSD and opportunistic data). The aggregate number of bytes used in each L1 frame may vary.

To send data over these channels, packets are encoded in a continuous byte stream. Successful packet delivery relies on the bearer channels to deliver the bytes in the same order that they were transmitted. Further, there is no synchronization between encoded byte streams and their bearer channels. The portion of these encoded byte streams, which a bearer channel transports in an L1 frame, may contain one or more packets, a portion of a single packet, portions of multiple packets, or fragments of multiple packets (when FEC is used).

The general structure of an encoded byte stream is shown in Figure 6-1.

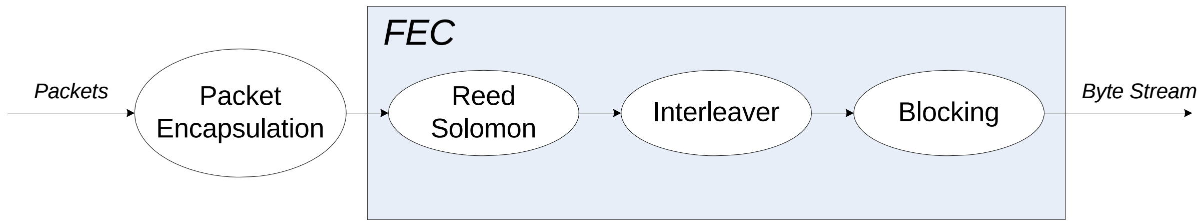

Figure 6-1: Byte Stream Encoding

Packet Encapsulation encodes AAS Data packets as a serial byte stream with embedded error detection.

Forward Error Correction (FEC) may be applied to the encoded packet stream to control packet loss and errors using the following methods:

- Reed Solomon block coding for error correction
- Byte Interleaving to protect against error bursts
- Block synchronization mechanism

6.2 Packet Encapsulation – PDU Generation

The packet encapsulation used by the HD RLS mechanism follows the HDLC-like framing employed by the Point-to-Point Protocol (PPP) as standardized by the IETF in RFC 1662 (Reference [24]). The following sections describe how the HDLC-like framing of PPP has been adapted for the HD Radio system.

The HDLC-like framing allows encapsulation of a packet within a byte stream, referred to as AAS PDU, that may be broadcast in segments of arbitrary size (for example, in each L1 frame). Reconstruction of the packet requires only concatenation of the segments. Depending on their size, a single L1 frame may contain multiple such encapsulated packets or a single portion of a large packet. The L1 frame rate would depend on which L1 logical channel is being used to transport the packets (References [1] and [2]).

6.2.1 PDU Structure

An AAS PDU is contained in an HDLC-like frame delimited by flags as shown in Table 6-1.

Table 6-1: AAS PDU Field Definition

<table>
<tr><th colspan="2">Field</th><th>Bytes</th><th>Description</th></tr>
<tr><td colspan="2">Flag</td><td>1</td><td>0x7E indicates the end of the previous PDU and the start of the current PDU</td></tr>
<tr><td colspan="2">Data Transport Packet Format (DTPF)</td><td>1</td><td>Indicates the format of the current PDU. Refer to 6.2.3 for details</td></tr>
<tr><td rowspan="3">Conditional</td><td>ECK Length</td><td>1</td><td rowspan="3">The ECK field, in whole, is optional and is only present in specific PDU formats that include restricted access to the AAS data.
These bytes are not used if DTPF = 0x21</td></tr>
<tr><td>ECK No.</td><td>1</td></tr>
<tr><td>ECK Data</td><td>2 - 32</td></tr>
<tr><td colspan="2">PORT</td><td>2*</td><td rowspan="3">AAS Data packets as defined in Section 5</td></tr>
<tr><td colspan="2">SEQ</td><td>2*</td></tr>
<tr><td colspan="2">Payload[ ]</td><td>1 – 8192*</td></tr>
<tr><td colspan="2">Frame Check Sequence (FCS)</td><td>2*</td><td>A 16-bit FCS is used for error detection in little-endian format.</td></tr>
<tr><td colspan="2">Flag</td><td>1</td><td>0x7E indicates the end of the current PDU and the start of the next PDU</td></tr>
</table>

* Payload is limited to a maximum size of 8192 bytes after the addition of any escape bytes as defined in Subsection 6.2.5. The PORT, SEQ, and FCS fields are nominally 2 bytes in length but may be 3 or 4 bytes in length if one or both bytes need to be escaped, respectively.

