ATSC Recommended Practice: Guidelines for the Physical Layer Protocol (A/327)

Doc. A/327:2018
2 October 2018

Advanced Television Systems Committee
1776 K Street, N.W.
Washington, D.C. 20006
202-872-9160

LGE 1022

The Advanced Television Systems Committee, Inc., is an international, non-profit organization developing voluntary standards and recommended practices for digital television. ATSC member organizations represent the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries. ATSC also develops digital television implementation strategies and supports educational activities on ATSC standards. ATSC was formed in 1983 by the member organizations of the Joint Committee on Inter-society Coordination (JCIC): the Electronic Industries Association (EIA), the Institute of Electrical and Electronic Engineers (IEEE), the National Association of Broadcasters (NAB), the National Cable Telecommunications Association (NCTA), and the Society of Motion Picture and Television Engineers (SMPTE). For more information visit www.atsc.org.

Note: The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. One or more patent holders have, however, filed a statement regarding the terms on which such patent holder(s) may be willing to grant a license under these rights to individuals or entities desiring to obtain such a license. Details may be obtained from the ATSC Secretary and the patent holder.

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
Recommended Practice approved	2 October 2018

Table of Contents

1. SCOPE 1
   1.1 Introduction and Background 1
   1.2 Organization 1
2. REFERENCES 1
3. DEFINITION OF TERMS 2
   3.1 Compliance Notation 2
   3.2 Acronyms and Abbreviations 2
   3.3 Terms 5
4. SYSTEM OVERVIEW AND GUIDELINES FOR PHYSICAL LAYER MODE 6
   4.1 System Overview 6
   4.2 Guidelines for Physical Layer Mode 8
      4.2.1 FFT Size 8
      4.2.2 Bandwidth and Bandwidth Reduction 8
      4.2.3 Pilot Pattern 9
         4.2.3.1 Separation of Pilot Carriers (D_x) 9
         4.2.3.2 Length of Pattern in Symbols (D_y) 10
      4.2.4 Pilot Boosting 10
      4.2.5 Frame and Subframe Length 12
         4.2.5.1 Frame Length 12
         4.2.5.2 Number of Preamble Symbols 13
         4.2.5.3 Subframe Configuration 13
      4.2.6 Symbol-Aligned or Time-Aligned Mode 14
      4.2.7 PLP Multiplexing 15
         4.2.7.1 PLP Cell Multiplexing 15
         4.2.7.2 Layered Division Multiplexing 16
      4.2.8 LDM Parameters 16
         4.2.8.1 LDM Injection Level 16
         4.2.8.2 LDM ModCod Combination 18
      4.2.9 Code Rate, Length and Constellation 18
      4.2.10 L1 Protection Mode 20
      4.2.11 Time Interleaver Mode 21
         4.2.11.1 Valid Conditions for CTI Mode 21
         4.2.11.2 Valid Conditions for HTI and No TI Mode 22
         4.2.11.3 Extended Time Interleaving 23
      4.2.12 Guard Interval 24
      4.2.13 Frequency Interleaver Mode 25
      4.2.14 Time Information Type 25
      4.2.15 Multiple Subframes within a Frame 26
      4.2.16 L1D_plp_fec_type 26
      4.2.17 Transmitter Identification (TxID) 27
5. GUIDELINES FOR TRANSMITTER IMPLEMENTATION 28
   5.1 Input Formatting 28
      5.1.1 Delivered Product in Multiple PLPs 28
   5.2 Bit Interleaved and Coded Modulation (BICM) 29
      5.2.1 Forward Error Correction (FEC) 29

5.2.1.1 Inner Encoding 30
5.2.2 Bit Interleaving 31
5.2.3 Constellation Mapping 32
5.2.4 Layered Division Multiplexing (LDM) 32
5.2.5 Protection for L1-Signaling 34
5.2.5.1 Common Block for L1-Basic and L1-Detail 34
5.2.5.2 L1-Detail Specific Block: L1 Segmentation 35
5.2.5.3 L1-Detail Specific Block: Additional Parity 35
5.3 Framing and Interleaving 36
5.3.1 Time Interleaving 36
5.3.1.1 Convolutional Time Interleaving (CTI) 36
5.3.1.2 Hybrid Time Interleaving (HTI) 36
5.3.2 Frame Structure 37
5.3.2.1 Example Scenario for Power Saving Aspect 37
5.3.2.2 Example Scenario for Performance Aspect 38
5.3.3 LDM and PLP Multiplexing 41
5.3.3.1 Definition of L1D_plp_start and L1D_plp_size 41
5.3.3.2 Indexing TI Groups 42
5.3.3.3 Injection Level (L1D_plp_ldm_injection_level) for Multiple Enhanced PLPs 43
5.3.3.4 Positioning Enhanced PLP(s) and Not Recommended LDM Cases 44
5.3.3.5 LDM Configuration with Different TI Modes 45
5.3.3.6 Combination with FDM 48
5.3.4 Frequency Interleaving 50
5.4 Waveform Generation 51
5.4.1 Pilot Insertion 51
5.4.1.1 Scattered Pilot Insertion 51
5.4.1.2 Continual Pilot Insertion 52
5.4.1.3 Edge Pilot Insertion 52
5.4.1.4 Preamble Pilot Insertion 52
5.4.1.5 Subframe Boundary Pilot Insertion 52
5.4.2 MISO 52
5.4.2.1 Signal Model 52
5.4.3 Inverse Fast Fourier Transform (IFFT) 53
5.4.4 Guard Interval 53
5.4.5 Bootstrap 54
5.5 Channel Bonding 55
5.5.1 Memory Considerations 55
5.5.2 Channel Bonding Examples 56
6. GUIDELINES FOR RECEIVER IMPLEMENTATION 57
6.1 Signal Discovery and Synchronization 57
6.1.1 Use of Bootstrap for Signal Acquisition and Synchronization 57
6.1.1.1 Signal Acquisition and Timing Synchronization 57
6.1.1.2 Fractional Frequency Offset (FFO) Estimation 59
6.1.1.3 Integer Frequency Offset (IFO) Estimation and Validation 60
6.1.2 Signaling Detection of Bootstrap Symbols 60
6.1.2.1 Time Domain Cyclic Shift Detection 60
6.1.2.2 Gray De-Mapping 62

6.2 Waveform Demodulation 62
6.2.1 Channel Estimation and Equalization 62
6.2.1.1 Channel Estimation for Mobile Reception 63
6.2.1.2 Channel Estimation for Fixed Reception 64
6.2.2 Removal of Peak to Average Power Ratio Reduction Techniques 64
6.3 De-Framing and De-Interleaving 64
6.3.1 Frequency De-Interleaving 64
6.3.1.1 Frequency De-Interleaving for 8K/16K FFT Size 65
6.3.1.2 Frequency De-Interleaving for 32K FFT Size 65
6.3.2 Time De-Interleaving 65
6.3.2.1 Extended Time De-Interleaving 65
6.3.2.2 Hybrid Time De-Interleaving 66
6.4 LDM Decoding 70
6.4.1 Decoding Process of Core and Enhanced Layers 71
6.4.2 Core PLP Decoding 71
6.4.3 Core PLP Cancellation and Enhanced PLP Decoding 72
6.5 Channel Decoding 73
6.5.1 LLR De-Mapping for Non-Uniform Constellations 74
6.5.1.1 De-Mapping for 1D-NUC 74
6.5.1.2 De-Mapping for 2D-NUC 76
6.5.2 Bit De-Interleaving 78
6.5.3 Inner and Outer Decoding 79
6.5.4 Decoding of L1 Signaling 82
7. GUIDELINES FOR MOBILE SERVICES 83
7.1 Input Formatting 83
7.2 Bit Interleaved Coding and Modulation (BICM) 83
7.2.1 BICM for Data Payload 83
7.2.2 Protection for L1-Signaling 84
7.3 Framing and Interleaving 84
7.3.1 Time Interleaving 84
7.3.1.1 Time Interleaver Modes 84
7.3.1.2 Time Interleaver Size 84
7.3.2 Framing 84
7.3.2.1 Frame Length 84
7.3.2.2 PLP Multiplexing 84
7.3.3 Frequency Interleaving 85
7.4 Waveform Generation 85
7.4.1 Pilot Insertion 85
7.4.2 Inverse Fast Fourier Transform (IFFT) 85
7.4.3 Guard Interval 85
ANNEX A : SYSTEM PERFORMANCE 87
A.1 Introduction 87
A.2 Channel Models 87
A.3 Simulation, Laboratory Test and Field Test Results 87
ANNEX B : ATSC 3.0 RECEIVER C/N MODEL 92
B.1 Introduction 92

B.2 Method **92**
B.2.1 Calculation of Boosted Pilot Correction Factor **93**
B.2.2 Estimation of Channel Estimation Loss **93**
B.2.3 Estimation of Average Value of $fINT$ **94**
B.3 Example C/N calculation **95**
B.4 Example Calculated values of ΔBP **96**
B.5 Calculated values of ΔRCE **97**
B.6 Expected receiver AWGN C/N based on Calculation Model **98**
***ANNEX C* : ATSC 3.0 SERVICE EXAMPLES** **114**
C.1 Single PLP Service **114**
C.2 Multiple PLP Subframe Service **115**
C.3 Multiple PLP TDM Service **116**
C.4 Multiple PLP LDM Service **117**

Index of Figures

Figure 4.1 High level ATSC 3.0 physical layer protocol diagram enabling an example of multiple-PLP architecture. 7
Figure 4.2 Equalized SNR performance for $\boldsymbol{Dy = 2}$ (left) and $\boldsymbol{Dy = 4}$ (right) with linear frequency interpolation (top) and DFT frequency interpolation (bottom). 12
Figure 4.3 Injection level control of Enhanced PLP(s). 17
Figure 5.1 Format of FEC frame when BCH/CRC is used (top) or not (bottom). 30
Figure 5.2 PCM structure of (*a*) Type A LDPC and (*b*) Type B LDPC. 31
Figure 5.3 Bit interleaver structure: Parity, group-wise and block interleavers. 31
Figure 5.4 A two-layer LDM transmitter configuration. 33
Figure 5.5 Grouping of FFT sizes: (*a*) Not recommended, random order, (*b*) Recommended, in 8K, 16K and 32K order. 38
Figure 5.6 Example at subframe boundary from SP3_2 to SP6_4. 38
Figure 5.7 Example at subframe boundary from SP3_2 to SP4_4. 39
Figure 5.8 Example at subframe boundary from SP12_2 to SP6_4. 39
Figure 5.9 Example showing channel at a receiver before interpolation with scattered pilots $D_x = 6$ and $D_y = 4$. 40
Figure 5.10 Example at a receiver showing that after time and frequency interpolation the channel can be well estimated up to Tu/6. 40
Figure 5.11 Example at a receiver with $D_x = 12$ and $D_y = 4$ showing that the receiver cannot estimate the channel well in this case, since the variation is greater than Tu/12 which is the maximum that can be interpolated. 40
Figure 5.12 Example at subframe boundary from SP16_2 to SP6_4. 41
Figure 5.13 L1D_plp_start and L1D_plp_size definitions for Core and Enhanced PLPs. 42
Figure 5.14 TI Group assignment for multiple Core PLPs. 43
Figure 5.15 Two Enhanced PLPs injected into a single Core PLP. 43
Figure 5.16 Not recommended LDM configuration example #1. 44
Figure 5.17 Not recommended LDM configuration example #2. 45
Figure 5.18 Allowed LDM configuration example #1 using the CTI mode. 46
Figure 5.19 Allowed LDM configuration example #2 using the CTI mode. 46
Figure 5.20 Allowed LDM configuration example using the HTI mode. 47
Figure 5.21 Recommended use of TI Blocks for HTI-based LTDM or LFDM configurations. 48
Figure 5.22 FLDM configuration example. 49
Figure 5.23 LFDM configuration example. 50
Figure 5.24 Scattered pilot pattern SP12_2 ($D_x = 12$, $D_y = 2$). 51
Figure 5.25 1-PLP 2-Channel Bonding example transmitter architecture. 56
Figure 5.26 1-PLP 2-Channel Bonding example receiver architecture. 56
Figure 5.27 4-PLP channel bonding example transmitter architecture. 57
Figure 5.28 4-PLP channel bonding example receiver architecture. 57

Figure 6.1 Delayed correlation diagram. 58
Figure 6.2 Memory reconfiguration to support the extended de-interleaving. 66
Figure 6.3 Example of input TI Block and output TI Block in a TBI operation: (*a*) The 0-th TI Block, (*b*) The 1-st TI Block. 68
Figure 6.4 Example of input TI Block and output TI Block in TBDI operation: (*a*) Memory data cells for the 0-th TI Block, (*b*) Address generation for reading the 0-th input TI Block, (*c*) Memory data cells for the 1-st TI Block, (*d*) Address generation for reading the 1-st input TI Block. 69
Figure 6.5 Example of time interleaving/de-interleaving with virtual FEC Blocks: (*a*) Output data cells after TBI operation, (*b*) Memory data cells after writing into TBDI memory. 70
Figure 6.6 LDM decoding block diagram. 71
Figure 6.7 LDM FEC Block decoding process of Core and Enhanced Layers. 71
Figure 6.8 CL signal re-generation: (1) Whole codeword, (2) Information + outer code parity, and (3) Information-only cases. 73
Figure 6.9 De-mapper structure for 1D-NUC. 74
Figure 6.10 De-mapper structure for 2D-NUC. 76
Figure 6.11 De-interleaving process of Type A block de-interleaver. 78
Figure 6.12 LDPC code performance based on various decoding algorithms (Code rate = 10/15, Code length = 64800, AWGN channel). 82

Index of Tables

Table 4.1 Scattered Pilot Pattern Overheads 10
Table 4.2 Maximum Recommended Number of Preamble Symbols for Different FFT Sizes 13
Table 4.3 Required SNRs (dB) for all ModCod Combinations, Long LDPC codes (64800 bits) and AWGN Channel 18
Table 4.4 Required SNRs (dB) for all ModCod Combinations, Long LDPC Codes (64800 bits) and i.i.d. Rayleigh Channel 18
Table 4.5 Required SNRs (dB) for all ModCod Combinations, Short LDPC Codes (16200 bits) and AWGN Channel 19
Table 4.6 Required SNRs (dB) for all ModCod Combinations, Short LDPC Codes (16200 bits) and i.i.d. Rayleigh Channel 19
Table 4.7 Mandatory ModCod Combinations for Long Codes (64800 bits) 19
Table 4.8 Mandatory ModCod Combinations for Short Codes (16200 bits) 19
Table 4.9 Definition of L1 Protection Modes 20
Table 4.10 Performance of L1-Basic and L1-Detail Modes Under AWGN and Rayleigh Channels (FER=10^{-4}) 21
Table 4.11 Required SNRs of the Preamble when TxID is Injected (AWGN Channel) 27
Table 5.1 Types of Pilots in Each Type of Symbol 51
Table 5.2 Allowed Combinations of GI and FFT Sizes 54
Table 5.3 Statistical Multiplexing Gains 55
Table 5.4 Example Parameters and Bit Rate for 1-PLP Channel Bonding 56
Table 6.1 Optimized Offset and Scaling Values 80
Table 6.2 Performance Difference from the Shannon Capacity Limit to the Theoretical Thresholds and Simulation Results (Short Code Cases) 80
Table 6.3 Performance Difference from the Shannon Capacity Limit to the Theoretical Thresholds and Simulation Results (Long Code Cases) 81
Table 7.1 Recommended ModCod Combinations for ATSC 3.0 Mobile Services (N_{inner} = 16200 bits, i.e., Short Codes) 84
Table 7.2 Recommended Scattered Pilot Patterns for ATSC 3.0 Mobile Services 85
Table 7.3 Recommended Guard Intervals for ATSC 3.0 Mobile Services 86
Table A.3.1 System Parameters for Simulations, Laboratory Tests, and Field Tests 88
Table A.3.2 Required C/N for BER = 10^{-6} After LDPC and BCH Decoding Under AWGN Channel 89
Table A.3.3 Required C/N for BER = 10^{-6} After LDPC and BCH Decoding Under Rician Channel 90
Table A.3.4 Required C/N for BER = 10^{-6} After LDPC and BCH Decoding Under Rayleigh Channel 91
Table B.4.1 Calculated Values of ΔBP 96
Table B.5.1 Calculated Values of ΔRCE 97
Table B.6.1 Gaussian Channel 8K FFT, QPSK Expected C/N Values 99

Table B.6.2 Gaussian Channel 8K FFT, 16QAM Expected C/N Values 100
Table B.6.3 Gaussian Channel 8K FFT, 64QAM Expected C/N Values 101
Table B.6.4 Gaussian Channel 8K FFT, 256QAM Expected C/N Values 102
Table B.6.5 Gaussian Channel 8K FFT, 1024 and 4096QAM Expected C/N Values 103
Table B.6.6 Gaussian Channel 16K FFT, QPSK Expected C/N Values 104
Table B.6.7 Gaussian Channel 16K FFT, 16QAM Expected C/N Values 105
Table B.6.8 Gaussian Channel 16K FFT, 64QAM Expected C/N Values 106
Table B.6.9 Gaussian Channel 16K FFT, 256QAM Expected C/N Values 107
Table B.6.10 Gaussian Channel 16K FFT, 1024QAM and 4096QAM Expected C/N Values 108
Table B.6.11 Gaussian Channel 32K FFT, QPSK Expected C/N Values 109
Table B.6.12 Gaussian Channel 32K FFT, 16QAM Expected C/N Values 110
Table B.6.13 Gaussian Channel 32K FFT, 64QAM Expected C/N Values 111
Table B.6.14 Gaussian Channel 32K FFT, 256QAM Expected C/N Values 112
Table B.6.15 Gaussian Channel 32K FFT, 1024QAM and 4096QAM Expected C/N Values 113
Table C.1.1 Example of Physical Layer Parameters for Single-PLP Service 114
Table C.2.1 Example of Physical Layer Parameters for 2-Subframe Service 115
Table C.3.1 Example of Physical Layer Parameters for 2-PLP TDM Service 116
Table C.4.1 Example of Physical Layer Parameters for 2-PLP LDM Service 117

ATSC Recommended Practice: Guidelines for the Physical Layer Protocol

1. SCOPE

This document provides recommended practices for the ATSC 3.0 physical layer protocol standards specified by A/321 [2] and A/322 [3]. The intent of this document is to make recommendations for physical layer operating modes so that readers can make informed decisions about physical layer configurations. Also, this document provides some implementation guidelines to aid with flexible configurations of physical layer design resources in transmitter and receiver manufacturers' equipment.

1.1 Introduction and Background

The ATSC 3.0 physical layer protocol is designed to provide a toolbox of technology that allows flexible operating modes for a variety of harsh channel conditions (e.g., indoor or mobile) while maintaining efficient use of spectrum resources. This document provides recommended parameter and technology choices in A/321 [2] and A/322 [3] so that broadcasters can optimally deliver intended service(s). It also contains detailed guidelines for transmitter and receiver design implementations based on engineering studies of the latest technologies in the ATSC 3.0 physical layer. Guidelines for broadcasters' mobile service(s) are provided with operating modes and parameter choices of A/322 [3] in aspects of robustness and power consumption. The ATSC 3.0 system performance and recommended service examples cover aspects of real field experiences and are intended to provide practical guidance for all readers.

1.2 Organization

This document is organized as follows:

- Section 1 – The scope of this document and general introduction
- Section 2 – References and applicable documents
- Section 3 – Definition of terms, acronyms, and abbreviations used
- Section 4 – System overview and guidelines for physical layer parameter choices
- Section 5 – Guidelines in detail for transmitter implementation
- Section 6 – Guidelines in detail for receiver implementation
- Section 7 – Guidelines for mobile services
- Annex A – ATSC 3.0 system performance: Simulation, laboratory and field test results
- Annex B – ATSC 3.0 receiver C/N models
- Annex C – ATSC 3.0 service examples

2. REFERENCES

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

The following documents, in whole or in part, as referenced in this document, contain specific provisions that should be followed in order to facilitate implementation and application of this Recommended Practice.

[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 System Committee, Washington, D.C., 23 March 2016.

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

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

[5] ATSC: "ATSC Standard: Scheduler / Studio to Transmitter Link," Doc. A/324:2018, Advanced Television System Committee, Washington, D.C., 2 January 2018.

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

[7] NorDig Unified Requirements for Integrated Receiver Decoders for use in cable, satellite, terrestrial and IP-based networks, January 2017.

[8] EBU Tech 3348: "Frequency and Network Planning Aspects of DVB-T2," 2014.

[9] DTG D-Book 9: "Digital Terrestrial Television Requirements for Interoperability," 2016.

[10] ETSI TS 102 831: "Digital Video Broadcasting (DVB); Implementation guidelines for a second generation digital terrestrial television broadcasting system (DVB-T2)," V1.2.1, August 2012.

[11] ETSI EN 300 744: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television," V1.6.2, October 2015.

[12] ISO/IEC: "Information Technology – Telecommunications and information exchange between systems – Part 3: Standard for Ethernet," Doc. ISO/IEC 8802-3:2017, International Organization for Standardization/International Electrotechnical Commission, Geneva, Switzerland.

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] is used. Where an abbreviation is not covered by IEEE practice or industry practice differs from IEEE practice, the abbreviation in question will be described in Section 3.2 of this document.

3.1 Compliance Notation

This section defines compliance terms for use by this document:

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

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

3.2 Acronyms and Abbreviations

The following acronyms and abbreviations are used within this document.

ACE Active Constellation Extension

ALP ATSC 3.0 Link layer Protocol

AP Additional Parity

ATSC Advanced Television Systems Committee

AWGN Additive White Gaussian Noise

BBP	BaseBand Packet
BCH	Bose, Chaudhuri, Hocquenghem
BER	Bit Error Rate
BICM	Bit-Interleaved and Coded Modulation
BIL	Bit InterLeaver
BSR	Baseband Sampling Rate
CDL	Convolutional Delay Line
CFO	Carrier Frequency Offset
CFR	Channel Frequency Response
CL	Core Layer
CLI	Cross-Layer Interference
Cod	Code rate
CP	Continual Pilot
CPM	Circulant Permutation Matrix
CRC	Cyclic Redundancy Check
CTDI	Convolutional Time De-Interleaver
CTI	Convolutional Time Interleaver
dB	decibel
DFT	Discrete Fourier Transform
DRAM	Dynamic Random Access Memory
EL	Enhanced Layer
FDI	Frequency De-Interleaver
FDM	Frequency Division Multiplexing
FEC	Forward Error Correction
FER	Frame Error Rate
FFO	Fractional Frequency Offset
FFT	Fast Fourier Transform
FI	Frequency Interleaver
FIFO	First-In-First-Out
GI	Guard Interval
GUR	Guard Utilization Ratio
HTDI	Hybrid Time De-Interleaver
HTI	Hybrid Time Interleaver
IFFT	Inverse Fast Fourier Transform
IFO	Integer Frequency Offset
i.i.d.	Independent and Identically Distributed
ISO	International Organization for Standardization
IU	Interleaving Unit
L1	Layer 1
LDM	Layered Division Multiplexing
LDPC	Low-Density Parity Check
LFDM	Layered Frequency Division Multiplexing

LLR	Log-Likelihood Ratio
LLS	Low Level Signaling
LMT	Link Mapping Table
LS	Least Square
LSB	Least-Significant Bit
LSI	Large Scale Integration
Mbps	megabits per second
MHz	megahertz
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
Mod	Modulation
ModCod	Modulation and Code Rate
MS	Min-Sum Algorithm
MSB	Most-Significant Bit
msec	milliseconds
N/A	Not Allowed
NMS	Normalized Min-Sum Algorithm
NoA	Number of Active (cells)
NoC	Number of (useful) Carriers
NUC	Non-Uniform Constellation
OFDM	Orthogonal Frequency Division Multiplexing
OMSA	Offset Min-Sum Algorithm
OTA	Over the Air
PAM	Pulse Amplitude Modulation
PAPR	Peak-to-Average Power Ratio
PCM	Parity Check Matrix
pdf	Probability Distribution Function
PH	Phase Hopping
PLP	Physical Layer Pipe
PRBS	Pseudo Random Bit Sequence
PSR	Number of Punctured Bits to Number of Shortened Bits Ratio
PTP	Precision Time Protocol
QAM	Quadrature Amplitude Modulation
QEF	Quasi Error Free
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RFU	Reserved for Future Use
SBS	Subframe Boundary Symbol
SFN	Single Frequency Network
SHVC	Scalable High Efficiency Video Coding
SINR	Signal-to-Interference plus Noise Ratio
SISO	Single Input Single Output

SLS Service Layer Signaling
SLT Service List Table
SNR Signal-to-Noise Ratio
SP Scattered Pilot
SPA Sum-Product Algorithm
SRAM Static Random Access Memory
SSM Subslice Multiplexing
STL Studio-Transmitter Link
STO Symbol Timing Offset
TBDI Twisted Block De-Interleaver
TBI Twisted Block Interleaver
TDCFS Transmit Diversity Code Filter Sets
TDM Time Division Multiplexing
TI Time Interleaver
TR Tone Reservation
TxID Transmitter Identification

3.3 Terms

The following terms are used within this document.

Base Field – The first portion of a Baseband Packet Header.

Base Layer – The smallest video sub-stream (first layer) of SHVC.

Baseband Packet – A set of $K_{payload}$ bits which form the input to a FEC encoding process. There is one Baseband Packet per FEC Frame.

Baseband Packet Header – The header portion of a Baseband Packet.

Block Interleaver – An interleaver where the input data is written along the rows of a memory configured as a matrix, and read out along the columns.

Cell – One pair of encoded I/Q components in a constellation.

Cell Interleaver – An interleaver operating at the cell level.

Combined PLP – A PLP comprised of the Layered Division Multiplexing of a Core PLP and one or more Enhanced PLP(s) after processing by the LDM injection block.

Concatenated code – A code having an outer code followed by an inner code.

Constellation – A pair of encoded (I component/Q component) points in the I/Q plane.

Core Layer – The first layer of a 2-layer LDM system. The only layer in a non-LDM system.

Core PLP – A PLP belonging to the Core Layer.

Data Payload Symbols – Data and Subframe Boundary Symbols (i.e., non-Preamble symbols).

Enhanced Layer – The second layer of a 2-layer LDM system.

Enhanced PLP – A PLP belonging to the Enhanced Layer.

Enhancement Layer – The larger video sub-stream (second layer) of SHVC.

Extension Field – The third portion of a Baseband Packet Header.

FEC Block – A FEC Frame after constellation mapping to cells.

FEC Frame – A single Baseband Packet with its associated FEC parity bits attached, having a total size of 64800 or 16200 bits (per FEC Frame).

Frequency Interleaver – An interleaver which takes cells and interleaves them in frequency within a particular OFDM symbol.

Inner Code – One code of a concatenated code system.

Interleaver – A device used to counteract the effect of burst errors.

Layered Division Multiplexing – A multiplexing scheme where multiple PLPs are combined in layers with a specific power ratio.

ModCod – A combination of modulation and code rate that together determine the robustness of the PLP and the size of the Baseband Packet.

Non-Uniform Constellation – A constellation with a non-uniform spread of constellation points.

Optional Field – The second portion of a Baseband Packet Header.

Outer Code – One code of a concatenated code system.

Physical Layer Pipe – A data-carrying structure specified to an allocated capacity and robustness that can be adjusted to broadcaster needs.

reserved – Set aside for future use by a Standard.

Systematic – A property of a code in which the codeword is composed of the original data in its sequential order followed by the parity data for the codeword.

TI Block – An integer number of FEC Blocks.

Time Interleaver – An interleaver which takes cells and interleaves them over a particular time period.

4. SYSTEM OVERVIEW AND GUIDELINES FOR PHYSICAL LAYER MODE

4.1 System Overview

The ATSC 3.0 physical layer protocol described by A/321 [2] and A/322 [3] provides distinct features and descriptions of the physical layer technologies. For a channel coding method, a well-optimized Bit-Interleaved Coded Modulation (BICM) chain, comprised of Low Density Parity Check (LDPC) codes (2/15 – 13/15 code rates), Bit Interleavers, and Non-Uniform Constellations (QPSK – 4096QAM modulation orders), is used to achieve near Shannon limit performance of channel error mitigation across a wide range of SNR values. Multiple data streams can be carried by one or multiple Physical Layer Pipes (PLPs) using flexible choices of framing and multiplexing techniques including Time Division Multiplexing (TDM), SubSlice Multiplexing (SSM), Frequency Division Multiplexing (FDM), and Layered Division Multiplexing (LDM). Waveform parameters based on Orthogonal Frequency Division Multiplexing (OFDM) modulation also support a wide range of guard intervals, pilot patterns, and FFT sizes for broadcasters to select optimal system performance of intended services. Optional technologies such as channel bonding, Multiple-Input Single-Output (MISO), Multiple-Input Multiple-Output (MIMO), and Transmitter Identification (TxID) are included as additional features in the ATSC 3.0 physical layer protocol.

Figure 4.1 shows a high-level ATSC 3.0 physical layer protocol diagram of an example multiple-PLP architecture. Multiple data streams intended to have different robustness levels are delivered by input PLPs. After each PLP is processed by independent Input Formatting and BICM blocks, further block processes are selected depending on the choices of multiplexing methods. TDM, SSM, and FDM can be achieved in the Framing and Interleaving block (bypassing the LDM Combining block) by using PLP multiplexing within a subframe. When different waveform parameters such as guard intervals, pilot patterns and FFT sizes are configured in the Waveform Generation block, TDM can also be enabled by constructing multiple subframes with each PLP

contained in a separate subframe. LDM can be achieved with data processing right after the BICM stage as it contains combinations of data cells. A 2-layer LDM may be combined with TDM and/or FDM to configure multiple PLPs as shown in Figure 4.1.

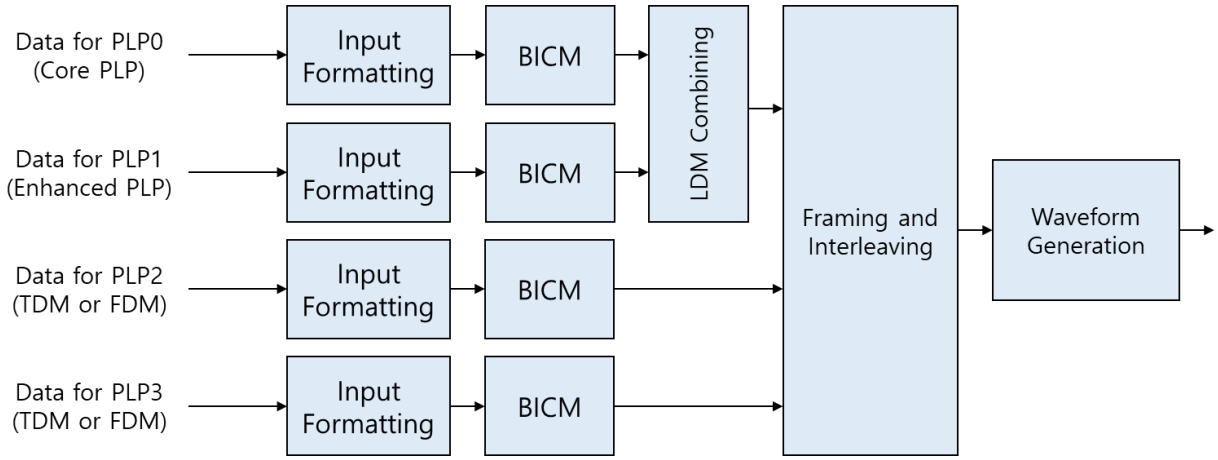

Figure 4.1 High level ATSC 3.0 physical layer protocol diagram enabling an example of multiple-PLP architecture.

The key technologies of the ATSC 3.0 physical layer protocol can be summarized as following.

- Bit-Interleaved and Coded Modulation (BICM)
  - FEC: LDPC (inner code), BCH / CRC (outer code)
  - Code rate: 2/15 – 13/15 (64800, 16200 bits)
  - Constellations: QPSK, 16QAM – 4096QAM (Non-uniform constellations)
- Time Interleaver
  - Convolutional Time Interleaver (CTI)
  - Hybrid Time Interleaver (HTI): Cell Interleaver, Twisted Block Interleaver, Convolutional Delay Line
  - Maximum TI memory size: 2^{19} cells
- Multiplexing
  - TDM, SSM, FDM, LDM
  - Maximum number of PLPs per RF channel: 64
  - Maximum number of PLPs per complete delivered product: 4
- Framing and Waveform
  - Time-aligned and symbol-aligned frames
  - Maximum frame length: 5 seconds
  - One or more subframes per frame
  - FFT size: 8K, 16K, and 32K
  - Pilot patterns: Scattered pilots (16 choices), Continual pilots, Edge pilots, and Subframe boundary pilots
  - Guard intervals: 12 choices
- Optional Technologies
  - Channel bonding

- MISO: TDCFS
- MIMO
- PAPR reduction: TR and ACE
- Transmitter Identification (TxID)

4.2 Guidelines for Physical Layer Mode

4.2.1 FFT Size

ATSC 3.0 defines three FFT sizes: 8K, 16K and 32K. The choice of FFT size defines the number of carriers included in a fixed bandwidth. FFT size influences the capacity, delay and mobility tolerance of the transmitted signal. For example, if a large FFT size (32K FFT) is used, the guard interval overhead is reduced compared to smaller FFT sizes for a fixed guard interval duration in microseconds. Moreover, a greater delay tolerance for the same fractional guard interval is obtained, allowing larger Single Frequency Networks (SFNs) to be constructed. Conversely, the use of a larger FFT size implies greater vulnerability to fast time-varying channels due to the narrower carrier spacing.

Therefore, for delivering high-bit-rate services where fixed reception is mainly targeted, the 32K FFT mode is recommended as the time variations of the channel in fixed reception scenarios are small. For mobile reception, the smaller FFT sizes such as 8K and 16K FFT modes are recommended. Note that in ATSC 3.0, multiple subframes per frame may be used with a different FFT size for each subframe. See Section 5.3.2 for recommendations when multiple FFT sizes are present in the same ATSC 3.0 frame.

4.2.2 Bandwidth and Bandwidth Reduction

The bandwidth of the system is indicated by the value of **system_bandwidth** signaled in bootstrap symbol 1 as described in Section 6.1.1.1 of [2]. Options available for the current version of the standard are 6MHz, 7MHz and 8MHz. A greater than 8 MHz option is also available to facilitate future operation using a system bandwidth greater than 8 MHz, but it is not intended to be used by the current version described by the present signaling set.

The number of useful carriers, denoted by NoC in [3], determines the occupied bandwidth of an ATSC 3.0 signal which is typically smaller than the system bandwidth. As described in Section 7.2.3 of [3], the NoC for each FFT size is configured by a carrier reduction coefficient C_{red_coeff} that indicates the number of control units to be deducted from the maximum configurable NoC for that FFT size, NoC_{max}. The carrier reduction coefficient ranges from 0 to 4 and the corresponding ranges of occupied bandwidths are from 5.832844 MHz to 5.508844 MHz for **bsr_coefficient** = 2 which is a typical value for 6 MHz system bandwidth.

Note that the control unit size of 96 carriers for an 8K FFT is the least common multiple of 24 and 32, which are the maximum configurable D_x values of base 3 and 4, respectively. Therefore, the selected control unit size of 96 carriers simplifies receiver design by eliminating the need for extrapolation during channel estimation for all possible combinations of NoC and scattered pilot patterns. For 16K and 32K FFT sizes, the control unit sizes are scaled by 2 and 4, respectively, to keep the control unit granularity in absolute frequency and signaling overhead the same.

The first Preamble symbol always uses the minimum NoC possible for the Preamble FFT size of the current frame to eliminate the need to signal the NoC of the first Preamble symbol in the bootstrap. When the Preamble comprises more than one Preamble symbol, i.e., **L1B_preamble_num_symbols**>0, all Preamble symbols except the first Preamble symbol have the same NoC as indicated by **L1B_preamble_reduced_carriers** (Section 9.2.2 of [3]) in L1-Basic. For

data symbols, all of the OFDM symbols within a subframe have the same NoC which is configured per subframe and is indicated by **L1D_reduced_carriers** (Section 9.3.3 of [3]) in L1-Detail. One exception is the NoC of all of the OFDM symbols in the first subframe which is indicated by **L1B_first_sub_reduced_carriers** (Section 9.2.2 of [3]) in L1-Basic. Carrying the NoC information of the first subframe in L1-Basic helps receivers to begin processing the first subframe without further waiting for L1-Detail decoding which would be the case if the NoC information of the first subframe were carried in L1-Detail.

Symmetric reduction of the occupied bandwidth or NoC is assumed in [3] in favor of smaller signaling overhead compared to asymmetric reduction. If asymmetric control of the occupied bandwidth or NoC is needed for some reason during the frequency-planning process, asymmetric control of the occupied bandwidth or NoC can still be implemented by combining symmetric control of NoC and an intentional frequency offset from transmitters. Then, with the aid of the bootstrap having a wide acquisition range of frequency offset, ATSC 3.0 receivers can acquire the frequency-shifted ATSC 3.0 signal without any additional information. Note that the ATSC 3.0 Scheduler [5] allows a carrier frequency offset in order to avoid co-channel interference among neighboring transmitters. The allowed carrier frequency offsets are -1 or +1 OFDM carriers in the 8K FFT size (see Section 9.3 of [5]).

Narrower bandwidth than the occupied bandwidth currently supported by the current version can be supported, if desired, in later versions by defining new values for the reserved fields of **L1B_preamble_reduced_carriers**, **L1B_first_sub_reduced_carriers** and **L1D_reduced_carriers**.

In general, a smaller value of the carrier reduction coefficient (larger number of carriers) is recommended for increased data capacity, while a larger carrier reduction coefficient (smaller number of carriers) is recommended for the case where there is severe adjacent channel interference.

4.2.3 Pilot Pattern

Scattered pilots (SPs) are carriers that do not contain payload data but whose value is known by receivers in order to get a proper channel estimation at the SP positions. Next, the channel estimates at data cells can be obtained by interpolation. The SP pattern defines the density and location of the SPs inside the ATSC 3.0 subframes.

As the Channel Frequency Response (CFR) varies with both time and frequency, the SP pattern is characterized by two terms, the frequency separation of pilots, D_x, and the length of the SP pattern in OFDM symbols, D_y. The 16 different SP patterns of ATSC 3.0 are obtained from 8 D_x values (3, 4, 6, 8, 12, 16, 24, and 32) and from 2 D_y values (2, 4). It is strongly recommended that the values of D_x and D_y be selected according to the CFR characteristics.

4.2.3.1 Separation of Pilot Carriers (D_x)

The last path that can contribute constructively so that it can be correctly equalized by a receiver depends on the channel delay spread, i.e., the coherence bandwidth. According to the Nyquist sampling theorem, this limit (when both time and frequency interpolation are implemented) is estimated as:

$$T_n = \frac{T_U}{D_x}$$

where T_n represents the Nyquist limit and T_U is the useful symbol duration. For ATSC 3.0 it has been assumed that receivers are only able to correctly equalize those signals with echoes up to

75% or 89% of Nyquist limit. That is, only those GIs with a length shorter than 75% or 89% of T_n are allowed. This ratio is also known as the Guard Utilization Ratio (GUR). It can be seen that the Nyquist limit is increased with the useful symbol duration, i.e., with an increased FFT size, while it is reduced with the SP spacing. Therefore, not all GIs are allowed for each combination of SP pattern and FFT size.

4.2.3.2 Length of Pattern in Symbols (D_y)

If the transmitted signal is expected to be received in mobile conditions, the propagation channel will vary across OFDM symbols. Thus, the pilots need to be inserted at a certain rate (D_y) that is a function of the coherence time, which is related to the Doppler shift limit. As OFDM symbols occur at the rate $f_D = 1/(T_U + T_G)\ Hz$, the Doppler shift limit for frequency channel variation, f_D, that can be measured is:

$$f_D = \frac{\pm 1}{2D_y \cdot (T_U + T_G)}\ Hz$$

where T_G is the GI length in time.

At first, it can be assumed that the densest SP pattern provides the most accurate channel estimation. Nevertheless, at the same time, it introduces the highest data rate overhead. From the expression above, it can be observed that the smaller the D_y, the GI, and the FFT size, the higher the Doppler shift limit. Hence, in order to support high speeds, $D_y = 4$ was discarded for 32K FFT size.

In summary, taking the Nyquist (D_x) and Doppler (D_y) limits into account, the permitted FFT/GI-SP combinations in ATSC 3.0 are presented in Table 8.3 of [3]. Note however that as the pilot density increases, the data overhead also increases. The SP overhead (which does not take into account Preamble, continual, edge and subframe boundary pilots) can be expressed as the simple fraction $1/(D_x \times D_y)$ as shown in Table 4.1.

Table 4.1 Scattered Pilot Pattern Overheads

SP	D_x	D_y	Overhead (%)	SP	D_x	D_y	Overhead (%)
SP3_2	3	2	16.67	SP12_2	12	2	4.17
SP3_4	3	4	8.33	SP12_4	12	4	2.08
SP4_2	4	2	12.50	SP16_2	16	2	3.13
SP4_4	4	4	6.25	SP16_4	16	4	1.57
SP6_2	6	2	8.33	SP24_2	24	2	2.08
SP6_4	6	4	4.17	SP24_4	24	4	1.04
SP8_2	8	2	6.25	SP32_2	32	2	1.57
SP8_4	8	4	3.13	SP32_4	32	4	0.78

4.2.4 Pilot Boosting

The pilot boost factor defines the pilot amplitude with respect to data carriers. The pilot boosting affects the reception performance as higher pilot boosting improves channel estimation accuracy. However, since the total signal power is fixed, a higher pilot boosting results in a reduced data carrier power. Thus, the choice of the pilot boosting is not straightforward.