This PDU structure follows that described in Reference [24] except for the following differences:

1. The Address and Control fields provide no useful function in the HD Radio system and have been eliminated in the interest of efficiency.
2. The Data Transport Packet Format (DTPF) field size is always eight bits
3. No padding is used.
4. The Frame Check Sequence (FCS) is always 16 bits.
5. An ECK (Encryption Control Key) field is conditionally added, depending on the value (format indication) in item 2.

Figure 6-2 shows the structure of the AAS PDU.

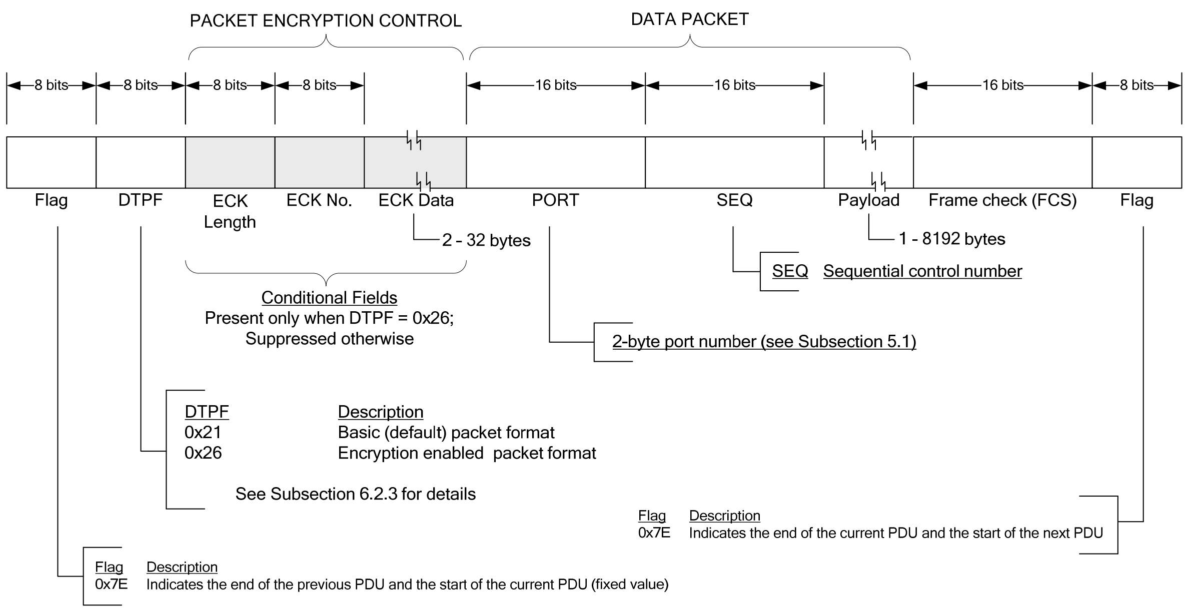

Figure 6-2: AAS PDU Structure

6.2.2 Flag Delimiters

Each PDU is delimited by Flag bytes having the value 0x7E. The Flag delimiters support the following capabilities:

- Only one byte is needed to delimit a packet of any length.
- A false Flag due to a payload error results only in the loss of a single PDU (data packet).
- A corrupted Flag cannot cause a loss of more than two PDUs (data packets).

A single L1 frame may consist of partial or multiple instances of such PDUs. The Flag bytes help in identifying and delimiting each PDU in such instances.

6.2.3 Data Transport Packet Format

The Data Transport Packet Format (DTPF) field, as described in Table 6-2, is used to define the packet format for that PDU. The available (un-used) numbering range allows new packet formats to be added in the future while retaining backward compatibility with older receivers.

Table 6-2: DTPF Allocated Values

DTPF	Description
0x21	Basic packet format. This is the configuration default
0x26	Encryption enabled packet format.
0x00 – 0x20	Reserved for system use
0x22 – 0x25	Reserved for system use
0x27 – 0x7C	Reserved for system use
0x7F – 0xFF	Reserved for system use
0x7D – 0x7E	Invalid. Never used due potential conflict with control flags.

DTPF value 0x21is the default. It indicates that the information contained in the PDU conforms to the basic packet structure defined in Section 5. DTPF value 0x26 indicates that the encapsulated packet format includes additional fields that carry encryption control information. That ECK (Encryption Control Key) includes the key data, key length, and key enumerator for integrity control. All the other DTPF values are unassigned and reserved for future use.

6.2.4 Frame Check Sequence

The Frame Check Sequence (FCS) uses a 16-bit CRC. The FCS is generated using the DTPF field and the fields in the packet structure (refer to Table 5-1) in accordance with Reference [24]. The FCS is in little-endian format.

6.2.5 Transparency

To prevent the values of 0x7E and 0x7D occurring in the data from being read as Flags, an escape mechanism is provided to replace these bytes with a special meaning with alternate values. This is done by replacing the byte with two bytes consisting of the control escape byte 0x7D followed by the original byte exclusive-or'ed with hexadecimal 0x20 (Reference [24]). The only two values that need to be escaped are:

0x7E which is encoded as 0x7D, 0x5E, (Flag Sequence)

0x7D which is encoded as 0x7D, 0x5D (Control Escape)

Since the escape mechanism requires two bytes to encode a single byte, it reduces efficiency by approximately 1% (average) for a long packet with a random data payload.

6.2.6 Idle Pattern

When no packet data is available to send, an idle pattern of repeating Flags may be sent. This is equivalent to a stream of zero length frames and is used so that there is always data to fill a bearer channel.

6.2.7 Application of IETF RFC 1662 for HD Radio Data Transport

Many of the features defined in IETF RFC 1662 (Reference [24]) are inapplicable or unnecessary for data transport in the HD Radio system. In particular, the following sections of the RFC 1662 are ***not*** applicable to the HD Radio system:

- Sections 4.4.2, 4.5.2, and 5 – All streams used for packet transport in the HD Radio system are octet-synchronous.
- Section 6 – No asynchronous-to-synchronous conversion is used.
- Section 7 – The Flag Sequence and Control Escape are the only control flags used in the HD Radio system. Negotiation of additional control characters is not possible and not required.
- Section A – LCP negotiations are not possible and are not used.
- Section B – The PPP frames identified are not valid HD Radio frames.

6.2.8 Encapsulated Packet – Example

Figure 6-3 shows an example of an encapsulated packet (PDU). The payload shown is an ID3 tag for PSD. However, this payload can also be present as AAS Data. As a bearer channel, PSD is also encapsulated in a similar fashion as described in the above sections by the HD RLS mechanism in the PSD Transport. The following elements are noted in the figure:

1. The beginning of the frame is indicated by a Flag Sequence (0x7E).
2. The first byte of the frame is the DTPF field which is set to 0x21, indicating the basic (default) packet format.
3. The next two bytes, following DTPF, contain the Port Number, 0x5100, in little-endian format.
4. A two-byte Sequence Number in little-endian format is shown next. The value of 0x0000 is meaningful only with respect to the sequence numbers of the previous and subsequent packets sent to Port 0x5100.
5. The payload is an ID3 tag that encodes the song title ("Analog Blues"), the artist ("J.Q. Public"), and the album name ("The Lost Sessions").
6. The payload is followed by a two-byte Frame Check Sequence (FCS) in little-endian format. It is computed over all bytes from the DTPF field through the last byte of the payload.
7. The end of frame is indicated by a Flag Sequence (0x7E).

Note: The byte stream shown could arrive as segments of arbitrary size as long as the byte order is preserved. Lost segments will result in short packets that fail FCS checking.