For the choice of pilot boost, the equalized data Signal-to-Noise Ratio ($SNR_{EQ,b}$) can be considered as a metric which obtains the best overall performance taking into account different receiver equipment. It is estimated as:

$$SNR_{EQ,b} = \frac{\sigma_s^2 \cdot k}{\sigma_N^2 + \sigma_N^2 \cdot f_{int}/b} = SNR \cdot \frac{k}{1 + f_{int}/b}$$

where σ_s^2 is the data signal variance, σ_N^2 is the noise variance, b is the SP boosting factor, k is the power normalization ($k = D_x \cdot D_y/(D_x \cdot D_y - 1) + b$), and $f_{int} = f_{int,time} \cdot f_{int,freq}$ is the noise reduction factor by time and frequency interpolation. The value of $f_{int,time}$ is set to $\{0.75, 0.6875\}$ for $D_y = \{2, 4\}$, which can be calculated from the distance between the two symbols to be linearly interpolated. Although the values of $f_{int,freq}$ vary depending on the frequency interpolator of each receiver implementation, ATSC 3.0 provides 5 boosting values by assuming $f_{int,freq} = \{0, 0.25, 0.5, 0.75, 1\}$. The available boosting values are shown in Table 9.14 of [3].

As noted, the choice of boosting value depends primarily on the frequency interpolator. For example, Figure 4.2 presents the $SNR_{EQ,b}/SNR$ for each SP pattern when a linear (top) or a DFT (bottom) frequency interpolator is employed in a receiver. The higher $SNR_{EQ,b}/SNR$ corresponds to a lower degradation due to the channel estimation. The corresponding boosting values and the optimum boosting value are marked.

As shown in Figure 4.2, since DFT $f_{int,freq} = 1$, the optimum $SNR_{EQ,b}/SNR$ matches with a boosting value of 4. In the case of linear interpolation, $f_{int,freq}$ varies from $\{0.667, 0.704\}$, so that the optimum $SNR_{EQ,b}/SNR$ is closer to a boosting value of 3.

Note that a high pilot boosting improves channel estimation accuracy, but it reduces the power of the data carriers, which will degrade the overall SNR of the system. This data cell power reduction is approximated as an SNR reduction (Δ_{BP}), and it can be estimated as:

$$\Delta_{BP}(dB) = 10 \cdot \log_{10} \frac{N_{data} + (N_{SP} + N_{EP}) \cdot A_{SP}^2 + N_{CP} \cdot A_{CP}^2}{N_{data} + N_{SP} + N_{EP} + N_{CP}}$$

where N_{data} refers to the number of data cells per OFDM symbol, N_{SP} refers to the number of scattered pilots per OFDM symbol, N_{EP} refers to the number of edge pilots per OFDM symbol, A_{SP}^2 refers to scattered pilot amplitude, N_{CP} refers to the number of continual pilots per OFDM symbol, and A_{CP}^2 refers to continual pilot amplitude. The corresponding data cell power reduction values for each combination of pilot boosting and SP pattern in ATSC 3.0 are shown in Table B.4.1 in Annex B.

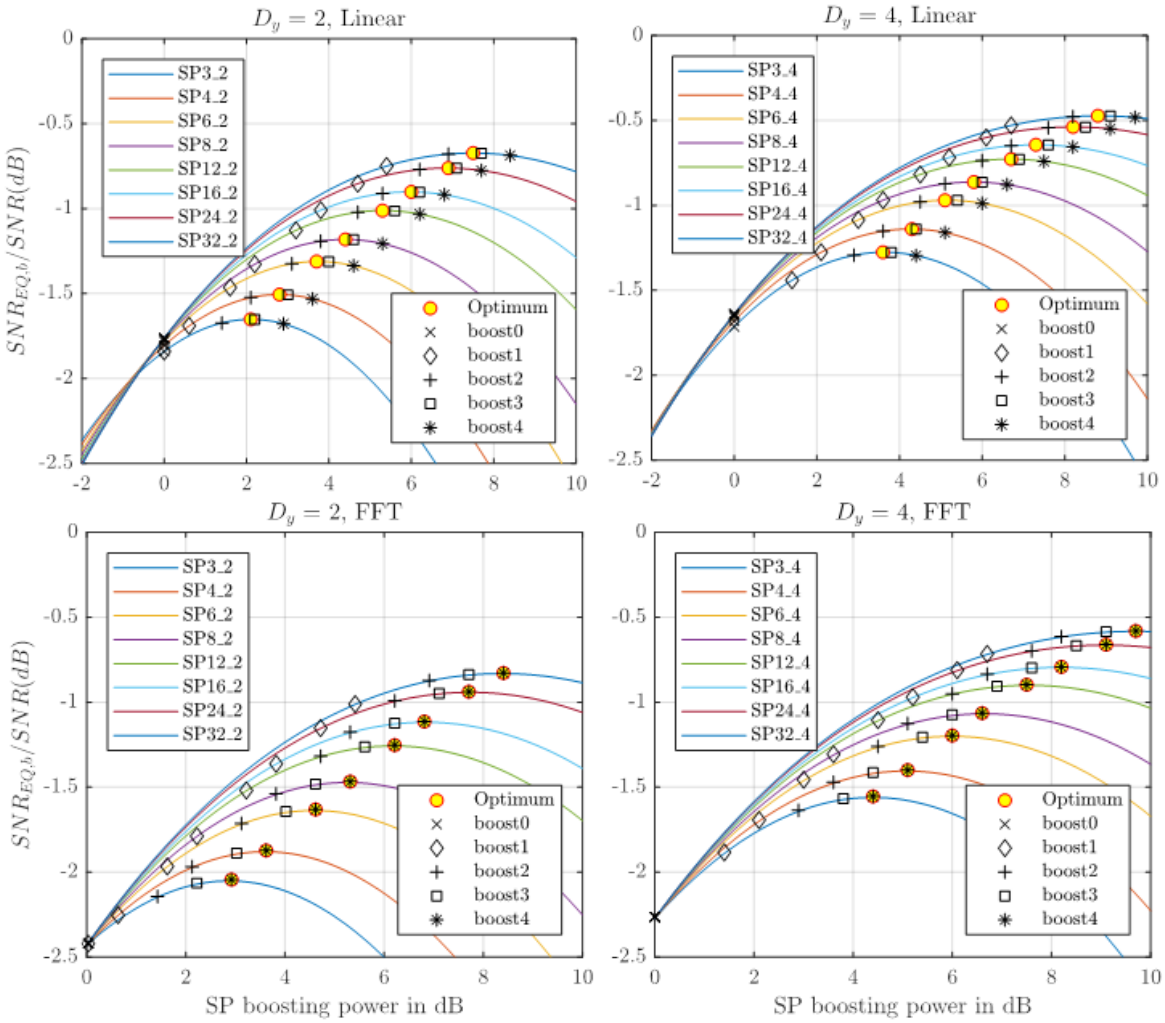

Figure 4.2 Equalized SNR performance for $\boldsymbol{D_y = 2}$ (left) and $\boldsymbol{D_y = 4}$ (right) with linear frequency interpolation (top) and DFT frequency interpolation (bottom).

4.2.5 Frame and Subframe Length

4.2.5.1 Frame Length

A frame comprises the bootstrap, Preamble and one or more subframe(s). The frame length is 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. The frame length is a configurable parameter that can be chosen as required by broadcasters and network operators. The range of the frame length that can be configured within the normative constraints imposed by [3] are from 50 ms to 5 seconds as specified in Section 7.2.2.2 of [3].

A longer frame length typically has the benefit of smaller percentage overhead because the bootstrap and Preamble always occur once per frame irrespective of the configured frame length. Conversely, a shorter frame length is preferable if features such as faster service acquisition and reacquisition are more critical. Note that receiver time-out depends not only on the frame length but also on the worst-case effect of the channel (multipath, Doppler) as well as receiver side

impairments (noise, frequency offset, crystal drift). When a long frame length is set, receivers may time-out before acquiring the transmitted signal in poor reception conditions, and longer frame lengths may also lead to a requirement for more memory. In general, when fast service acquisition and reacquisition are required, it is recommended that ATSC 3.0 frame length be set at less than or equal to 250 ms.

The exact value of the frame length depends on the choice of frame type i.e., choice of symbol-aligned or time-aligned mode. **L1B_frame_length_mode** in L1-Basic indicates the frame type of the current frame. **L1B_frame_length_mode**=0 indicates time-aligned mode and **L1B_frame_length_mode**=1 indicates symbol-aligned mode. For time-aligned mode, **L1B_frame_length** in L1-Basic indicates the exact value of the frame length in 5 ms units. Given the configurable range of frame length from 50 ms to 5 seconds, **L1B_frame_length** ranges from 10 to 1000.

For the symbol-aligned mode (**L1B_frame_length_mode**=1), the overall frame length is the summation of each of the time durations of the bootstrap, Preamble and subframe(s). The bootstrap consists of a number of bootstrap symbols. Each bootstrap symbol has a fixed time duration of 500 µs (Section 5.1 of [2]). The total time duration in microseconds of the bootstrap is 500 µs multiplied by the total number of bootstrap symbols.

4.2.5.2 Number of Preamble Symbols

The Preamble consists of one or more Preamble symbol(s). The time duration of the Preamble is the time duration of one Preamble symbol multiplied by the number of Preamble symbols constituting the Preamble. The **preamble_structure** in the bootstrap indicates the FFT size and guard interval of the Preamble as described in Section 9.1.4 and Annex H of [3] and **L1B_preamble_num_symbols** in L1-Basic indicates one less than the number of Preamble symbols in the Preamble. In theory, the number of Preamble symbols in a Preamble ranges from 1 to 8 according to the value assigned to **L1B_preamble_num_symbols** in L1-Basic. Note, however, that the memory size required to implement the block interleaver/de-interleaver for L1-Detail signaling data is proportional not only to the number of Preamble symbols but also is proportional to the FFT size of the Preamble symbols (Section 7.2.5.2 of [3]). An arbitrary combination of the number of Preamble symbols and the FFT size, e.g., eight Preamble symbols with 32K FFT size, renders the ATSC 3.0 device unnecessarily costly, due to the huge memory size required to implement the block interleaver/de-interleaver for L1-signaling data. Therefore, it is recommended to set upper limits on the maximum supported number of Preamble symbols per each Preamble FFT size as described in Table 4.2.

Table 4.2 Maximum Recommended Number of Preamble Symbols for Different FFT Sizes

Preamble FFT Size	Maximum Recommended Number of Preamble Symbols
8K	8
16K	4
32K	2

4.2.5.3 Subframe Configuration

The time duration of each subframe can be configured by varying the number of OFDM symbols in the subframe. Related to the subframe length, two normative constraints are imposed by [3]. Firstly, for a subframe configured with a 32K FFT, the sum of the number of data and subframe

boundary symbols present in the subframe is always even, except for the first subframe of a frame where the sum of the number of Preamble, subframe boundary and data symbols is even (Section 7.2.4.2 of [3]). This is to limit the amount of memory used by the frequency interleaver and frequency de-interleaver. Secondly, the minimum duration of a subframe is the greater of 20ms or the duration in ms of $4\times D_y$ data and subframe boundary symbols of that subframe. At least $4\times D_y$ data and subframe boundary symbols are present in every subframe. (Section 7.2.2.3 of [3]). This is because channel estimation is difficult for very short subframes, and very short subframes are also less efficient.

L1B_num_subframes in L1-Basic indicates the number of subframes present within the current frame. **L1B_first_sub_fft_size** and **L1B_first_sub_guard_interval** in L1-Basic indicate the FFT size and guard interval associated with the first subframe, respectively. **L1B_first_sub_num_ofdm_symbols** in L1-Basic signals the number of OFDM symbols present within the first subframe. Similarly, **L1D_fft_size** and **L1D_guard_interval** in L1-Detail indicate, for each subframe following the first subframe, the FFT size and guard interval of that subframe, respectively. **L1D_num_ofdm_symbols** in L1-Detail indicates, for each subframe following the first subframe, the number of OFDM symbols present within the associated subframe.

In the case that additional samples are added to the end of a symbol-aligned frame to facilitate sampling clock adjustment in possible future versions of ATSC 3.0 (Section 9.2.1 of [3]), the time duration of additional samples is added to the frame length calculation. **L1B_additional_samples** in L1-Basic indicates the number of additional samples added at the end of a symbol-aligned frame to facilitate sampling clock alignment.

4.2.6 Symbol-Aligned or Time-Aligned Mode

ATSC 3.0 physical layer frames are configured as either symbol-aligned or time-aligned frames, as stated in Section 7.2.2.2 of [3]. The bootstrap and Preamble symbols of both types of frames are identical in structure. The primary difference between the two frame types lies in the guard intervals of the OFDM symbols belonging to the subframe(s) within a frame.

In a symbol-aligned frame, the guard interval lengths for the OFDM symbols belonging to a particular subframe are exactly equal to the length signaled by either **L1B_first_sub_guard_interval** (Section 9.2.3 of [3]) or **L1D_guard_interval** (Section 9.3.3 of [3]) for the corresponding subframe.

A time-aligned frame has a total length, in samples, exactly equal to an integer multiple of 5 ms with that frame length being indicated by **L1B_frame_length** (Section 9.2.1 of [3]). The frame length is achieved by padding a configured frame out to the desired length, and this padding is performed by adding an appropriate number of extra samples to the guard interval of each and every OFDM symbol belonging to each and every subframe within the frame. The same number of excess samples is added to the guard interval of each of these OFDM symbols regardless of the guard interval length signaled by **L1B_first_sub_guard_interval** or **L1D_guard_interval**. The number of extra samples to be added to each guard interval is calculated as specified in Section 8.5.1 of [3] and is signaled by **L1B_excess_samples_per_symbol** (Section 9.2.1 of [3]). Note that excess samples are not added to any of the Preamble symbols, since the Preamble contains the **L1B_excess_samples_per_symbol** signaling field and the Preamble should therefore be decodable without knowledge of the contents of this field. There will also likely be a non-zero number of padding samples that cannot be equally distributed to the OFDM symbol guard intervals, and these padding samples are used to construct a cyclic postfix on the final OFDM symbol of the corresponding frame. Note that the length, in samples, of any such cyclic postfix should necessarily be less than the number of non-Preamble OFDM symbols within the frame (since otherwise it

would be possible to equally distribute at least one more excess sample to each of the relevant guard intervals within the frame).

The frame type of the current frame is signaled in L1-Basic by **L1B_frame_length_mode** (Section 9.2.1 of [3]).

Time-aligned frames are useful when it is desired to have physical layer frame boundaries aligned on specific time points (e.g., exactly on a 1 ms GPS time tick). Potential use cases for this configuration include the time-division multiplexing of different wireless services within a single RF channel. There is a very small loss of efficiency by using time-aligned frames, due to the extra padding samples that are used to achieve the time alignment, but this efficiency loss should be minimal with a carefully chosen frame configuration.

If a broadcaster has no requirement for the time alignment of physical layer frames, then it is recommended that symbol-aligned frames be configured since this will maximize the efficient use of the configured physical layer resources.

At the present time, there is no known use case for multiplexing symbol-aligned and time-aligned frames within the same RF channel, so it is recommended that **L1B_frame_length_mode** maintain the same value from one physical layer frame to the next. In the infrequent event of switching a flow of frames from symbol-aligned to time-aligned frames within a given RF channel, it is recommended that the broadcaster temporarily suspend the transmitted signal of symbol-aligned frames and recommence transmitting time-aligned frames at an appropriate time boundary. Switching from time-aligned to symbol-aligned frames (also expected to be a rare event) can be achieved via an immediate cut-over with no requirement to suspend transmission since there will be no need to align the frame edges of symbol-aligned frames with any kind of time boundary.

4.2.7 PLP Multiplexing

4.2.7.1 PLP Cell Multiplexing

In the context of PLP cell multiplexing and as defined in Section 7.2.6.6 of [3], there are two basic types of PLPs in the ATSC 3.0 physical layer: non-dispersed PLPs and dispersed PLPs. All of the data cells in a non-dispersed PLP within a subframe are contiguous in the one-dimensional indexed data cell space that is defined in Section 7.2.6.1 of [3]. Conversely, a dispersed PLP is subsliced (see Section 7.2.6.8 of [3]) such that equal-sized blocks (or subslices) of that PLP appear at non-contiguous periodic intervals in the indexed data cell space of a subframe.

The two different PLP types lead to two basic approaches when multiplexing the cells of more than one PLP into a subframe.

Non-dispersed PLPs can be multiplexed in a time-division multiplexed manner. Here, for example, a first non-dispersed PLP would occupy a certain contiguous block in the indexed data cell space, followed by a second non-dispersed PLP occupying a different contiguous block within the indexed data cell space. TDM is a relatively simple multiplexing scheme that can be used for different applications and has the benefit of potentially reducing power consumption at a receiver, especially in battery-powered receivers. A receiver could access and decode an ATSC 3.0 signal only during the portion where the TDMed PLP of interest is transmitted. During unwanted portions of the transmission (e.g. other TDMed PLPs), the receiver could fall into an idle mode to reduce battery energy consumption. An example of TDMed PLPs is shown in Section 7.2.7.3 of [3].

Dispersed PLPs can be cell multiplexed in a subslice multiplexed (SSM) manner. Here, the subslices of multiple (at least two) dispersed PLPs are essentially interwoven with each other across a greater period of time than would be otherwise occupied by only one of those PLPs if that PLP was configured as a non-dispersed PLP. This leads to greater time diversity by spreading the

same amount of data over a longer period of time. The basic requirement for subslice multiplexing multiple dispersed PLPs together is that all of those PLPs are configured with the same subslice interval value. Note that the frequency-division multiplexed example shown in Section 7.2.7.5 of [3] is a special case of SSM where the subslice interval is configured to be equal to the number of available data cells in one OFDM symbol. However, SSM is more general than the limited case of FDM and is not limited just to an FDM configuration.

TDM and SSM PLP configurations can be combined within a subframe when both non-dispersed and dispersed PLPs are present. One such example is the case of time/frequency-division multiplexing (TFDM) shown in Section 7.2.7.6 of [3]. Note again, however, that the FDM portion of this example is a special case of SSM and that SSM is not limited just to FDM configurations.

The special case of FDM allocates dispersed PLPs within a subframe such that each PLP is delivered within a narrower bandwidth than the overall system bandwidth, and may therefore allow receivers to use lower-power narrower-bandwidth operations for the body of that subframe. However, the bootstrap and the preamble symbols occupy the full system bandwidth (subject to any carrier reduction that is applied), so a receiver will need to process the full system bandwidth at the beginning of each physical layer frame.

4.2.7.2 Layered Division Multiplexing

LDM is a PLP multiplexing scheme where multiple PLPs share the same time and frequency resources, but are multiplexed together with different power levels. In terms of performance, PLPs that are strictly layered-division multiplexed together make more efficient use of physical layer resources than do PLPs that are strictly time-division or frequency-division multiplexed together. In a receiver implementation of LDM, it is necessary to cancel out the Core Layer signal in order to decode the Enhanced Layer signal, so there is an increase in computational complexity and memory use as compared to the decoding of a single PLP.

It is important to note that LDM is a PLP multiplexing scheme, but is not a PLP cell multiplexing scheme for framing, since LDM occurs earlier in the transmit chain (immediately following modulation) than does PLP cell multiplexing. Consequently, LDM and the PLP cell multiplexing schemes discussed further above are actually complementary (or non-orthogonal) PLP multiplexing schemes, in that they can be used in conjunction with each other. For example, LDM can be combined with TDM as TLDM (i.e., each TDMed PLP is a Combined PLP consisting of a Core PLP and an Enhanced PLP that have been LDMed together) or LTDM (i.e., each of the Core Layer and Enhanced Layer uses TDM methods independently of the other layer). LDM can also be combined with FDM (i.e., FLDM or LFDM) and/or, more generally, with SSM. Combining all three multiplexing methods (LDM, TDM, SSM/FDM) is also theoretically possible. Note, however, that as more than two multiplexing methods are combined, more hardware complexity may be required. Particular care that should be taken when LDM is combined with TDM and/or SSM/FDM, including memory considerations, is described in Section 5.3.3.

4.2.8 LDM Parameters

4.2.8.1 LDM Injection Level

When LDM is used, the injection level of an Enhanced PLP is signaled by **L1D_plp_ldm_injection_level** to determine the Enhanced PLP's power level relative to the associated Core PLP(s). The injection level of Enhanced PLP(s) can be varied from 0 dB to 25 dB below the associated Core PLP(s) as described in Section 6.4.2 and 9.3.5 of [3]. As shown in Figure 4.3, a lower injection level of an Enhanced PLP implies that the transmission power of the Enhanced PLP increases. However, this Enhanced PLP signal becomes additional noise to the associated

Core PLP(s), which will impact the performance of the Core PLP(s). Conversely, a higher injection level implies less transmission power allocated to an Enhanced PLP and more robustness to the associated Core PLP(s). Note that the total power after the injection of Enhanced PLP(s) is normalized to be equal to the power of a single PLP configuration. The required SNR of the Core and Enhanced PLPs after LDM combining depends on the required SNR of Core and Enhanced PLPs before LDM combining as well as the injection level. The required SNR of Core PLP (SNR_{CL_AC}) and Enhanced PLP (SNR_{EL_AC}) after LDM combining can be calculated as:

$$\mathrm{SNR_{CL_AC}} = \mathrm{SNR_{CL_BC}} + 10\log_{10}\left(1 + 10^{\frac{-\mathrm{IL}}{10}}\right) - 10\log_{10}\left(1 - 10^{\frac{\mathrm{SNR_{CL_BC}} - \mathrm{IL}}{10}}\right)$$

$$\mathrm{SNR_{EL_AC}} = \mathrm{SNR_{EL_BC}} + 10\log_{10}\left(1 + 10^{\frac{\mathrm{IL}}{10}}\right)$$

where IL is the injection level in dB scale, SNR_{CL_BC} is the required SNR value in dB of the PLP that is assigned to the Core Layer (i.e., the required SNR before LDM combining), and SNR_{EL_BC} is the required SNR value in dB of the PLP that is assigned to the Enhanced Layer (i.e., the required SNR before LDM combining). Note that when multiple Enhanced PLPs are associated with one Core PLP, the same value of injection level should be used for all of those multiple Enhanced PLPs (see Section 5.3.3). Therefore, there will be no multiple values of required SNRs for a Core PLP.

When the injection level of an Enhanced PLP is chosen, it is important to consider the required SNR of Core PLP(s) before LDM combining. Since the injected Enhanced PLP acts as additional noise to the associated Core PLP(s), the Core PLP(s) will not be decodable if the injection level is less than the required SNR of Core PLP(s) before LDM combining. Furthermore, even if the injection level is slightly higher (e.g., 1 dB) than the required SNR of Core PLP(s) before LDM combining, there is still an effect on the SNR performance of the Core PLP(s) after LDM combining. Therefore, in order to provide enough headroom between the threshold of required SNR of Core PLP(s) and the injection level, it is recommended that the injection level (IL) of Enhanced PLP(s) be chosen as the following condition:

$$\mathrm{IL} \geq \mathrm{SNR_{CL_BC}} + 3$$

Note that this recommended headroom also considers possible implementation losses. Note also that when multiple Core PLPs are associated with an Enhanced PLP, the least robust Core PLP should be used to choose the condition of the injection level.

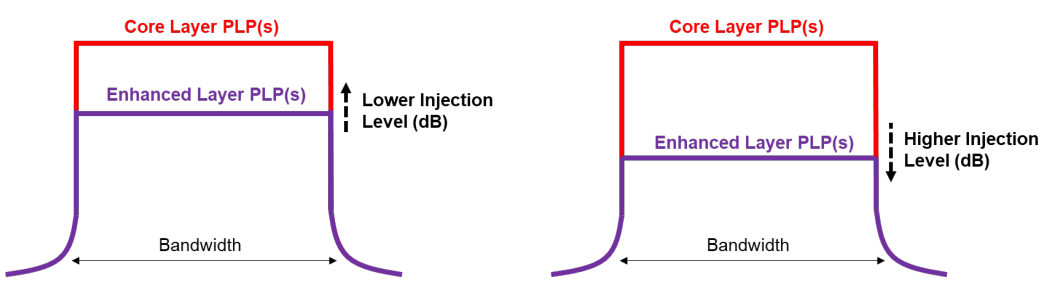

Figure 4.3 Injection level control of Enhanced PLP(s).

4.2.8.2 LDM ModCod Combination

As described in Section 4.2.8.1, particular care is needed to choose a ModCod combination for the Core Layer depending on the injection levels of the Enhanced Layer. Therefore, it is recommended that a Core PLP be configured with QPSK, 16QAM, or 64QAM constellations. When 64QAM is used for a Core PLP, it is recommended that code rates no higher than 7/15 be used to avoid high required SNRs in the Core Layer. An Enhanced PLP may be configured with any choice of ModCod combination described in Section 6.3.2 of [3].

4.2.9 Code Rate, Length and Constellation

The total number of modulation and code rate (ModCod) combinations in ATSC 3.0 is 120 (72 and 48 for long and short LDPC codes, respectively). Among these 120 choices of ModCod, there are ModCod combinations that have equivalent throughputs with other combinations, and therefore, Section 6.3.2 of [3] describes the mandatory ModCod combinations from all of the ModCod combinations given those duplicated throughputs. Table 4.3 and Table 4.4 show the required SNRs for AWGN and i.i.d. Rayleigh channels, respectively, in long code (64800 bits) cases. In these tables, capacity-equivalent ModCods are marked by the same color, and for each color, the ModCods that show the lowest required SNRs are marked in bold. ModCods that do not have equivalent throughputs are not marked. Similarly, the required SNRs for short LDPC codes (16800 bits) are also shown in Table 4.5 and Table 4.6, for AWGN and i.i.d Rayleigh channels, respectively.

Table 4.3 Required SNRs (dB) for all ModCod Combinations, Long LDPC codes (64800 bits) and AWGN Channel

Constellation	Code Rate											
	2/15	**3/15**	**4/15**	**5/15**	**6/15**	**7/15**	**8/15**	**9/15**	**10/15**	**11/15**	**12/15**	**13/15**
QPSK	-6.23	-4.32	**-2.89**	-1.70	**-0.54**	0.30	**1.16**	**1.97**	**2.77**	3.60	4.49	5.53
16QAM	-2.73	-0.25	1.46	2.82	4.21	5.21	**6.30**	**7.32**	**8.36**	9.50	10.57	11.83
64QAM	-0.26	2.27	**4.15**	5.96	7.66	8.92	**10.31**	11.55	**12.88**	14.28	15.57	17.03
256QAM	1.60	4.30	6.57	8.53	10.61	12.10	13.91	**15.55**	**17.13**	18.76	**20.44**	22.22
1024QAM	3.23	6.17	8.77	11.07	13.46	15.30	17.46	19.45	21.35	23.43	**25.52**	27.62
4096QAM	4.58	7.85	10.73	13.45	16.04	18.22	20.69	23.05	25.55	28.11	30.34	32.83

Table 4.4 Required SNRs (dB) for all ModCod Combinations, Long LDPC Codes (64800 bits) and i.i.d. Rayleigh Channel

Constellation	Code Rate											
	2/15	**3/15**	**4/15**	**5/15**	**6/15**	**7/15**	**8/15**	**9/15**	**10/15**	**11/15**	**12/15**	**13/15**
QPSK	-5.72	-3.62	**-1.97**	-0.55	0.86	1.95	3.16	4.35	5.62	7.05	8.76	10.97
16QAM	-1.84	**0.81**	**2.69**	**4.32**	5.98	7.21	8.63	9.94	11.40	12.78	14.60	16.85
64QAM	0.86	**3.61**	**5.88**	**7.74**	**9.72**	11.10	**12.75**	14.25	15.81	17.44	19.39	21.82
256QAM	2.89	5.97	**8.46**	**10.59**	12.92	14.58	16.54	**18.23**	**20.06**	21.94	24.01	26.62
1024QAM	4.65	8.04	10.85	13.25	15.91	17.84	20.13	22.34	24.47	26.61	28.82	31.59
4096QAM	6.23	9.83	12.95	**15.75**	18.79	21.03	**23.67**	26.37	**28.64**	31.18	33.82	36.54

Table 4.5 Required SNRs (dB) for all ModCod Combinations, Short LDPC Codes (16200 bits) and AWGN Channel

Constellation	Code Rate											
	2/15	3/15	4/15	5/15	6/15	7/15	8/15	9/15	10/15	11/15	12/15	13/15
QPSK	-5.55	-3.73	**-2.32**	-1.30	**-0.33**	0.56	**1.38**	**2.20**	**2.94**	3.82	4.70	5.76
16QAM	-2.15	0.35	1.99	3.16	**4.45**	5.51	**6.51**	**7.58**	**8.59**	9.74	10.81	12.09
64QAM	0.35	2.85	4.65	6.30	7.93	9.29	**10.56**	11.83	13.13	14.52	15.86	17.33
256QAM	2.27	4.78	7.19	8.93	10.91	12.57	14.25	**15.80**	17.45	19.08	20.78	22.55

Table 4.6 Required SNRs (dB) for all ModCod Combinations, Short LDPC Codes (16200 bits) and i.i.d. Rayleigh Channel

Constellation	Code Rate											
	2/15	3/15	4/15	5/15	6/15	7/15	8/15	9/15	10/15	11/15	12/15	13/15
QPSK	-5.06	-2.97	**-1.36**	-0.08	**1.15**	2.30	3.44	4.70	5.97	7.46	9.15	11.56
16QAM	-1.14	1.45	**3.41**	**4.78**	**6.27**	7.58	**8.96**	10.28	11.73	13.22	14.97	17.44
64QAM	1.60	**4.30**	6.55	8.29	**10.05**	11.54	**13.09**	14.62	16.20	17.87	19.82	22.44
256QAM	3.60	6.79	9.32	**11.16**	13.29	15.15	16.95	**18.64**	20.50	22.40	24.54	27.23

The mandatory ModCod combinations for long codes (shown in Table 4.7) and short codes (shown in Table 4.8) are obtained based on the performance results in AWGN and i.i.d. Rayleigh channels only. Note that these mandatory ModCod combinations may be useful when broadcasters have a desired physical layer throughput and need to choose the best-performing ModCod combination among all the possible ModCod combinations that have the same throughput. For maximum broadcaster flexibility, and possible future extensibility of ATSC 3.0, it is strongly recommended that all the ModCod combinations (i.e., 72 choices for long codes and 48 choices for short codes) be implemented in ATSC 3.0 transmitters as well as in ATSC 3.0 receivers.

Table 4.7 Mandatory ModCod Combinations for Long Codes (64800 bits)

Constellation	Code Rate											
	2/15	3/15	4/15	5/15	6/15	7/15	8/15	9/15	10/15	11/15	12/15	13/15
QPSK	✓	✓	✓	✓	✓	✓	✓	✓		✓		
16QAM			✓	✓		✓	✓	✓		✓		
64QAM		✓	✓	✓	✓	✓	✓	✓	✓	✓		
256QAM			✓	✓		✓	✓	✓	✓	✓	✓	✓
1024QAM				✓		✓	✓	✓	✓	✓	✓	✓
4096QAM						✓		✓		✓	✓	✓

Table 4.8 Mandatory ModCod Combinations for Short Codes (16200 bits)

Constellation	Code Rate											
	2/15	3/15	4/15	5/15	6/15	7/15	8/15	9/15	10/15	11/15	12/15	13/15
QPSK	✓	✓	✓	✓	✓	✓	✓	✓				
16QAM				✓	✓	✓	✓			✓		
64QAM				✓	✓	✓	✓	✓	✓	✓		
256QAM				✓		✓	✓	✓	✓	✓	✓	✓

4.2.10 L1 Protection Mode

The L1-Basic Mode is indicated via **preamble_structure** of the bootstrap signaling together with the FFT size, guard interval, and pilot pattern of the Preamble as described in Table H.1.1 of [3]. Note that Section 6.5 of [3] defines seven L1-Basic Modes (i.e., from L1-Basic Mode 1 to L1-Basic Mode 7); however, the current **preamble_structure** of the bootstrap signaling can indicate only five L1-Basic Modes (i.e., from L1-Basic Mode 1 to L1-Basic Mode 5).

The L1-Detail Mode is indicated via **L1B_L1_Detail_fec_type** of the L1-Basic signaling. Section 6.5 of [3] defines the seven L1-Detail Modes (i.e., from L1-Basic Mode 1 to L1-Basic Mode 7), and the current **L1B_L1_Detail_fec_type** from the L1-Basic signaling can select all the choices of L1-Detail Modes.

The length of L1-Basic data is fixed at 200 bits, while the length of L1-Detail data is variable. Since the L1 protection uses the LDPC codes of 3/15 rate (Type A) and 6/15 rate (Type B) from data PLP, zero padding (i.e., shortening LDPC information bits) is performed before LDPC encoding. Furthermore, parity repetition and puncturing are also performed to achieve desired robustness after LDPC encoding. The protection mode of L1-Basic and L1-Detail signaling is determined based on the LDPC code rate, modulation order, parity repetition and parameters such as the number of punctured parity bits and the number of zero-padded bits, as shown in Table 4.9. Note that the parameters *A* and *B* in Table 4.9 indicate the number of punctured bits to the number of shortened bits ratio (i.e., zero-padding bits) and the constant integer for parity puncturing, respectively. More detailed parameters for L1 protection modes are given in Tables 6.17, 6.18, 6.23, 6.24 and 6.25 of [3].

Table 4.9 Definition of L1 Protection Modes

L1 Protection Mode		LDPC Code Rate	Modulation	Parameters for Puncturing		Parity Repetition
				A	B	
L1-Basic	Mode 1	R=3/15 (Type A)	QPSK	0	9360	✓
	Mode 2		QPSK		11460	*N/A*
	Mode 3		QPSK		12360	*N/A*
	Mode 4		16QAM		12292	*N/A*
	Mode 5		64QAM		12350	*N/A*
	Mode 6		256QAM		12432	*N/A*
	Mode 7		256QAM		12776	*N/A*
L1-Detail	Mode 1		QPSK	7/2	0	✓
	Mode 2		QPSK	2	6036	*N/A*
	Mode 3	R=6/15 (Type B)	QPSK	11/16	4653	*N/A*
	Mode 4		16QAM	29/32	3200	*N/A*
	Mode 5		64QAM	3/4	4284	*N/A*
	Mode 6		256QAM	11/16	4900	*N/A*
	Mode 7		256QAM	49/256	8246	*N/A*

The required SNRs of the L1-Basic and L1-Detail Modes under AWGN and Rayleigh channels are shown in Table 4.10. Note that the Preamble (including L1-Basic and L1-Detail) does not provide time diversity unless the Additional Parity for L1-Detail is used. Therefore, when selecting the L1-Basic and L1-Detail Modes, it is always recommended that both L1-Basic and L1-Detail Modes be more robust than the data payload modes. The general recommendation of L1-Basic and L1-Detail Modes is to use Mode 1 to Mode 3 to provide much higher robustness than the data

payload. It is also recommended that the L1-Detail Mode use the same mode as the L1-Basic Mode indicated by the bootstrap signaling. For example, when the L1-Basic Mode is selected as Mode 1, then L1-Detail Mode 1 is also recommended.

Table 4.10 Performance of L1-Basic and L1-Detail Modes Under AWGN and Rayleigh Channels (FER=10^{-4})

Mode	L1-Basic (dB)		L1-Detail (dB)	
	AWGN	Rayleigh	AWGN	Rayleigh
1	-9.2	-9.0	-7.8	-7.5
2	-2.7	-1.8	-1.4	-0.3
3	1.0	2.8	0.5	3.3
4	5.4	7.5	6.1	8.3
5	9.9	12.3	10.2	12.9
6	Not used		15.8	18.6
7			22.5	27.2

4.2.11 Time Interleaver Mode

Each PLP transmitted at the physical layer is configured with one of three time interleaver modes, as specified in Section 7.1.1 of [3], with the available time interleaver modes being: no time interleaving, Convolutional Time Interleaver (CTI) mode, and Hybrid Time Interleaver (HTI) mode. The time interleaver mode for a particular PLP is indicated via the **L1D_plp_TI_mode** signaling field for that PLP (see Section 9.3.4 of [3]). The time interleaver mode is signaled on a per PLP basis for all PLPs, even when Layered Division Multiplexing (LDM) is in use and Core and Enhanced PLPs are present. That is, **L1D_plp_TI_mode** is signaled for Enhanced PLPs, even though such Enhanced PLPs pass through time interleaving as a combined part of the Core PLP(s) with which they have been layered division multiplexed.

When LDM is configured, the time interleaver mode signaled for an Enhanced PLP is the same as for the time interleaver mode signaled for the Core PLP(s) with which that Enhanced PLP is layered division multiplexed. Note that an Enhanced PLP can be layered division multiplexed with multiple Core PLPs (see Section 5.3.3 for details).

4.2.11.1 Valid Conditions for CTI Mode

A PLP can be configured with any time interleaver mode that is considered valid for that particular PLP's situation. Time interleaver mode validity for a PLP is determined on the basis of what is referred to in [3] as a complete delivered product. A complete delivered product represents all of the data (e.g., video, audio, captions, meta-data) that goes together to form a complete product (e.g., program, movie, sportscast, etc) presented to an end user at a receiver. For example, this data could all be multiplexed together into a single PLP that carries video and audio data packets together. Alternatively, video data for a particular program could be carried in one PLP and the corresponding audio data for the same program could be carried in a second PLP. Consequently, a complete delivered product consists of one or more PLPs carrying related components of the same end-user service. Note that multiple programs may be multiplexed together in the same RF channel, so different PLPs within the same RF channel may belong to the same or different complete delivered products.

Valid time interleaver modes for the PLPs of a particular complete delivered product are determined independently of any other distinct complete delivered products present within the same RF channel. That is, only the characteristics of PLP(s) belonging to a particular complete

delivered product are considered when determining which time interleaver modes are considered as valid for the PLP(s) of that complete delivered product.

The CTI mode can only be, but does not necessarily need to be, used for the PLP(s) of a complete delivered product if either of the following two conditions is satisfied. (These two conditions are mutually exclusive so a maximum of only one of them can be satisfied.)

- The complete delivered product is composed of a single PLP that has a constant cell rate as measured at the input to the time interleaver.
- The complete delivered product is composed of a single Core PLP and one or more Enhanced PLPs layered division multiplexed with that Core PLP, with all of those PLPs having a constant cell rate as measured at the input to the time interleaver.

In the context of the above two conditions, constant cell rate means that over a given period of time (e.g., 1 second) that begins at any time point within the data flow, the same amount of data always arrives (e.g., from the studio) at the input to the time interleaver. For example, a 4 Mbps constant cell rate PLP would mean that during any one-second time period, 4 Mbits of data would arrive at the modulator for that PLP.

Note that it is important to consider the cell rate at the input to the time interleaver, rather than at the output from the time interleaver. Depending upon the exact physical layer frame configuration, the cell rate for a PLP at the output from the time interleaver may be bursty rather than constant. For example, consider the same 4 Mbps constant cell rate PLP. Suppose that each physical layer frame is configured to have a length of 1 second and that each such frame contains two subframes with each subframe having a length of 0.5 seconds. Assume that the PLP is carried only in the first subframe of each frame. Then, at the output of the time interleaver, the observed cell rate for the PLP will be 8 Mbps (4 Mbits divided by the first subframe's length of 0.5 seconds) during the first subframe, but 0 Mbps (0 Mbits divided by the second subframe's length of 0.5 seconds) during the second subframe. Thus, it is necessary to consider a PLP's data rate at the input to the time interleaver, rather than at the output, when determining whether or not that PLP possesses a constant cell rate. The other key condition, of course, is that over a certain period of time (e.g., 1 second), the number of cells arriving at the input of the time interleaver (e.g., from the studio) equals the number of cells leaving the output of the time interleaver (i.e., being multiplexed into physical layer subframes). This ensures that the CTI never overflows or underflows.

When the CTI mode is valid for a particular complete delivered product, this does not imply that the CTI mode is mandated to be used for the PLP(s) belonging to that complete delivered product. Instead, the PLP(s) may be configured, if so desired, with one of the other time interleaver modes (HTI mode or no time interleaving mode). However, it is recommended that PLP(s) for which the CTI mode is considered valid should be configured with the CTI mode for optimum performance.

4.2.11.2 Valid Conditions for HTI and No TI Mode

Any PLPs can be configured with either the HTI mode or the no time interleaving mode. Multiple Core and/or non-layered-division-multiplexed PLPs belonging to the same complete delivered product can be configured individually with either the HTI mode or the no time interleaving mode. That is, when a complete delivered product contains multiple Core and/or non-layered-division-multiplexed PLPs belonging to the same complete delivered product, each of those PLPs can be configured with either the HTI mode or the no time interleaving mode, regardless of the time interleaver mode(s) configured for other PLPs within the same complete delivered product.

Additionally, when a complete delivered product contains multiple Core and/or non-layered-division-multiplexed PLPs belonging to the same complete delivered product, each of those PLPs can be configured with the same or different time interleaver parameters with some restrictions, as described in more detail below.

For use cases considered at the present time it is anticipated that multiple Core and/or non-layered-division-multiplexed PLPs belonging to the same complete delivered product would benefit from all being configured with the HTI mode (see also the discussion on the no time interleaving mode further below).