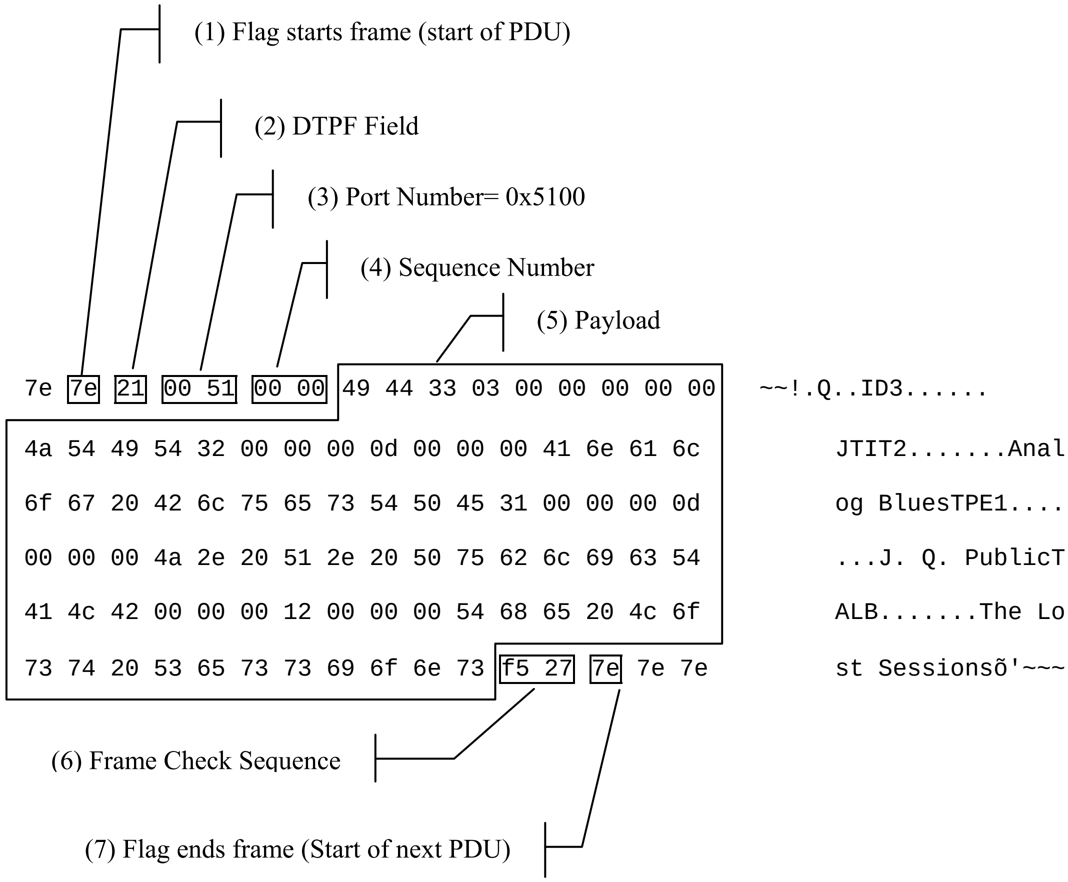

Figure 6-3: Encapsulated Packet – Example

AAS PDUs may span multiple transfer frames as defined in [1] and [2]. Figure 6-4 shows this example. Constraints such as number of streams, size of bearer channel, and others may allow for this scenario. In other scenarios, multiple AAS PDUs may be transmitted within one L1 frame.

L1 Frame 1

Start of AAS PDU	DTPF Field	PORT	Sequence Number	Start of Payload
0x7E	0x21	0xHHHH	0x0001	

L1 Frame 2

Middle section of Payload

L1 Frame 3

End of Payload	FCS	Start Next AAS PDU	DTPF Field	Next PDU
	2 bytes	0x7E	0x21	

Figure 6-4: AAS PDUs Spanning Multiple L1 Frames

6.3 Reed Solomon Coding

All Reed-Solomon (RS) coding in HD RLS uses the extended Galois Field over 2^8 using the characteristics shown below. All codeword blocks are 255 bytes long (that is, no shortened blocks are used). Key characteristics of the Reed-Solomon coding are:

- Primitive polynomial is $x^8+x^4+x^3+x^2+1$ (100011101 in binary notation, most significant bit on the left).
- The RS coding may be configured with any number of roots (parity elements), $p \leq 64$.
- Generator polynomial g(x) = $\prod_{i=1}^{i=p}\left(x - a^i\right)$, where a is a root of the primitive polynomial.
- The first byte of the RS coding block is the leftmost in Figure 6-2. The parity bytes are the rightmost in the payload of each coding block shown in Figure 6-2.

Figure 6-5 shows a block of data, coded using RS (255, 223), which contains 223 data bytes and 32 parity bytes in each block (that is, codeword). The 223-byte data block contains a single “1” followed by 222 zeros. The 32 RS parity bytes are located at the end of the block and are shaded in gray.

```
 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 8b
1b e9 a3 e3 cb 72 1b ba 1c 2e 5c 06 8b 93 b1 03
93 37 e7 b7 d4 ca e3 61 9c f4 e1 de 74 8d f3
```

Data Bytes

RS Parity Bytes

Figure 6-5: Reed-Solomon (255, 223) Coding Example

6.3.1 Coding Rates

The AAS configuration mechanism supports a wide range of RS encoder configurations, resulting in a wide range of coding rates. The encoder may be configured from (255,253) to (255,191).

Due to practical considerations, such as simplified configuration control and reduced throughput versus error correction benefits, a default configuration has been chosen. The default configuration is (255,223), thus providing a coding rate of r = 0.875.

It is noted that the default configuration applies only if and when FEC is applied.

6.4 Byte Interleaving

6.4.1 Overview

The HD RLS mechanism employs a convolutional byte interleaver. An interleaver row consists of 255 columns (bytes), which is the size of one RS codeword. Thus, the FEC stream is a series of 255-byte RS encoded blocks, and the interleaver design allows decoding to start at any FEC block in the sequence. The interleaver column size depends on the error protection level.

The RS codeword bytes are mapped to the interleaver matrix. Write operations are sequential in rows, while read operations are addressed by the applicable equations given in Subsections 6.4.2.1 and 6.4.2.2. Every (entire) codeword is written into a row, starting from the first row and first column, so that the rows and the codewords are aligned. Consecutive codewords are placed in consecutive rows.

6.4.2 Interleaver Equations

6.4.2.1 Interleaver Column Selection – Read Operation

Apply the following equation using the applicable parameters:

$$Column(i) = (i \cdot N_s)MOD(255)$$

where N_s is 53

6.4.2.2 Interleaver Row Selection – Read Operation

Apply the following equation using applicable parameters:

$$Row(i) = [i - 254 \cdot INT(i/255)]MOD(R_w)$$

where R_w is the number of rows in the interleaver matrix

6.4.3 Interleaver Timing

The interleaving operation starts by writing to the first location of the interleaver, marked as (0, 0). After the first write operation, every write operation has to be followed by a read operation. This means that the first location of each row is being written into and read out immediately, experiencing no delay.

The process continues, while the write operations are by row and the read operations are by using the equations above.

6.4.4 Interleaving Range

The interleaver may be configured to a span (“depth”) ranging from one code word (meaning no interleaving) to 64 code words. The default operation setting is “no interleaving.” No “optimized” recommendations or default interleaving span values (“depths”) are applied since interleaving performance is related to the bit rate allocation for the data stream and may result in a different interleaving time for the same interleaver span.

6.5 Block Synchronization

FEC and interleaving occur after every *n* blocks of 255 bytes (the length of a Reed Solomon codeword). The exact number *n* depends on the specific configuration, as further discussed. To enable the receiver to identify the start of these blocks, a four-byte Block Boundary Marker (BBM) is regularly inserted into the stream to indicate the start of a block, as shown in Figure 6-6.

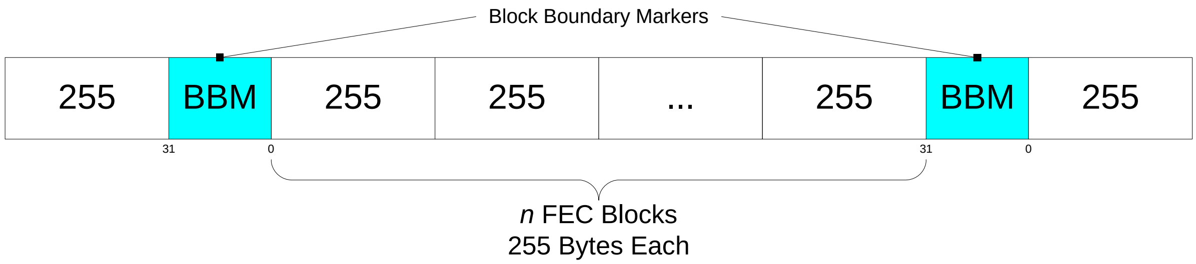

Figure 6-6: FEC Blocking with Block Boundary Markers

The number of FEC blocks between BBMs is set at $n=1$ for Opportunistic bearer channels, and $n=4$ for Fixed bearer channels.