When a complete delivered product contains multiple Core and/or non-layered-division-multiplexed PLPs and all of those PLPs are configured with the HTI mode, then the normative constraint is that all of those PLPs use the same subframe interleaving mode within the HTI. That is, those PLPs either all use intra-subframe interleaving or all use inter-subframe interleaving; it is not allowed for some of those PLPs to use intra-subframe interleaving while other PLPs within the same complete delivered product use inter-subframe interleaving. Additionally, when those PLPs use inter-subframe interleaving, then the normative constraint is that those PLPs all use the same time interleaving unit (N_{IU}).

When a complete delivered product contains multiple Core and/or non-layered-division-multiplexed PLPs and at least one of those PLPs is configured with the no time interleaving mode, then the use of the HTI mode with inter-subframe interleaving is disallowed for any of the other PLPs within the same complete delivered product. In this situation (with at least one PLP within a complete delivered product being configured with no time interleaving), the normative constraint is that all of the PLPs configured with HTI mode within that same complete delivered product use the same subframe interleaving mode, which is intra-subframe interleaving. Inter-subframe interleaving is not allowed for any of the PLPs within a complete delivered product if at least one PLP within that same complete delivered product is configured with the no time interleaving mode.

As mentioned earlier, any PLP can be configured with the no time interleaving mode. When the no time interleaving mode is configured for a PLP, that PLP does not consume any time interleaver memory resources. There may be certain use cases where it may be desirable to completely disable time interleaving for one or more PLPs, since time interleaving does introduce an additional amount of time latency (dependent upon the time interleaver depth). Note, however, that disabling time interleaving also removes any performance benefits obtained from time diversity. It is therefore recommended that broadcasters do not configure the no time interleaving mode unless they are very certain that it is applicable to the particular use case under consideration.

4.2.11.3 Extended Time Interleaving

Extended time interleaving is specified in Section 7.1.3 of [3] and is indicated on a per PLP basis via **L1D_plp_TI_extended_interleaving** (see Section 9.3.9 of [3]). Extended interleaving can only be used when the corresponding PLP is configured with QPSK modulation and is not layered division multiplexed as either a Core or Enhanced PLP. Extended interleaving can be used with either CTI mode or HTI mode, but has no real meaning for the no time interleaving mode since no time interleaver memory is associated with that latter mode. When a complete delivered product is comprised of multiple PLPs, a subset of those PLPs can be configured for extended time interleaving provided that the other conditions discussed in this section are satisfied.

The purpose behind extended interleaving is to extend the number of cells that can be stored in the time interleaver memory provisioned within an ATSC 3.0 device. As specified in Section 7.1.2 of [3], ATSC 3.0 receivers should be able to store up to 2^{19} cells (per complete delivered

product) in time interleaver memory for regular time interleaving and up to 2^{20} cells for extended time interleaving. When a complete delivered product is comprised of multiple PLPs, with at least one PLP configured for regular time interleaving and at least one PLP configured for extended time interleaving, then the sum of the total time interleaver memory (in cells) required for PLPs configured with regular time interleaving and half of the total time interleaver memory (in cells) required for PLPs configured with extended time interleaving should be less than or equal to 2^{19} cells. There are two primary approaches that can be used to provision the time interleaver memory in an ATSC 3.0 receiver.

- The ATSC 3.0 receiver can be provisioned with sufficient memory to store 2^{20} regular data cells (per complete delivered product), which meets the requirement of storing up to 2^{19} cells in time interleaver memory for regular time interleaving and up to 2^{20} cells for extended time interleaving.
- The ATSC 3.0 receiver can be provisioned with sufficient memory to store 2^{19} regular data cells (per complete delivered product). When extended interleaving is configured, a coarser quantization resolution can be used for QPSK constellations (as compared to higher-order QAM constellations) such that the memory normally used to store one regular data cell can instead store two extended interleaving data cells.

The use of extended time interleaving, when allowed for a particular PLP, should increase the time diversity benefits experienced by that PLP. At the same time, however, the extended time depth of the interleaver will result in additional time latency experienced by the PLP's data passing through the time interleaver at a transmitter and the time de-interleaver at a receiver.

4.2.12 Guard Interval

Each OFDM symbol within a physical layer frame has a guard interval prepended to it. There are twelve available guard interval lengths as listed in Section 8.5 of [3], ranging from short (192 samples) to long (4864 samples). Note that the guard interval length is limited to a maximum of one-quarter of the corresponding configured FFT size.

The guard interval length is configured per subframe and is indicated by **L1B_first_sub_guard_interval** (Section 9.2.3 of [3]) and **L1D_guard_interval** (Section 9.3.3 of [3]), depending upon the applicable subframe.

The guard interval provides protection against inter-symbol interference that arises when multiple time-delayed copies of the same OFDM symbol arrive at a receiver. Such time-delayed copies can arise from one or both of the following causes.

- Multi-path propagation. When the local terrain includes various "reflectors" such as hills, mountains, buildings, etc., the originally transmitted signal may reflect off these features and cause a time-delayed copy of the signal to arrive at a receiver.
- Multiple transmitters in a Single Frequency Network (SFN). In an SFN, all transmitters transmit the same signal at the same time. However, the propagation distances to a receiver from each transmitter will, in general, be different, and thus multiple copies of the same signal will arrive with differing time delays.

The end result, as viewed by a receiver, of both of the above two causes is essentially the same: multiple copies of the desired signal will arrive with differing time delays. The range, in time, between these different echoes is termed the delay spread of the propagation channel. For a single transmitter scenario (no SFN), the presence and length of the delay spread will be dependent upon the local terrain and should be able to be measured using a professional receiver. Flat terrain with

no reflectors will likely result in little or no delay spread, while the presence of many reflectors will result in a larger delay spread of the propagation channel.

In an SFN, the apparent delay spread will primarily be a function of the distance between adjacent transmitters since this will also result in copies of the same signal arriving with different time delays at a receiver. The delay spread in this scenario can be estimated using the propagation distances from different transmitters to various example receiver locations and/or can be measured using a professional receiver.

The configured guard interval length for a given deployment is recommended to be equal to or greater than the observed and/or estimated delay spread for that same deployment in order to minimize the performance impacts of inter-symbol interference. At the same time, the guard interval represents overhead so selecting a guard interval that is significantly longer than is actually required would result in the unnecessary loss of efficiency.

When multiple subframes are configured within a frame, there is a choice whether to configure all subframes with the same guard interval length or to configure different guard interval lengths for different subframes within the same frame, since the guard interval length is configured per subframe. It is generally recommended that the same guard interval length be configured for all subframes, with the possible exception of the following scenario. In an SFN, transmitted signals might contain both region-wide content (transmitted by all transmitters in the SFN) and local content (different content transmitted by each individual transmitter). For subframes containing region-wide content, the configured guard interval length needs to take into account the distance between adjacent transmitters. For subframes containing only local content, the configured guard interval length may only need to consider the delay spread introduced by local terrain features.

4.2.13 Frequency Interleaver Mode

After PLP data has been mapped to the data cells of OFDM symbols in the frequency domain and prior to pilot insertion, those OFDM symbol data cells can be frequency interleaved within each OFDM symbol as described in Section 7.3 of [3]. Frequency interleaving can be enabled or disabled on a per subframe basis, and this enabling/disabling is indicated for each subframe by **L1D_frequency_interleaver** (Section 9.3.3 of [3]).

When enabled, frequency interleaving changes the order in which carriers are shuffled in the frequency dimension every one or two OFDM symbols, depending upon the configured FFT size. As such, frequency interleaving is intended to introduce frequency diversity into an ATSC 3.0 physical layer transmission, which should result in improved performance.

There may be possible future use cases that would benefit from or that would require frequency interleaving to be disabled. See, for example, the quasi-frequency-division-multiplexing and quasi-time-frequency-division-multiplexing examples in Sections 7.2.7.5 and 7.2.7.6, respectively, of [3]. However, at the present time and for presently-envisioned use cases it is recommended that frequency interleaving be enabled for all transmitted subframes in order to obtain the benefits of frequency diversity.

4.2.14 Time Information Type

It is recommended that time information be always sent in every frame. There are several choices for the components of the time information. It is recommended that the most detailed information available be sent, that is, if nanosecond time information is available, this should be sent, i.e., **L1B_time_info_flag** $=11_2$. This allows receivers to use the maximum time accuracy of transmitters.

4.2.15 Multiple Subframes within a Frame

There are many possible ways to transmit multiple PLPs according to the ATSC 3.0 specification. Particularly broadcasters may wonder about whether to transmit multiple PLPs in a single subframe or in multiple subframes.

It is recommended that the number of subframes used be limited to the minimum number possible, since there is a small loss of efficiency when SBS (subframe boundary symbols) are used. Therefore, if there are 2 PLPs that both use 8K FFT size and the same pilot pattern and there are no other considerations (for example different broadcasters operating different services), although separate subframes with the same parameters are possible, it makes sense to combine the 2 PLPs into the same subframe.

Furthermore, care should be taken with the order of the subframes, as described in Section 5.3.2. The most common envisaged use of subframes is to allow completely different OFDM parameters, specifically different FFT sizes and/or different pilot patterns. For example, a typical scenario for a 2-PLP TDM may be to have one service targeting mobile reception, and another service targeting stationary reception. With multiple subframes, the mobile service could use the 8K or 16K FFT size for greater resistance to Doppler fading and thus high speed and mobility. A denser pilot pattern can provide better time and frequency interpolation of the channel leading to better channel estimation and higher mobile performance. Conversely, the service targeting stationary reception could use a 32K FFT for better efficiency and a less dense pilot pattern since mobility is not a priority for that service. In this way, there is great flexibility in combining TDM with multiple subframes.

Some common examples for combinations of subframes and multiple PLPs are as below:

- 1 PLP in subframe 1 (8K FFT), 1 PLP in subframe 2 (32K FFT, example above)
- 1 PLP in subframe 1 (8K FFT), 3 PLPs in subframe 2 (32K FFT)
- 2 PLPs in subframe 1 (8K FFT), 2 PLPs in subframe 2 (32K FFT)
- 4 PLPs in one subframe (16K FFT)

While not common, the following example may also be used

- 4 PLPs, each in their own subframe

While the use of PLPs spanning multiple subframes within the same frame is allowed by the specification, this option should be used with care. Note that Section 7.2.2.1 of [3] mandates that subframes containing the same PLP are of the same type (e.g., it is not allowed to map the same PLP to subframes containing different FFT sizes). Therefore, it is recommended that broadcasters consider that up to 4 subframes per frame are likely to be supported by receivers, where the PLP(s) are completely contained within each subframe (and do not span multiple subframes within the same frame).

4.2.16 L1D_plp_fec_type

In the current version of [3], **L1D_plp_fec_type** has defined values 0 to 5 and reserved values 6 to 15. At the present time, **L1D_plp_fec_type** should not be set to a value greater than or equal to 6 since those values are currently reserved. However, legacy receivers should be able to correctly parse a future version of the L1-Detail signaling in the event that additional FEC types are defined at a later time. Note that legacy receivers would not be expected to decode such new FEC types, but would be expected to correctly handle the L1-Detail signaling.

For the current version of [3], when the indicated value of **L1D_plp_fec_type** for a particular PLP is greater than or equal to 6 in Table 9.8 of [3], the signaling fields **L1D_plp_mod** and **L1D_plp_cod**

will not be included and will therefore have unspecified values for that PLP. There is a later condition in Table 9.8 of [3] that is evaluated to determine whether or not to include the **L1D_plp_TI_extended_interleaving** signaling field for each PLP. This condition is based, in part, on testing the indicated value of **L1D_plp_mod**. When **L1D_plp_mod** has an unspecified value for a particular PLP (as could arise if a currently-reserved value of **L1D_plp_fec_type** is indicated), it should be assumed that this condition is not satisfied and that therefore **L1D_plp_TI_extended_interleaving** is not configured for that PLP. This assumption will allow receivers to ensure correct interpretation of future versions of the L1-Detail signaling.

4.2.17 Transmitter Identification (TxID)

TxID is signaled from the Scheduler that is commonly located in a studio, and is injected into ATSC 3.0 host signals for unique identification of individual transmitters. The TxID, injected only in the first Preamble symbol (see Annex N of [3]), should appear as only a small amount of interference to ordinary ATSC 3.0 consumer receivers, and should therefore be transparent to such receivers. Precisely, the receiving performance of the Preamble will be degraded depending on the injection level of the TxID, but the data payload part will not be affected by the TxID. Table 4.11 shows the required SNRs of the Preamble according to the TxID injection.

Table 4.11 Required SNRs of the Preamble when TxID is Injected (AWGN Channel)

Injection Level [dB]	Preamble Decoding Performance [dB]											
	L1-Basic					L1-Detail						
	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Off	-9.20	-7.80	-2.70	-1.40	1.00	1.50	5.40	6.10	9.90	10.20	16.50	27.00
45	-9.20	-7.80	-2.70	-1.20	0.90	2.00	5.40	6.50	9.70	11.00	16.51	27.07
42	-9.20	-7.80	-2.70	-1.20	0.90	2.00	5.40	6.50	9.70	11.00	16.51	27.14
39	-9.20	-7.80	-2.70	-1.20	0.90	2.00	5.40	6.50	9.71	11.01	16.52	27.28
36	-9.20	-7.80	-2.70	-1.20	0.90	2.00	5.40	6.50	9.71	11.01	16.55	27.58
33	-9.20	-7.80	-2.70	-1.20	0.90	2.00	5.41	6.51	9.72	11.03	16.60	28.26
30	-9.20	-7.80	-2.70	-1.20	0.91	2.01	5.42	6.52	9.74	11.06	16.70	30.02
27	-9.20	-7.80	-2.70	-1.19	0.91	2.01	5.43	6.54	9.78	11.11	16.91	N/A
24	-9.20	-7.80	-2.69	-1.19	0.92	2.03	5.46	6.58	9.86	11.22	17.35	N/A
21	-9.20	-7.79	-2.68	-1.17	0.94	2.06	5.52	6.66	10.03	11.46	18.40	N/A
18	-9.19	-7.79	-2.66	-1.15	0.99	2.11	5.65	6.82	10.40	11.97	21.85	N/A
15	-9.18	-7.78	-2.63	-1.09	1.07	2.22	5.90	7.16	11.22	13.20	N/A	N/A
12	-9.17	-7.75	-2.55	-0.99	1.25	2.46	6.47	7.94	13.56	17.87	N/A	N/A
9	-9.13	-7.71	-2.41	-0.76	1.63	2.97	7.89	10.09	N/A	N/A	N/A	N/A

Note that N/A in Table 4.11 implies that the TxID cannot be injected because of the performance limitation of the corresponding L1-Basic or L1-Detail Modes. For example, the required SNR of L1-Detail Mode 7 is 27 dB, and therefore, any TxID injection level less than or equal to 27 dB (i.e., 9 – 27 dB) cannot be used because the injected TxID would cause sufficient interference in those cases such that no receiver would be able to decode L1-Basic or L1-Detail successfully. Therefore, all the N/A modes in Table 4.10 should not be used by the Scheduler.

Broadcasters should turn on the TxID only when network design, adjustment, monitoring, or measurements are needed in SFNs since otherwise the injected TxID will cause unnecessary

interference to the first Preamble symbol. Specialized receivers will be needed to obtain channel impulse response information given by the TxID for such monitoring and measurement, but ordinary consumer ATSC 3.0 receivers will not necessarily decode the TxID.

5. GUIDELINES FOR TRANSMITTER IMPLEMENTATION

5.1 Input Formatting

The input formatting consists of three blocks: encapsulation and compression of data, baseband formatting, and the Scheduler. Encapsulation and compression of data blocks creates ALP (ATSC 3.0 Link layer Protocol) packets which are defined in [4]. Baseband formatting consists of three blocks (i.e., Baseband Packet Construction, Baseband Packet Header Addition, and Baseband Packet Scrambling) and creates Baseband Packets as defined in Section 5.2 of [3]. The Scheduler that is defined in [5] oversees the generation of Baseband Packets, and those Baseband Packets are encapsulated in the dedicated format by [5] and sent to ATSC 3.0 transmitters through a studio to transmitter link (STL).

5.1.1 Delivered Product in Multiple PLPs

When a complete delivered product is assembled and delivered by multiple PLPs, those multiple PLPs are required to be simultaneously recovered in ATSC 3.0 receivers, and therefore, the maximum number of PLPs required to assemble a complete delivered product is limited to four, while also satisfying the condition of the maximum TI memory size of receivers, i.e., 2^{19} cells.

Depending on broadcasters' intended services, a complete delivered product carried by multiple PLPs may contain a number of different data streams such as signaling, multiple audios, video, enhanced video, or application data. Care should be taken when such multiple data streams requiring simultaneous recovery in receivers (i.e., complete delivered product) are carried in multiple PLPs. In principle, additional PLP(s) are required when different robustness levels and different coverage areas are intended by broadcasters. If multiple data streams are required to have the same robustness level and the same coverage area (e.g., multiple audio streams in multiple languages), it is recommended that those data streams be carried in the same PLP. Note that multiplexing these multiple data streams prior to the physical layer may even be beneficial due to the statistical multiplexing gain as long as the same robustness for those multiple data streams is intended. Therefore, when all of the components of a transmitted product are carried by multiple PLPs, it is strongly recommended that the maximum number of PLPs for that product be limited to four at the transmitter, and those four PLPs consume the full TI memory size (i.e., 2^{19} cells) in order to maximize the time diversity of individual PLPs. It is not recommended that all of the components of a transmitted product be carried by more than four PLPs at the transmitter, even though a 4-PLP subset within the complete delivered product of more than four PLPs would meet the TI memory requirement at receivers.

The IP-level signaling information such as Low Level Signaling (LLS), including Service List Table (SLT) and Service Layer Signaling (SLS) [6], and Link Mapping Table (LMT) [4] may be configured in a separate PLP in order to provide different robustness for the signaling information. In such cases, it is strongly recommended that the IP-level signaling information be present in the most robust PLP out of multiple PLPs carrying a complete delivered product. When LDM is used, it is recommended that the IP-level signaling information be present in the most robust Core PLP out of multiple PLPs carrying a complete delivered product.

A complete delivered product that may be composed of one or more PLPs should contain the appropriate IP-level signaling information including LLS and LMT. Note that in [4] and [6], the

minimum requirement of the LLS and LMT delivery is to be repeated in every 5 seconds. For fast service acquisition, it is recommended that the LLS and LMT be sent in every physical layer frame. This would result in setting **L1B_lls_flag** = 1 (for LLS carried in the current frame) and **L1D_plp_lls_flag** = 1 (for LLS carried in the corresponding PLP) in every frame.

When a complete delivered product is carried by multiple PLPs, a possible configuration is to use multiple subframes. One example of such a service scenario is a robust audio service delivered in an 8K or 16K FFT subframe and a video service delivered in a 32K FFT subframe. Therefore, it is recommended that ATSC 3.0 receivers be able to simultaneously recover multiple subframes that may carry a complete delivered product. Note that in such cases, the CTI mode is not allowed by the current version of specification since the CTI mode is used when a complete delivered product is composed of only a single Core PLP with constant cell rate (see Section 7.1.1 of [3]). Further guidelines for multiple PLPs configured in multiple subframes are described in Section 4.2.15.

When LDM is used, multiple PLPs that are layered division multiplexed share the common subframe parameters such as FFT size, pilot pattern, and guard interval. Therefore, it is not recommended that a complete delivered product be delivered by multiple subframes that are all LDM configured, so that receivers do not need simultaneous recovery of multiple LDM configured subframes. Further guidelines for multiple PLPs in combination with LDM are described in Section 5.3.3.

5.2 Bit Interleaved and Coded Modulation (BICM)

The ATSC 3.0 BICM chain consists of three processes: the forward error correction (FEC), the bit interleaver, and the constellation mapping. For protection of input data, an ATSC 3.0 transmitter performs the FEC with outer and/or inner coding. Then, the bit interleaving is performed with three steps such that (1) the parity interleaver interleaves only the LDPC parity bits, (2) the group-wise interleaver interleaves bit groups which are obtained by splitting an LDPC codeword, and (3) the block interleaver interleaves the bit groups interleaved by the group-wise interleaver. Finally, the interleaved codeword bits are mapped onto constellation points according to the selected modulation scheme. When LDM is used, a superposition technique of constellations is further performed (see Section 6.4 of [3]).

5.2.1 Forward Error Correction (FEC)

The FEC of the BICM is formed by concatenation of an outer code and an inner code with the information part. The outer code is either a Bose, Chaudhuri, Hocquenghem (BCH) code, a Cyclic Redundancy Check (CRC) or none, while the inner code is a Low-Density Parity Check (LDPC) code. Both BCH and LDPC codes are systematic codes, such that the information part is contained within the codeword. The resulting codeword is thus a concatenation of information or payload part, BCH or CRC parities and LDPC parities, as shown in Figure 5.1 (top). The codeword when no BCH or CRC parity is used is also depicted in Figure 5.1 (bottom).

To provide maximum flexibility and achieve higher throughput when sufficient error correction capability of the inner code is expected, the CRC may be used as an outer code or even no outer code may be selected. Note that when the CRC is used for improved efficiency, it does not provide any error correction capability, but only provides error detection. Since the use of BCH provides additional 12 bit error correction and detection capability that can prevent possible error floors depending on the LDPC decoding methods of receivers, it is recommended that BCH be used as an outer code (see Section 6.5.3 for the performance analysis of the decoding methods).

When BCH is used, the length of M_{OUT}=192 bits for N_{IN}=64800 and M_{OUT} =168 bits for N_{IN}=16200. When the CRC is used, the length of M_{OUT}=32 bits. When no outer code is used, M_{OUT} =0.

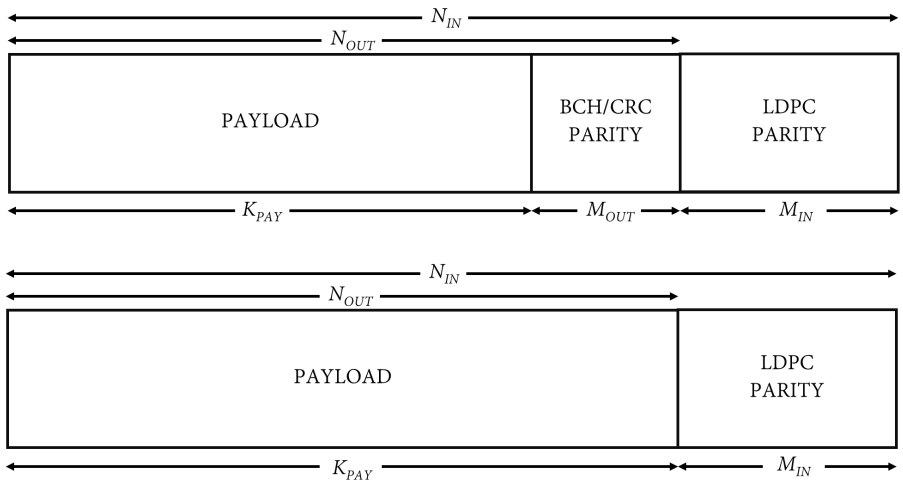

Figure 5.1 Format of FEC frame when BCH/CRC is used (top) or not (bottom).

Regarding the inner LDPC, two different structures are defined in ATSC 3.0. The Type A LDPC code structure is mostly used in lower code rates, and the Type B LDPC code structure is used in medium and higher code rates. Two different lengths of LDPC code (N_{IN}) are defined: 64800 and 16200 bits. The 64800 bit codes provide better performance than the 16200 bit codes (see Annex A and Annex B for details of performance results). As the 16200 bit codes provide lower latency and memory use, it is recommended that 16200 bit codes be used for mobile services when consumer receivers have limited power resources.

5.2.1.1 Inner Encoding

The ATSC 3.0 LDPC codes are defined by a parity check matrix (PCM) as shown in Figure 5.2. For both Type A and Type B LDPC codes, the sub-matrix of PCM corresponding to the information bits consists of row/column-permuted quasi-cyclic matrices such as circulant permutation matrices (CPMs). Due to this quasi-cyclic structure, the ATSC 3.0 LDPC codes based on the Type A and Type B structures can be efficiently implemented in the aspects of partial parallel encoding and decoding with the parallel factor of CPM size (=360).

The PCM of a Type A LDPC code consists of 5 sub-matrices A, B, C, D, and Z as shown in Figure 5.2 (*a*). For each combination of code length and code rate, the lengths of the first parity part corresponding to matrix B and of the second parity part corresponding to matrix D are deterministic values as specified in Table 6.5 of [3]. Once the length of information bits N_{outer}, the first parity part M_1, and the second parity part M_2 are derived, the size of each matrix is defined as follows. The sizes of matrices A and C are $M_1 \times N_{outer}$ and $M_2 \times (N_{outer} + M_1)$, respectively. Matrix Z is a zero matrix having a size of $M_1 \times M_2$, matrix D is an identity matrix having a size of $M_2 \times M_2$, and matrix B is a dual diagonal matrix having a size of $M_1 \times M_1$. In matrix B, all elements are 0 except elements along a diagonal line and neighboring elements.

In the case of the Type A LDPC encoding process, the accumulation is performed at parity bit addresses corresponding to the values of each row of the sequence specified in Annex A of [3]. For a given code length and code rate, the rows of the sequence are repeatedly changed by the CPM size (=360) of the PCM. In Type A LDPC, each sequence has a number of rows equal to the

sum of the value obtained by dividing the length of the information part N_{outer} by the CPM size 360 and the value obtained by dividing the length of the first parity part M_1 by the CPM size 360.

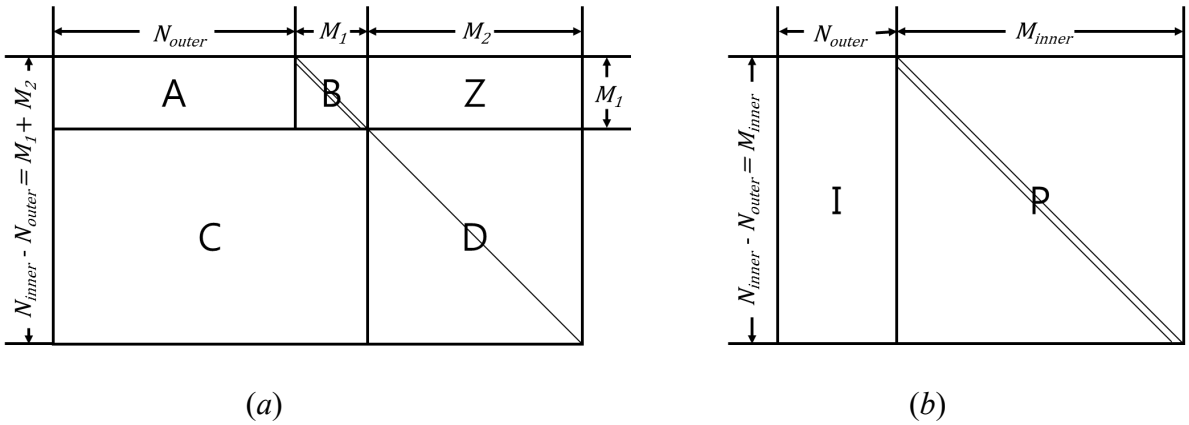

Figure 5.2 PCM structure of (*a*) Type A LDPC and (*b*) Type B LDPC.

The PCM of a Type B LDPC code, consisting of the I and P sub-matrices in Figure 5.2 (*b*), has the same structure as the sub-matrix [A B] that is a part of the PCM of Type A LDPC codes. The size of matrix I is $M_{\text{inner}} \times N_{outer}$, and matrix P is a dual diagonal matrix having a size of $M_{\text{inner}} \times M_{\text{inner}}$. Similar to the Type A LDPC codes, the Type B LDPC codes generate the parity bits using the PCM determined by the parity bit indices in Annex A of [3].

In ATSC 3.0, the PCM for each LDPC code (Type A and B) is described by an indices list (see Annex A of [3]). The indices in the i-th row in the indices list denote the positions of weight-1 in the 0-th column of the i-th column group. Note that each column group consists of 360 columns, and for the Type A LDPC code, the column groups corresponding to the indices list represent the sub-matrix of the PCM corresponding to the information bits and the first parity bits. For the Type B LDPC code, all the column groups corresponding to the indices list represent the sub-matrix of the PCM corresponding to information bits only. The weight-1s in the other 359 columns of each column group are arranged based on the indices list in a quasi-cyclic manner. The indices list is equivalent to a set of cyclic shift values for non-zero CPMs since the list is defined by the number, location and cyclic shift values of the CPMs. Each indices list is obtained by adjusting cyclic shift values of non-zero CPMs (this is known as lifting), which can be derived from the algebraic properties such as cycle characteristics or degree distribution.

5.2.2 Bit Interleaving

The bit interleaver in the ATSC 3.0 BICM chain distributes burst errors occurring in a physical propagation channel by interleaving the LDPC codeword using the interleaving sequence defined in Section 6.2 and Annex B of [3]. ATSC 3.0 has a 3-stage bit interleaver, which consists of a parity interleaver followed by a group-wise interleaver followed by a block interleaver, as shown in Figure 5.3. This structure allows for parallel LDPC decoding while optimizing the performance of the FEC codes to any constellation.

Figure 5.3 Bit interleaver structure: Parity, group-wise and block interleavers.

The role of the parity interleaver is to convert the staircase structure of the parity-part of the LDPC PCM into a quasi-cyclic structure similar to the information-part of the matrix enabling parallel decoding. Note that Type A LDPC codes already form a quasi-cyclic structure in the LDPC encoding process, and hence, the parity interleaver is only used for the Type B LDPC codes.

The group-wise interleaving allows optimizing the combination between the FEC code and the constellation, and hence the group-wise interleaver is optimized for each ModCod combination. Since the group-wise interleaving is performed on a bit group basis, an LDPC codeword is first split into multiple bit groups consisting of 360 bits (i.e., parallel factor of the LDPC codeword or CPM size). Then, the bit groups are interleaved based on the equation in Section 6.2.2 of [3] and the interleaving order in Annex B of [3].

The group-wise interleaved codeword is then interleaved by the block interleaver. There are two block interleaver types: Type A and Type B, and the type used for each ModCod combination is defined in Section 6.2.3 of [3]. Note that some Type A FEC and constellation combinations use a Type A block interleaver while other Type A FEC and constellation combinations use a Type B block interleaver, and similarly for Type B FEC and constellation combinations, and Type A and Type B block interleavers. That is, Type A FEC and constellation combinations are not necessarily always paired with a Type A block interleaver, and Type B FEC and constellation combinations are not necessarily always paired with a Type B block interleaver. The block interleaving is performed using memory containers (i.e., columns for Type A block interleaver and rows for Type B block interleaver) with write/read operations. The Type A block interleaver uses the column-writing/row-reading operation, and the Type B block interleaver uses the row-writing/column-reading operation, as specified in Section 6.2.3 of [3].

The containers for both Type A and Type B block interleavers are divided into two parts: Part 1 and Part 2 as specified in Section 6.2.3 of [3]. The configurations for Part 1 and Part 2 block interleaving are determined based on each modulation format and code length. In the case of the Type A block interleaver, for example, if an LDPC code of length 64800 and 256QAM modulation are used, the numbers of rows of Part 1 and Part 2 are 7920 and 180, respectively. If an LDPC code of length 16200 and 256QAM modulation are used, the numbers of rows of Part 1 and Part 2 are 1800 and 225, respectively. Note that the number of columns for Type A block interleaver (N_c) is also determined by the modulation format. Note also that the numbers of rows and bits to be written in Part 1, N_{r1} and $N_{r1} \times N_c$, are multiples of 360 (= bit group size) and $360 \times N_C$, respectively. Consequently, the numbers of rows and bits to be written in Part 2 are easily obtained by $(\frac{N_{inner}}{N_c} - N_{r1})$ and $(N_{inner} - N_{r1} \times N_c)$, respectively. If $\left(\frac{N_{inner}}{N_c} - N_{r1}\right) > 0$, at least one bit group of 360 bits is interleaved in Part 2.

5.2.3 Constellation Mapping

There is one non-uniform constellation (NUC) defined for each ModCod combination (see Section 6.3 and Annex C of [3]). The interleaved LDPC codeword as the output bits of the block interleaver is mapped onto the defined constellation points. For implementation, a look-up table with the defined constellation mapping for each ModCod combination can be used.

5.2.4 Layered Division Multiplexing (LDM)

LDM is a power-based non-orthogonal-multiplexing technology using constellation superposition to combine multiple PLPs at different power levels. It provides a significant capacity increase compared with traditional TDM and FDM technologies when delivering multiple services with different quality of service (QoS) requirements over the same RF channel.

A block diagram of a two-layer LDM structure is shown in Figure 5.4 (see Section 6.4.1 of [3] for details). When LDM is used, it is recommended that Core PLP(s) belonging to the Core Layer (CL) use robust ModCod combinations intended for mobile, handheld, and indoor services (see Section 4.2.8.2 for recommended Core PLP ModCods). Enhanced PLP(s) belonging to the Enhanced Layer (EL) use equal or less robust ModCod combinations as specified in Section 6.4.1 of [3], and hence, it is recommended that Enhanced PLP(s) use ModCod combinations delivering higher capacity (i.e., less robust) intended for fixed services.

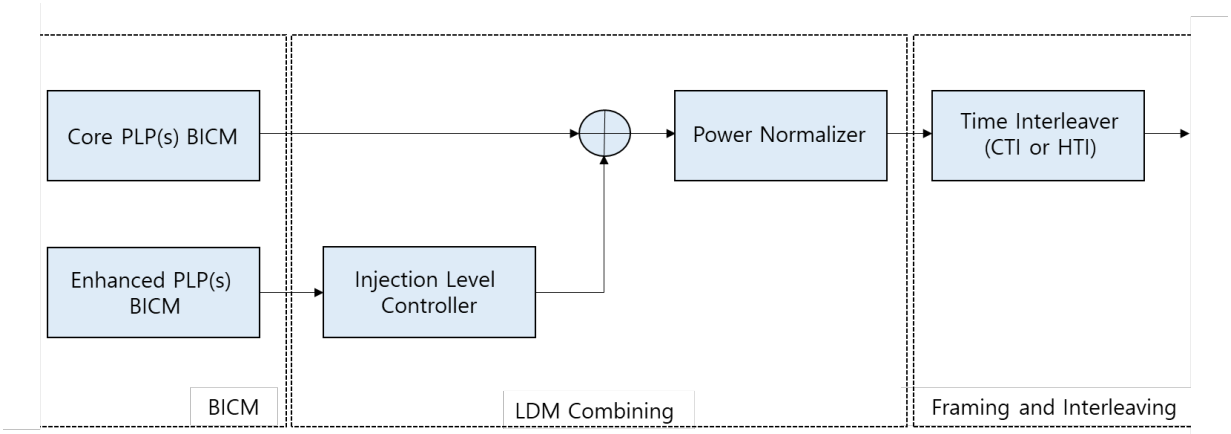

Figure 5.4 A two-layer LDM transmitter configuration.

The code length of the inner code may also be different for Core and Enhanced PLPs. As Core PLP(s) are targeted for mobile, pedestrian and indoor services, short codes (i.e., 16200 bits) may be used, resulting in simplicity and low power consumption in ATSC 3.0 receivers (see Section 7.2.1 for mobile guidelines). Enhanced PLP(s) for higher data rate fixed services may use long codes (64800 bits) for better performance. For simpler receiver complexity and less memory use, ATSC 3.0 is designed such that CL and EL share the same time and frequency interleavers as well as waveform parameters, including the pilot pattern, FFT size, and guard intervals.

The Enhanced PLP injection level relative to CL is selectable from 0 dB to 25 dB below the CL. Relative power distributions of the two layers as a function of the allowed injection levels are listed in Table 6.15 in [3]. The SNR performance of Core and Enhanced PLPs is highly related to the injection level. Smaller injection levels will bring the powers of the two layers closer. Under such conditions, while the Enhanced PLP will have higher transmission power for better SNR performance, it will cause more interference to the CL. Conversely, a higher injection level difference will reduce the Enhanced PLP power resulting in deteriorated SNR performance and reduced interference to the CL, making the Core PLP more robust. Therefore, it is recommended that the injection level be selected based on the QoS requirements of the Core and Enhanced PLPs services and a balanced performance.

LDM uses the power normalizer to normalize the average power of the combined CL and EL signals to unity. In Table 6.16 of [3], the scaling and normalization factors as a function of the injection level are shown. As noted in the table, when the injection level of an Enhanced PLP becomes larger (i.e., less power for the Enhanced PLP), the scaling factor decreases and the normalizing factor increases in order to equate the average power of the combined signal to unity.

For L1 signaling, the Enhanced PLP injection level relative to that of the CL is indicated by the L1-Detail signaling field **L1D_plp_ldm_injection_level** in Table 9.8 in [3]. The signaling format of **L1D_plp_ldm_injection_level** is indicated in Table 9.22 in [3].

In the CTI mode, with LDM, the field **L1D_plp_CTI_fec_block_start** needs to be signaled separately for both Core and Enhanced PLPs, as specified in Section 9.3.9.1 of [3].

When LDM is used, **L1D_plp_type** is signaled only for Core PLPs and Enhanced PLPs will not have a specific non-dispersed or dispersed PLP type as described in Section 7.2.2.6 of [3]. Further implementation details and examples with respect to the LDM based framing and PLP multiplexing are described in Section 5.3.3.

5.2.5 Protection for L1-Signaling

ATSC 3.0 L1 protection performs zero padding for the LDPC information part and puncturing for the LDPC parity part with a set of optimized patterns in addition to parity repetition according to the protection mode. Additional parity can be optionally used to further enhance time diversity of L1-Detail.

5.2.5.1 Common Block for L1-Basic and L1-Detail

The signaling information in L1-Basic and L1-Detail signaling is protected by concatenation of the BCH outer code and LDPC inner code. L1-Basic and L1-Detail are first scrambled and then BCH-encoded, as described in Sections 6.5.2.2 and 6.5.2.3 of [3]. Note that the number of BCH parity-check bits for BCH encoding is fixed at 168 bits.

The concatenated signaling and BCH parity-check bits are further protected by a shortened and punctured 16K LDPC code. When the length of the concatenated signaling and BCH parity-check bits is smaller than that of LDPC information bits K_{ldpc}, zero-padding (i.e., shortening of LDPC information bits) is performed to fill the rest of K_{ldpc} LDPC information bits. Since the zero-padding operation is performed on bit-group basis, bit-grouping of LDPC information bits is preceded, as described in 6.5.2.4 of [3]. Note that the number of bit-groups is different according to LDPC codes.

The coding performance of a shortened and punctured LDPC code depends on the number of punctured bits to the number of shortened bits ratio (PSR). Moreover, the optimum zero-padding patterns are different according to the PSR. Table 6.20 of [3] shows that the optimum pattern for zero-padding is determined based on the L1 protection mode corresponding to each PSR, as defined in Table 4.9 of Section 4.2.10. Since each LDPC code rate and PSR for L1-Basic and L1-Detail is different, the optimum shortening patterns for L1-Basic and L1-Detail are also different, as shown in Table 6.20 of [3].

After the zero-padding operation, LDPC encoding is performed, followed by the parity permutation. The parity permutation is performed only for the parity part of an LDPC codeword. The purpose of the parity permutation is to easily support an optimum puncturing of LDPC parity bits for each L1 protection mode. Note that for L1-Detail Mode 3 to Mode 7, parity interleaving is performed before the parity permutation. The parity interleaving enables the parity part of the LDPC parity-check matrix to have the same structure as the information part, i.e., almost quasi-cyclic structure with parallel factor 360. Due to this quasi-cyclic structure, the parity permutation is performed on bit-group basis to provide stable and good coding performance.

After the parity permutation, in the case of L1-Basic Mode 1 and L1-Detail Mode 1 only, the repetition of parity is additionally performed before parity puncturing, as described in Section 6.5.2.7 of [3]. The number of repeated bits is determined based on the number of the concatenated signaling and BCH parity-check bits. In the case of L1-Basic Mode 1 and L1-Detail Mode 1, the

repetition of parity bits allows enhanced decoding performance due to the parity-check matrix structure of the 16K LDPC code of rate 3/15.

After the parity permutation and repetition, portions of LDPC parity bits are punctured to provide desired robustness while minimizing unnecessary overhead. Note that these punctured bits are not transmitted. The number of punctured bits is determined based on PSR and a constant integer corresponding to the minimum temporary size of puncturing bits for each L1 protection mode. Once the final number of punctured bits N_{punc} is determined, the last N_{punc} bits of the entire LDPC codeword with parity permutation and repetition are punctured, as described in Section 6.5.2.8 of [3]. Finally, the ATSC 3.0 transmitter transmits the final LDPC codeword excluding zero-padding bits and puncturing bits, as described in Section 6.5.2.9 of [3].

5.2.5.2 L1-Detail Specific Block: L1 Segmentation

The number of signaling information bits in L1-Detail is variable and depends mainly on the number of subframes and PLPs. Therefore, one or more FEC Frames may be required for transmission of the total amount of signaling information in L1-Detail. In ATSC 3.0, when the size of L1-Detail signaling exceeds a threshold number K_{seg}, the L1-Detail signaling is segmented into multiple smaller blocks. Furthermore, each segmented block is protected by concatenation of BCH outer and LDPC inner codes, and the coded bits are mapped to appropriate modulation symbols, as described in Sections 6.5.2 of [3]. Note that in the case of L1-Detail Mode 1, the repetition of portions of parity bits is used before parity puncturing after LDPC encoding of each segmented block.