The BBM is defined as follows,

BBM = [01111101001110101110001001000010]

where bit 0 (least significant bit (LSB)) is on the right and the left-most bit is sent first.

7 Bearer Channels

The HD Radio broadcast system supports multiple bearer channels to transport packets in one or more encoded byte streams. The AAS PDUs are transmitted on the fixed and opportunistic bearer channels and the PSD PDUs are transmitted on the PSD bearer channel. The HD RLS enables the encapsulation of the data packets (both AAS and PSD) as PDUs before they are sent over the different bearer channels to Layer 2 for further processing.

7.1 Layer 2 PDU Packing

Figure 7-1 shows how the digital audio and data are packed into a Layer 2 PDU. In addition to transporting the data, Layer 2 also provides an indication whether the Layer 2 PDU contains audio, opportunistic data, and/or fixed data.

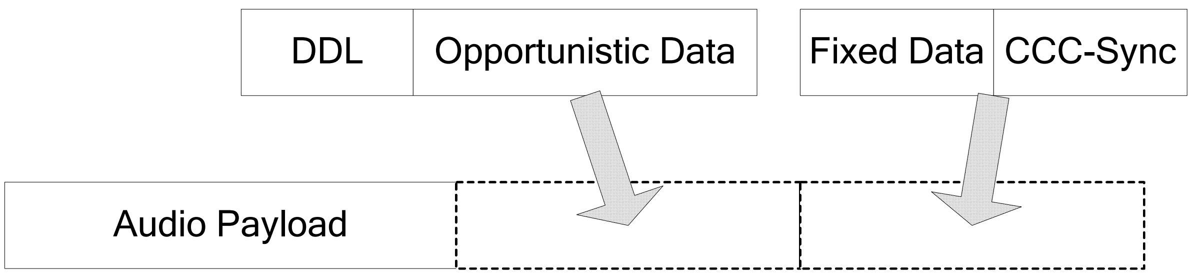

Figure 7-1: Layer 2 PDU Packing

Reference [3] provides a detailed description of the Layer 2 transmit processing.

7.2 Bearer Channel Comparison

Table 7-1: Bearer Channel Comparison

Bearer Channel Type	Bearer Channel Description	Reed Solomon	Interleaver	BBM Frequency
PSD	Bytes allocated in MPS/SPS PDU	FEC not used		
OPPORTUNISTIC	Unused bytes allocated to audio programs Variable capacity	(255,223)	None	1 : 1
FIXED	Uses allocated segment(s) of L2 frame “Infinitely” variable FEC	(255,255) to (255,191)	0 to 64 Blocks	1 : 4

7.3 PSD Bearer Channel

A PSD bearer channel uses bytes allocated within the Audio Transport for the transmission of program-related packet data. The encoded byte stream transmitted over a PSD bearer channel contains encapsulated packets (PSD PDUs) from the PSD Transport with no forward error correction. The Audio Transport provides the mechanism to identify the number and position of the encoded bytes in each transmitted Audio Transport PDU.

7.4 Opportunistic Bearer Channel

During silence and simple audio passages the encoded digital audio might require less than its allocated bandwidth. When this occurs, the unused capacity may be used to send data packets over the opportunistic bearer channel. The size of the opportunistic payload is determined on the basis of whether the audio programs use their full allocated capacity or not.

The data packet encapsulation is described in Subsection 6.2.

The FEC chain is defined in Section 7.4.1 and the settings are not configurable.

7.4.1 FEC Coding for Opportunistic Bearer Channels

For opportunistic bearer channels the byte stream uses:

- A fixed Reed Solomon coding rate of (255,223)
- No byte interleaving
- A block boundary marker (BBM) for every Reed Solomon codeword

This choice of settings provides robust error tolerance and minimal latency.

7.4.2 Opportunistic Data Identification/Delimiting

To allow opportunistic data to be identified in the Layer 2 PDU (Reference [3]), a five-byte data delimiter (DDL) field is used to identify the start of the opportunistic data in the PDU. The end of the opportunistic data is either the end of the PDU or the start of the fixed data payload (if present).

The five-byte DDL sequence has a binary value of [1001110100101111001110010000010111011000] where ddl_0 (LSB) is on the right.