K_{seg} is determined based on the LDPC code rate and PSR which affect coding performance. In Table 6.25 of [3], L1-Detail Mode 1 and Mode 2 use the rate 3/15 LDPC code, and L1-Detail Mode 3 to Mode 7 use the rate 6/15 LDPC code. Therefore, K_{seg} has different values for each LDPC code. Despite the use of the same LDPC code for L1-Detail Mode 1 and Mode 2, the reason why their values for K_{seg} are different is that Mode 1 and Mode 2 have different targeted robustness and PSR. As shown in Table 4.9, the PSR for L1-Detail Mode 1 is 7/2, while that for L1-Detail Mode 2 is 2.

5.2.5.3 L1-Detail Specific Block: Additional Parity

Additional parity may be used when the performance of L1-Detail under mobile or indoor conditions needs to be improved. This method basically adds time diversity to L1-Detail by including a certain number of additional parity bits in the previous frame as specified in Section 6.5.3.2 of [3]. The ratio of additional parity bits to be included in the previous frame is either 1, 1/2, or 0 as signaled in L1-Basic (**L1B_L1_Detail_additional_parity_mode**), and the amount of bits may be configured depending on the channel conditions.

The additional parity bits are selected among the punctured bits in the current frame in order to exploit not only diversity gain but also coding gain. In ATSC 3.0, the number of additional parity bits is defined to be greater than that of punctured bits. As specified in Section 6.5.3.2 of [3], the repeated bits are selected when the number of additional parity bits exceeds the number of punctured bits.

For the implementation aspect, additional parity for L1-Detail Mode 1, 2, and 3 has relatively low hardware complexity in the receiver. Conversely, additional parity for L1-Detail Mode 4, 5, 6, and 7 will require more hardware complexity in the receiver compared to L1-Detail Mode 1, 2, and 3. Therefore, when additional parity is used, it is recommended that L1-Detail Mode 1, 2, or 3 be used. Note that receivers can ignore any transmitted additional parity bits in situations where

those additional parity bits are not required (e.g., the receiver knows that it is not at the edge of the broadcast area based on the measured SINR).

5.3 Framing and Interleaving

5.3.1 Time Interleaving

5.3.1.1 Convolutional Time Interleaving (CTI)

It is recommended that the CTI mode be always used for single Core PLP operation with constant cell rate. The CTI operation provides twice the time interleaving depth of a block interleaver, and for the single Core PLP case where both the HTI and CTI modes are allowed, it is recommended that the CTI mode be always used. The CTI mode is also more efficient because insertion of dummy modulation values to obtain an integer number of FEC Blocks per subframe is not required. Further details of the valid conditions for the CTI mode are described in Section 4.2.11.1.

It is recommended that the maximum CTI depth (1024 rows) be used. Unless the small savings in latency are critical to the broadcaster operation, the maximum CTI depth provides the best performance. ATSC 3.0 receivers should be able to support the maximum time interleaving depth, and therefore, the choice of smaller depths of the CTI from transmitters is not desired.

5.3.1.2 Hybrid Time Interleaving (HTI)

The hybrid time interleaver is composed of a cell interleaver, a twisted block interleaver (TBI), and a convolutional delay line (CDL) (see Section 7.1.5 of [3]). Different from the CTI mode, the HTI mode can support all frame configurations regardless of the number of PLPs and a delivered product type, i.e., a particular complete delivered product and a complete delivered product as described in Section 4.2.11. The following are the key properties of HTI component blocks:

- In principle, the HTI optimally interleaves subframes by accomplishing an intra-subframe interleaving (inner interleaving) and an inter-subframe interleaving (outer interleaving). The role of the cell interleaver is to interleave the cells (output from mapper block) within a single FEC Block. The TBI for the intra-subframe interleaving interleaves FEC blocks in a single TI Block consisting of FEC Blocks (cell-interleaved FEC Blocks or non-cell-interleaved FEC Blocks). The CDL for the inter-subframe interleaving interleaves TI Blocks consisting of the output TI Blocks of the TBI.
- In HTI mode, the TBI is always used while the cell interleaver and CDL are each optionally used. The use of the cell interleaver and CDL is signaled by **L1D_plp_HTI_cell_interleaver** and **L1D_plp_HTI_inter_subframe**, respectively. The cell interleaver and/or CDL may be used to further enhance the robustness to burst errors in a severe block/multipath fading environment.
- The cell interleaver takes input cells in FEC Blocks and arranges them into TI Blocks, each of which contains one or more FEC Blocks. Cell interleaving is accomplished via a linear writing operation and pseudo-random reading operation.
- In the TBI, the FEC Blocks are serially written column-wise into a TBI memory and then cells are read out diagonal-wise. This diagonal-wise reading operation increases the interleaving depth (or robustness to a burst error) compared to a conventional row-wise reading operation. In particular, note that any virtual FEC Blocks to support a non-constant cell-rate PLP are located ahead of data FEC Blocks in order to guarantee the twisted block de-interleaving (TBDI) in a given single-memory at a receiver (see Section 6.3.3.2.2). The concept of virtual FEC Blocks is similarly applied to the CDL operation.

- The CDL is a kind of a convolutional interleaver as used in the CTI mode. Different from the CTI, a FIFO register in each branch of the CDL delays a group of cells. The key role of the CDL is to spread FEC Blocks within a TI Block over multiple subframes after splitting a TI Block into interleaving units (IUs). Note that when the CDL is enabled by **L1D_plp_HTI_inter_subframe**=1, the field **L1D_plp_HTI_num_ti_blocks** denotes the number of subframes used for the interleaving while the number of TI Blocks is set to 1. This will in general consume a large TI memory, and therefore, each PLP size within a subframe will necessarily be limited in order to meet the TI memory requirement (2^{19} cells). Another constraint when using the CDL is that when a complete delivered product is composed of multiple PLPs, the normative condition is that all the PLPs use the same IU in order to have the same decoding time.

5.3.2 Frame Structure

There are many possible combinations of frame structure in ATSC 3.0. This section provides some recommendations and examples of frame structure combinations in the aspects of power saving and performance.

It is recommended that the Preamble and the first subframe meet the following relationship $Tu_p/Dx_p \geq Tu_0/Dx_0$, where Tu_p is the useful symbol time of the Preamble, Dx_p is the preamble pilot spacing in the frequency direction of the Preamble, Tu_0 is the useful symbol time of the first subframe, and Dx_0 is the unit of scattered pilot spacing in the frequency direction of the first subframe. The time Tu/Dx is the maximum interpolation length that receivers can cope with when using both time and frequency interpolation. Note that this is always longer than when only frequency interpolation is used i.e., $Tu/(DxDy)$, and hence, the frequency-only interpolation case is not considered further here.

It is recommended that when multiple subframes are used, the order of the subframes are arranged in order with the relationship $Tu_i/Dx_i \geq Tu_{i+1}/Dx_{i+1}$ maintained where *i* represents the *i*th subframe, and where the useful symbol time of that subframe is Tu_i, and the unit of scattered pilot spacing in the frequency direction of that subframe is Dx_i. In most cases, this works out to keeping the order of the subframes as

- First 8K FFT subframe(s), followed by 16K FFT subframe(s), followed by 32K FFT subframe(s), and
- Within each FFT size grouping, the order of the subframes should be based on the unit of scattered pilot spacing in the frequency direction D_x being from most dense to least dense (that is, from the smallest D_x to the largest D_x).

It is recommended that the number of subframes be as low as possible in order to increase efficiency by reducing the need for dummy cells and subframe boundary symbols. This also enables receivers to do simple intermittent operation for power saving and to improve interpolation at subframe boundaries for performance improvement.

5.3.2.1 Example Scenario for Power Saving Aspect

Figure 5.5 shows an example scenario including ATSC 3.0 mobile services that are restricted to using lower FFT sizes (see Section 7.4.2). As shown in Figure 5.5 (*a*), if the order of the subframes is random, ATSC 3.0 mobile receivers that only receive lower FFT sizes will switch on and off several times, which wastes power on the mobile receivers. Note that receivers cannot switch on exactly at the beginning of a subframe but in general should start some number of symbols before the start. Conversely, if the subframes are ordered as shown in Figure 5.5 (*b*), mobile receivers can

stay in a power-on state to receive mobile subframes after the bootstrap and Preamble and then switch off for the rest of the frame, which allows simpler operation and lower power consumption.

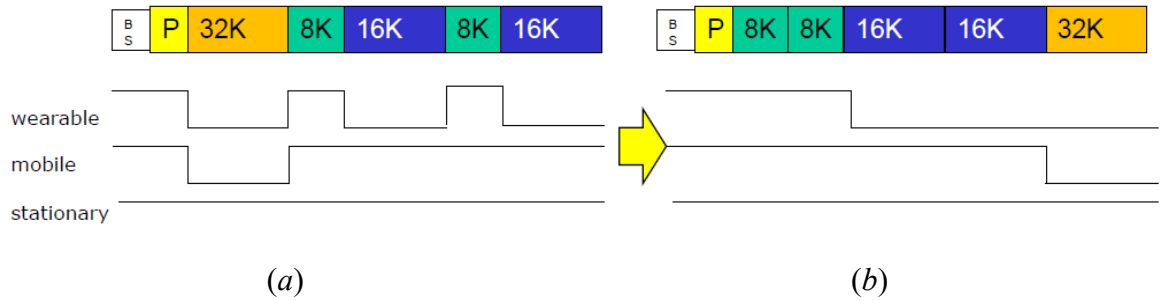

Figure 5.5 Grouping of FFT sizes: (*a*) Not recommended, random order, (*b*) Recommended, in 8K, 16K and 32K order.

5.3.2.2 Example Scenario for Performance Aspect

Example scenarios for the performance improvement aspect are shown from Figure 5.6 to Figure 5.12. In these figures, blue dots represent scattered pilots, and orange dots represent SBS pilots. The arrows represent how interpolation is performed.

(1) Dense to Sparse, Integer Multiples of D_x (Recommended Order)

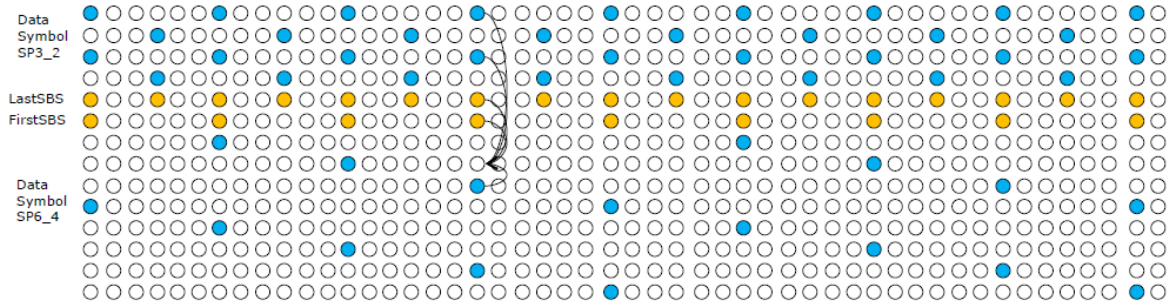

Figure 5.6 Example at subframe boundary from SP3_2 to SP6_4.

As shown in Figure 5.6, for integer multiples of D_x and dense to sparse pilot subframe order, at the subframe boundary, the previous subframe's pilots can be used when estimating the channel for the sparse subframe, which improves the performance.

(2) Dense to Sparse, Non-Integer Multiples of D_x (Less Recommended Order)

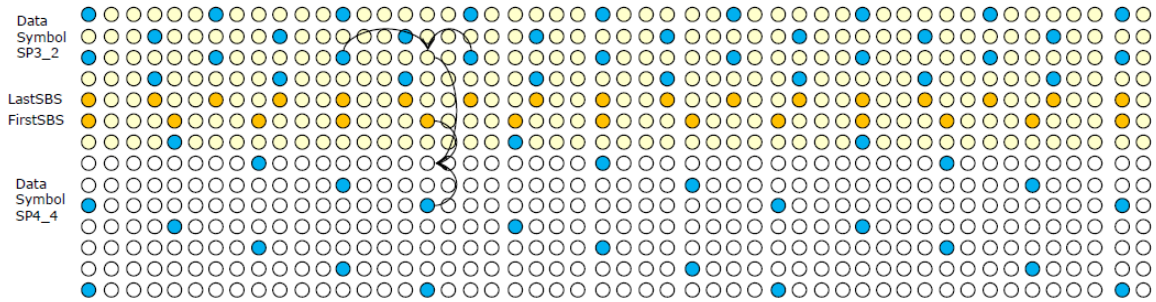

Figure 5.7 Example at subframe boundary from SP3_2 to SP4_4.

As shown in Figure 5.7, for non-integer multiples of D_x and dense to sparse pilot subframe order, the previous subframe's frequency-interpolated carriers can be reused when estimating the channel for the sparse subframe, which improves the performance. This is not as accurate as using the known pilots as shown in (1), but still can help to improve the time interpolation of the sparse subframe. Therefore, (1) is preferred over (2).

(3) Sparse to Dense, Integer Multiples of D_x (Not Recommended)

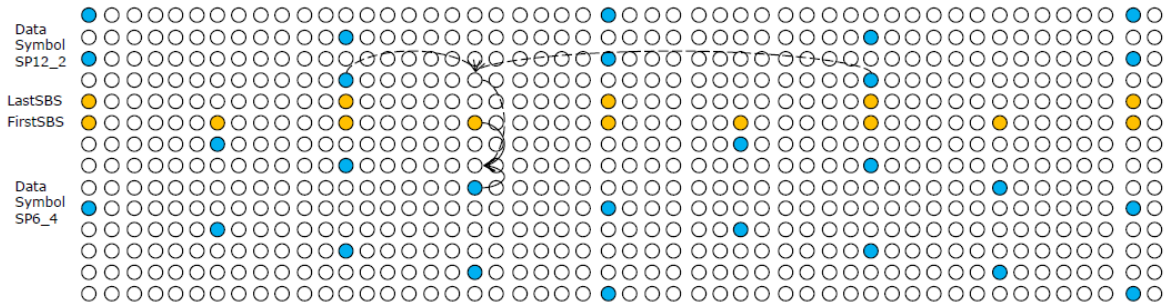

Figure 5.8 Example at subframe boundary from SP12_2 to SP6_4.

As shown in Figure 5.8, for integer multiples of D_x and sparse to dense pilot subframe order, the previous subframe's frequency-interpolated carriers cannot be used to accurately estimate the channel for the dense pilot subframe, leading to worse performance. The time-interpolation performance of the first D_y symbols may be poor, since it is not possible to separate out the channel effects from noise in a time/frequency varying channel. This can be seen in the following example:

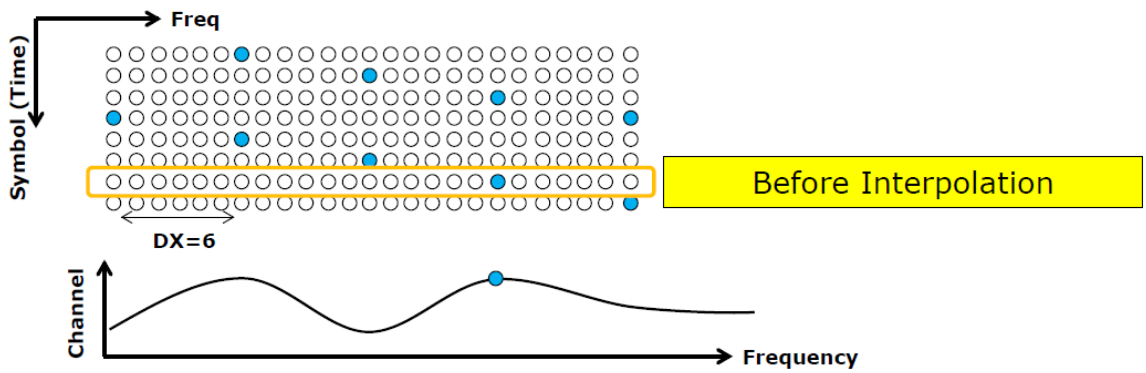

Figure 5.9 Example showing channel at a receiver before interpolation with scattered pilots $D_x = 6$ and $D_y = 4$.

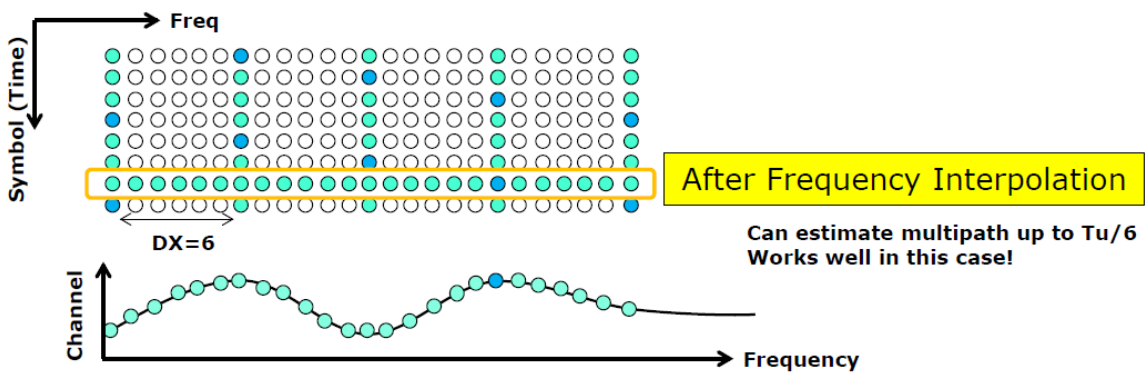

Figure 5.10 Example at a receiver showing that after time and frequency interpolation the channel can be well estimated up to Tu/6.

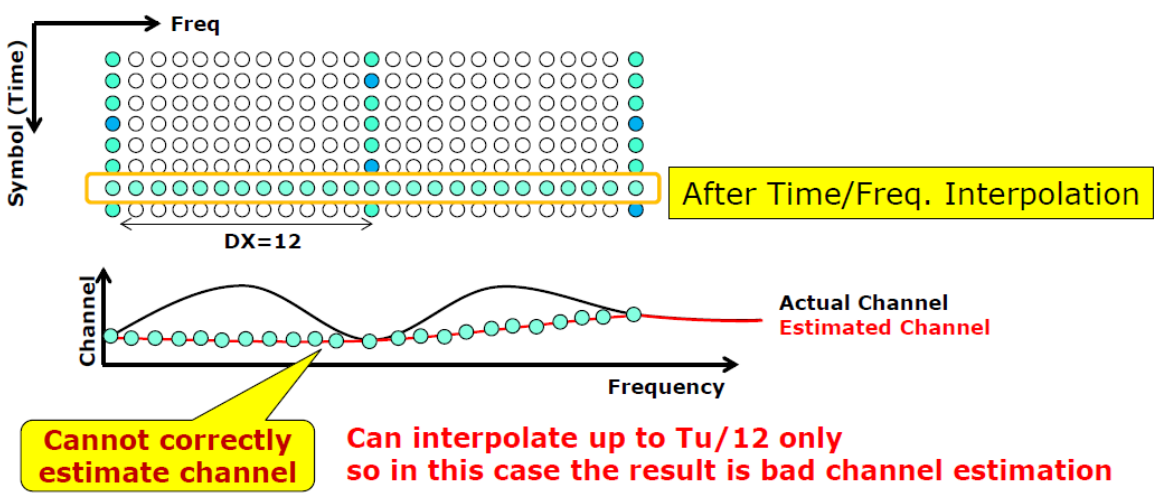

Figure 5.11 Example at a receiver with $D_x = 12$ and $D_y = 4$ showing that the receiver cannot estimate the channel well in this case, since the variation is greater than Tu/12 which is the maximum that can be interpolated.

It can be seen that using the interpolated results of a sparse pilot subframe (D_x=12 in the example) can lead to incorrect interpolation results in a channel with strong multipath. Therefore, if the strong multipath channel is estimated using a sparse pilot subframe first, the result cannot be used to improve the channel estimate for the dense pilot subframe as the result may be even worse than beginning channel estimation from the subframe boundary itself.

(4) Sparse to Dense, Non-Integer Multiples of D_x (Not Recommended)

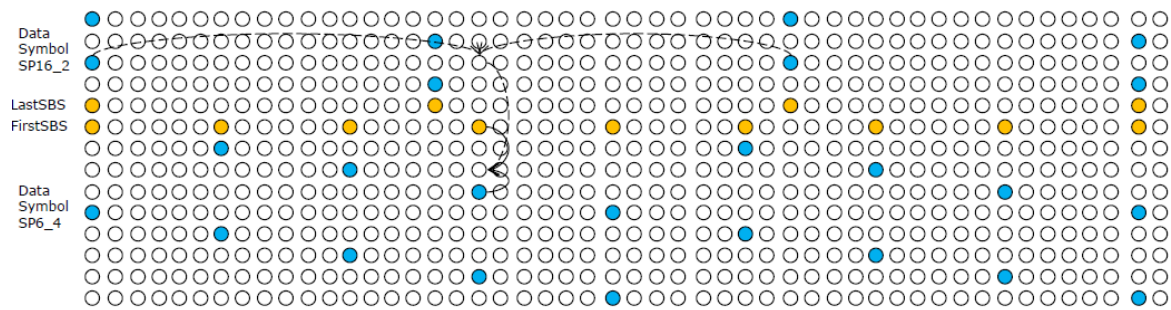

Figure 5.12 Example at subframe boundary from SP16_2 to SP6_4.

As shown in Figure 5.12, for non-integer multiples of D_x and sparse to dense pilot subframe order, the previous subframe's frequency-interpolated carriers and pilots cannot be used to accurately estimate the channel for the dense pilot subframe, leading to worse performance. The time-interpolation performance of the first D_y symbols may be poor, since it is not possible to separate out channel effects from noise in time/frequency varying channel.

5.3.3 LDM and PLP Multiplexing

ATSC 3.0 allows up to two-layer LDM that can be combined with TDM and/or FDM configurations in order to configure multiple PLPs within a subframe. When LDM is configured with more than 2 PLPs, it can be formed as one of the configurations such as Layered Time Division Multiplexing (LTDM), Time Layered Division Multiplexing (TLDM), Layered Frequency Division Multiplexing (LFDM), or Frequency Layered Division Multiplexing (FLDM).

For the implementation to have low complexity and low memory use, A/322 [3] specifies that the time interleaver block including the memory is shared by both Core and Enhanced PLPs, and due to this transmission system architecture, there are certain cases that are allowed or disallowed by A/322 when LDM is combined with TDM or FDM. Note that the example configurations in Section 5.3.3.1 to Section 5.3.3.5 are assumed to be TLDM or LTDM configurations. FLDM or LFDM configurations can be regarded as special cases of TLDM or LTDM respectively, by subslicing dispersed Core PLPs as described in Section 5.3.3.6. Note also that the example configurations in this section show up to 4 PLPs that may be assembled as a complete delivered product requiring simultaneous recovery in receivers according to the constraint in Section 5.1.1 of [3].

5.3.3.1 Definition of L1D_plp_start and L1D_plp_size

L1D_plp_start and **L1D_plp_size** are the signaling fields that determine the position and placement of each PLP within a subframe. Regardless of the use of LDM, these signaling fields of Core PLP(s) are defined with respect to after time interleaving (CTI or HTI) and framing. However, when LDM is used, **L1D_plp_start** and **L1D_plp_size** of Enhanced PLP(s) are defined with respect to before time interleaving because the LDM combining is performed after constellation mapping and before time

interleaving as described in Section 6.4.1 of [3]. Further details of the signaling definitions of **L1D_plp_start** and **L1D_plp_size** for Core and Enhanced PLPs are illustrated in Figure 5.13. As presented in the figure, those cells indicated by the signaling fields of Enhanced PLPs are spread over a subframe after time interleaving, which will not be meaningful information at the framing stage. In receivers, Enhanced PLPs decoded after time de-interleaving and cancellation of Core PLP will be recovered based on the corresponding **L1D_plp_start** and **L1D_plp_size** of the Enhanced PLPs in L1-Detail.

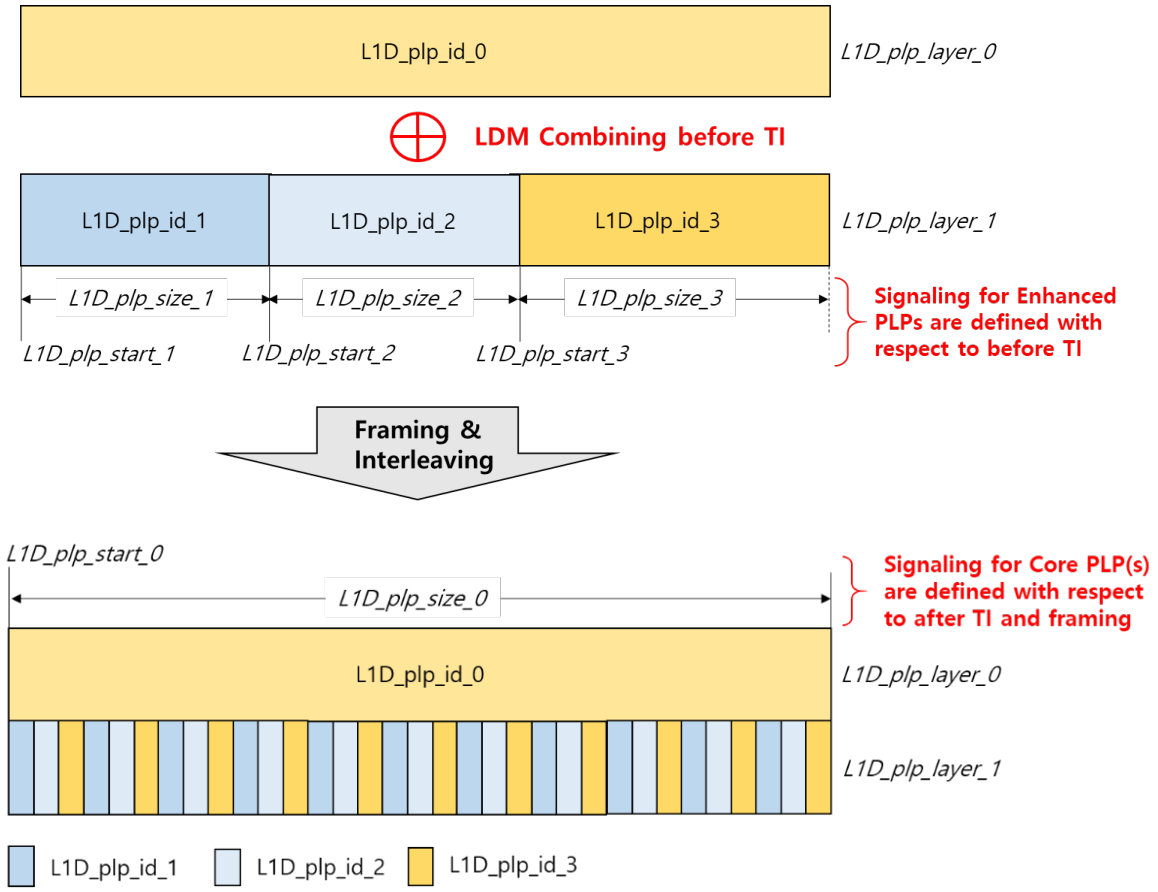

Figure 5.13 **L1D_plp_start** and **L1D_plp_size** definitions for Core and Enhanced PLPs.

5.3.3.2 Indexing TI Groups

When LDM is configured with multiple Core PLPs, each Core PLP represents a time interleaving group (TI Group) as defined in Section 7.2.7.4 of [3] and as shown in the example of Figure 5.14. These TI Groups are implicitly indexed in ascending order according to the order in which the corresponding Core PLPs appear within the L1-Detail control signaling. For the efficient use of receiver memory, it is strongly recommended that the Core PLPs for each subframe be arranged within that subframe in ascending order. That is, **L1D_plp_start** of the first Core PLP in a subframe's L1-Detail control signaling indexed as TI_Group_0 should have the lowest cell index of the **L1D_plp_start** values for all Core PLPs within that subframe. Then, **L1D_plp_start** of the second Core PLP in that subframe's L1-Detail control signaling indexed as TI_Group_1 should be larger than that of the first Core PLP (TI_Group_0) and smaller than that of the third Core PLP (TI_Group_2),

and so on. Note that **L1D_plp_id** of each PLP is independent of the TI_Group values, so that it is not necessary for **L1D_plp_id** values to be arranged in ascending order within the L1-Detail control signaling (as in Figure 5.14).

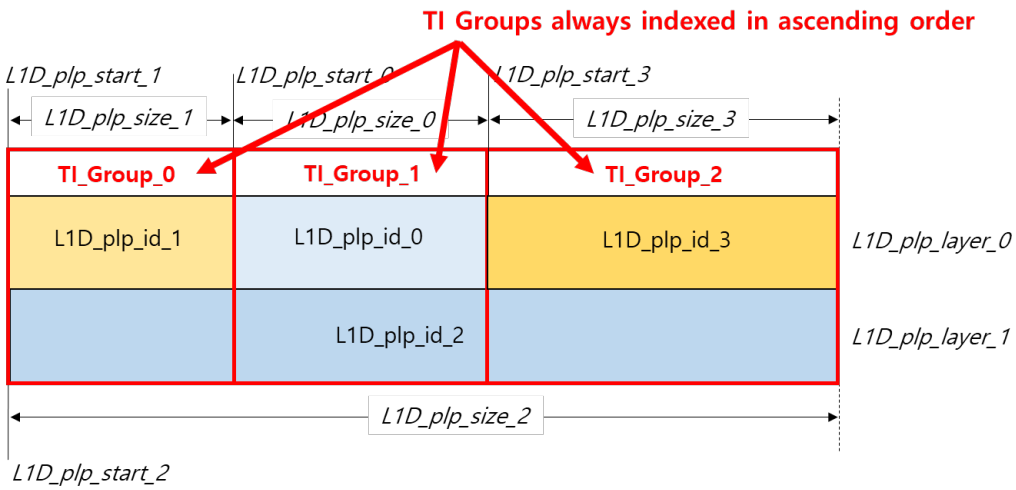

Figure 5.14 TI Group assignment for multiple Core PLPs.

5.3.3.3 Injection Level (**L1D_plp_ldm_injection_level**) for Multiple Enhanced PLPs

When multiple Enhanced PLPs are associated with a single Core PLP (e.g., L1D_plp_id_0) as shown in the example of Figure 5.15, those Enhanced PLPs should all have the same LDM injection level as signaled by **L1D_plp_ldm_injection_level**. Note that if those Enhanced PLPs have different injection levels, the portions of the Core PLP associated with the two different injection levels (e.g., L1D_plp_id_0 of Figure 5.15) will have different robustness, which will eventually provide different coverage areas for a single content.

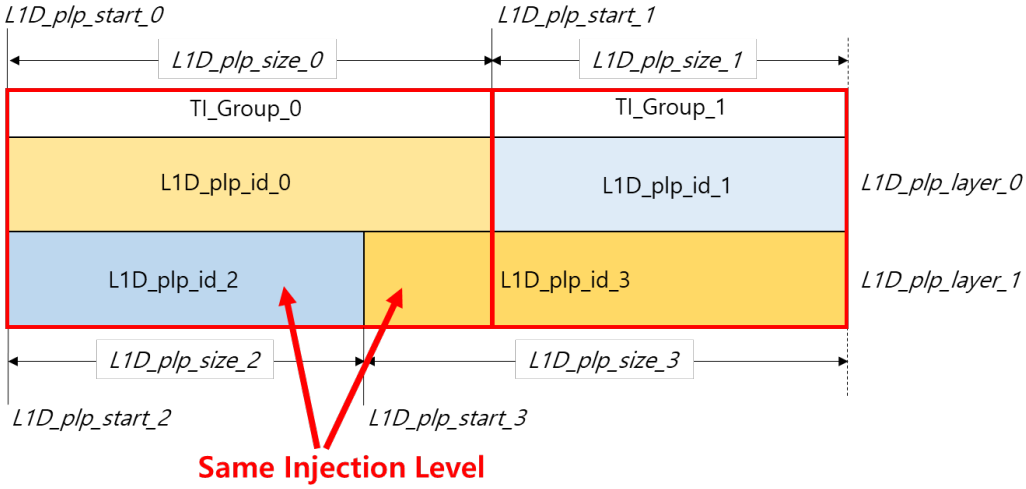

Figure 5.15 Two Enhanced PLPs injected into a single Core PLP.

5.3.3.4 Positioning Enhanced PLP(s) and Not Recommended LDM Cases

When Enhanced PLP(s) are positioned and associated with Core PLP(s) within a subframe, **L1D_plp_start** and **L1D_plp_size** of Enhanced PLP(s) should be carefully determined such that cells of Enhanced PLP(s) are continuously associated with cells of Core PLP(s). Figure 5.16 is an example case that is not recommended as an LDM configuration. As shown in the example, **L1D_plp_start** of the Enhanced PLP (L1D_plp_id_1) is not the same as that of the Core PLP (L1D_plp_id_0). For the same reason as described in Section 5.3.3.3, portions of the Core PLP in this configuration will have different robustness, and therefore, such a configuration is not recommended. Furthermore, **L1D_plp_size** of the Enhanced PLP should not be either shorter or longer than the Core PLP. Note that when the HTI mode is used, all the PLPs configured by the HTI mode use integer numbers of FEC Blocks. In such cases, the number of actual data cells in the Enhanced Layer is only allowed to be less than or equal to those in the Core Layer, and Enhanced Layer dummy modulation values described in Section 7.2.7.4.6 of [3] are used when necessary. More details of the Enhanced Layer dummy modulation values are described in Section 5.3.3.5.2.

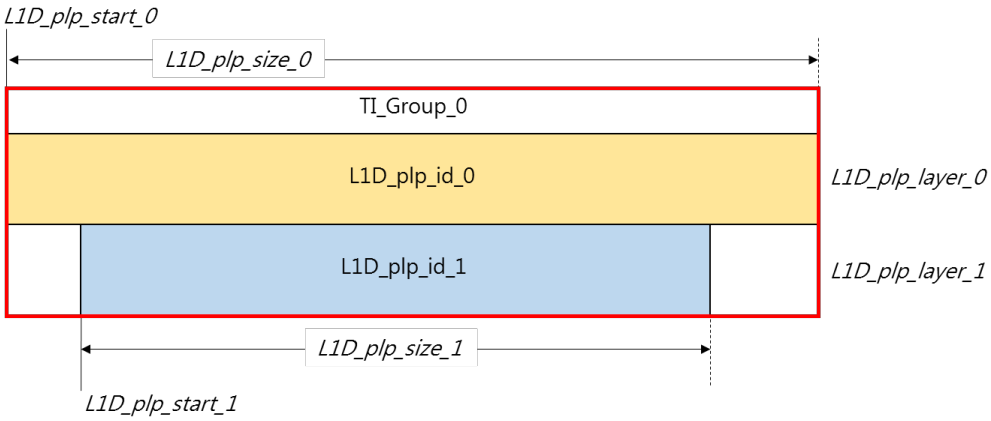

Figure 5.16 Not recommended LDM configuration example #1.

Figure 5.17 is another example case that is not recommended as an LDM configuration. When multiple PLPs are configured in Enhanced Layer, **L1D_plp_start** of the following Enhanced PLP in a subframe (i.e., L1D_plp_start_3 of Figure 5.17) should correspond to the immediately following cell of the preceding Enhanced PLP (i.e., L1D_plp_id_2), to prevent gaps between Enhanced PLPs.

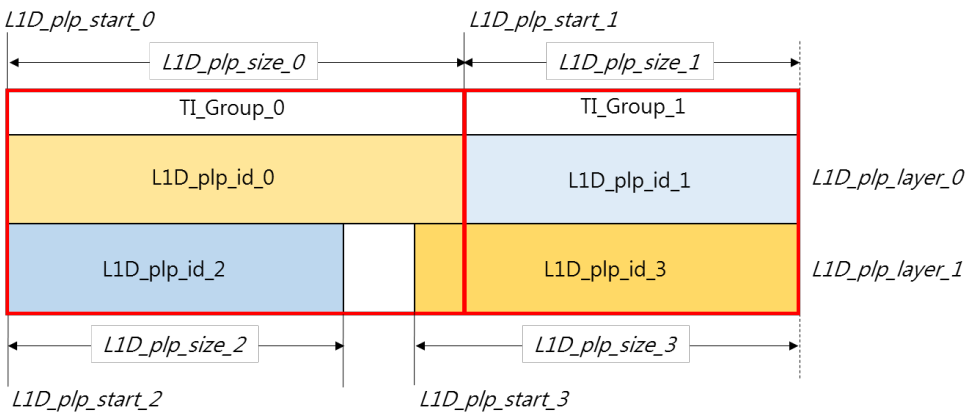

Figure 5.17 Not recommended LDM configuration example #2.

5.3.3.5 LDM Configuration with Different TI Modes

5.3.3.5.1 CTI Mode

The CTI mode may be used when a complete delivered product consists of a single Core PLP that has a constant cell rate. A simple LDM example described in Section 7.2.7.4.1 of [3] is one of the common service scenarios using the CTI mode, as both Core and Enhanced PLPs of such simple LDM have the same **L1D_plp_start** and **L1D_plp_size** with a constant cell rate. When the CTI mode is used, it is not necessary to achieve an integer number of FEC Blocks per subframe so it is therefore recommended that the data cells in both the Core and Enhanced Layers be completely filled in a subframe.

Figure 5.18 is another LDM configuration that is allowed by the CTI mode. When a subframe consists of multiple Core PLPs (i.e., multiple TI Groups) representing different delivered products as shown in this figure, each PLP within that subframe should have a constant cell rate to be configured with the CTI mode. Furthermore, any Enhanced PLPs associated with Core PLPs should not be spread over multiple TI Groups as exemplified in Figure 5.18. Note that in the CTI mode, an incomplete FEC Block at the last part of a Core PLP is time interleaved and it should be buffered until the next subframe to be decoded in a receiver. In such a case, the FEC Block of the Enhanced PLP associated with that incomplete FEC Block of the Core PLP should be also buffered until the next subframe. This implies that when an Enhanced PLP is spread over multiple TI Groups, any subsequent FEC Blocks of that Enhanced PLP associated with the following TI Group(s) should be buffered as well, which will consume a large amount of memory in a receiver. Due to this reason, the example configurations shown in Figure 5.14 and Figure 5.15, in which an Enhanced PLP is spread over multiple TI Groups, are not allowed when the CTI is used. Note also that in the CTI mode, when an LDM configuration as shown in Figure 5.18 is used, building a complete delivered product composed of PLPs belonging to different TI Groups is not allowed as specified in Section 7.1.1 of [3]. A complete delivered product should be composed of the Core and Enhanced PLPs within the same TI Group when the CTI mode is used.

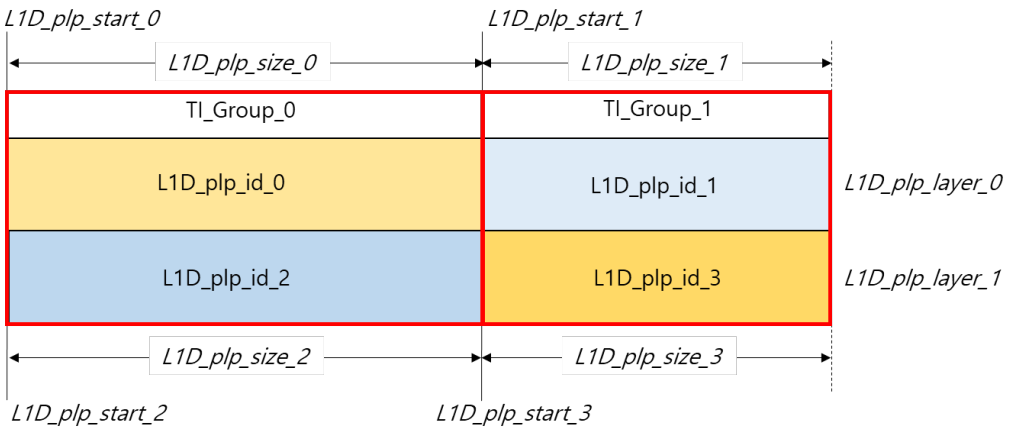

Figure 5.18 Allowed LDM configuration example #1 using the CTI mode.

Figure 5.19 is another LDM case allowed by the CTI mode. As shown in this figure, a Core PLP that is associated with multiple Enhanced PLPs may be configured with a single CTI. When such a configuration is used, it is required that each of the Enhanced PLPs associated with the Core PLP also have a constant cell rate. This implies that **L1D_plp_start** and **L1D_plp_size** of each Enhanced PLP should remain constant from one subframe to the next subframe, so that receivers do not need to track these signaling values of the previous subframes.

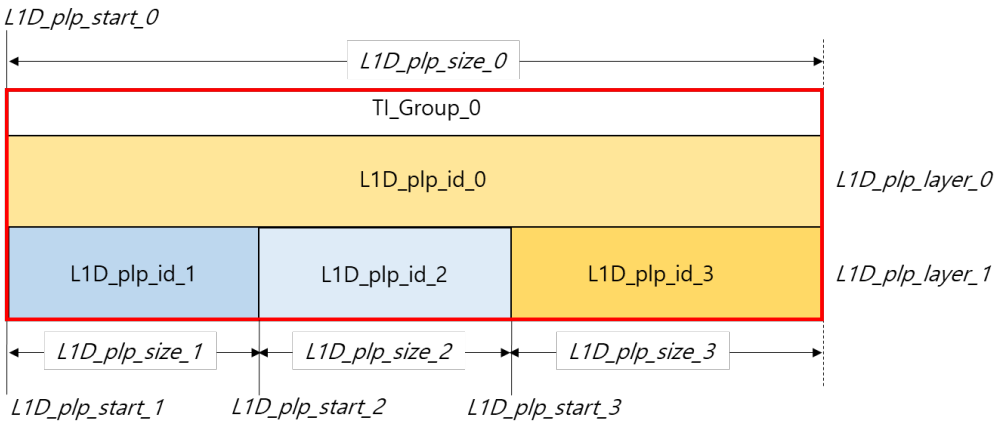

Figure 5.19 Allowed LDM configuration example #2 using the CTI mode.

5.3.3.5.2 HTI Mode

When the HTI mode is used, each of the Core and Enhanced PLPs is composed of an integer number of FEC Blocks, and therefore, the case that an Enhanced PLP is spread over multiple TI Groups can be implemented without further receiver memory requirement. Figure 5.20 is an allowed LDM configuration when two Core PLPs associated with an Enhanced PLP are configured with the HTI mode. The 4-PLP configurations as shown in Figure 5.14 and Figure 5.15 are also cases where an Enhanced PLP is spread over multiple TI Groups, which is allowed by the HTI mode. Note that using different TI modes for different TI Groups associated with the same Enhanced PLP is not allowed. Note also that when the HTI mode is used in those cases that the same Enhanced PLP is spread over multiple TI Groups, the use of the Convolutional Delay Line (CDL) is not allowed. If the CDL were used, the two Core PLPs might have different interleaving

depths over multiple subframes, and therefore, the portions of the single Enhanced PLP associated with those two Core PLPs might have different decoding times, which eventually would require a large amount of memory in receivers.

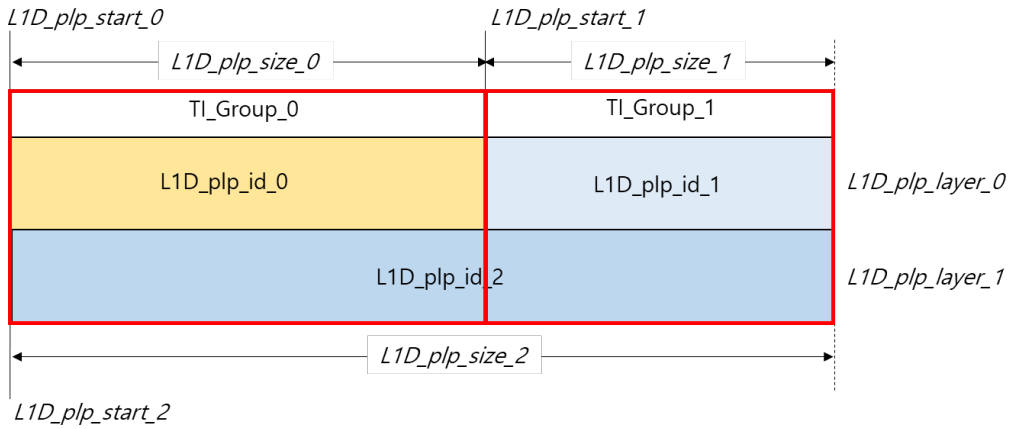

Figure 5.20 Allowed LDM configuration example using the HTI mode.

Since the HTI mode requires an integer number of FEC Blocks for each PLP, the total number of cells of the Core PLP(s) may be different from that of the Enhanced PLP(s) depending on the ModCod choice for each PLP. In such cases, Enhanced Layer dummy modulation values as described in Section 7.2.7.4.6 of [3] are used, so that CL and EL have the same number of cells. Note that these Enhanced Layer dummy modulation values are only allowed to be inserted after the last Enhanced PLP in a 'PLP group' (defined in Section 7.2.7.4.6 of [3]). Furthermore, these dummy modulation values have the same constellation mapping and injection level as the preceding Enhanced PLP, so that the Core PLP associated with the preceding Enhanced PLP and the dummy modulation values has a uniform robustness.

When an Enhanced PLP is spread over multiple TI Groups (i.e., HTI-based LTDM or LFDM configurations), an extra frame buffer may be needed when the TI depths of Core PLPs belonging to multiple TI Groups are substantially different. For example, if the TI depth of the first Core PLP is larger than that of the second Core PLP, the portion of the Enhanced PLP associated with the second Core PLP should be buffered due to the larger TI depth of the first Core PLP. Note that for buffering such a portion of the Enhanced PLP, an extra frame buffer (e.g., DRAM) can be used, rather than consuming the TI memory (e.g., SRAM). In order to avoid the extra frame buffer for Enhanced PLP, it is recommended that multiple Core PLPs associated with an Enhanced PLP use the same or similar TI depth as shown in Figure 5.21. This can be achieved when the number of FEC Blocks per TI Block is the same or similar for the multiple Core PLPs associated with an Enhanced PLP. This is recommended not only to avoid extra frame memory, but also to provide the same performance for a single Enhanced PLP.

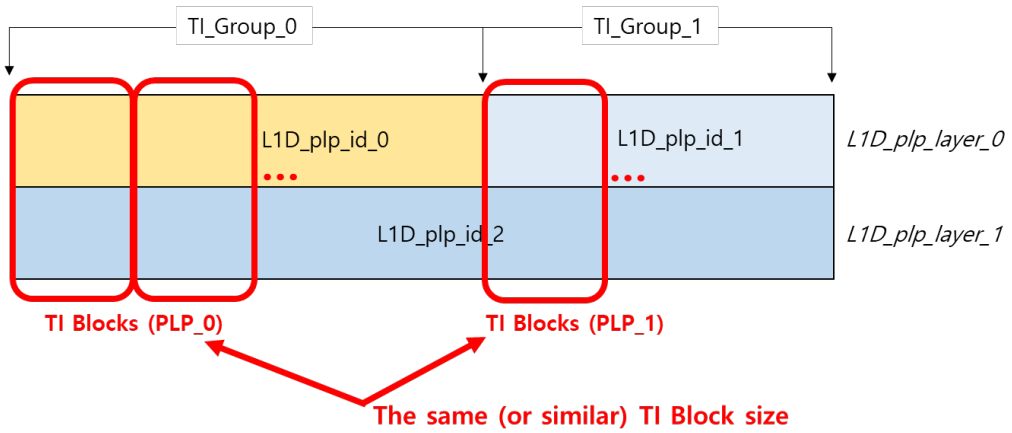

Figure 5.21 Recommended use of TI Blocks for HTI-based LTDM or LFDM configurations.

5.3.3.5.3 No TI Mode

The no TI mode is generally not recommended regardless of the use of LDM, even though it is allowed by the specification. Refer also to Section 4.2.11.2 which discusses the use of the no TI mode. When the no TI mode is used with LDM, those configurations described in Section 5.3.3.5.1 and Section 5.3.3.5.2 are all allowed. However, when an Enhanced PLP is spread over multiple TI Groups, the Core PLPs associated with that Enhanced PLP should have integer numbers of FEC Blocks due to the reason described in Section 5.3.3.5.1.

5.3.3.6 Combination with FDM

FDM, which is enabled by cell multiplexing methods described in Section 7.2.7 of [3], may be combined with LDM such as FLDM or LFDM configurations, and such cell multiplexing methods controlled by **L1D_plp_type**, **L1D_plp_subslice _interval**, and **L1D_plp_num_subslices** are applied with respect to Core PLP(s) only. An FLDM configuration is relatively simple because Core and Enhanced PLPs given the FDM cells have the same number of cells as shown in Figure 5.22. Note that when the cell multiplexing for FDM is applied, dummy modulation values described in Section 7.2.6.5 of [3] are required in many cases because the defined subslice interval (**L1D_plp_subslice _interval**) may not be an integer multiple of the total number of data cells. Note also that the cell multiplexing of FDM is generally not applied to the data part of the Preamble and SBS as specified in [3].

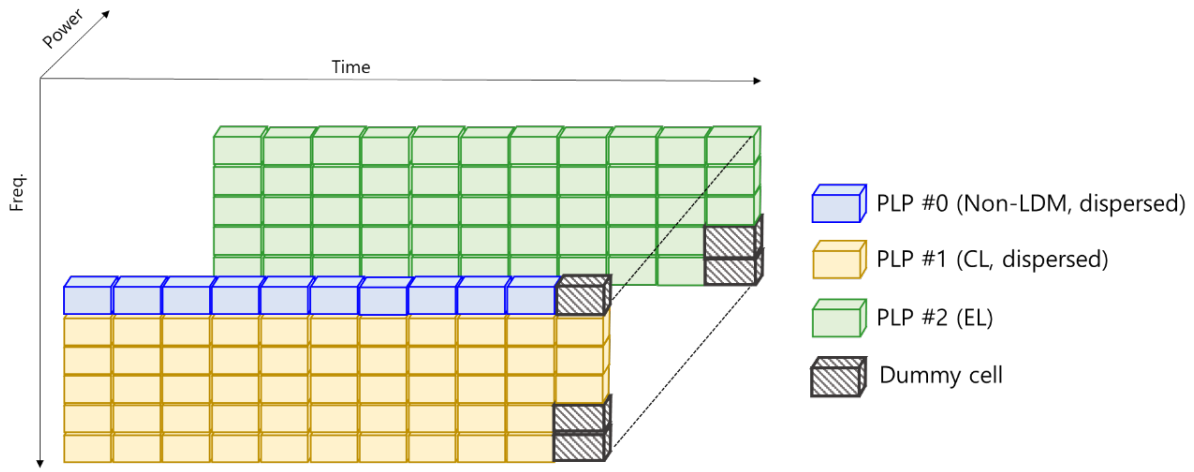

Figure 5.22 FLDM configuration example.

Another type of FDM combination with LDM is an LFDM configuration as shown in Figure 5.23. Since the subslicing parameters are only applied to Core PLPs, the cell writing order of the associated Enhanced PLPs follows the cell writing order of those Core PLPs as described in Figure 5.23. As shown in this LFDM example, when the cell multiplexing for FDM requires a PLP to span the whole subframe duration, receivers will also need to buffer Enhanced PLP(s) for the whole subframe duration because the Enhanced PLP(s) that follow the cell writing order of Core PLP(s) should be re-ordered. Note that for buffering the Enhanced PLP(s), an extra frame buffer (e.g., DRAM) can be used, rather than consuming the TI memory (e.g., SRAM). Due to this additional frame memory requirement, it is recommended that the number of cells of Enhanced PLP(s) buffered in a frame memory be less than or equal to 2^{20} when such an LFDM configuration is used. That is, in the example of Figure 5.23, the sum of **L1D_plp_size** of PLP #0 and PLP #1 in the CL (which is equal to the sum of **L1D_plp_size** of PLP #2 and PLP #3 in the EL) should not exceed 2^{20} cells due to the limitation of frame buffer in receivers. Note that in the example configuration shown in Figure 5.23, the Enhanced PLP (PLP #3) is spread over multiple TI Groups, and therefore the CTI mode is disallowed.

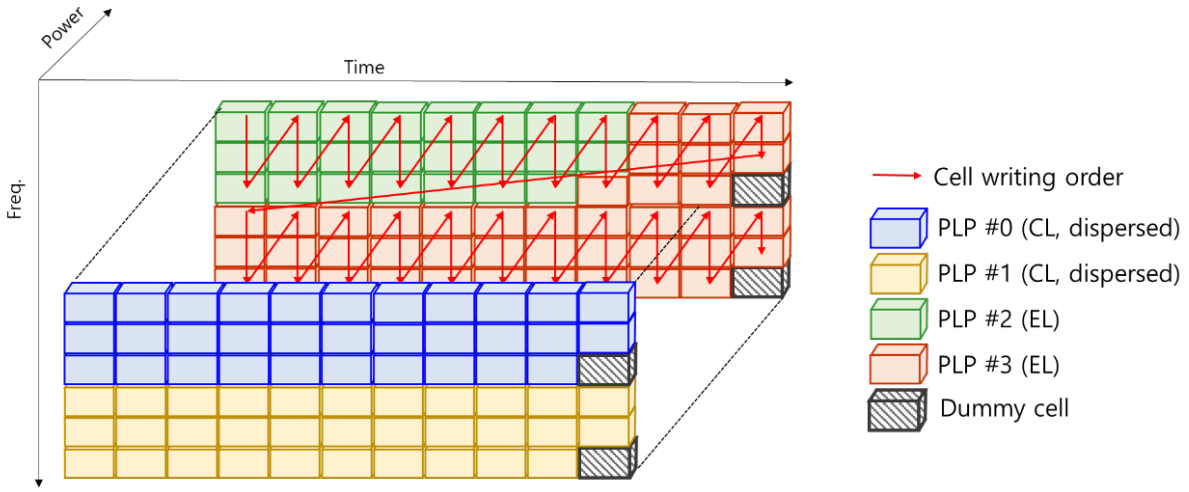

Figure 5.23 LFDM configuration example.

5.3.4 Frequency Interleaving

The use of a frequency interleaver (FI) is recommended to provide robustness to a burst error on particular frequency-bins in block/multipath fading channels. In particular, in order to enhance the stability of FI over a burst error, the FI applies a different interleaving sequence every OFDM symbol within a subframe. The key properties of FI are:

- The FI operates on the data cells (not pilots) of one OFDM symbol, i.e., Preamble symbols, data symbols, and subframe boundary symbols. The FI is optionally used for data and subframe boundary symbols, while it is always used for Preamble symbols. The use of FI on a per subframe basis is indicated by **L1D_frequency_interleaver**.
- Three FI structures are provided for 8K, 16K, and 32K FFT sizes, respectively (see Section 7.3 of [3]). When FI is not used, input data cells are output in the same order as they are input.
- Each FI structure mainly consists of an interleaving sequence generator and a symbol offset generator. In particular, the symbol offset generator shifts cyclically the output of interleaving sequence generator. The cyclic shift is accomplished prior to the following address-check block in order to guarantee the frequency de-interleaving in a single-memory at a receiver side, even if the number of data cells in a symbol pair is different from each other (see Section 6.3.1).
- In the 32K FFT size case, for even symbols, a linear writing operation into memory is performed and then a pseudo-random reading operation from memory is performed. For odd symbols, a pseudo-random writing operation into memory is performed and then a linear reading operation (in a sequence order) from memory is performed. Conversely, in case of 8K and 16K FFT sizes, two memories are used alternately for the frequency interleaving of odd and even symbols.

In frequency interleaving, the normative conditions related to frame configuration are as follows:

- In the 32K FFT size case, the sum of the number of data and subframe boundary symbols in the subframe is always even, except for the first subframe of a frame where the sum of the number of Preamble, subframe boundary and data symbols is even (Section 7.2.4.2 of

[3]). This normative condition guarantees frequency de-interleaving in a single-memory at a receiver side.

- On the first Preamble symbol of the frame, the FI is reset to the initial state of interleaving. Such a reset operation is accomplished by setting the symbol index count to zero and setting the PRBS registers of the interleaving sequence generator and the symbol offset generator to the initial values.
- With the exception of the first subframe in the frame, the FI is reset to the initial state of interleaving on the first symbol of the second and subsequent subframes in the frame.

5.4 Waveform Generation

5.4.1 Pilot Insertion

ATSC 3.0 employs scattered, continual, edge, Preamble and subframe boundary pilots. These cells are modulated with reference information whose transmitted values are known to ATSC 3.0 receivers. The pilots can be used for frame synchronization, frequency synchronization, time synchronization, channel estimation, and phase noise tracking. Table 5.**1** gives an overview of the different types of pilots and the symbols in which they appear, where a check mark indicates the presence of a particular pilot type in that symbol type.

Table 5.1 Types of Pilots in Each Type of Symbol

Symbol Type	Preamble Pilot	Scattered Pilot	Subframe Boundary Pilot	Common Continual Pilot	Additional Continual Pilot	Edge Pilot
Preamble	✓			✓		
Data		✓		✓	✓	✓
Subframe Boundary			✓	✓	✓	✓

5.4.1.1 Scattered Pilot Insertion

ATSC 3.0 defines sixteen scattered pilot (SP) schemes. The terminology employed is SPa_b, where a = D_x is the separation of pilot bearing carriers (i.e., in the frequency direction) and b = D_y is the number of symbols forming one scattered pilot sequence (i.e., time direction), as shown in Figure 5.24.

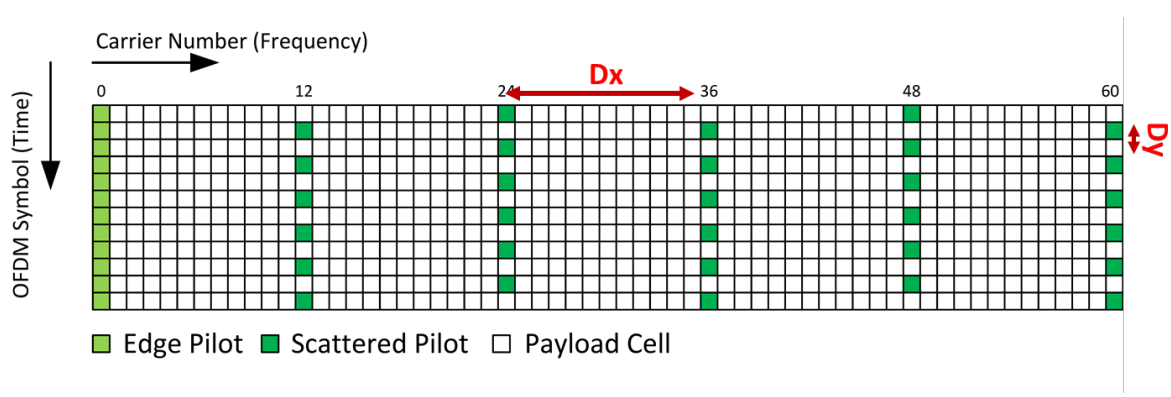

Figure 5.24 Scattered pilot pattern SP12_2 (D_x = 12, D_y = 2).

To enhance the channel estimation quality without degrading the data cells, the SP power is optionally boosted. For each SP pattern, the boosting factor can be selected from five possible

values and is signaled by **L1B_first_sub_scattered_pilot_boost** and **L1D_scattered_pilot_boost**. For example, the boosting power for SP8_4 ranges from 0 dB (no boosting) to 6.6 dB. The different boosting values can be obtained from Table 9.14 of [3].

5.4.1.2 Continual Pilot Insertion

The minimum number of common continual pilots (CPs) is set to 45 for 8K FFT when the minimum number of carriers (NoC) is configured and doubles with each doubling of the FFT size. The number of common continual pilots is augmented slightly with the NoC increase. The indices of the CP locations (see Annex D of [3]) were selected to be random and evenly distributed within the OFDM symbol and occupy roughly 0.7% of the cells. They are transmitted at a boosted power level of 8.52 dB.

5.4.1.3 Edge Pilot Insertion

The edge carriers, that is carriers with relative carrier indices k=0 and k= NoC – 1, are edge pilots in every symbol except for the Preamble symbol(s). They are inserted in order to allow frequency interpolation of channel estimates up to the edge of the spectrum.

5.4.1.4 Preamble Pilot Insertion

The D_x value used for the Preamble pilots of a frame is less than or equal to the D_x value used for the scattered pilots of the first subframe of the same frame, in order to provide more accurate equalization for the Preamble symbols.

5.4.1.5 Subframe Boundary Pilot Insertion

The pilots for subframe boundary symbols are essentially denser than for the adjacent normal data symbols of the same subframe since the same D_x value is used but D_y is assumed equal to 1 for subframe boundary symbols.

5.4.2 MISO

ATSC 3.0 includes an optional Multiple-Input Single-Output (MISO) antenna scheme, known as Transmit Diversity Code Filter Sets (TDCFS), in order to minimize the possibility of cross-interference among the transmitted signals in an SFN over the entire reception area. TDCFS introduces a frequency pre-distortion of the common waveforms from the different transmitters of an SFN in such a way that special signal processing at receivers is not necessary, since the frequency pre-distortion appears as part of the propagation channel to receivers. The coding is done using linear filters so that the decoding in receivers can be implemented as part of the traditional equalizer process. Furthermore, the coding filters are designed in such a way as to provide robust performance at receivers over a wide range of expected multipath behavior in the transmission environment.

5.4.2.1 Signal Model

A transmit diversity code filter set is a set of unique transmit filters used to filter a common modulated signal. These filtered signals are transmitted from separate transmitters in an SFN. The design of the transmit diversity code filter set is based on creating all-pass filters using a minimized metric, called the Peak Side Lobe (PSL), over all filter pairs within the constraints of the number of transmitters M and the length of the filters L. The use of such filter code sets creates an improved overall signal condition at receivers taking into account the likelihood of multipath transmission conditions.

The common modulated signal $S[k]$ is individually filtered through M strategically optimized filter vectors $\{C_1[k],C_2[k],...,C_M[k]\}$ to create uniquely pre-distorted signals $\{F_1[k], F_2[k],...,$

$F_M[k]\}$. The code filters are implemented as a multiplication of fixed coefficients across all carrier frequencies, indexed within each block by k. Each pre-distorted signal block is transformed through an Inverse Fast Fourier Transform (IFFT) and cyclically extended with a guard interval in an OFDM manner before being transmitted through its respective transmit antenna. These transmitted signals will then be modified through different channel conditions represented in the time domain by the channel impulse responses $\{h_1[n], h_2[n],..., h_M[n]\}$. The superposition of these signals combined with AWGN $w[n]$ is received by an antenna at a receiver as function $g[n]$.

If the FFT block size of the OFDM system is N_B and, assuming proper synchronization of symbol blocks, the system can be described as:

$$G[k] = \left(\sum_{i=1}^{M} H_i[k]C_i[k]\right)S[k] + W[k]$$

where:

$G[k] = \mathrm{FFT}(g[n], N_B)$

$H_i[k] = \mathrm{FFT}(h_i[n], N_B)$ with $i \in \{1..M\}$

$W[k] = \mathrm{FFT}(w[n], N_B)$

The system in the time domain can be described as:

$$g[n] = \left(\sum_{i=1}^{M} h_i[n] * c_i[n]\right)s[n] + w[n]$$

where * is the circular convolution operator over length N_B, and:

$c_i[n] = \mathrm{IFFT}(C_i[k], N_B)$ with $i \in \{1..M\}$

$s[n] = \mathrm{IFFT}(s[n], N_B)$

5.4.3 Inverse Fast Fourier Transform (IFFT)

As an OFDM system, the IFFT algorithm for Preamble and subframe symbols is used as described in Section 8.3 of [3]. Given the three different choices of FFT sizes (i.e., 8K, 16K and 32K) allowed by ATSC 3.0, recommended choices of FFT sizes depending on broadcasters' intended services are described in Section 4.2.1.

5.4.4 Guard Interval

The guard interval (GI) is a cyclic extension of the useful portion of the time domain symbol which consists of the samples belonging to the last part of the OFDM symbol. The guard interval is prepended as a prefix to the original symbol Tu (useful symbol length), leading to a new symbol of length Ts (total symbol length). GI insertion is used to prevent inter-symbol interference (ISI) at the expense of introducing overhead. When signals arrive outside the GI, ISI degradation increases until a certain value, Tp, is reached. Tp is defined as the Nyquist limit and can be considered as a limit to the channel extent that can be tolerated by the system when more than one signal is received. Furthermore, it has to be taken into account that the GI does not remove Self-Symbol Interference, i.e., the interference between the samples of the same OFDM symbol, but that can be easily equalized at receivers. GI is appended at the transmitter and discarded at the receiver by means of the FFT window positioning, which has a length of Tu. The GI is also used for synchronization aspects at receivers.

ATSC 3.0 offers twelve possible selections of GI which can be configured with the three different values of FFT sizes. However, ATSC 3.0 does not allow certain combinations of GI and FFT size due to excessive overhead. The lengths in number of samples of the different GI values and the FFT allowed combinations are presented in Table 5.2 where check marks indicate allowed combinations.

Table 5.2 Allowed Combinations of GI and FFT Sizes

GI Samples	FFT Size		
	8K	16K	32K
192	✓	✓	✓
384	✓	✓	✓
512	✓	✓	✓
768	✓	✓	✓
1024	✓	✓	✓
1536	✓	✓	✓
2048	✓	✓	✓
2432	N/A	✓	✓
3072	N/A	✓	✓
3648	N/A	✓	✓
4096	N/A	✓	✓
4864	N/A	N/A	✓

5.4.5 Bootstrap

An ATSC 3.0 frame starts with the highly robust bootstrap, which provides a universal entry point into a digital transmission signal. Broadcasters anticipate providing multiple wireless-based services, in addition to just broadcast television, in the future. A highly robust signal is needed to indicate, at a low level, the type or form of a waveform that is being transmitted during a particular time period, so that a receiver can discover and identify the signal. The bootstrap with extensible signaling transmission is suitable as a service indicator.

The bootstrap consists of a number of symbols, beginning with a synchronization symbol positioned at the start of each frame period to enable signal discovery, coarse synchronization, frequency offset estimation, and initial channel estimation. The remainder of the bootstrap contains sufficient control signaling to permit the reception and decoding of the remainder of the frame to begin. The bootstrap is first generated in the frequency domain and is then transformed into the time domain by Inverse Fast Fourier Transforming the frequency domain sequence. Then, the signaling information is communicated by a cyclic shift of the time domain signal. There are two variants of the time domain structure: C-A-B and B-C-A. The postfix or hyper prefix, part B, and the prefix, part C, are each composed of samples taken from the time domain main body signal, part A, with a frequency shift (equal to the carrier spacing). The initial symbol of the bootstrap is provided for sync detection and employs the C-A-B variant. The remaining bootstrap symbols conform to the B-C-A variant.

The generation of the bootstrap is defined in [2]. The syntax and semantics of the signaling fields for each bootstrap symbol are described in [2].

5.5 Channel Bonding

Channel bonding allows a single PLP to be spread across multiple RF channels, and the primary purpose of channel bonding is to enable total service data rates that exceed the net capacity of a single RF channel and that are up to twice what can be transmitted in one RF channel.

Other benefits of channel bonding include greater frequency diversity and more efficient statistical multiplexing. The performance of statistical multiplexing depends on the transmission bandwidth, the number of multiplexed services, the statistical properties of service traffic, and implementation. An example of the possible gains is shown in Table 5.3.

Table 5.3 Statistical Multiplexing Gains

Available Capacity (Mbps)	Number of Services (%StatMux Efficiency Increase)		
	4K UHD @15 Mbps	HD 1080p @5 Mbps	HD 720p @2.5 Mbps
10	0 (0%)	2 (7.5%)	4 (15%)
20	1 (0%)	4 (15%)	8 (25%)
40	2 (7.5%)	8 (25%)	16 (31%)

It is possible to bond RF channels from both the VHF and UHF bands (e.g., VHF/UHF), as well as different RF channels in the same band (e.g., UHF/UHF).

Channel bonding is described in detail in Annex K of [3]. There are two different types of channel bonding described in the specification, plain channel bonding and SNR averaging. Plain channel bonding can be realized at a receiver with two separate demodulators and the addition of combining logic and memory, and is more likely to be supported than SNR averaging, which requires exchange of data in the middle of the receiver data path.

5.5.1 Memory Considerations

Channel bonding places additional memory constraints on receivers in the reconstruction circuit to absorb any delays from the two different paths that the data will take after partitioning at the transmitter, so it is important that the transmitter make efforts to reduce these delays as much as possible. For example, if one RF channel uses time interleaving and the other RF channel does not use time interleaving, it can be easily seen that memory to accommodate the delay equivalent to twice the time interleaving depth will be required at receivers, since both the transmitter time interleaving delay and receiver time de-interleaving delay should be considered.

Therefore, it is recommended that after partitioning a PLP, the same PLP parameters should be used for both RF channels. That is, the same modulation, coding, time interleaving, number of symbols and so on should be used. This is required if channel bonding with SNR averaging is used, and is recommended if plain channel bonding is used.

The channel bonding specification in Annex K of [3] also limits the reconstruction memory size by requiring that Baseband Packets are assigned to the different RF channels in proportion to their respective PLP rates. Furthermore, it is specified that no more than 5 consecutive Baseband Packets are assigned to one of the RF channels before at least one Baseband Packet is assigned to the other RF channel.

In the case where the two RF channels use exactly the same parameters, it is recommended that the delay on each RF channel (and thus the required receiver memory) be minimized by sending consecutive Baseband Packets alternatively to each RF channel. That is, a Baseband Packet sent to RF channel 1 should be followed by a Baseband Packet sent to RF channel 2,

followed by a Baseband Packet to RF channel 1, and so on. This reduces the latency of the overall output and minimizes the need for buffering at receivers.

5.5.2 Channel Bonding Examples

An example of plain channel bonding with a single PLP split over two RF channels is shown below in Figure 5.25, which shows the transmitter side, and Figure 5.26, which shows the receiver side.

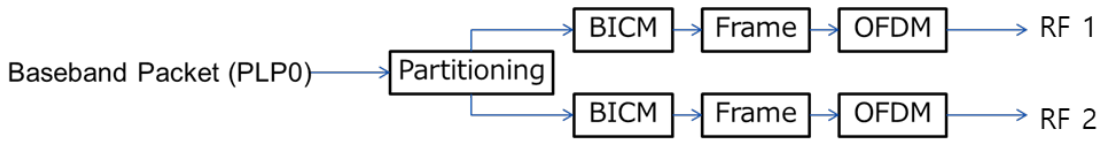

Figure 5.25 1-PLP 2-Channel Bonding example transmitter architecture.

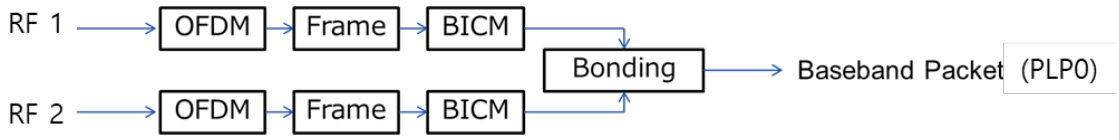

Figure 5.26 1-PLP 2-Channel Bonding example receiver architecture.

An example with parameter values, relevant bit rates and order of Baseband Packet partitioning is shown in Table 5.4. Note that since the data rates for each of the RF channels are the same, Baseband Packets are assigned to each of the RF channels in an alternating manner. This reduces latency through the channel bonding process.

Table 5.4 Example Parameters and Bit Rate for 1-PLP Channel Bonding

	Input PLP	**RF 1**	**RF 2**	**Output PLP**
Data Rate	57.69 Mbps	28.845 Mbps	28.845 Mbps	57.69 Mbps
BBP Order	1,2,3,4,5,6,7,8,9…	1,3,5,7,9 …	2,4,6,8, …	1,2,3,4,5,6,7,8,9…
PLP0 Modulation	-	256QAM	256QAM	-
PLP0 Coding Rate	-	10/15	10/15	-
PLP0 FEC Length	-	64K	64K	-
PLP0 FEC Outer	-	BCH	BCH	-
PLP0 TI Depth	-	1024	1024	-
PLP0 TI Type	-	CTI	CTI	-
PLP0 FFT Size	-	32K	32K	-
PLP0 Guard Interval	-	5_1024	5_1024	-
Pilot Pattern	-	SP24_2	SP24_2	-
Num. Symbols	-	49	49	-
Preamble FFT	-	32K	32K	-
Preamble GI	-	5_1024	5_1024	-
Preamble D_x	-	12	12	-
Preamble L1B	-	Mode 5	Mode 5	-
Preamble L1D	-	Mode 5	Mode 5	-

An example of plain channel bonding with 4 PLPs split over 2 RF channels is shown in Figure 5.27 and Figure 5.28. The Baseband Packets are partitioned and channel bonded on a PLP basis.

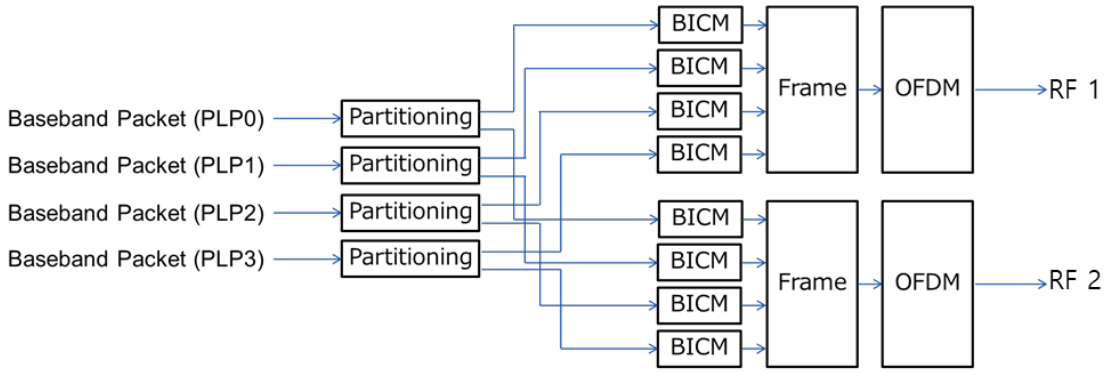

Figure 5.27 4-PLP channel bonding example transmitter architecture.

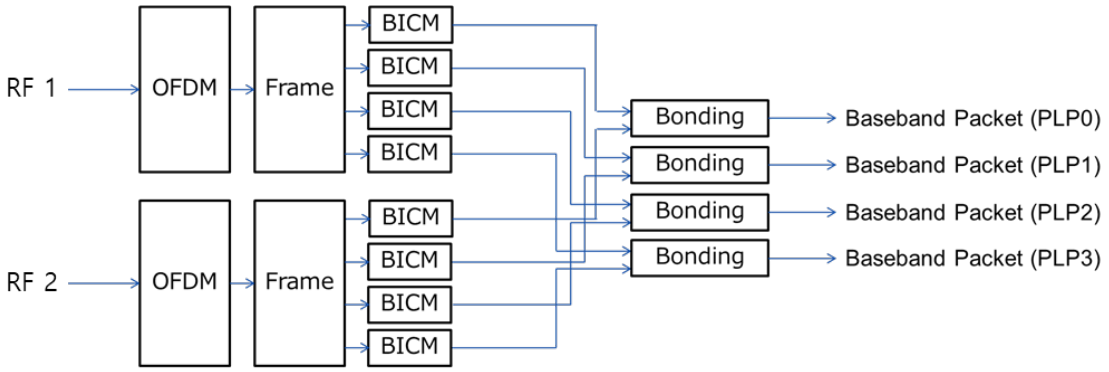

Figure 5.28 4-PLP channel bonding example receiver architecture.

6. GUIDELINES FOR RECEIVER IMPLEMENTATION

6.1 Signal Discovery and Synchronization

For ATSC 3.0 receivers, symbol timing offset (STO) and carrier frequency offset (CFO) have to be estimated and compensated before conducting data detection. The bootstrap inserted at the very beginning of each frame is utilized for fast time-frequency tracking and for minimal initial signaling transmission. The bootstrap has the following four main purposes:

1) To enable receivers to detect and validate the presence of a standardized frame.
2) To enable receivers to achieve coarse frequency and timing synchronization in a very short time.
3) To signal some transmission parameters describing the associated physical layer frame structure to permit the reception of the succeeding portion of the frame.
4) To identify the standard version used when generating the signal

6.1.1 Use of Bootstrap for Signal Acquisition and Synchronization

6.1.1.1 Signal Acquisition and Timing Synchronization

The bootstrap acquisition aims to detect the presence of bootstrap symbols. In general, receivers can achieve signal acquisition and coarse timing synchronization simultaneously by employing the same metric. The signal detected at the correct position has a relatively high power, which indicates the presence of an expected signal. The number of OFDM symbols in the bootstrap is extensible, which enables dynamic signaling transmission capacity. In ATSC 3.0, the four-symbol bootstrap,

which begins with a synchronization symbol and proceeds with three successive signaling symbols, is used. A, B, and C, respectively, denote the time domain main body signal, the postfix or hyper prefix, and the prefix as described in Section 5.4.5.

6.1.1.1.1 Delayed Correlation

Delayed Correlation can be used to achieve fast and coarse synchronization. The implementation diagram shown in Figure 6.1 performs the correlation for all guard intervals of four bootstrap symbols. The diagram comprises three branches, one for correlating parts A and C, another for correlating parts B and C, and the third for correlating parts A and B. The correlation results are denoted as $R_{DC,A}(\hat{\theta})$, $R_{DC,A+B}(\hat{\theta})$, and $R_{DC,B}(\hat{\theta})$, respectively, where DC denotes delayed correlation. The sum of the correlation values above is:

$$R_{DC}(\hat{\theta}) = R_{DC,A}(\hat{\theta}) + R_{DC,A+B}(\hat{\theta}) + R_{DC,B}(\hat{\theta})$$

which is the simplest combination method. The timing estimation θ is obtained by locating the peak of $R_{DC}(\hat{\theta})$ presented as:

$$\theta = \arg\max_{\hat{\theta}}\{R_{DC}(\hat{\theta})\}$$

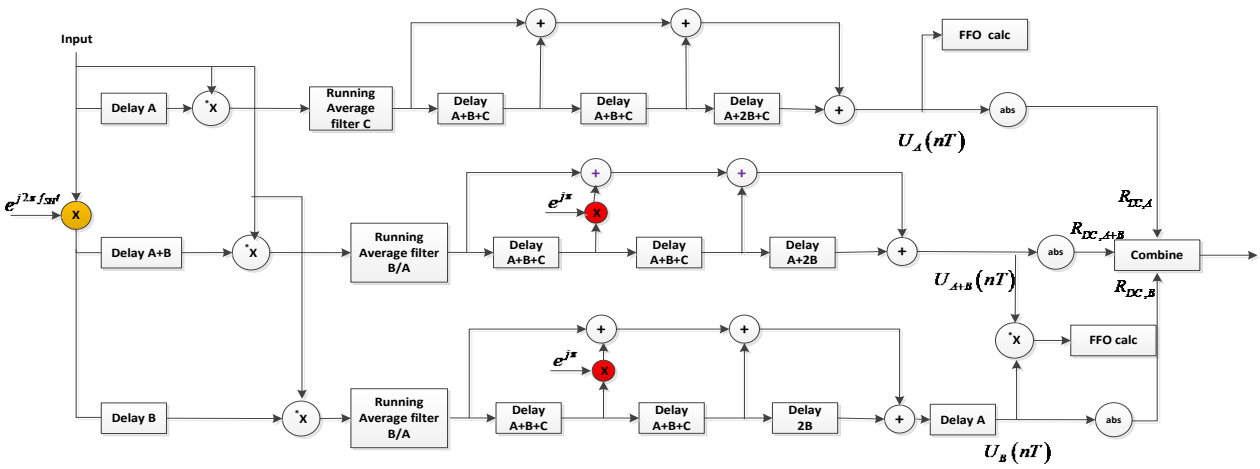

Figure 6.1 Delayed correlation diagram.

6.1.1.1.2 Local Sequence Based Correlation

As the first bootstrap symbol carries limited information for detection by receivers, the waveform of the first bootstrap symbol can be generated locally at a receiver and received samples correlated with this local sequence. Since this method requires more complexity in receivers, it may be used to search within a smaller range after finding a rough range of valid signal by the delayed correlation method. To find the local sequence correlation, a received signal can be described as:

$$r_n = s_{n-\theta} e^{j2\pi\Delta f n}$$

where Δf is the normalized frequency offset with respect to the carrier spacing, and θ is the delay of samples. First, differentiate the received samples by:

$$Z_n^{(D)} = r_n r_{n-D}^* = (s_{n-\theta} \cdot s_{n-D-\theta}^*) e^{j2\pi\Delta f D}$$

where D denotes the differential value. The phase shift caused by carrier frequency offset (CFO) is transformed to a fixed phase term $e^{j2\pi\Delta f D}$. Meanwhile, differentiate the local sequence as:

$$c_n^{(D)} = s_n \cdot s_{n-D}^*, n = D, \dots, L-1$$

Correlating the two differentiated sequences results in:

$$R_{LSBC,\hat{\theta}}^{(D)} = \sum_{n=D}^{L-D} Z_{n+\hat{\theta}}^{(D)} [c_n^{(D)}]^*$$

The correlation sequence $R_{LSBC,\hat{\theta}}^{(D)}$ has a significant peak and will not be influenced by CFO. However, the differential operation will amplify any noise resulting in SNR degradation. To minimize the influence of SNR degradation, multiple differentiations can be calculated as $R_{LSBC,\hat{\theta}}^{(D_0)}, \dots, R_{LSBC,\hat{\theta}}^{(D_{N-1})}$, where $D_0, D_1, \dots, D_{N-1}$ are the differential values with common difference K, e.g., 16, 32, 48, etc. Taking each adjacent differential correlation and doing conjugate multiplication yields:

$$RM_{i,\hat{\theta}} = R_{LSBC,\hat{\theta}}^{(D_i)} \cdot \left(R_{LSBC,\hat{\theta}}^{(D_{i+1})}\right)^*, i = 0,1,2, \dots, N-2$$

This operation can turn different phase terms $e^{j2\pi\Delta f D_i}$ into identical terms $e^{j2\pi\Delta f K}$; thus all correlation values can be summed equally to get the overall local sequence based correlation as:

$$R_{LSBC,\hat{\theta}} = \sum_{i=0}^{N-2} RM_{i,\hat{\theta}}$$

The timing estimation is obtained as:

$$\theta = \arg\max_{\hat{\theta}} \{R_{LSBC}(\hat{\theta})\}$$

6.1.1.2 Fractional Frequency Offset (FFO) Estimation

FFO is estimated using results of delayed correlation, $R_{DC,A}(\theta)$, $R_{DC,A+B}(\theta)$ and $R_{DC,B}(\theta)$ in Figure 6.1, where θ is the estimation of time sync position. The FFO estimation is obtained as:

$$FFO_1 = angle(\frac{R_{DC,A}(\theta)}{2\pi})$$

$$FFO_2 = angle(\frac{R_{DC,A+B}(\theta) \cdot (R_{DC,B}(\theta))^*}{2\pi})$$

$$FFO = \frac{FFO_1 + FFO_2}{2}$$

6.1.1.3 Integer Frequency Offset (IFO) Estimation and Validation

The FFT of the received first bootstrap symbol including pre-/post-fix (note that residual timing offset may exist) is taken, and then the FFT results are circularly shifted for a given value in the set $I = [-N: N]$. Each circularly-shifted sequence is conjugate multiplied with the FFT of the local replica of the first bootstrap symbol (also pre/post-fix inclusive), and then the IFFT of each conjugate multiplication result is taken.