7.5 Fixed Bearer Channel

A fixed bearer channel contains the following:

- A Synchronization Channel (SYNC)
- A Configuration Control Channel (CCC)
- From one to four sub-channels containing fixed byte streams with different FEC

Figure 7-2: Fixed Bearer Channel Structure

7.5.1 Synchronization Channel

The one-byte wide Synchronization Channel, as described in Table 7-2, encodes the Configuration Control Channel width and timing information to allow AAS PDU synchronization between the transmitter and receiver. The bytes are sent in the following sequence: count, width, width, width, ..., count, width, width, width.

Table 7-2: Synchronization Channel

Synchronization Channel		
Element	**Type**	**Description**
width	UINT8	Valid values for the Configuration Control Channel sequence width (in bytes) are any even number from 2 through 30, or one. These values are represented in the Synch Width byte in the form 0xNN, where N is one-half the number of bytes of the Configuration Control Channel. The byte 0xNN consists of two four-bit nibbles, each of the value 0xN. Thus, for N=0x1 through 0xF, the Configuration Control Channel width takes the value of an even number from 2 through 30 bytes (0xN · 2) bytes. For example, 0x44 indicates a CCC width of 8 bytes. For N=0, a CCC width of one byte is defined.
count	UINT8	The initial value of the count is zero. Each time the count is sent, it is incremented by 4 modulo 256. The count is intended to allow for synchronizing fixed channel reconfiguration in the future.

7.5.2 Configuration Control Channel

The Configuration Control Channel (CCC), as described in Table 7-3, sends a repeating message describing the number, width, and FEC configuration of the fixed sub-channels. The Configuration Control Channel can describe from one to four fixed sub-channels using from one to 20 bytes to allow for the optimum compromise between channel overhead and the time to process a complete fixed configuration message. The message is encapsulated to provide framing and error detection. Note that the mode, length, and CRC-16 elements may need additional escape bytes to be applied, as defined in Subsection 6.2.5. However, these escape bytes are not counted in the computation of the Synchronization Channel width element.

Table 7-3: Configuration Control Channel

HD RLS Configuration Control Channel Message		
Element	**Type**	**Description**
0x7E	UINT8	HDLC Flag When computing the value of the Synchronization Channel width element, this byte is counted
0x00	UINT8	Dummy byte for padding in order to maintain even width
mode[0]	UINT16	A 16-bit value indicating the FEC encoding of the 0^{th} sub-channel
length[0]	UINT16	A 16 bit value indicating the number of bytes in the 0^{th} sub-channel (in little-endian format)
…		Additional mode and length pairs
mode[n]	UINT16	Mode of n^{th} sub-channel
length[n]	UINT16	Length of n^{th} sub-channel in bytes (in little-endian format)
CRC-16	UINT16	16 bit FCS supplied by HDLC provides error detection
0x7E	UINT8	HDLC Flag for next message

The mode bits are defined in Figure 7-3. The default value of 0x0000 indicates no FEC encoding or interleaving.