For each IFFT result, significant peaks corresponding to main paths are picked and combined to get the total power $h_{pwr}(k)$ where k belongs to I. Among all $2N + 1$ combined powers, the largest one h_{pwr_peak} is picked, and the corresponding shift value K is regarded as the IFO.

As for the validation operation, the peak-to-average-ratio, which is defined as h_{pwr_peak}/h_{pwr_ave}, is used for validation where h_{pwr_ave} denotes the average of $2N + 1$ powers. Only when such peak-to-average-ratio is larger than a pre-defined threshold, are the results of acquisition and synchronization validated.

6.1.2 Signaling Detection of Bootstrap Symbols

Once the bootstrap is correctly detected and centered, the signaling detection and decoding is achieved through two steps, time domain cyclic shift detection and Gray de-mapping.

6.1.2.1 Time Domain Cyclic Shift Detection

6.1.2.1.1 Detection Method A

The bootstrap signaling information is conveyed through the use of cyclic shifts in the time domain after the IFFT of the originating frequency domain sequence. Specifically, the signaling is carried by the relative cyclic shift between two consecutive bootstrap symbols, which can eliminate any side effect brought in by false detection of a previous symbol. The relative cyclic-shift value can be detected as follows. Note that it is assumed that the channel transfer functions of consecutive bootstrap symbols are almost equal. The method is performed in the frequency domain to take full advantage of the channel transfer function to eliminate the channel effects.

Step 1. Remove the prefix and postfix of the received bootstrap symbol and take the FFT of the remaining portion of the symbol to get the frequency domain equivalent of the symbol as $\tilde{r}_n(k)$ where the subscript n denotes the index of the bootstrap symbol.

Step 2. Divide $\tilde{r}_n(k)$ by a known reference $s_n(k)$ to get the transfer function $H_n(k)$ of the bootstrap symbol.

Step 3. Eliminate the channel effects of the symbol $\tilde{r}_n(k)$ by multiplying it by the conjugate of the preceding symbol's transfer function $H_{n-1}(k)$ to get $r_n(k)$.

Step 4. To leave out the extra IFFT operation by taking advantage of the circular time-shifting theorem, the cyclic shift in time domain applied in each symbol period can be considered as a phase rotation in the frequency domain; thus the relative cyclic shift M_n can be obtained as:

$$\widetilde{M}_n = \max_{\widehat{M}_n \in [1:2048]} \{ \sum_{k=0}^{2047} r_n(k) [s_n(k) \exp(-\frac{j2\pi k \widehat{M}_n}{2048})]^* \}$$

6.1.2.1.2 Detection Method B: Iterative Detection

As described in Section 6.1.2.1.1, the channel estimates of consecutive bootstrap symbols are used for the detection of bootstrap symbols. The reliability of these channel estimates affects the detection performance of the bootstrap symbols. In order to improve the reliability of channel

estimates, the channel estimates for two consecutive bootstrap symbols can be iteratively averaged. The iterative detection consists of forward and backward detections.

Step 1. Forward detection

The original channel estimate $H_n(k)$ can be obtained by multiplying the received signal $\tilde{r}_n(k)$ by a known reference $s_n(k)$. The channel estimate $H_{n-1}(k)$ of the preceding symbol is used for the detection of the *n*-th symbol. Then, the relative cyclic shift $\widetilde{M}_n$ can be obtained as:

$$\widetilde{M}_n = \arg\min_{\widetilde{m}_n \in \chi_r} \sum_{k=0}^{N_{FFT}-1} \left| \tilde{r}_n(k) - H_{n-1}(k) S_n(k) e^{j\frac{2\pi k \widetilde{m}_n}{N_{FFT}}} \right|^2$$
$$= \arg\max \mathrm{Re}\left\{ IFFT\{\tilde{r}_n^*(k) H_{n-1}(k) S_n(k)\} \right\}$$

where χ_r denotes the set of all possible values of the relative cyclic shift.

Conversely, after detecting the relative cyclic shift $\widetilde{M}_n$, another estimate of the channel gain can be calculated as $H_{n-1}(k)e^{j\frac{2\pi k \widetilde{M}_n}{N_{FFT}}}$ for the *n*-th symbol using the detected $\widetilde{M}_n$. Then, by averaging the original channel estimate $H_n(k)$ and the calculated channel estimate $H_{n-1}(k)e^{j\frac{2\pi k \widetilde{M}_n}{N_{FFT}}}$, the channel estimate can be updated as:

$$\widetilde{H}_n(k) = \frac{1}{2}\left(H_{n-1}(k) e^{j\frac{2\pi k \widetilde{M}_n}{N_{FFT}}} + H_n(k) \right)$$

Using the updated channel estimate $\widetilde{H}_n(k)$, the subsequent symbol can be detected. In this manner, all symbols are successively detected up to the last relative cyclic shift. Accordingly, channel estimates are updated using original channel estimates and detected relative cyclic shifts.

Step 2. Backward detection

Since the detected relative cyclic shift is used for the update of the channel estimate in the forward detection, the symbol detection affects the reliability of the updated channel estimate. Therefore, in order to improve the reliability of the channel estimate, it can be further updated along the backward direction. For backward detection, the updated channel estimate $\widetilde{H}_n(k)$ is used for the detection of the *n*-th symbol using the preceding received signal $\tilde{r}_{n-1}(k)$. Then, the relative cyclic shift $\widetilde{M}_n$ can be obtained as:

$$\widetilde{M}_n = \arg\min_{\widetilde{m}_n \in \chi_r} \sum_{k=0}^{N_{FFT}-1} \left| \tilde{r}_{n-1}(k) - \widetilde{H}_n(k) S_{n-1}(k) e^{j\frac{2\pi k(-\widetilde{m}_n)}{N_{FFT}}} \right|^2$$
$$= \arg\max \mathrm{Re}\left\{ IFFT\{\tilde{r}_{n-1}^*(k) \widetilde{H}_n(k) S_{n-1}(k)\} \right\}$$

The backward detection is started from the last symbol. Using the updated channel estimate, the last relative cyclic shift $\widetilde{M}_{N_s-1}$ is re-detected as:

$$\widetilde{M}_{N_s-1} = \arg\max \mathrm{Re}\left\{ IFFT\{\tilde{r}_{N_s-2}^*(k) \widetilde{H}_{N_s-1}(k) S_{N_s-2}(k)\} \right\}$$

Conversely, similar to the forward detection, another estimate of the channel gain for the $(n-1)$-th symbol can be calculated as $\widetilde{H}_n(k)e^{j\frac{2\pi k(-\widetilde{M}_n)}{N_{FFT}}}$ with the detected relative cyclic shift $\widetilde{M}_n$. Then, by averaging the original channel estimate $H_{n-1}(k)$ and the calculated channel estimate $\widetilde{H}_n(k)e^{j\frac{2\pi k(-\widetilde{M}_n)}{N_{FFT}}}$, the channel estimate can be updated as:

$$\widetilde{H}_{n-1}(k) = \frac{1}{2}\left(\widetilde{H}_n(k)e^{j\frac{2\pi k(-\widetilde{M}_n)}{N_{FFT}}} + H_{n-1}(k)\right)$$

Using the updated channel estimate $\widetilde{H}_{n-1}(k)$, the subsequent symbol is detected along the backward direction. In this manner, all symbols are successively detected up to the first relative cyclic shift. Accordingly, channel estimates are updated using the original channel estimates and detected relative cyclic shifts. Moreover, the channel estimate can be iteratively updated in the forward and backward directions in turn.

6.1.2.2 Gray De-Mapping

In [2], a Gray code mapping of signaling bit values to a corresponding relative cyclic shift value for transmitter operation is specified. The corresponding de-mapping operation is present as follows.

Let the detected relative cyclic shift $\widetilde{M}_n$ be represented in binary form as $\widehat{m}_{10}^n\widehat{m}_9^n\ldots\widehat{m}_1^n\widehat{m}_0^n$. The signaling bit values expected by a receiver can be estimated as follows, where $\oplus$ represents the logical exclusive OR operator.

$$\hat{b}_i^n = \begin{cases} \widehat{m}_{10}^n, & i = 0 \\ \widehat{m}_{11-i}^n \oplus \widehat{m}_{10-i}^n, & 1 \le i < N_b^n \\ 0, & N_b^n \le i < 11 \end{cases}$$

Note that N_b^n denotes the number of actual signaling bits for a given bootstrap symbol. After the signaling bits $\hat{b}_0^n\hat{b}_1^n\ldots\hat{b}_7^n$ have been decoded, the transmission parameters can be interpreted by looking up the relevant syntax and semantics tables.

6.2 Waveform Demodulation

6.2.1 Channel Estimation and Equalization

ATSC 3.0 uses OFDM modulation, which is a form of multi-carrier transmission, where each carrier delivers coherently modulated complex symbols. In a receiver, the received constellation symbol for the *k*th carrier of the *l*th OFDM symbol may be expressed as:

$$\tilde{R}_l(k) = S_l(k) \cdot H_l(k) + z_l(k)$$

where $S_l(k)$ is the transmitted symbol, $H_l(k)$ is the channel response of the *k*th carrier in the *l*th OFDM symbol, and $z_l(k)$ is the system noise. Given the received symbol, equalization may be performed as:

$$\tilde{S}_l(k) = \frac{\tilde{R}_l(k)}{\widetilde{H}_l(k)}$$

where $\widetilde{H}_l(k)$ is the estimate of the channel response, $H_l(k)$.

In ATSC 3.0, scattered pilots (see Section 8.1.3 of [3]) are transmitted on dedicated carriers, which are designed for receivers to obtain estimates of the channel status of each carrier. Each pilot pattern is defined by the spacing of the scattered pilots in the frequency domain and the number of OFDM symbols forming one scattered pilot sequence. The scattered pilots are allocated to different carrier locations in multiple consecutive OFDM symbols in order to better compensate for frequency selectivity. Since scattered pilots are transmitted in every OFDM symbol and are evenly distributed over the active frequency band, a commonly used channel estimation method is to perform frequency domain interpolation. The least square (LS) channel estimates can be first obtained as:

$$\widetilde{H}_l(k_P) = \frac{\widetilde{R}_l(k_P)}{X_l(k_P)}$$

where $\widetilde{R}_l(k_P)$ is the received pilot symbol, $X_l(k_P)$ is the transmitted pilot symbol, and k_p is the carrier index assigned in the lth OFDM symbol for scattered pilot. The most popular method is linear interpolation, which may be carried out as:

$$\widetilde{H}_l(k) = \frac{k_P^+ - k}{\Delta_P} \cdot \widetilde{H}_l(k_P^-) + \frac{k - k_P^-}{\Delta_P} \cdot \widetilde{H}_l(k_P^+)$$

where k_P^- is the closest pilot carrier index lower than k, and k_P^+ is the closest pilot carrier index larger than k. In addition, no matter what scattered pilot pattern is used, known symbols are also transmitted on the two edge carriers. These are called edge pilots and are used to obtain accurate channel estimates on carriers close to either end of the active spectrum. Note that continual pilots (see Section 8.1.4 of [3]), which are also inserted in each OFDM symbol, are usually used for system frequency synchronization, rather than equalization.

Linear interpolation performs well for propagation channels with limited frequency selectivity, i.e., with small multipath delay spread. In this case, the channel frequency response between two adjacent pilot carriers can be well approximated by a linear curve. However, for wireless channels with multipath components of large delays, the channel response may change significantly on carriers between two adjacent pilot carriers. Linear interpolation may degrade the channel estimation performance. For more accurate channel estimate, frequency domain interpolation methods such as the cubic-spline interpolation, DFT-interpolation, or Wiener interpolation methods may be used.

6.2.1.1 Channel Estimation for Mobile Reception

ATSC 3.0 mobile receivers for handheld, portable, and vehicular receptions may experience very challenging channel conditions such as:

- Channel most likely consisting of many multipath components;
- Large multipath delay spread;
- Fast channel time-variation, which introduces high-level Doppler noise;
- Slow channel time-variation, which introduces long-lasting time domain fade;
- Low signal strength, especially for indoor reception.

For the improvement of reception performance under these challenging mobile channel environments, a short time domain window may be beneficial if a receiver uses a time domain

smoothing filter. A longer time domain smoothing filter will be less effective with the time variation of a fast moving channel. The higher-order cubic-spline interpolation or DFT-interpolation in the frequency domain will outperform linear interpolation if there is significant frequency selectivity due to large delay spread.

6.2.1.2 Channel Estimation for Fixed Reception

For receiving high throughput services with stationary powerful antennas (e.g., rooftop antennas), the received signal has the following characteristics:

- Strong received signal power, resulting in high SNR;
- Little or no time-variation;
- For signal received from one transmitter, the channel may be mostly modeled as either a near line-of-sight (LOS) channel or a Rician channel with a strong LOS component;
- Possibly large delay spread, especially in SFN environment.

Channel estimation for fixed services can be quite accurate because the channels usually follow the predictable LOS or Rician models, which are generally easier to estimate. Furthermore, if a typical value of SNR for fixed signal detection is around 15 dB, which is equivalent to pilots with a very high power boosting, such high SNR on pilot symbols can provide accurate channel estimates for normal propagation conditions with limited multipath delay spread. In order to achieve further improvement on channel estimation, or to overcome channels in SFN with very long delay spread, a longer time domain window may be beneficial if a receiver uses a time domain smoothing filter.

6.2.2 Removal of Peak to Average Power Ratio Reduction Techniques

Tone Reservation (TR) and/or Active Constellation Extension (ACE) techniques may be applied to reduce the Peak to Average Power Ratio (PAPR) of the transmitted ATSC 3.0 signal, as indicated by **L1B_papr_reduction** (Section 9.2.1 of [3]) in L1-Basic.

When TR is enabled, a set of carriers is reserved for PAPR reduction and this set of reserved carriers does not overlap with pilot carriers. Therefore, ATSC 3.0 receivers can simply discard the symbols in the corresponding reserved carriers. The calculation of indices of reserved carriers is the same as the transmitter side, which is described in Section 8.4.1 of [3].

When ACE is enabled, no specific operation at ATSC 3.0 receivers is required.

6.3 De-Framing and De-Interleaving

6.3.1 Frequency De-Interleaving

To minimize the use of de-interleaving memory at a receiver, the frequency de-interleaving method in a single memory is recommended, where the single memory size should be set to the value greater than or equal to $N_{data}^{max} = \max(N_{data,32K}^{max}, 2N_{data,16K}^{max}, 2N_{data,8K}^{max})$:

- $N_{data,32K}^{max}$: the maximum number of data cells carried in one 32K OFDM symbol.
- $N_{data,16K}^{max}$: the maximum number of data cells carried in one 16K OFDM symbol.
- $N_{data,8K}^{max}$: the maximum number of data cells carried in one 8K OFDM symbol.

For data recovery at a receiver, the frequency de-interleaver (FDI) performs the inverse of FI operation. Note that the de-interleaving sequences used for FDI operation are identical to the interleaving sequences used for the FI operation, which implies the address generator for frequency interleaving is identically used for generating the de-interleaving sequences.

6.3.1.1 Frequency De-Interleaving for 8K/16K FFT Size

In the 8K/16K FFT size, frequency de-interleaving is accomplished by switching alternately between two memories. That is, the first input OFDM symbol is written to the first memory for the de-interleaver. The second input OFDM symbol is then written to the second memory while the first memory is being read.

6.3.1.2 Frequency De-Interleaving for 32K FFT Size

From the interleaving operations for even and odd OFDM symbols in 32K FFT size, the FDI should perform the following two basic operations:

- For even symbols, a pseudo random writing operation into memory is performed and then a linear reading operation (in a sequence order) from memory is performed.
- For odd symbols, a linear writing operation into memory is performed and then a pseudo random reading operation from memory is performed.

Furthermore, note that in order to perform frequency de-interleaving in a single memory of size N_{data}^{max} without losing data cells (or overwriting between data cells) even if a sequence varies every symbol pair and each symbol length (in units of data cells) in a symbol pair is different (data symbol and subframe boundary symbol, for example), the FDI should comply with the following operations:

- In a received symbol pair, the address sequence for writing the odd symbol should match to the address sequence for reading the even symbol. Using the address generator used for frequency interleaving, the matching condition can be satisfied, especially by cyclically shifting the output address of the interleaving sequence generator, and then checking the availability of the output address in the address check block after the symbol offset generator.
- The cell of the previous symbol should be read out from the memory location to which the cell of the current symbol is to be written. Otherwise, some data cells of the even symbol will be overwritten.

6.3.2 Time De-Interleaving

6.3.2.1 Extended Time De-Interleaving

As explained in Section 4.2.11.3, the key role of the extended interleaving mode is to double the interleaving depth of the CTI and HTI modes and the implementation of each mode depends on manufacturer choice. In the support of the extended de-interleaving mode at a receiver, an efficient memory management may be used in order to reduce the cost of time de-interleaving memory. An efficient memory management can be achieved by adjusting the bit resolution of each cell of de-interleaving memory and appropriately quantizing the input values of that memory. Figure 6.2 illustrates the reconfiguration of a linear (sequential) de-interleaving memory after adjusting the bit resolution of each cell. Note that a linear memory configuration may be considered practically for convolutional time de-interleaving (CTDI) and hybrid time de-interleaving (HTDI). From Figure 6.2, it is shown that the number of memory cells doubles after memory reconfiguration, which means that $M_{TI} = 2^{19}$ cells (without extended interleaving mode) are sufficient to support the extended de-interleaving mode.

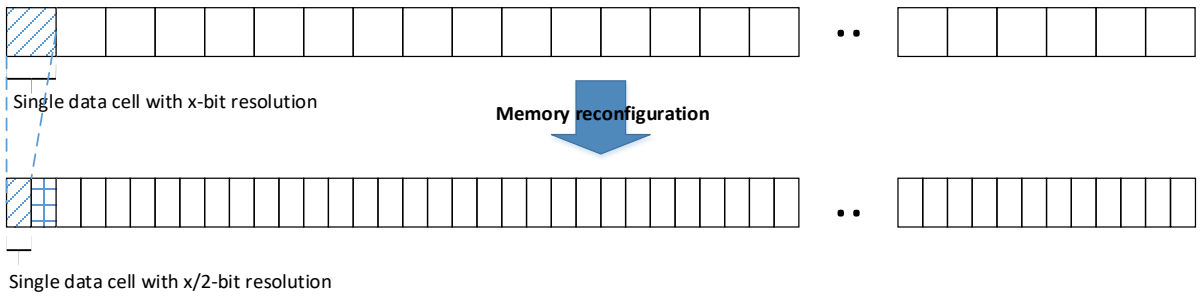

Figure 6.2 Memory reconfiguration to support the extended de-interleaving.

6.3.2.2 Hybrid Time De-Interleaving

The hybrid time interleaver can be flexibly configured with a combination among the cell interleaver, TBI and CDL. Note that the use of the TBI is mandatory if the HTI mode is enabled, while the use of the cell interleaver and CDL is optional and their use is indicated to a receiver via **L1D_plp_HTI_cell_interleaver** and **L1D_plp_HTI_inter_subframe**, respectively (see Section 7.1.5 of [3]). The configuration of the hybrid time de-interleaver (HTDI) depends on the HTI. For data recovery at a receiver, the HTDI performs the inverse of HTI operation. That is, the inverse CDL corresponding to the CDL used for HTI is first performed if **L1D_plp_HTI_inter_subframe**=1. Twisted block de-interleaving is then performed and cell de-interleaving is lastly accomplished if **L1D_plp_HTI_cell_interleaver**=1. In the following, the inverse CDL, TBDI and cell de-interleaver are described separately since each de-interleaving operation is unique and independent of the others (see Section 7.1.5 of [3]).

6.3.2.2.1 Inverse CDL

From Section 5.3.1.2, the CDL follows a typical convolutional interleaving operation (ref. CTI mode) except that a FIFO register in CDL delays a group of cells defined as a block unit. Note that the information needed for calculating the FIFO register size is delivered to a receiver via **L1D_plp_HTI_num_ti_blocks**, **L1D_plp_HTI_num_fec_blocks**, and **L1D_plp_HTI_num_fec_blocks_max**. The inverse CDL at a receiver simply performs the inverse of the CDL operation at a transmitter and its entire operation and implementation is similar to the CDL. The key role of the inverse CDL is to collect the spread FEC Blocks over multiple subframes in order to feed a complete TI Block to the TBDI.

6.3.2.2.2 Twisted Block De-Interleaver

An efficient TBDI implementation method is described by using a single memory that is only one TI-block's worth of memory. The input of the TBDI can be the output of the de-framing process or the inverse CDL. In the following, to simplify the description, it is assumed that **L1D_plp_HTI_inter_subframe**=0. The de-interleaving method for a constant cell rate will be described in detail since it can be a basis to understand the core of the method for the variable cell rate (non-constant cell rate) case. The de-interleaver method for the variable cell rate case is almost identical to the constant cell rate case. As the TI Block is synchronized to subframe boundary in the time direction, receivers can begin time de-interleaving after subframe detection. A receiver first detects the Preamble to get information about TBI configuration via **L1D_plp_HTI_num_fec_blocks** and **L1D_plp_HTI_num_fec_blocks_max**. Then, the receiver can immediately start de-interleaving using an allocated TBDI memory.

For a comprehensive understanding of TBDI operation in a single-memory, a constant cell rate case is first considered, in which a virtual FEC Block is not required. An example case is given where a TBDI operation on a single linear memory array of size $M(= N_r \times N_c)$ is considered where N_r and N_c denote the number of cells in a FEC Block and the number of FEC Blocks in a TI Block, respectively. Note that an address generation needed for accomplishing the exact reading-and-writing operation on a single linear memory array may be different depending on implementation methods, and this example is based on a 2-dimensional memory array such as a TBI operation. A basic TBDI operation can be achieved as follows.

1) After initial subframe synchronization, data cells of the first TI Block are received and written into a memory in an interleaving sequence order. During this TI Block only, there are no output data cells. Once the whole TI Block is stored in the memory, data cells are ready to be output.
2) While the input data cells of the second TI Block are being received, the data cells of the first TI Block are read out (in a de-interleaving sequence order). Note that the addresses for reading will also be used to write the new data cells belonging to the second TI Block, which leads to the efficient use of TBDI memory.
3) When the data cells of the second input TI Block are all stored (and the first TI Block is completely read out), those data cells will also need to be read out.
4) Repeat the second and third steps for the following input TI Blocks.

The following describes the address generation to perform a successful reading-and-writing operation for TI Blocks continuously being input into TBDI. By referring to the principle of the address generation used in the TBI (see Section 7.1.5.4 of [3]), the address for reading-and-writing on a single linear memory array can be achieved. As aforementioned, it is assumed that the first input TI Block to TBDI is written into a linear memory array in an interleaving sequence order. In the following input TI Blocks, the address $a_{i,j}$ needed for reading-and-writing operation can be generated as follows:

$$a_{i,j} = N_r \times C_{i,j} + R_{i,j}, \text{ for } i = 0, \cdots, M-1$$

where i is the input cell index and j is the input TI Block index. The parameters of $R_{i,j}$ and $C_{i,j}$ are calculated as follows:

$$R_{i,j} = i \bmod N_r,$$

$$T_{i,j} = S_j \times R_{i,j} \bmod N_c,$$

$$C_{i,j} = \left(T_{i,j} + \left\lfloor \frac{i}{N_r} \right\rfloor\right) \bmod N_c.$$

Here, the parameters of $R_{i,j}$ and $C_{i,j}$ denote the row and column indices under the assumption of a 2-dimensional memory array such as a TBI operation, and $T_{i,j}$ is a twisting parameter with the parameter S_j defined as:

$$s_j = mod(s_{j-1} - 1, N_c), \ where \ s_{-1} = 0.$$

The following example illustrates an example of TBDI operation on a single linear memory array. In this example, two TI Blocks are considered, where each TI Block has size of $N_r \times N_c = 4 \times 3$, and the TI Block after interleaving is directly fed to the TBDI. Figure 6.3 illustrates the input TI Blocks and output TI Blocks during the TBI operation.

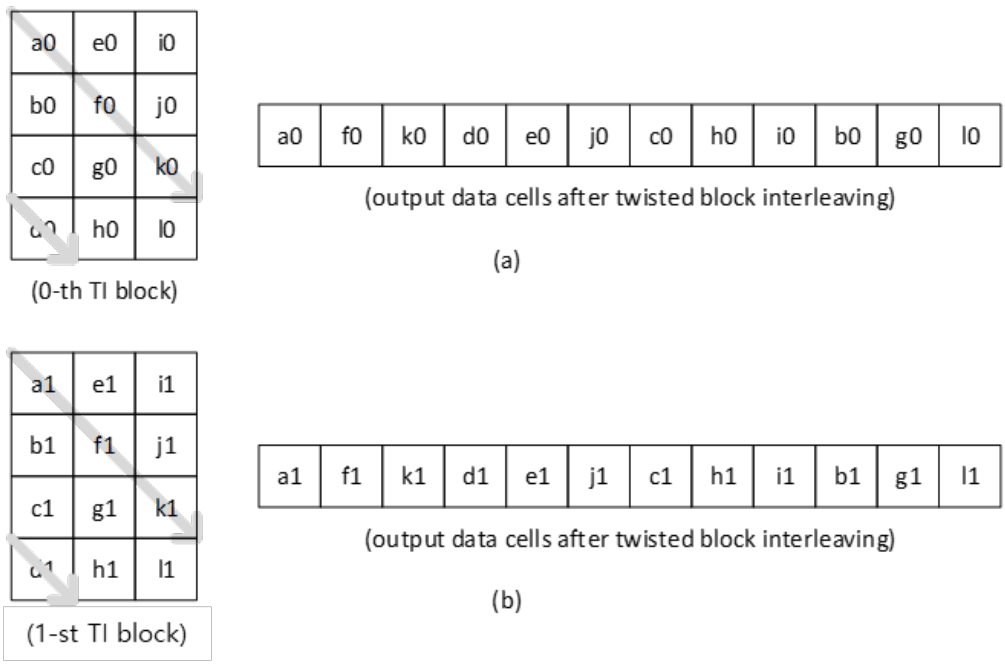

Figure 6.3 Example of input TI Block and output TI Block in a TBI operation: (*a*) The 0-th TI Block, (*b*) The 1-st TI Block.

Then, an example of the de-interleaving process using a single linear memory array as shown in Figure 6.4 is as follows.

1) Data cells of the first TI Block are written into a memory in an interleaving sequence order as shown in Figure 6.4 (*a*). During this TI Block only, no output data cells are produced.
2) When data cells of the second TI Block are being received, the data cells of the first TI Block will be read out. The addresses used for reading will also be used to write the new data cells belonging to the second TI Block. Figure 6.4 (*b*) denotes the addresses for reading-and-writing on the memory after the first step and those addresses are calculated by using the aforementioned address generation.
3) When the data cells of the second input TI Block are all stored (and data cells of the first TI Block are completely read out) as shown in Figure 6.4 (*c*), those data cells will also need to be read out. Figure 6.4 (*d*) denotes the addresses for reading the data cells after the second step.
4) The second and third steps will be repeated for the following input TI Blocks.

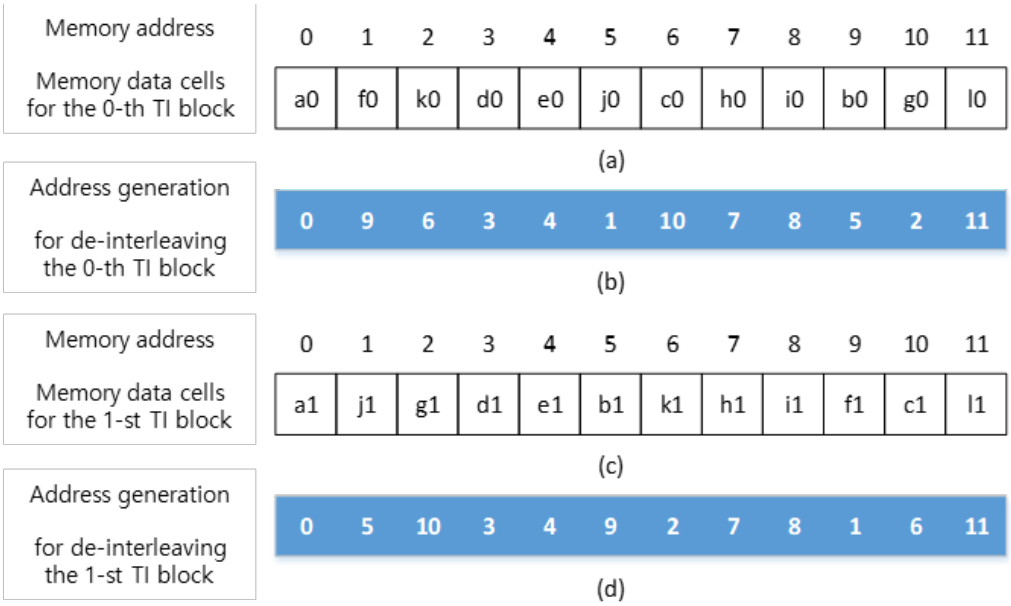

Figure 6.4 Example of input TI Block and output TI Block in TBDI operation: (*a*) Memory data cells for the 0-th TI Block, (*b*) Address generation for reading the 0-th input TI Block, (*c*) Memory data cells for the 1-st TI Block, (*d*) Address generation for reading the 1-st input TI Block.

In the case of a variable cell rate PLP that is, the number of FEC Blocks varies every TI Block, the position of virtual FEC Blocks (or virtual cells) which are not actually transmitted should be known for correct de-interleaving in receivers. Note that such position information of virtual cells as well as actual data cells is needed to generate the memory addresses for reading-and-writing operations. Figure 6.5 shows an example of time interleaving and de-interleaving operations for the variable cell rate case. In particular, this example illustrates the position of virtual cells when an input TI Block is written into a linear memory array at receivers. After writing, the following de-interleaving operations on a single linear memory array are similar with the constant cell rate case. In this example, it is assumed that a TI Block has size $N_r \times N_c = 4 \times 3$, and the maximum number of FEC Blocks and the current number of FEC Blocks are 3 and 2, respectively (i.e., a single virtual FEC Block is used). The interleaving and de-interleaving operations are illustrated as follows.

1) Figure 6.5 (*a*) shows the output TI Block after TBI operation at a transmitter, where virtual cells are not transmitted. In practice, such skip operation of virtual cells may be realized by neglecting the memory addresses of virtual cells during reading operation. That is, the actual skipping of virtual cells is performed by an address generation, where a memory address calculation is neglected since the virtual cells are positioned ahead of the actual data cells at the TBI input.
2) Figure 6.5 (*b*) shows the memory data cells after writing the interleaved TI Block into the TBDI memory. Here, note that the position information of virtual cells skipped in the TBI reading process is recovered in the TBDI writing operation. In practice, the virtual cells are not inserted as actual data cells and the memory addresses corresponding to the position of virtual cells are not necessarily generated as well.

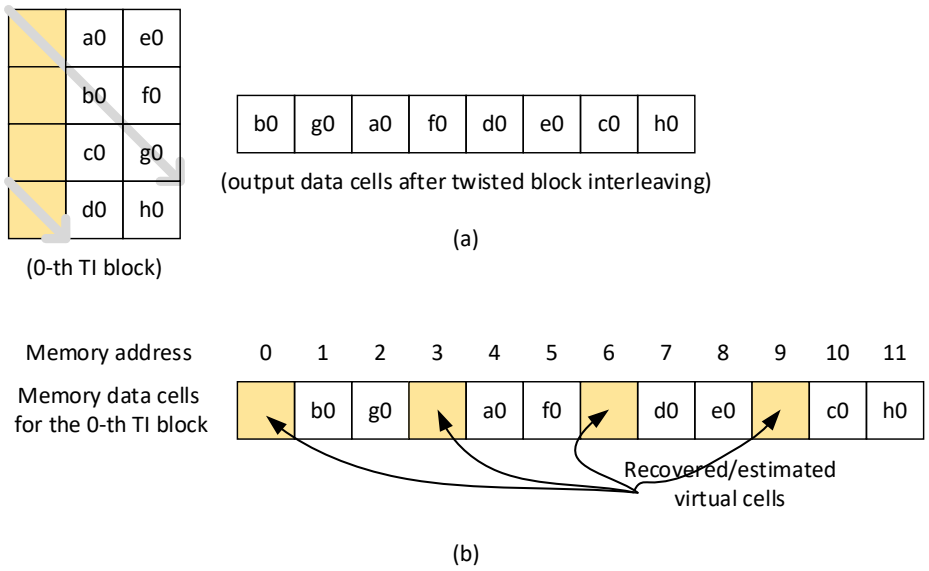

Figure 6.5 Example of time interleaving/de-interleaving with virtual FEC Blocks: (*a*) Output data cells after TBI operation, (*b*) Memory data cells after writing into TBDI memory.

6.3.2.2.3 Cell De-Interleaver

The output cells of the TBDI correspond to the input cells of the cell de-interleaver. The cell de-interleaver simply performs the inverse of the cell interleaver: 1) Pseudo-random writing operation and 2) linear reading operation. From the insight of cell interleaving operation (see Section 7.1.5.2 of [3]), it is analyzed that two memories required for cell de-interleaving may be replaced by two memories for FEC decoding. For example, before the pseudo-random writing operation of cell de-interleaving (into a FEC memory), the cell de-mapper operation is performed on-the-fly. This immediate de-mapping is possible since the cell mapper block operates independently on each cell at a transmitter. From the cell de-interleaving operation, each output vector of the cell de-mapper block is pseudo-randomly transferred into FEC memory, on-the-fly (without memory required). After a single FEC Block is stored in FEC memory, the bit de-interleaving and FEC decoding are performed, sequentially. Note that such cell de-interleaving and cell de-mapping on-the-fly is possible since the cell interleaver performs the linear writing operation and pseudo-random reading operation. As another example (see Clause 10.4.4 of [10]), it may be possible to use the same memory as for the preceding or following stages (time de-interleaver or bit de-interleaver).

6.4 LDM Decoding

The block diagram for LDM decoding in an ATSC 3.0 receiver is shown in Figure 6.6. In ATSC 3.0, since both Core Layer (CL) and Enhanced Layer (EL) signals share the same OFDM signal structure, including FFT size, Guard Interval length, and pilot pattern, only one process for synchronization, channel estimation, frequency / time de-interleaving is needed for the decoding of services in both CL and EL.

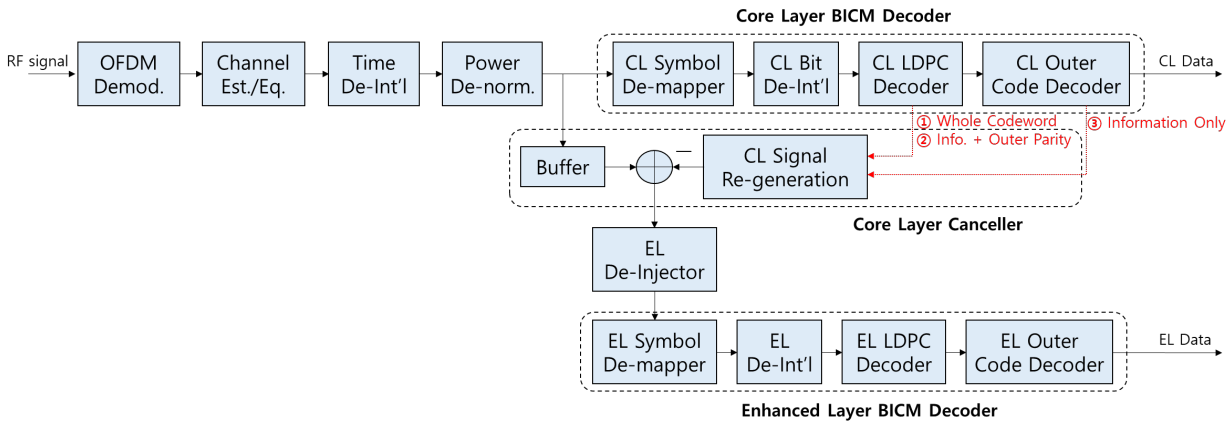

Figure 6.6 LDM decoding block diagram.

6.4.1 Decoding Process of Core and Enhanced Layers

When ATSC 3.0 receivers process combined cells of CL and EL, the CL should be decoded first treating the associated EL as additional noise, and then the EL is decoded through the CL re-generation and cancellation. When LDM is used, such decoding process of CL and EL should be performed with minimal memory consumption as shown in Figure 6.7. As shown in the figure, the first FEC Block of CL in a subframe is decoded first, and then the first FEC Block of EL is decoded through the re-generation and cancellation of the first FEC Block of CL. Then, the same process is performed for the second FEC Block, and so on. Note that FEC Block sizes of CL and EL may be different in general depending on the choice of ModCod combinations in CL and EL, and therefore, additional memory in receivers will be required in order to buffer an entire FEC Block. However, this additional memory will be no more than the size of one FEC Block regardless of subframe length because of the decoding process shown in Figure 6.7.

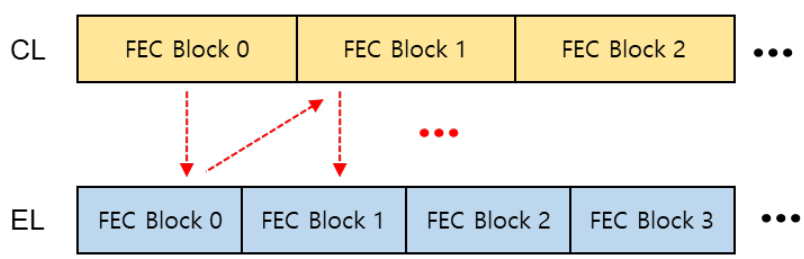

Figure 6.7 LDM FEC Block decoding process of Core and Enhanced Layers.

6.4.2 Core PLP Decoding

As shown in Figure 6.6, decoding a service carried in Core PLP(s) follows the same process as a non-LDM receiving process with the addition of a power de-normalization step prior to symbol de-mapping. The received two-layer LDM symbols on each carrier can be expressed as:

$$\mathrm{R}_l(k) = \beta \cdot S_C^l(k) \cdot H_l(k) + \beta \cdot \alpha \cdot S_E^l(k) \cdot H_l(k) + N_l(k)$$

where l is the OFDM symbol index, k is the carrier index. S_C and S_E are the Core and Enhanced PLP signals, respectively, α and β are scaling and normalizing factors as defined in Section 6.4 of

[3], $H_l(k)$ is the channel response of the kth carrier in the lth OFDM symbol, and $N_l(k)$ is the system noise. Note that when decoding the CL signal, the EL signal is simply treated as interference.

The CL decoding is performed using the output of the channel equalization followed by time de-interleaving and power de-normalization. For the derivation of input values for the channel decoding, the SNR of CL signal can be calculated as:

$$\gamma_C = \frac{P_C}{P_E + P_{ICI} + P_0}$$

where P_C is the power of the received CL signal, P_E is the power of the received EL signal, P_{ICI} is the power of inter-carrier-interference (ICI) generated by time-variation in OFDM signals, and P_0 is the power of system noise.

6.4.3 Core PLP Cancellation and Enhanced PLP Decoding

As shown in Figure 6.6, to decode the services delivered in EL, the received CL signal needs to be removed first, which requires the CL signal detection and re-generation. In general, the EL is used to deliver high throughput services that require high SNR conditions. Therefore, the CL signal cancellation needs to be quite accurate in order to meet the EL signal SNR requirements. Fortunately, when there is sufficient SNR to support the EL reception, the successful recovery of CL is guaranteed.

The CL signal cancellation can be directly affected by the channel estimation performance, where the power of the residual CL signal after cancellation is calculated as:

$$\gamma_{CLI} = \frac{\sigma_H^2}{\alpha^2}$$

where γ_{CLI} is the power of the cross-layer-interference (CLI), and σ_H^2 is the mean square error of the channel estimate. Therefore, the channel estimation for EL signal detection needs to be highly accurate in order to support the high SNR conditions. The CL signal cancellation can be performed as:

$$\tilde{R}_E^l(k) = R_l(k) - \alpha \cdot \tilde{S}_C^l(k) \cdot \widetilde{H}_l(k)$$

where $\tilde{R}_E^l(k)$ is the EL symbol after CL cancellation, $\tilde{S}_C^l(k)$ is the re-generated CL signal, and $\widetilde{H}_l(k)$ is the channel estimate. At higher SNR for EL signal decoding, it can be safely assumed that the CL signal can be perfectly detected.

Figure 6.8 shows three different methods for CL signal re-generation in the Core PLP canceller block of Figure 6.6. As shown in Figure 6.8, the CL signal can be re-generated by one of the following cases:

1) Whole codeword case: Information and the parities of LDPC and outer code are directly used from the Core Layer LDPC decoding block. LDPC re-encoding in the CL re-generation is not needed.
2) Information + outer code parity case: Information and the outer code parities are used from the Core Layer LDPC decoding block. Outer code (e.g., BCH) re-encoding in the CL re-generation is not needed.

3) Information only case: Only information is used from the Core Layer outer code decoding block. The entire re-encoding process (i.e., outer and LDPC re-encoding) in the CL re-generation is needed.

For fixed reception with a high SNR environment (e.g., 15 dB or higher), a Core PLP will be configured with a robust ModCod to have sufficient SNR headroom between the CL and EL decoding. In such cases, the CL decoding in a fixed reception area can be easily achieved with quasi error free (QEF) performance, and therefore, the CL signal re-generation can be performed using the (1) whole codeword case or (2) information + outer code parity case.

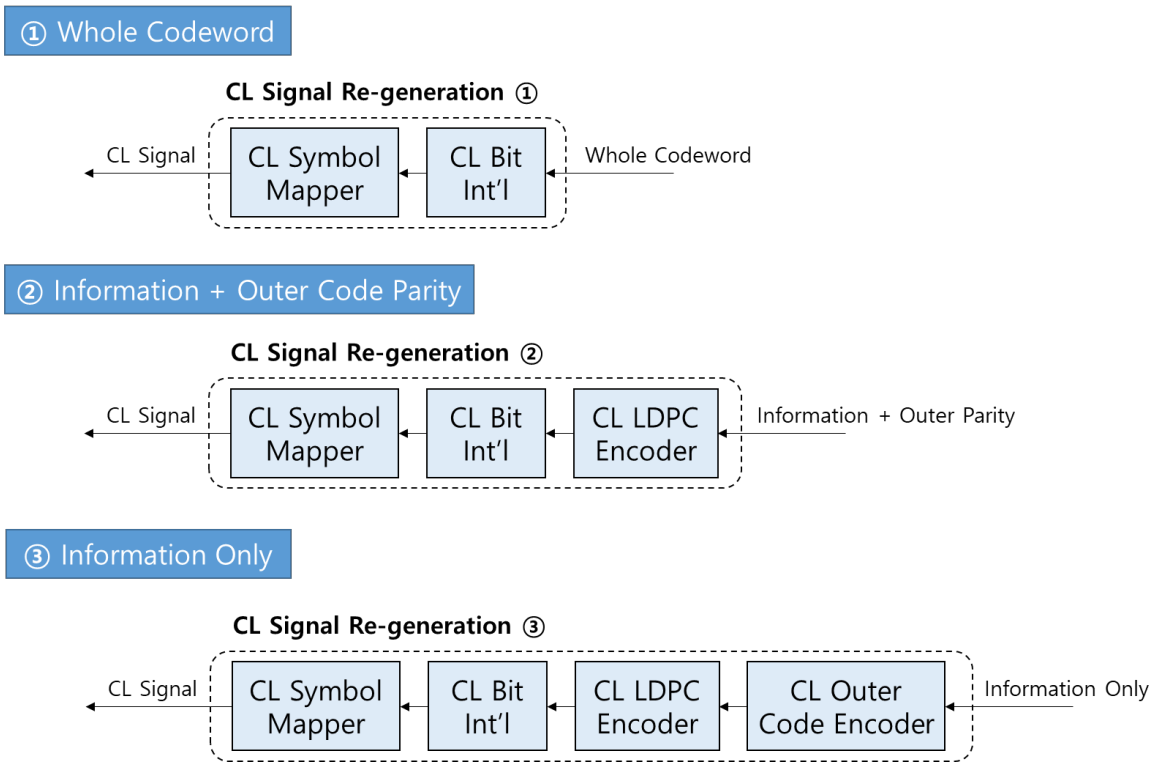

Figure 6.8 CL signal re-generation: (1) Whole codeword, (2) Information + outer code parity, and (3) Information-only cases.

After the CL signal cancellation, which is performed on the output of the channel equalization followed by the time de-interleaving and power de-normalization, EL decoding is performed using the output of CL signal cancellation. For the derivation of input values for the channel decoding, the SNR of EL signal can be calculated as:

$$\gamma_E = \frac{P_E}{P_0 + P_{CLI}} \approx \frac{P_E}{P_0}$$

where P_{CLI} is the power of CLI after the CL signal cancellation, and P_0 is the power of thermal noise.

6.5 Channel Decoding

The channel decoding process includes the de-mapping of NUCs, bit de-interleaving, and inner/outer decoding. After the de-mapping of NUCs, values which correspond to the bit-

interleaved codewords in the transmitter are fed to the bit de-interleaver, and then to the inner/outer decoders. Note that these values are often called channel values and they may be obtained in hard or soft decision form in receivers.

6.5.1 LLR De-Mapping for Non-Uniform Constellations

For optimal performance in decoding, the de-mapping of NUCs may be performed via soft decision such as Log Likelihood-Ratios (LLRs) calculation. In ATSC 3.0, two types of non-uniform constellations (NUC) are used: 1-dimensional (1D-NUC) and 2-dimensional (2D-NUC). When 1D-NUCs are used, the LLRs can be calculated using the well-known one-dimensional de-mapper described in Section 6.5.1.1. In the case of 2D-NUCs, the derivation of the LLRs is shown in Section 6.5.1.2.

6.5.1.1 De-Mapping for 1D-NUC

In the cases of 1D-NUCs (QPSK, 1024QAM and 4096QAM), a receiver may form LLRs as a function of two non-uniform 1-dimensional pulse amplitude modulation (PAM) constellations representing the in-phase (I) and quadrature (Q) components of the 1D-NUC. A 2-dimensional de-mapper can be also applied but with higher computational expense. Considering that a QAM constellation conveys m coded bits, i.e., each constellation component has $\sqrt{2^m}$ possible states, the structure of the de-mapper is shown in Figure 6.9.

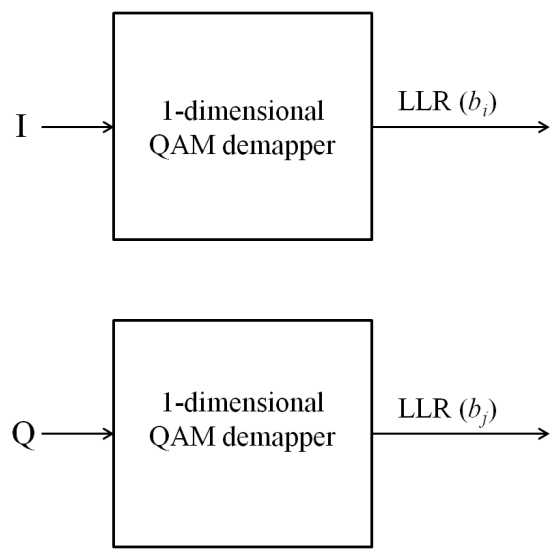

Figure 6.9 De-mapper structure for 1D-NUC.

6.5.1.1.1 Computation of Perfect LLR for 1D-NUC

Given the de-mapper structure of Figure 6.9, LLRs may be defined from the received constellation components I and Q. Note that index i refers to the bits de-mapped from the I components and index j from the Q components, and these expressions describe the LLRs for the ith bit b_i and jth bit b_j:

$$\mathrm{LLR}(b_i) = \ln\left(\frac{Pr(b_i = 1 \mid I)}{Pr(b_i = 0 \mid I)}\right)$$

$$\mathrm{LLR}(b_j) = \ln\left(\frac{Pr(b_j = 1 \mid Q)}{Pr(b_j = 0 \mid Q)}\right)$$

A positive LLR indicates that b_i or b_j was more probably transmitted as 1, a negative LLR that b_i or b_j was more probably transmitted as 0. A particular constellation point x is transmitted with coordinates I_x and Q_x. Ideally, the received constellation components I and Q are expected to be identical. In practice, however, they are different because the cells in which I and Q travel are subjected to amplitude-fading factors ρ_I and ρ_Q, respectively, and additionally to noise. Therefore, given that the point x was transmitted, the conditional probability distribution function (pdf) of receiving a particular I and Q can be expressed as:

$$p(I \mid x\ was\ transmitted) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(I-\rho_I I_x)^2}{2\sigma^2}}$$

$$p(Q \mid x\ was\ transmitted) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Q-\rho_Q Q_x)^2}{2\sigma^2}}$$

Considering reception of a particular bit b_i or b_j, if 1 was transmitted, then this implies that any of $\frac{\sqrt{2^m}}{2}$ possible states was transmitted, which depends on the state of the other $(\frac{m}{2} - 1)$ bits. If 0 was transmitted, then one of the other $\frac{\sqrt{2^m}}{2}$ possible states was transmitted. Let C_i^k denote the set of constellation points x for which the ith bit, b_i, takes the value k (0 or 1), and C_j^k denote the set of constellation points x for which the jth bit, b_j, takes the value k (0 or 1). The conditional pdf for the received values I and Q, given that b_i or b_j was transmitted as 1 is thus given by the following expression, under the assumption that all of the $\frac{\sqrt{2^m}}{2}$ possible transmitted states for which $b_i = 1$ and $b_j = 1$ are transmitted with equal probability. In other words, each of the other $(\frac{m}{2} - 1)$ bits takes the values 0 and 1 with equal frequency.

$$p(I \mid b_i = 1) = \frac{1}{\frac{\sqrt{2^m}}{2}\sqrt{2\pi}\sigma} \sum_{x \in \mathrm{C}_\mathrm{i}^1} e^{-\frac{(I-\rho_I I_x)^2}{2\sigma^2}}$$

$$p(Q \mid b_j = 1) = \frac{1}{\frac{\sqrt{2^m}}{2}\sqrt{2\pi}\sigma} \sum_{x \in \mathrm{C}_\mathrm{j}^1} e^{-\frac{(Q-\rho_Q Q_x)^2}{2\sigma^2}}$$

The conditional pdf for receiving I and Q, given that b_i *or* b_j was transmitted as 0 is the same except that the summation is taken over points $x \in C_i^0$ and $x \in C_j^0$. If Bayes' theorem is used, and further assuming that the transmitted bit b_i *or* b_j is itself equally likely to be 0 or 1, then the LLR can be obtained as:

$$\mathrm{LLR}(b_i) = \ln\left(\frac{Pr(b_i = 1 \mid I)}{Pr(b_i = 0 \mid I)}\right) = \ln\left(\frac{p(I \mid b_i = 1)}{p(I \mid b_i = 0)}\right)$$

$$= \ln\left(\frac{\sum_{x \in C_i^1} e^{-\frac{(I-\rho_I I_x)^2}{2\sigma^2}}}{\sum_{x \in C_i^0} e^{-\frac{(I-\rho_I I_x)^2}{2\sigma^2}}}\right)$$

$$\mathrm{LLR}(b_j) = \ln\left(\frac{Pr(b_j = 1 \mid Q)}{Pr(b_j = 0 \mid Q)}\right) = \ln\left(\frac{p(Q|b_j = 1)}{p(Q|b_j = 0)}\right)$$

$$= \ln\left(\frac{\sum_{x \in C_j^1} e^{-\frac{(Q-\rho_Q Q_x)^2}{2\sigma^2}}}{\sum_{x \in C_j^0} e^{-\frac{(Q-\rho_Q Q_x)^2}{2\sigma^2}}}\right)$$

6.5.1.1.2 Computation of Simplified LLR for 1D-NUC

The LLR computation may be simplified by applying the Max-Log approximation as:

$$\ln(e^{a_1} + \cdots + e^{a_k}) \approx \max_{i=1\ldots k}(a_i)$$

Then, the approximate LLR may be obtained as:

$$LLR(b_i) \approx \frac{1}{2\sigma^2}\left[\min_{x \in C_i^0}((I - \rho_I I_x)^2) - \min_{x \in C_i^1}((I - \rho_I I_x)^2)\right]$$

$$LLR(b_j) \approx \frac{1}{2\sigma^2}\left[\min_{x \in C_j^0}\left(\left(Q - \rho_Q Q_x\right)^2\right) - \min_{x \in C_j^1}\left(\left(Q - \rho_Q Q_x\right)^2\right)\right]$$

Note that this simplification of LLR computation for 1D-NUC may be suitable for hardware implementation due to low complexity.

6.5.1.2 De-Mapping for 2D-NUC

In the case of 2D-NUCs (16QAM, 64QAM and 256QAM), LLRs may be calculated as a 2-dimensional function. The structure of the de-mapper is shown in Figure 6.10.

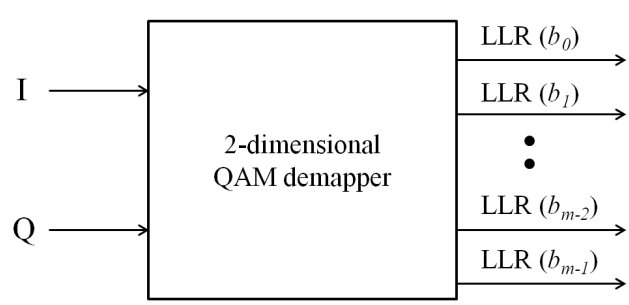

Figure 6.10 De-mapper structure for 2D-NUC.

6.5.1.2.1 Computation of Perfect LLR for 2D-NUC

Given the de-mapper structure of Figure 6.10, the LLRs may be defined from the received constellation I and Q, which describes the LLR for the ith bit b_i as:

$$\mathrm{LLR}(b_i) = \ln\left(\frac{Pr(b_i = 1 \mid I, Q)}{Pr(b_i = 0 \mid I, Q)}\right)$$

A positive LLR indicates that b_i was more probably transmitted as 1, and a negative value indicates that b_i was more probably transmitted as 0. A particular constellation point x is transmitted with coordinates I_x and Q_x. Ideally, the received constellation I and Q are expected to be identical. In practice, however, they are different because the cells in which I and Q travel are subjected to amplitude-fading factors ρ_I and ρ_Q, respectively, in addition to noise. Therefore, given that point x was transmitted, the pdf of receiving particular I and Q may be expressed as:

$$p(I, Q \mid x\ was\ transmitted) = \frac{1}{2\pi\sigma^2} e^{-\frac{(I-\rho_I I_x)^2 + \left(Q-\rho_Q Q_x\right)^2}{2\sigma^2}}$$

Considering reception of a particular bit b_i, if 1 was transmitted, then any of 2^{m-1} possible states is transmitted, which depends on the state of the other (m - 1) bits. If 0 was transmitted, then one of the other 2^{m-1} possible states was transmitted. Let C_i^j denote the set of constellation points x for which the ith bit, b_i, takes the value j (0 or 1). The conditional pdf for the received values I and Q, given that b_i is transmitted as 1 is thus given by the following expression, under the assumption that all of the 2^{m-1} possible transmitted states for which $b_i = 1$ are transmitted with equal probability. In other words, each of the other (m - 1) bits takes the values 0 and 1 with equal frequency.

$$p(I, Q \mid b_i = 1) = \frac{1}{2^m \pi\sigma^2} \sum_{x \in \mathrm{C}_\mathrm{i}^1} e^{-\frac{(I-\rho_I I_x)^2 + \left(Q-\rho_Q Q_x\right)^2}{2\sigma^2}}$$

The conditional pdf for receiving I and Q, given that b_i was transmitted as 0 is the same except that the summation is taken over points $x \in C_i^0$. If Bayes' theorem is used, and assuming that the transmitted bit b_i is itself equally likely to be 0 or 1, then the LLR may be obtained as:

$$\mathrm{LLR}(b_i) = \ln\left(\frac{Pr(b_i = 1 \mid I, Q)}{Pr(b_i = 0 \mid I, Q)}\right) = \ln\left(\frac{p(I, Q|b_i = 1)}{p(I, Q|b_i = 0)}\right)$$

$$= \ln\left(\frac{\sum_{x \in \mathrm{C}_\mathrm{i}^1} e^{-\frac{(I-\rho_I I_x)^2 + \left(Q-\rho_Q Q_x\right)^2}{2\sigma^2}}}{\sum_{x \in \mathrm{C}_\mathrm{i}^0} e^{-\frac{(I-\rho_I I_x)^2 + \left(Q-\rho_Q Q_x\right)^2}{2\sigma^2}}}\right)$$

6.5.1.2.2 Computation of Simplified LLR for 2D-NUC

The LLR computation may be simplified by applying the Max-Log approximation as:

$$\ln(e^{a_1} + \cdots + e^{a_k}) \approx \max_{i=1\ldots k}(a_i)$$

Then, the approximated LLR may be obtained as:

$$LLR(b_i) \approx \frac{1}{2\sigma^2}\left[\min_{x \in C_i^0}\left((I-\rho_I I_x)^2 + \left(Q-\rho_Q Q_x\right)^2\right) - \min_{x \in C_i^1}\left((I-\rho_I I_x)^2 + \left(Q-\rho_Q Q_x\right)^2\right)\right]$$

Note that this simplification of LLR computation for 2D-NUC may be suitable for hardware implementation due to low complexity.

6.5.2 Bit De-Interleaving

The bit de-interleaving is a reverse operation of the bit interleaving operation in the transmitter. Note that the bit de-interleaving in receivers is performed on channel values (e.g., multi-bit LLR values) while the bit interleaving in the transmitter is performed on individual codeword bits.

The channel values such as LLR values are first de-interleaved by the block de-interleaver. The type and configurations of the block de-interleaver should be the same as those of the block interleaver in the transmitter for correct decoding. The block de-interleaving can be performed using memory containers (e.g., columns for Type A block de-interleaver and rows for Type B block de-interleaver) based on the write/read operations. As shown in Figure 6.11, when the Type A and Type B block interleavers are implemented based on the column-writing/row-reading and row-writing/column-reading operations, respectively, the Type A and Type B block de-interleavers should be implemented based on the row-writing/column-reading and column-writing/row-reading operations, respectively. Note that the column and row operations can be implemented by switching each other.

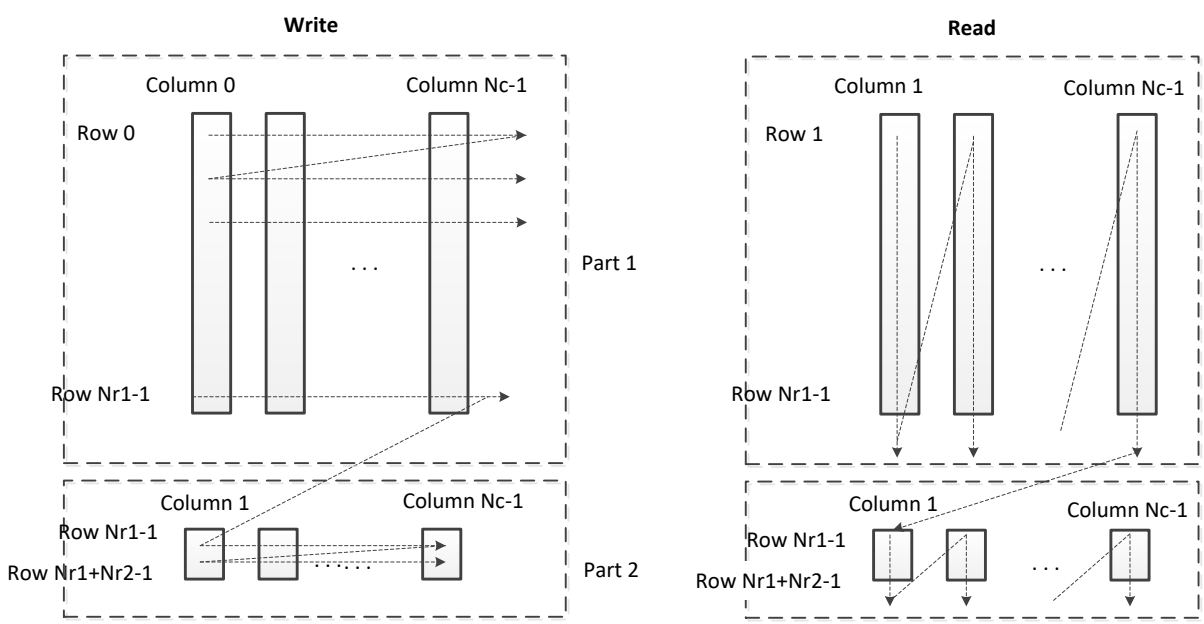

Figure 6.11 De-interleaving process of Type A block de-interleaver.

Similar to the block interleaver in the transmitter, the containers for the Type A and Type B block de-interleavers are divided into two parts: Part 1 and Part 2, and the configurations for Part 1 and Part 2 block de-interleaving are also determined based on each modulation format and code length used in the transmitter. In the case of Type A block de-interleaver, for example, if an LDPC code of length 64800 and 256QAM modulation are used in the transmitter, the numbers of rows

in Part 1 and Part 2 are 7920 and 180, respectively. If an LDPC code of length 16200 and 256QAM modulation are used in the transmitter, the numbers of rows in Part 1 and Part 2 are 1800 and 225, respectively. Note that the unit for de-interleaving is the channel values.

The number of columns of the block de-interleaver is determined by the modulation format used in the transmitter. In the case of Type A de-interleaver, the number of rows in Part 2 is obtained by $(\frac{N_{inner}}{N_c} - N_{r1})$ where N_c and N_{r1} denote the numbers of columns and rows in Part 1, respectively. If $\left(\frac{N_{inner}}{N_c} - N_{r1}\right) > 0$, the channel values corresponding to at least one bit group of transmitters are divided in at least two columns and de-interleaved in Part 2.

Following the block de-interleaver, the group-wise de-interleaver de-interleaves the output of the block de-interleaver. Since the group-wise de-interleaving is performed on a group basis, the block de-interleaved channel values are first split into multiple channel value groups. Note that each channel value group has the same size as the parallel factor of the LDPC codeword (i.e., 360) so as to correspond to a bit group (360 bits) in transmitters. The channel value groups are de-interleaved based on the equation and de-interleaving order defined in Section 6.2.2 and Annex B of [3], respectively. Since the de-interleaving order is a reverse of the interleaving order, the de-interleaving is also performed based on the pattern defined in Annex B of [3].

When a Type A LDPC code is used in the transmitter, the output of the group-wise de-interleaver is the input to the LDPC decoder. When a Type B LDPC code is used, the output of the group-wise de-interleaver is further parity de-interleaved, and the parity de-interleaved values are the input to the LDPC decoder.

6.5.3 Inner and Outer Decoding

For inner decoding, the LDPC decoding corresponding to the PCM is performed to correct errors over a physical channel. There are a number of LDPC decoding algorithms which may provide tradeoffs between the decoding performance and hardware complexity.

The sum-product algorithm (SPA) is known as the best performing and the most complex decoding algorithm. In order to reduce decoding complexity of the SPA, simplified decoding schemes can be used. First, the min-sum algorithm (MSA) is a simplified method that can greatly reduce decoding complexity compared to the SPA, but it may introduce substantial performance degradation in terms of bit error rate (BER) or frame error rate (FER). The modified MSAs can be used to improve the BER and FER performance of the MSA, and there are the normalized min-sum algorithm (NMSA) and offset min-sum algorithm (OMSA), which use normalization and offset terms for the output of check node operations, respectively.

Given the four different LDPC decoding algorithms (i.e., SPA, MSA, OMSA, and NMSA), the simulation performance of ATSC 3.0 LDPC codes in an AWGN channel was analyzed. For the NMSA and OMSA algorithms, decoding parameters for each combination of code rate and code length should be appropriately determined. Table 6.1 shows the optimized scaling and offset values of OMSA and NMSA that support the best decoding performance with extensive simulation.

Table 6.1 Optimized Offset and Scaling Values

Code Rate	16200 Bit		64800 Bit	
	Offset Value	Scaling Value	Offset Value	Scaling Value
2/15	0.31	0.9	0.46	0.63
3/15	0.47	0.65	0.48	0.63
4/15	0.47	0.67	0.46	0.62
5/15	0.5	0.65	0.48	0.64
6/15	0.5	0.8	0.51	0.8
7/15	0.57	0.78	0.44	0.73
8/15	0.57	0.76	0.6	0.77
9/15	0.6	0.78	0.54	0.75
10/15	0.6	0.72	0.57	0.75
11/15	0.6	0.73	0.56	0.73
12/15	0.6	0.72	0.58	0.72
13/15	0.63	0.65	0.6	0.72

The performance of ATSC 3.0 LDPC codes was analyzed given the theoretical thresholds and computer simulation results. Note that the theoretical thresholds of each code were calculated by Density Evolution algorithm, and the computer simulation results were obtained using 50 iterations for LDPC decoding. Note also that a threshold is defined as the theoretical performance limit that a given LDPC code can achieve when the code length goes to infinity. Table 6.2 and Table 6.3 show the performance differences from the Shannon limits to the theoretical thresholds and simulation results in the cases of ATSC 3.0 short and long codes, respectively. Note that the target FER for LDPC codes in computer simulation was set to 10^{-4} since the concatenation of LDPC and BCH codes provides overall FER performance lower than 10^{-6}.

Table 6.2 Performance Difference from the Shannon Capacity Limit to the Theoretical Thresholds and Simulation Results (Short Code Cases)

Code Rate	Performance Difference from Theoretical Thresholds (dB)				Performance Difference from Simulation Results (dB) @ FER=10^{-4}			
	SPA	MSA	OMSA	NMSA	SPA	MSA	OMSA	NMSA
2/15	0.5	2.47	0.78	0.96	1.35	2.09	1.58	1.69
3/15	0.5	2.07	0.9	0.86	1.14	2.52	1.49	1.46
4/15	0.56	1.85	0.86	0.82	1.1	2.07	1.37	1.34
5/15	0.41	1.61	0.69	0.71	0.93	2.18	1.35	1.34
6/15	0.24	1.66	0.58	1.1	0.91	2.09	1.18	2.32*
7/15	0.23	1.52	0.54	0.94	0.82	1.93	1.08	2.16*
8/15	0.16	1.37	0.41	0.8	0.74	1.78	0.95	1.96*
9/15	0.25	1.16	0.4	0.54	0.72	1.5	0.86	1.04*
10/15	0.15	1.12	0.32	0.49	0.65	1.49	0.77	1.44*
11/15	0.25	0.91	0.35	0.36	0.64	1.22	0.72	0.82*
12/15	0.19	0.8	0.28	0.29	0.62	1.12	0.68	0.75
13/15	0.23	0.75	0.31	0.29	0.62	1.08	0.68	0.71*

Table 6.3 Performance Difference from the Shannon Capacity Limit to the Theoretical Thresholds and Simulation Results (Long Code Cases)

Code Rate	Performance Difference from Theoretical Thresholds (dB)				Performance Difference from Simulation Results (dB) @ FER=10^{-4}			
	SPA	MSA	OMSA	NMSA	SPA	MSA	OMSA	NMSA
2/15	0.19	2.28	0.70	1.00	0.71	2.62	1.66	2.37
3/15	0.19	2.00	0.57	0.65	0.63	2.27	1.01	1.09
4/15	0.21	1.75	0.50	0.47	0.6	1.99	0.89	0.93
5/15	0.22	1.58	0.46	0.44	0.6	1.79	0.81	0.81
6/15	0.29	1.67	0.58	1.09	0.73	1.87	1.05	2.59*
7/15	0.25	1.23	0.43	0.45	0.58	1.39	0.74	0.76
8/15	0.24	1.38	0.44	0.76	0.58	1.56	0.84	2.03*
9/15	0.19	1.17	0.36	0.54	0.53	1.35	0.69	1.63*
10/15	0.18	1.08	0.33	0.47	0.5	1.25	0.65	1.05*
11/15	0.18	0.90	0.29	0.33	0.46	1.05	0.56	0.82*
12/15	0.17	0.82	0.27	0.30	0.43	0.99	0.52	0.58
13/15	0.17	0.68	0.24	0.25	0.41	0.85	0.48	0.52*

As shown in Table 6.2 and Table 6.3, the performance differences from the Shannon limits to the theoretical thresholds in the use of the SPA algorithm are 0.15 to 0.56 dB for short LDPC code cases, and 0.17 to 0.29 dB for long LDPC codes. This implies that the ATSC 3.0 short and long LDPC codes are well-designed to approach the Shannon capacity limits. The results also show that the long LDPC codes have approximately 0.15 to 0.64 dB better performance than the short LDPC codes because of length benefit. The SPA algorithm outperforms all other algorithms in the theoretical thresholds and computer simulation results for all cases, although it has the highest decoding complexity. The OMSA also provides stable and good decoding performance close to the SPA. Conversely, the NMSA has significant performance degradation for some cases marked by asterisks in Table 6.2 and Table 6.3. In those cases, the NMSA performs similarly or worse than the MSA. This is caused by NMSA's inevitable error floors due to the channel mismatch effect. To resolve this problem, BCH codes can be used in conjunction with LDPC codes. Therefore, it is recommended that an ATSC 3.0 transmitter use the concatenation of BCH outer code and LDPC inner code to support good decoding performance at ATSC 3.0 receivers.

Figure 6.12 shows the Shannon limit and the FER performance for each LDPC decoding algorithm for code rate 10/15 and code length 64800. As shown in this figure, the SPA shows the best performance while the MSA provides the worst performance. The performance of the OMSA and NMSA is similar in high FER region; however, the NMSA has a serious error floor at FER lower than 10^{-4} and can predict that there is a cross-over near FER = 10^{-5} between the MSA and NMSA. Conversely, the OMSA provides stable performance without any error floor in all FER regions. The decoding complexity of the OMSA is slightly higher than the NMSA, but it can provide a stable and superior coding performance in all the cases.

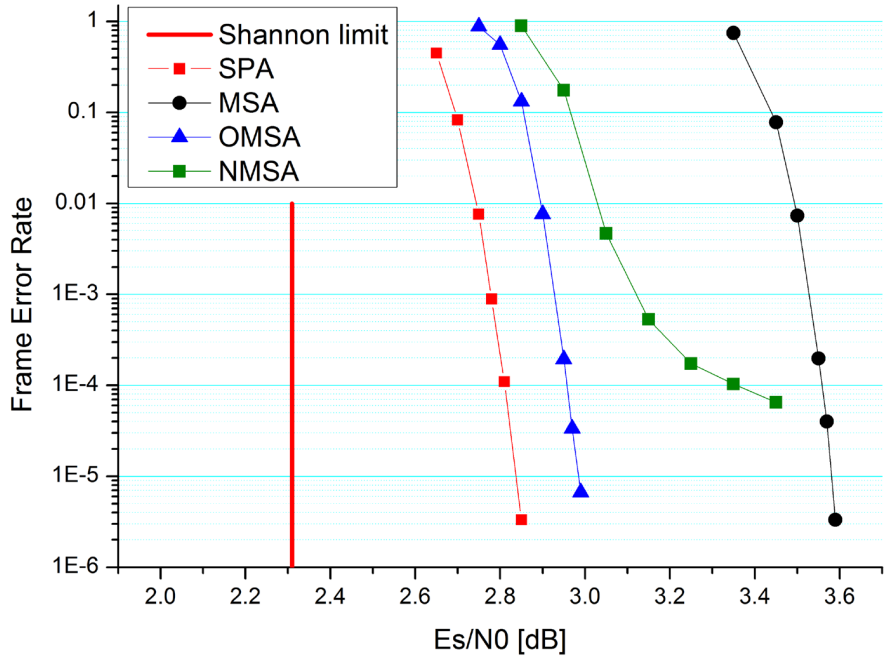

Figure 6.12 LDPC code performance based on various decoding algorithms (Code rate = 10/15, Code length = 64800, AWGN channel).

6.5.4 Decoding of L1 Signaling

Decoding of L1 signaling is the reverse operation of L1 protection operation in the transmitter. First of all, receivers can acquire the information about the L1-Basic Mode from the bootstrap. Since the length of L1-Basic signaling is fixed at 200 bits, receivers can easily calculate the number of cells occupied by L1-Basic signaling bits in the first Preamble symbol. After decoding L1-Basic, ATSC 3.0 receivers acquire the information about the L1-Detail Mode and the length of L1-Detail. Note that the L1 protection mode is determined based on combination of the LDPC code rate, modulation order, and the number of punctured bits to the number of shortened bits ratio (PSR), as shown in Table 4.9 of Section 4.2.10.

Since the L1 signaling is protected by a shortened and punctured LDPC code at the transmitter, the LDPC decoder at a receiver determines the length of shortened (i.e., zero-padding) and punctured bits in an LDPC codeword as well as appropriate shortening and puncturing patterns. For each L1 protection mode, the shortening and puncturing patterns are determined based on the LDPC code rate, modulation order, and PSR. Note that the shortening and puncturing patterns are defined on a bit-group basis.

Based on the determined shortening length and pattern, the decoder inserts a fixed channel value at the positions corresponding to shortened bits. Here, the fixed channel value is usually set to the maximum LLR value defined in the LDPC decoder since a zero-padding bit can be treated as a signal with LLR = ∞, theoretically. Similarly, based on the determined puncturing length and pattern, the decoder inserts another fixed channel value at the punctured positions. Here, the latter fixed channel value is usually set to LLR = 0 since a punctured bit is an erasure, i.e., a lost signal. When the repetition of parity bits is applied at the transmitter for L1-Basic Mode 1 and L1-Detail Mode 1, the decoder combines the LLR values for the repeated parity bits. The combining of the LLR values is simply achieved by adding the LLR values of repeated parity bits for the same parity.

Note that the length of the repeated parity bits is determined based on the size of concatenation of L1 signaling and BCH parity-check bits.

After the insertion of appropriate LLR values for the shortened and punctured bits and combining of LLR values for repeated parity bits, the LDPC decoding is performed. After the LDPC decoding and zero removal for the output of LDPC decoding, the decoder performs BCH decoding, and finally, the L1 signaling is obtained by de-scrambling the output of BCH decoding.

When the size of L1-Detail signaling exceeds a segmentation reference value K_{seg}, the L1-Detail signaling is segmented into multiple smaller blocks at the transmitter and the L1-Detail signaling is transmitted by one or more FEC Frames. Therefore, when applying the segmentation of L1-Detail signaling at the transmitter, ATSC 3.0 receivers combine the segmented multiple blocks based on the value K_{seg} after decoding each segmented block. K_{seg} for each L1-Detail mode is determined based on the LDPC code rate and PSR, as described in Section 5.2.5.2.

When additional parity is configured at the transmitter and the number of additional parity bits exceeds the number of punctured bits, the repeated bits are selected. Then, similar to the repetition of parity, the decoder combines the LLR values for the repeated parity. The combining of the LLR values is simply achieved by adding the LLR values of repeated parity bits for the same parity.

7. GUIDELINES FOR MOBILE SERVICES

As the ATSC 3.0 physical layer protocol offers flexibility, it allows broadcasters to choose a wide range of physical layer parameters from ultra-robust modes to high-capacity modes. This section provides a guideline for ATSC 3.0 mobile services, by taking into account ATSC 3.0 physical layer parameter choices enabling robust reception and lower power consumption. ATSC 3.0 mobile receivers that may have limited power resources and require lower power consumption (e.g., portable or handheld devices) should be able to support the physical layer parameter choices described in this section.

7.1 Input Formatting

The same Input Formatting guidelines as in Section 5.1.1 should be applied for ATSC 3.0 mobile services. When a complete delivered product is composed of multiple PLPs, ATSC 3.0 mobile receivers may decode only a portion of the delivered product. For example, when SHVC is used such that the Base Layer is carried in a robust PLP and the Enhancement Layer is carried in a high-capacity PLP (i.e., less robust), ATSC 3.0 mobile receivers may decode only the robust PLP.

7.2 Bit Interleaved Coding and Modulation (BICM)

7.2.1 BICM for Data Payload

The recommended ModCod combinations for ATSC 3.0 mobile services are described in Table 7.1. Note that Table 7.1 defines a subset of the N_{inner} = 16200 bits (i.e., short codes) ModCod combinations of Table 4.8 with the higher code rates of 12/15 and 13/15, as well as the higher-order modulation of 256QAM, being omitted due to lower robustness. It is also recommended that BCH be always used as the outer code in order to provide additional error correction capability and prevent possible error floors. Although a check mark ✔ in Table 7.1 represents the mandatory ModCod combinations (see Section 4.2.9), it is recommended that all the ModCod combinations described in Table 7.1 be implemented in ATSC 3.0 mobile receivers as well as ATSC 3.0 transmitters for the full range of flexibility within this limited set of Table 7.1.

Table 7.1 Recommended ModCod Combinations for ATSC 3.0 Mobile Services
(N_{inner} = 16200 bits, i.e., Short Codes)

Code Rate/ Constellation	2/ 15	3/ 15	4/ 15	5/ 15	6/ 15	7/ 15	8/ 15	9/ 15	10/ 15	11/ 15
QPSK	✔	✔	✔	✔	✔	✔	✔	✔		
16QAM				✔	✔	✔	✔			✔
64QAM				✔	✔	✔	✔	✔	✔	✔

7.2.2 Protection for L1-Signaling

The recommended L1 protection modes for ATSC 3.0 mobile services are L1-Basic Mode 1, Mode 2, or Mode 3, and L1-Detail Mode 1, Mode 2, or Mode 3. Additional parity may be used to further enhance time diversity of L1-Detail. When additional parity is used, it is also recommended that L1-Detail Mode 1, Mode 2, or Mode 3 be used.

7.3 Framing and Interleaving

7.3.1 Time Interleaving

7.3.1.1 Time Interleaver Modes

It is recommended that each PLP intended for an ATSC 3.0 mobile service be configured with either the CTI mode (see Section 7.1.4 of [3]) or the HTI mode (see Section 7.1.5 of [3]). When a complete delivered product for an ATSC 3.0 mobile service is configured with a single Core PLP that has a constant cell rate, it is recommended that the CTI mode with the maximum depth (1024 rows in the non-extended mode) be used to maximize time diversity. When a complete delivered product of an ATSC 3.0 mobile service is configured with multiple Core PLPs (up to four), it is recommended that the HTI mode be used.

7.3.1.2 Time Interleaver Size

As the maximum size of time interleaver memory for a single, complete delivered product is 2^{19} cells (see Section 7.1.2 of [3]), ATSC 3.0 mobile receivers should be able to support up to 2^{19} cells for time de-interleaving.

7.3.2 Framing

7.3.2.1 Frame Length

It is recommended that ATSC 3.0 frame length be set less than or equal to 250 ms when fast service acquisition is required. It is also recommended that both time-aligned frames and symbol-aligned frames be supported for ATSC 3.0 mobile services.

7.3.2.2 PLP Multiplexing

It is recommended that all the multiplexing techniques that the ATSC 3.0 physical layer protocol offers (i.e., TDM, SSM, FDM, and LDM) be supported for ATSC 3.0 mobile services. When a multiplexing technique is used, ATSC 3.0 mobile receivers should be able to decode PLP(s) that are intended for mobile service(s). For example, when TDM is used and two subframes are configured to deliver mobile and fixed services, ATSC 3.0 mobile receivers will decode the first subframe that is intended for mobile services (see Section 5.3.2). When LDM is used, ATSC 3.0 mobile receivers will decode Core PLP(s) intended for mobile services (see Section 5.2.4).

7.3.3 Frequency Interleaving

It is recommended that the frequency interleaver address generation schemes for the 8K or 16K FFT sizes be used. The frequency interleaver address generation scheme for the 32K FFT size should not be used as the 32K FFT size is not recommended for ATSC 3.0 mobile services.

7.4 Waveform Generation

7.4.1 Pilot Insertion

It is recommended that scattered pilots for ATSC 3.0 mobile services always use $D_y = 2$. The recommended choices of scattered pilot patterns for ATSC 3.0 mobile services are described in Table 7.2. Note that configuration combinations which are not allowed are indicated by N/A.

The pilot boost signaled by **L1B_first_sub_scattered_pilot_boost** and **L1D_scattered_pilot_boost** may be used for improved channel estimation performance in ATSC 3.0 mobile services. Details of the pilot boost recommendations are described in Section 4.2.4.

Table 7.2 Recommended Scattered Pilot Patterns for ATSC 3.0 Mobile Services

GI Pattern	Samples	8K FFT	16K FFT
GI1_192	192	SP32_2, SP16_2	SP32_2
GI2_384	384	SP16_2, SP8_2	SP32_2, SP16_2
GI3_512	512	SP12_2, SP6_2	SP24_2, SP12_2
GI4_768	768	SP8_2, SP4_2	SP16_2, SP8_2
GI5_1024	1024	SP6_2, SP3_2	SP12_2, SP6_2
GI6_1536	1536	SP4_2	SP8_2, SP4_2
GI7_2048	2048	SP3_2	SP6_2, SP3_2
GI8_2432	2432	*N/A*	SP6_2, SP3_2
GI9_3072	3072	*N/A*	SP4_2
GI10_3648	3648	*N/A*	SP4_2
GI11_4096	4096	*N/A*	SP3_2

7.4.2 Inverse Fast Fourier Transform (IFFT)

It is recommended that 8K or 16K FFT sizes be used for ATSC 3.0 mobile services. 32K FFT size should not be used for ATSC 3.0 mobile services due to the narrower carrier spacing. Note that this recommendation applies to both Preamble and subframe symbols.

7.4.3 Guard Interval

The recommended guard interval choices for ATSC 3.0 mobile services are described in Table 7.3. Note that the guard interval choices in Table 7.3 are only listed for 8K and 16K FFT sizes. Note also that guard intervals allowed for each FFT size are indicated by a check mark ✔, and disallowed guard intervals are indicated by N/A.