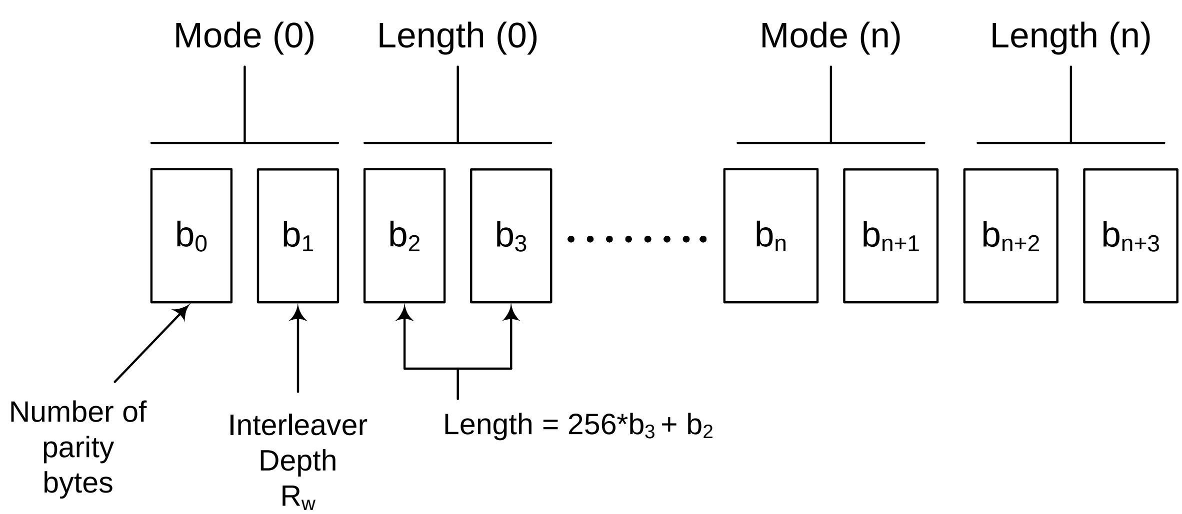

Figure 7-3: Mode Bit Definitions

In Figure 7-3, b_0 and b_1 are the high-byte and low-byte of Mode (0); b_2 and b_3 are the high-byte and low-byte of Length (0).

Where,

- b_0 = Number of Parity Bytes – This is the number of parity bytes in each Reed-Solomon codeword. A value of N implies a coding of (255-N):N
- b_1 = Interleaver Depth – Number of interleaver rows, R_w
- b_2 and b_3 = Length (in little-endian format) – The number of bytes the sub-channel is allocated in each Layer 2 PDU

A fixed portion of the Layer 2 PDU is allocated for every fixed sub-channel byte stream. The length – b2 and b3 – refers to the actual message length of the fixed sub-channel byte stream in the Layer 2 PDU structure.

The Configuration Control Channel messages require two HDLC flags to delimit them. However, as noted in Table 7-2, valid values for the CCC width are any even number from 2 through 30 bytes, or one byte. Therefore, additional dummy bytes are padded in order to maintain the even width and the byte counting as shown in Table 7-3.

The minimum number of bytes needed to transmit a single Configuration Control Channel message is eight: 1 flag, 1 padding byte, mode, length, and CRC-16. Therefore, a one-byte Configuration Control Channel message can specify a single fixed sub-channel payload configuration in 8 L1/L2 PDU structures. An 8-byte wide Configuration Control Channel accomplishes this in a single L1/L2 Frame PDU structure.

The fixed bearer channel Configuration Control Channel can describe from one to four fixed sub-channels. The total required number of Configuration Control Channel bytes is listed in Table 7-4.

Table 7-4: Required CCC Length

Number of Sub-Channels	CCC Total Length [bytes]
1	8
2	12
3	16
4	20

7.5.3 Fixed Sub-Channel Byte Stream

Each fixed sub-channel byte stream uses:

- A fixed Reed Solomon coding rate of (255, 255-N_P), where N_P is the number of parity bytes specified in the configuration message and $N_P \leq 64$.
- Byte interleaver depth of N_I, where N_I is the length of the interleaver in units of 255-byte blocks. $N_I \leq 64$ and is specified in the configuration message.
- A block boundary marker (BBM) for every four FEC block codewords.

Figure 7-4 shows an example of the fixed sub-channel byte stream for a 32-bit Reed Solomon parity.

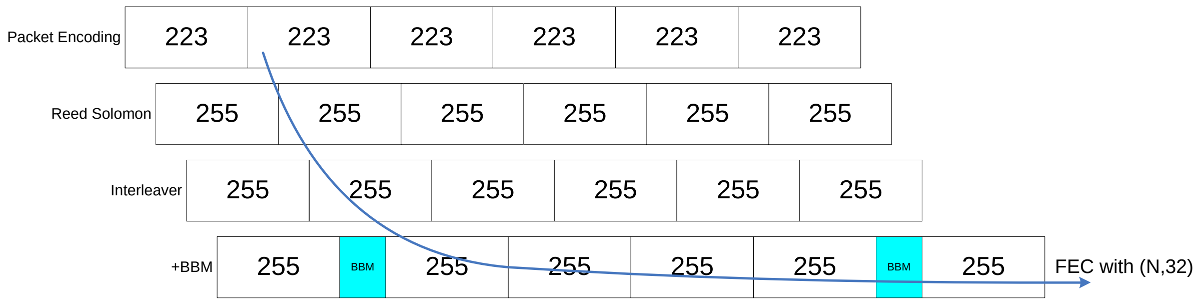

Figure 7-4: Fixed Sub-Channel Byte Stream