Table 7.3 Recommended Guard Intervals for ATSC 3.0 Mobile Services

GI Pattern	Duration in Samples	8K FFT	16K FFT
GI1_192	192	✔	✔
GI2_384	384	✔	✔
GI3_512	512	✔	✔
GI4_768	768	✔	✔
GI5_1024	1024	✔	✔
GI6_1536	1536	✔	✔
GI7_2048	2048	✔	✔
GI8_2432	2432	*N/A*	✔
GI9_3072	3072	*N/A*	✔
GI10_3648	3648	*N/A*	✔
GI11_4096	4096	*N/A*	✔

Annex A: System Performance

A.1 INTRODUCTION

This section describes the system performance of the ATSC 3.0 physical layer including BICM, OFDM, and time and frequency interleavers in various channel conditions. The system performance was measured through computer simulations, laboratory tests and field tests.

Note that the computer simulation results shown in Annex A are the theoretical limits that each ModCod combination in ATSC 3.0 can achieve. The laboratory and field test results which were obtained through professional ATSC 3.0 equipment show possible implementation losses compared to the theoretical limits. Note also that when pilot boosting is used, the measured values can be re-computed by adding the power reduction values shown in Table B.4.1.

A.2 CHANNEL MODELS

The system performance was measured under three different channel models such as AWGN, Rician (DVB-F1), and Rayleigh (DVB-P1) channel models. Rician and Rayleigh channel models are commonly used to simulate fixed reception and portable reception, respectively. Rician channel consists of one main line-of-sight (LOS) path and 20 multipaths without Doppler shift. Rayleigh channel has one main non-LOS path and 19 multipaths without Doppler shift. The details of each channel model are described in [11].

A.3 SIMULATION, LABORATORY TEST AND FIELD TEST RESULTS

The system performance of the ATSC 3.0 physical layer was measured through computer simulations, laboratory tests, and field tests. In computer simulations and laboratory tests, the channel models were accurately simulated and reflected. In the real field tests, the reception locations were selected by measuring the real channel conditions that were close to the channel models used for computer simulations and laboratory tests. Nevertheless, note that the Rician-like channel in the field environment was worse than the Rician model used for computer simulation and laboratory test. And the Rayleigh-like channel in the field was milder than the Rayleigh model used for computer simulation and laboratory test. The detailed system parameter choices for the ModCod performance results are shown in Table A.3.1. The system performance results in AWGN, Rician, and Rayleigh channels are shown in Table A.3.2, Table A.3.3, and Table A.3.4, respectively.

Table A.3.1 System Parameters for Simulations, Laboratory Tests, and Field Tests

Frame Length (Symbol Aligned Mode)		255.33ms (Including Bootstrap)
Bandwidth		6 MHz
Preamble Parameters	FFT Size	16K
	Guard interval	GI7_2048
	Pilot Pattern	SP_D_x = 3
	Signaling Protection	L1-Basic Mode 1
	# of Preamble Symbols	1
Payload OFDM Parameters	FFT Size	16K
	Guard interval	GI7_2048
	Pilot pattern	SP_ D_x = 3, SP_D_y = 4
	Pilot boosting	No pilot boosting
	# of payload symbols	94
	Time interleaver	CTI (1024 depth)
	Frequency interleaver	On
Payload BICM Parameters	Inner/outer code	LDPC 2/15~13/15 (64800, 16200) /BCH
	Constellation	QPSK ~ 4096QAM

Table A.3.2 Required C/N for BER = 10^{-6} After LDPC and BCH Decoding Under AWGN Channel

			2/15	3/15	4/15	5/15	6/15	7/15	8/15	9/15	10/15	11/15	12/15	13/15
QPSK	Long	Lab test	-4.0	-3.1	-2.0	-1.0	0.5	0.9	2.0	2.8	3.7	4.2	5.1	6.0
		Simulation	-6.1	-4.2	-2.8	-1.6	-0.4	0.4	1.3	2.1	2.9	3.7	4.6	5.6
		Field test	-3.9	-2.9	-1.9	-0.7	0.7	1.1	2.2	2.9	3.9	4.4	5.3	6.2
	Short	Lab test	-4.2	-2.8	-1.6	-0.7	0.7	1.4	2.1	2.8	3.7	4.2	5.1	6.1
		Simulation	-5.5	-3.7	-2.3	-1.2	-0.2	0.6	1.5	2.3	3.1	3.9	4.8	5.8
		Field test	-4.1	-2.7	-1.5	-0.5	0.9	1.6	2.3	2.9	3.9	4.5	5.3	6.3
16QAM	Long	Lab test	-1.8	0.7	2.1	3.6	5.5	5.9	7.2	8.3	9.2	10.1	11.1	12.3
		Simulation	-2.6	-0.1	1.6	2.9	4.4	5.4	6.5	7.5	8.5	9.6	10.7	11.9
		Field test	-1.6	0.9	2.4	3.7	5.7	6.1	7.3	8.5	9.4	10.5	11.4	12.6
	Short	Lab test	-1.4	1.0	2.6	3.8	5.5	6.3	7.4	8.3	9.4	10.2	11.3	12.5
		Simulation	-2.1	0.4	2.0	3.3	4.6	5.7	6.6	7.7	8.7	9.8	10.9	12.2
		Field test	-1.3	1.2	2.9	3.9	5.8	6.5	7.6	8.6	9.6	10.5	11.8	12.8
64QAM	Long	Lab test	0.5	3.0	4.9	6.5	8.9	9.5	11.3	12.6	13.8	15.0	16.3	17.6
		Simulation	-0.2	2.4	4.3	6.1	7.8	9.1	10.5	11.7	13.0	14.4	15.7	17.3
		Field test	0.6	3.2	5.2	6.9	9.2	9.6	11.4	12.8	13.9	15.2	16.6	17.7
	Short	Lab test	0.9	3.5	5.3	6.9	9.1	10.4	11.7	12.7	14.0	15.1	16.4	17.9
		Simulation	0.4	2.8	4.7	6.8	8.1	9.5	10.7	12.0	13.2	14.7	16.0	17.4
		Field test	1.2	3.7	5.5	7.1	9.3	10.6	11.9	12.9	14.2	15.5	16.5	18.1
256QAM	Long	Lab test	2.3	4.8	7.2	9.1	11.7	12.7	14.9	16.5	18.3	19.7	21.3	23.0
		Simulation	1.7	4.4	6.7	8.7	10.9	12.2	14.1	15.7	17.3	18.9	20.6	22.4
		Field test	2.5	5.0	7.3	9.3	11.8	12.9	15.0	16.8	18.5	19.9	21.6	23.4
	Short	Lab test	2.7	5.3	7.8	9.5	11.8	13.8	15.4	16.6	18.4	19.8	21.5	23.3
		Simulation	2.4	4.8	7.2	9.0	11.1	12.8	14.4	16.1	17.5	19.2	20.9	22.8
		Field test	2.9	5.6	8.1	9.8	12.1	14.1	15.6	16.8	18.5	19.9	21.7	23.4
1024QAM	Long	Lab test	3.8	6.8	9.4	11.7	14.4	16.0	18.5	20.6	22.5	25.3	26.4	28.4
		Simulation	3.4	6.3	8.9	11.2	13.8	15.4	17.7	19.7	21.6	23.6	25.7	27.8
		Field test	4.1	7.2	9.7	11.9	14.5	16.4	18.7	20.8	22.8	25.2	26.6	28.5
4096QAM	Long	Lab test	5.2	8.5	11.5	14.1	17.0	19.1	21.7	24.0	26.5	28.7	30.8	33.4
		Simulation	4.7	8.0	10.8	13.6	16.3	18.3	20.9	23.3	25.8	28.1	30.5	33.1
		Field test	5.4	8.8	11.9	14.2	17.1	19.3	22.1	24.2	26.7	28.9	31.5	34.0

Table A.3.3 Required C/N for BER = 10^{-6} After LDPC and BCH Decoding Under Rician Channel

			2/15	3/15	4/15	5/15	6/15	7/15	8/15	9/15	10/15	11/15	12/15	13/15
QPSK	Long	Lab test	-4.3	-2.9	-1.8	-0.8	0.9	1.2	2.4	3.2	4.1	4.7	5.7	6.8
		Simulation	-6.0	-4.2	-2.7	-1.5	-0.2	0.7	1.6	2.5	3.4	4.2	5.2	6.4
		Field test	-3.7	-2.3	-1.2	0.0	1.4	2.1	3.2	4.2	5.1	5.8	7.0	8.7
	Short	Lab test	-3.9	-2.7	-1.4	-0.4	1.1	1.7	2.5	3.2	4.2	4.8	5.8	7.0
		Simulation	-5.5	-3.6	-2.2	-1.1	0.1	1.0	1.8	2.7	3.5	4.4	5.4	6.6
		Field test	-3.6	-2.1	-0.9	0.4	1.8	2.3	3.2	3.9	5.0	6.1	7.2	8.5
16QAM	Long	Lab test	-1.6	0.9	2.4	3.8	5.8	6.2	7.7	8.9	9.8	10.8	11.9	13.1
		Simulation	-2.5	0.1	1.8	3.2	4.7	5.6	6.8	7.9	9.0	10.1	11.2	12.6
		Field test	-1.0	1.5	2.7	4.5	6.1	6.9	8.6	9.9	10.6	11.8	13.1	14.4
	Short	Lab test	-1.1	1.1	2.9	4.1	6.0	6.9	7.9	8.8	9.9	10.9	11.9	13.4
		Simulation	-2.0	0.6	2.2	3.6	4.9	5.9	7.0	8.1	9.2	10.3	11.4	12.9
		Field test	-0.5	1.8	3.7	4.8	6.7	7.5	8.6	9.8	11.3	12.2	13.6	14.9
64QAM	Long	Lab test	0.7	3.3	5.1	7.0	9.4	10.0	11.9	13.2	14.5	15.7	17.1	18.6
		Simulation	0.1	2.6	4.6	6.4	8.2	9.3	10.9	12.1	13.5	14.9	16.2	17.8
		Field test	1.3	3.8	5.9	7.5	10.0	10.7	12.4	14.0	15.3	16.4	18.0	19.5
	Short	Lab test	1.2	3.8	5.7	7.2	9.6	11.0	12.2	13.4	14.6	15.8	17.2	18.7
		Simulation	0.6	3.1	5.1	7.2	8.5	9.9	11.1	12.4	13.7	15.1	16.5	18.1
		Field test	2.0	4.5	6.6	8.1	10.5	11.8	13.1	14.5	15.7	17.2	18.7	20.5
256QAM	Long	Lab test	2.6	5.2	7.6	9.6	12.3	13.3	15.6	17.1	19.1	20.2	21.8	23.5
		Simulation	2.0	4.7	7.0	9.0	11.2	12.6	14.5	16.1	17.7	19.4	21.0	22.9
		Field test	3.3	6.3	8.3	10.4	13.1	14.1	16.5	18.1	19.5	21.3	23.1	24.8
	Short	Lab test	3.1	5.8	8.3	10.0	12.6	14.4	16.3	17.3	18.8	20.3	21.9	23.8
		Simulation	2.5	5.2	7.6	9.4	11.4	13.1	14.8	16.5	18.0	19.7	21.3	23.3
		Field test	3.9	6.6	9.1	10.9	13.4	15.6	17.3	18.1	20.2	21.5	23.3	25.2
1024QAM	Long	Lab test	4.2	7.2	9.9	12.2	15.0	16.7	18.9	21.0	23.0	25.9	27.6	29.7
		Simulation	3.5	6.6	9.3	11.6	14.2	15.8	18.0	20.2	22.0	24.1	26.2	28.3
		Field test	5.1	8.5	10.9	13.2	15.9	17.6	20.0	22.2	23.9	26.5	28.3	30.8
4096QAM	Long	Lab test	5.6	8.9	12.1	14.8	17.4	19.9	22.2	25.0	27.8	29.9	32.9	35.8
		Simulation	5.0	8.3	11.2	14.0	16.7	18.8	21.4	23.8	26.2	28.4	31.0	33.5
		Field test	6.6	9.6	13.1	15.8	18.7	21.0	23.4	26.0	28.9	31.0	33.9	37.1

Table A.3.4 Required C/N for BER = 10^{-6} After LDPC and BCH Decoding Under Rayleigh Channel

			2/15	3/15	4/15	5/15	6/15	7/15	8/15	9/15	10/15	11/15	12/15	13/15
QPSK	Long	Lab test	-3.1	-1.8	-0.4	1.1	2.9	3.1	4.4	5.6	6.6	7.9	9.7	12.0
		Simulation	-5.7	-3.7	-2.0	-0.6	0.8	1.8	3.2	4.3	5.5	6.8	8.5	10.7
		Field test	-3.9	-2.5	-1.0	0.1	1.7	2.2	3.5	4.6	5.6	6.6	8.3	10.1
	Short	Lab test	-3.4	-1.4	0.2	1.6	3.0	3.5	4.8	5.7	6.9	8.1	10.0	12.6
		Simulation	-5.1	-3.1	-1.5	-0.2	1.0	2.3	3.3	4.5	5.7	7.2	8.9	11.2
		Field test	-4.0	-2.1	-0.7	0.4	1.8	2.3	3.3	4.4	5.6	6.5	8.5	10.6
16QAM	Long	Lab test	-0.1	2.6	3.7	5.0	7.3	7.9	9.6	11.1	12.5	13.6	15.4	17.7
		Simulation	-1.9	0.7	2.6	4.2	5.9	7.1	8.5	9.8	11.3	12.7	14.3	16.6
		Field test	-1.2	1.6	2.9	4.4	6.5	7.1	8.7	10.2	11.3	12.0	14.2	15.7
	Short	Lab test	0.5	2.8	4.1	5.7	7.5	8.6	9.8	11.2	12.6	13.9	15.7	18.2
		Simulation	-1.2	1.3	3.3	4.6	6.3	7.5	8.8	10.1	11.5	13.0	14.7	17.0
		Field test	-0.5	1.8	3.2	4.6	6.3	7.5	8.7	10.0	11.2	12.5	14.1	16.5
64QAM	Long	Lab test	2.7	4.4	6.5	8.4	11.0	11.8	13.7	15.4	16.9	18.4	20.3	22.7
		Simulation	0.8	3.5	5.8	7.6	9.8	11.1	12.7	14.1	15.6	17.4	19.2	22.6
		Field test	1.6	3.7	5.7	7.8	9.9	10.7	13.0	14.6	15.9	16.6	18.5	21.1
	Short	Lab test	2.9	4.9	7.1	8.9	11.2	12.7	14.3	15.5	17.0	18.5	20.5	23.2
		Simulation	1.5	4.1	6.4	8.1	10.0	11.5	13.0	14.4	15.9	17.6	19.5	22.0
		Field test	1.6	4.0	6.0	8.1	10.1	11.5	13.2	14.4	16.0	17.4	19.6	21.7
256QAM	Long	Lab test	4.1	6.8	9.2	11.3	14.2	15.3	17.7	19.3	21.3	23.0	24.2	27.5
		Simulation	2.8	5.8	8.3	10.5	13.0	14.4	16.4	18.1	20.0	21.7	23.8	26.4
		Field test	2.9	6.0	8.1	10.5	13.3	14.5	16.4	18.0	20.3	21.8	23.2	26.0
	Short	Lab test	4.5	7.4	10.0	11.9	14.3	16.7	18.4	19.4	21.4	23.1	24.5	28.0
		Simulation	3.6	6.5	9.1	11.1	13.3	15.1	16.8	18.6	20.2	22.2	24.2	26.8
		Field test	3.5	6.6	9.2	11.0	13.5	15.4	17.3	18.3	20.2	21.8	23.8	26.4
1024QAM	Long	Lab test	5.4	8.8	11.6	14.0	17.1	18.7	21.3	23.7	25.9	27.7	29.6	32.2
		Simulation	4.7	7.9	10.9	13.1	15.9	17.7	20.2	22.4	24.3	26.4	28.6	31.3
		Field test	5.0	8.2	11.1	12.9	16.1	17.9	20.2	22.8	24.8	26.8	29.6	32.0
4096QAM	Long	Lab test	6.9	10.6	13.7	16.5	20.0	22.0	24.6	27.8	29.8	32.5	35.2	38.0
		Simulation	6.1	9.8	12.9	15.7	18.9	20.9	23.6	26.4	28.6	31.1	33.6	36.3
		Field test	6.5	10.0	12.9	16.1	18.9	21.0	23.9	26.9	29.5	32.5	35.8	38.9

Annex B: ATSC 3.0 Receiver C/N Model

B.1 INTRODUCTION

This section describes a method of calculating the ATSC 3.0 receiver C/N specification for different modulation parameter settings, assuming that a receiver implements channel estimation in the time domain using linear time interpolation. This method has been adopted for DVB-T2 in Clause 3.4.10.3 of [7], Clause 2.5.4 of [8] and in [9]. The description of the boosted pilot correction factor calculation is based on Clause 14.5 of [10].

Note that the performance values shown in Annex B are computed based on the assumptions of receiver implementation losses, concerning the worst possible reception case. Therefore, the performance results of a receiver implemented according to the specification [3] are expected to be better than the values shown in Annex B.

B.2 METHOD

The overall receiver tuner input C/N can be expressed by the following equation:

$$C/N = -10\, Log_{10}\left[10^{-(C/N_{raw} + A + \Delta_{BP} + IM)/10} - 10^{P_x/10}\right] \; [dB] \qquad \text{[eq.1]}$$

where

C/N_{raw} = required raw BICM C/N for BER=10^{-6} after LDPC decoding (see Table 4.3 to Table 4.6 in Section 4.2.9 for these values)

A = 0.5dB. The additional C/N in dB to achieve QEF BER=10^{-11} after LDPC decoding

Δ_{BP} = correction for pilot boosting as defined in [8] (see Table B.4.1 for these values)

IM is the Implementation Margin. This is the C/N loss due to real channel estimation (Δ_{RCE}), plus various other losses such as imperfect LDPC decoding, synchronization, demodulator fixed point losses and other imperfections not considered part of the back-stop noise. These other losses are estimated to be approximately 0.5 dB.
Thus IM = Δ_{RCE} + 0.5 dB. Δ_{RCE} is estimated in [eq. 6] below.

P_x = backstop or excess noise of transmitter and receiver. In this document, the transmitter contribution has been set to -50 dBc, representing a good quality transmitter with minimal contribution to the overall P_x. The receiver contribution has been set to the standard value of -33 dBc for QPSK, 16QAM, 64QAM and 256QAM. A tighter value of -38dBc was used for 1024QAM and 4096QAM. For further information on P_x see the note below.

Note on P_x:

In the transmitter, the contribution to P_x represents the non-ideal characteristics of the transmitted signal, such as the presence of intermodulation products.

In a receiver, P_x corresponds mainly to the effect of the noise floor of the tuner plus local oscillator phase noise and quantization effects of the demodulator Analog to Digital Converter (ADC). These have the effect of raising the required receiver tuner input C/N so that the demodulator sees the correct C/N to demodulate the signal. P_x is more dominant at high values of

C/N_{raw} which when combined with the other terms of [eq. 1] start to approach the value of $-P_x$. The value of P_x for 4096QAM needs more verification from actual receiver measurements when they become available but for the time being it is set to the same value as 1024QAM to allow comparisons of performance requirements. Note that for a Rayleigh channel, even -38 dBc is not sufficient for 4096QAM 13/15 code rate because the other terms of the C/N in [eq. 1] exceed 38 dBc in a Rayleigh channel, that is, a value better than -38 dBc might be needed for 4096QAM if used in a Rayleigh channel.

B.2.1 Calculation of Boosted Pilot Correction Factor

The ideal E_s/N_o or C/N_{raw} simulation values are for BICM only and do not take into account the increased signal power due to boosted pilots. This means the receiver C/N specification is increased by a small amount Δ_{BP} to compensate for the fact that the data carriers are of lower power compared to the total signal power.

$$\Delta_{BP} = 10\, Log_{10}\left(\frac{Total\ Symbol\ Power}{Non\ Boosted\ Symbol\ Power}\right)$$

So:

$$\Delta_{BP} = 10\, Log_{10}\left(\frac{Total\ Symbol\ Power}{\mathrm{NoC}}\right) \qquad \text{[eq. 2]}$$

where:

Total symbol power is given in Tables I.2.1 to I.2.5 of the ATSC 3.0 physical layer specification A/322 [3]. The total symbol power given in these tables also varies with the amplitude value of the scattered pilot boost factor **L1D_scattered_pilot_boost** as listed in Table 9.15 of [3].

NoC = number of useful carriers/symbol (each carrier has a nominal unboosted power of 1). NoC varies with the value of C_{red_coeff} so that must also be taken into account. NoC is given in Table 7.1 of [3]. Pre-calculated values of Δ_{BP} are given in Table B.4.1.

B.2.2 Estimation of Channel Estimation Loss

In a practical receiver, each received data carrier is divided by the channel frequency response estimate at that particular carrier frequency. However, errors in the interpolation of the scattered pilots used to calculate the channel estimate and the effects of added noise cause the resulting data carrier estimate to contain an error, which increases the implementation loss of receivers. The accuracy of this noisy channel estimate derived from the scattered pilots depends on the degree of pilot boosting as well as the accuracy of scattered pilot interpolation. However, summing the contributions from different pilots in the interpolator also has the effect of averaging the noise slightly resulting in a small noise gain. This noise gain can also be taken into account in estimating the channel estimation loss.

Assume that the interpolator changes the noise-power in the pilot-based measurements by a 'noise-gain' factor f_{INT}, so that $f_{INT} < 1$ corresponds to a reduction in the noise power. Assuming that the noise added to all carriers, pilot and data, has the same power, it follows that the SNRs (in linear power ratios) for the data carriers and interpolated channel estimates can be related as follows:

$$SNR_{channel-estimate} = \frac{L1D_scattered_pilot_boost^2 \ SNR_{data}}{f_{INT}} \qquad \text{[eq. 3]}$$

f_{INT} actually varies from carrier to carrier due to the different phases within the two-dimensional frequency/time structure of the scattered pilot patterns, but it is possible to estimate an average value of f_{INT}.

B.2.3 Estimation of Average Value of f_{INT}

Assuming the pilot patterns are chosen in relation to the guard intervals to allow operation approaching the Nyquist limit, there is very little noise gain in the frequency interpolator. Therefore, it is assumed that the noise-gain contribution from frequency interpolation is approximately unity and can be ignored.

The time interpolator is assumed to be a linear interpolator which does offer some noise gain benefit depending upon the value of D_y.

1) For scattered pilot patterns with D_y of 2, the two phases of the interpolator will be either lying on a pilot, or be half way between two pilots. Therefore the average noise gain is:

$$f_{INT} = \frac{1^2 + (0.5^2 + 0.5^2)}{2} = \mathbf{0.75} \qquad \text{[eq. 4]}$$

2) For scattered pilot patterns with D_y of 4, the four phases of the interpolator will be either lying on a pilot, or be quarter or half way between two pilots. Therefore the average noise gain f_{INT} is:

$$f_{INT} = \frac{1^2 + (0.75^2 + 0.25^2) + (0.5^2 + 0.5^2) + (0.25^2 + 0.75^2)}{4} = \mathbf{0.6875} \qquad \text{[eq. 5]}$$

The SNR of the channel estimates will affect the SNR of the equalized data carriers. Using the normal method of combining SNRs, the SNR of the equalized data carriers can be estimated as:

$$\frac{1}{SNR_{eq-data}} = \frac{1}{SNR_{data}} + \frac{1}{SNR_{channel_estimate}}$$

So from [eq. 3]

$$\frac{1}{SNR_{eq-data}} = \frac{1}{SNR_{data}} \left(1 + \frac{f_{INT}}{L1D_scattered_pilot_boost^2}\right)$$

So the channel estimation loss Δ_{RCE} can be written as:

$$\Delta_{RCE} = \frac{SNR_{data}}{SNR_{eq-data}}$$

So:

$$\Delta_{RCE} = 10\ Log_{10}\left(1 + \frac{f_{INT}}{L1D_scattered_pilot_boost^2}\right) \qquad \text{[eq. 6]}$$

Pre-calculated values of Δ_{RCE} are given in Table B.5.1.
Tables of C/N based on this model are given in Table B.6.1-Table B.6.15.

B.3 EXAMPLE C/N CALCULATION

An example calculation is given below assuming the following parameters:

32K FFT, 256QAM, 13/15 code rate, $D_y = 2$, Guard interval 1024 samples, SP12_2, 64K LDPC size

L1D_scattered_pilot_boost = 1, $C_{red_coeff} = 0$, $D_y = 2$

Receiver P_x = –33 dBc for 256QAM. Transmitter P_x = –50 dBc

$$C/N = -10\,Log_{10}\left[10^{-(C/N_{raw} + A + \Delta_{BP} + IM)/10} - 10^{P_x/10}\right] \quad [dB]$$

C/N_{raw} = **22.224 dB**

A = 0.5 dB

$$\Delta_{BP} = 10\,Log_{10}\left(\frac{Total\ Symbol\ Power}{\text{NoC}}\right) = 10\,Log_{10}\left(\frac{30080.16 \quad (from\ table\ I.2.1\ [3])}{27649 \quad (from\ table\ 8.8\ [3])}\right) = \mathbf{0.366\ dB}$$

$$\Delta_{RCE} = 10\,Log_{10}\left(1 + \frac{f_{INT}}{L1D_scattered_pilot_boost^2}\right)$$

f_{INT} = **0.75** for $D_y = 2$

$L1D_scattered_pilot_boost$ = 3.2dB or 1.445 in linear terms from Table 9.14 of [3]

So $\Delta_{RCE} = 10\,Log_{10}\left(1 + \frac{0.75}{1.445^2}\right)$ = **1.332 dB**

so $IM = \Delta_{RCE} + 0.5 = \mathbf{1.832\ dB}$

$P_x = 10\,Log_{10}\left[10^{-33/10} + 10^{-50/10}\right]$ = **-32.914 dB**

So $CNR = -10\,Log_{10}\left[10^{-(\mathbf{22.224 + 0.5 + 0.366 + 1.832})/10} - 10^{-\mathbf{32.914}/10}\right]$ = 25.7 dB

B.4 EXAMPLE CALCULATED VALUES OF Δ_{BP}

Table B.4.1 Calculated Values of Δ_{BP}

C_{red_coeff}		0					1					2					3					4				
L1D_scattered_pilot_boost		**000**	**001**	**010**	**011**	**100**	**000**	**001**	**010**	**011**	**100**	**000**	**001**	**010**	**011**	**100**	**000**	**001**	**010**	**011**	**100**	**000**	**001**	**010**	**011**	**100**
8K FFT	SP3_2	0.18	0.18	0.44	0.62	0.80	0.18	0.18	0.44	0.62	0.80	0.18	0.18	0.44	0.62	0.80	0.18	0.18	0.44	0.62	0.80	0.18	0.18	0.44	0.62	0.79
	SP3_4	0.18	0.31	0.50	0.64	0.75	0.18	0.31	0.50	0.65	0.76	0.18	0.31	0.50	0.65	0.76	0.18	0.31	0.50	0.64	0.75	0.18	0.31	0.50	0.64	0.75
	SP4_2	0.18	0.26	0.49	0.67	0.81	0.18	0.26	0.50	0.67	0.81	0.18	0.26	0.50	0.67	0.81	0.18	0.26	0.49	0.67	0.81	0.18	0.26	0.49	0.67	0.81
	SP4_4	0.18	0.34	0.51	0.62	0.73	0.18	0.34	0.51	0.62	0.73	0.18	0.34	0.51	0.62	0.73	0.18	0.34	0.51	0.62	0.73	0.18	0.34	0.51	0.62	0.73
	SP6_2	0.18	0.33	0.53	0.68	0.79	0.18	0.34	0.53	0.68	0.79	0.18	0.33	0.53	0.68	0.79	0.18	0.33	0.53	0.68	0.79	0.18	0.33	0.53	0.68	0.79
	SP6_4	0.18	0.35	0.49	0.59	0.68	0.18	0.36	0.49	0.60	0.68	0.18	0.35	0.49	0.60	0.68	0.18	0.35	0.49	0.59	0.68	0.18	0.35	0.49	0.59	0.68
	SP8_2	0.18	0.35	0.53	0.65	0.76	0.18	0.35	0.53	0.65	0.77	0.18	0.35	0.53	0.65	0.77	0.18	0.35	0.53	0.65	0.76	0.18	0.35	0.53	0.65	0.76
	SP8_4	0.18	0.35	0.47	0.56	0.63	0.18	0.35	0.47	0.56	0.63	0.18	0.35	0.47	0.56	0.63	0.18	0.35	0.47	0.56	0.63	0.18	0.35	0.47	0.56	0.63
	SP12_2	0.18	0.37	0.51	0.62	0.70	0.18	0.37	0.51	0.62	0.70	0.18	0.37	0.51	0.62	0.70	0.18	0.37	0.51	0.62	0.70	0.18	0.37	0.51	0.62	0.70
	SP12_4	0.18	0.34	0.44	0.52	0.57	0.18	0.34	0.44	0.52	0.58	0.18	0.34	0.44	0.52	0.58	0.18	0.34	0.44	0.52	0.58	0.18	0.34	0.44	0.51	0.57
	SP16_2	0.18	0.36	0.48	0.58	0.65	0.18	0.36	0.49	0.58	0.65	0.18	0.36	0.48	0.58	0.65	0.18	0.36	0.48	0.58	0.65	0.18	0.36	0.48	0.58	0.65
	SP16_4	0.18	0.33	0.42	0.49	0.54	0.18	0.33	0.42	0.49	0.54	0.18	0.34	0.42	0.49	0.55	0.18	0.33	0.42	0.49	0.54	0.18	0.33	0.42	0.49	0.54
	SP24_2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SP24_4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SP32_2	0.18	0.34	0.43	0.49	0.56	0.18	0.34	0.43	0.49	0.56	0.18	0.34	0.43	0.49	0.56	0.18	0.34	0.43	0.49	0.56	0.18	0.34	0.43	0.49	0.56
	SP32_4	0.18	0.31	0.37	0.42	0.46	0.18	0.31	0.37	0.42	0.45	0.18	0.31	0.37	0.42	0.46	0.18	0.31	0.37	0.42	0.46	0.18	0.31	0.37	0.42	0.46
16K FFT	SP3_2	0.18	0.18	0.44	0.62	0.80	0.18	0.18	0.44	0.62	0.80	0.18	0.18	0.44	0.62	0.79	0.18	0.18	0.44	0.62	0.80	0.18	0.18	0.44	0.62	0.79
	SP3_4	0.18	0.31	0.50	0.64	0.75	0.18	0.31	0.50	0.64	0.75	0.18	0.31	0.50	0.64	0.75	0.18	0.31	0.50	0.64	0.75	0.18	0.31	0.50	0.64	0.75
	SP4_2	0.18	0.26	0.49	0.67	0.81	0.18	0.26	0.50	0.67	0.81	0.18	0.26	0.49	0.67	0.81	0.18	0.26	0.49	0.67	0.81	0.18	0.26	0.49	0.67	0.81
	SP4_4	0.18	0.34	0.51	0.62	0.73	0.18	0.34	0.51	0.62	0.73	0.18	0.34	0.51	0.62	0.73	0.18	0.34	0.51	0.62	0.73	0.18	0.34	0.50	0.62	0.73
	SP6_2	0.18	0.33	0.53	0.68	0.79	0.18	0.34	0.53	0.68	0.79	0.18	0.33	0.53	0.68	0.79	0.18	0.33	0.53	0.68	0.79	0.18	0.33	0.53	0.68	0.79
	SP6_4	0.18	0.35	0.49	0.59	0.67	0.18	0.35	0.49	0.59	0.68	0.18	0.35	0.49	0.59	0.67	0.18	0.35	0.49	0.59	0.67	0.18	0.35	0.49	0.59	0.67
	SP8_2	0.18	0.35	0.53	0.65	0.76	0.18	0.35	0.53	0.65	0.77	0.18	0.35	0.53	0.65	0.76	0.18	0.35	0.53	0.65	0.76	0.18	0.35	0.53	0.65	0.76
	SP8_4	0.18	0.35	0.46	0.56	0.63	0.18	0.35	0.47	0.56	0.63	0.18	0.35	0.46	0.56	0.63	0.18	0.35	0.46	0.56	0.63	0.18	0.35	0.46	0.55	0.63
	SP12_2	0.18	0.37	0.51	0.62	0.70	0.18	0.37	0.51	0.62	0.70	0.18	0.37	0.51	0.62	0.70	0.18	0.37	0.51	0.62	0.70	0.18	0.37	0.51	0.62	0.70
	SP12_4	0.18	0.34	0.44	0.51	0.57	0.18	0.34	0.44	0.51	0.57	0.18	0.34	0.43	0.51	0.57	0.18	0.34	0.44	0.51	0.57	0.18	0.34	0.43	0.51	0.57
	SP16_2	0.18	0.36	0.48	0.58	0.65	0.18	0.36	0.48	0.58	0.65	0.18	0.36	0.48	0.58	0.65	0.18	0.36	0.48	0.58	0.65	0.18	0.36	0.48	0.58	0.65
	SP16_4	0.18	0.33	0.42	0.48	0.54	0.18	0.33	0.42	0.49	0.54	0.18	0.33	0.42	0.48	0.54	0.18	0.33	0.42	0.49	0.54	0.18	0.33	0.42	0.48	0.54
	SP24_2	0.18	0.35	0.45	0.53	0.59	0.18	0.35	0.45	0.53	0.59	0.18	0.35	0.45	0.53	0.59	0.18	0.35	0.45	0.53	0.59	0.18	0.35	0.45	0.53	0.59
	SP24_4	0.18	0.32	0.39	0.44	0.49	0.18	0.32	0.39	0.45	0.49	0.18	0.32	0.39	0.44	0.49	0.18	0.32	0.39	0.44	0.49	0.18	0.31	0.39	0.44	0.49
	SP32_2	0.18	0.34	0.43	0.49	0.55	0.18	0.34	0.43	0.49	0.56	0.18	0.34	0.43	0.49	0.55	0.18	0.34	0.43	0.49	0.55	0.18	0.34	0.43	0.49	0.55
	SP32_4	0.18	0.30	0.37	0.41	0.45	0.18	0.30	0.37	0.41	0.45	0.18	0.30	0.37	0.41	0.45	0.18	0.30	0.36	0.41	0.45	0.18	0.30	0.37	0.41	0.45
32K FFT	SP3_2	0.18	0.18	0.44	0.62	0.79	0.18	0.18	0.44	0.62	0.80	0.18	0.18	0.44	0.62	0.79	0.18	0.18	0.44	0.62	0.79	0.18	0.18	0.44	0.62	0.79
	SP3_4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SP4_2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SP4_4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SP6_2	0.18	0.33	0.53	0.68	0.79	0.18	0.34	0.53	0.68	0.79	0.18	0.33	0.53	0.68	0.79	0.18	0.33	0.53	0.68	0.79	0.18	0.33	0.53	0.68	0.79
	SP6_4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

	SP8_2	0.18	0.35	0.53	0.65	0.76	0.18	0.35	0.53	0.65	0.76	0.18	0.35	0.53	0.65	0.76	0.18	0.35	0.53	0.65	0.76	0.18	0.35	0.53	0.64	0.76
	SP8_4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SP12_2	0.18	0.37	0.51	0.62	0.70	0.18	0.37	0.51	0.62	0.70	0.18	0.37	0.51	0.61	0.70	0.18	0.37	0.51	0.62	0.70	0.18	0.36	0.51	0.61	0.70
	SP12_4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SP16_2	0.18	0.36	0.48	0.58	0.65	0.18	0.36	0.48	0.58	0.65	0.18	0.36	0.48	0.57	0.65	0.18	0.36	0.48	0.58	0.65	0.18	0.36	0.48	0.57	0.65
	SP16_4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SP24_2	0.18	0.35	0.45	0.53	0.59	0.18	0.35	0.45	0.53	0.59	0.18	0.35	0.45	0.53	0.59	0.18	0.35	0.45	0.53	0.59	0.18	0.35	0.45	0.52	0.59
	SP24_4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SP32_2	0.18	0.34	0.43	0.49	0.55	0.18	0.34	0.43	0.49	0.55	0.18	0.34	0.43	0.49	0.55	0.18	0.34	0.43	0.49	0.55	0.18	0.34	0.43	0.49	0.55
	SP32_4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

B.5 CALCULATED VALUES OF Δ_{RCE}

Table B.5.1 Calculated Values of Δ_{RCE}

		Calculated Real Channel Estimation Loss Δ_{RCE} (dB)							
Pilot Pattern SISO	**L1D_scattered_pilot_boost:**	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
SP3_2		*2.430*	*2.430*	*1.885*	*1.619*	*1.413*	*RFU*	*RFU*	*RFU*
SP3_4		*2.272*	*1.755*	*1.312*	*1.094*	*0.968*	*RFU*	*RFU*	*RFU*
SP4_2		*2.430*	*2.183*	*1.651*	*1.386*	*1.230*	*RFU*	*RFU*	*RFU*
SP4_4		*2.272*	*1.535*	*1.140*	*0.968*	*0.837*	*RFU*	*RFU*	*RFU*
SP6_2		*2.430*	*1.815*	*1.359*	*1.135*	*1.004*	*RFU*	*RFU*	*RFU*
SP6_4		*2.272*	*1.286*	*0.948*	*0.786*	*0.692*	*RFU*	*RFU*	*RFU*
SP8_2		*2.430*	*1.619*	*1.181*	*1.004*	*0.868*	*RFU*	*RFU*	*RFU*
SP8_4		*2.272*	*1.140*	*0.837*	*0.692*	*0.609*	*RFU*	*RFU*	*RFU*
SP12_2		*2.430*	*1.332*	*0.983*	*0.816*	*0.718*	*RFU*	*RFU*	*RFU*
SP12_4		*2.272*	*0.948*	*0.692*	*0.570*	*0.501*	*RFU*	*RFU*	*RFU*
SP16_2		*2.430*	*1.181*	*0.868*	*0.718*	*0.632*	*RFU*	*RFU*	*RFU*
SP16_4		*2.272*	*0.819*	*0.596*	*0.490*	*0.430*	*RFU*	*RFU*	*RFU*
SP24_2		*2.430*	*0.983*	*0.718*	*0.593*	*0.521*	*RFU*	*RFU*	*RFU*
SP24_4		*2.272*	*0.677*	*0.490*	*0.403*	*0.353*	*RFU*	*RFU*	*RFU*
SP32_2		*2.430*	*0.850*	*0.619*	*0.521*	*0.447*	*RFU*	*RFU*	*RFU*
SP32_4		*2.272*	*0.596*	*0.430*	*0.353*	*0.309*	*RFU*	*RFU*	*RFU*

B.6 EXPECTED RECEIVER AWGN C/N BASED ON CALCULATION MODEL

This section contains the RF guidelines for expected receiver performance with 8K, 16K and 32K FFT modes. Tables are color-coded such that a cell with a background color of GREEN indicates a mandatory ModCod combination, while a cell with a background color of GRAY indicates a non-mandatory ModCod combination.

All entries are for the following conditions:

L1D_scattered_pilot_boost = 001,

$C_{red_coeff} = 0$,

LDPC code length = 64800

$D_y = 4$ for 8K and 16K FFT, $D_y = 2$ for 32K FFT,

Receiver P_x = –33 dBc for QPSK, 16QAM, 64QAM and 256QAM.

Receiver P_x = –38 dBc for 1024QAM and 4096QAM (preliminary placeholder value– requires further study),

Transmitter P_x = –50 dBc

Compared to the expected C/N range of current digital television receivers, A/322 covers a much larger range of C/N values (see Section 3.4.10.3 of [7]), both in the very noisy range below 0dB and also the very high C/N range above 35dB. Care should be exercised when expecting consumer receivers to conform to the values in these ranges and further study of the performance of actual commercial receivers and refinement of the values may be needed.

Table B.6.1 Gaussian Channel 8K FFT, QPSK Expected C/N Values

FFT Size: 8K			GI: 192 samples (D_x=16)	GI: 384 samples (D_x=8)	GI: 512 samples (D_x=6)	GI: 768 samples (D_x=4)	GI:1024 samples (D_x=3)	GI: 1536 samples (D_x=4)	GI: 2048 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
QPSK	64800	2/15	-4.1	-3.7	-3.6	-3.4	-3.2	-3.4	-3.2
		3/15	-2.2	-1.8	-1.7	-1.4	-1.3	-1.4	-1.3
		4/15	-0.7	-0.4	-0.3	0.0	0.2	0.0	0.2
		5/15	0.5	0.8	0.9	1.2	1.4	1.2	1.4
		6/15	1.6	1.9	2.1	2.3	2.5	2.3	2.5
		7/15	2.5	2.8	2.9	3.2	3.4	3.2	3.4
		8/15	3.3	3.7	3.8	4.0	4.2	4.0	4.2
		9/15	4.1	4.5	4.6	4.9	5.0	4.9	5.0
		10/15	4.9	5.3	5.4	5.7	5.8	5.7	5.8
		11/15	5.8	6.1	6.3	6.5	6.7	6.5	6.7
		12/15	6.7	7.0	7.1	7.4	7.6	7.4	7.6
		13/15	7.7	8.0	8.2	8.4	8.6	8.4	8.6
QPSK	16200	2/15	-3.4	-3.1	-2.9	-2.7	-2.5	-2.7	-2.5
		3/15	-1.6	-1.2	-1.1	-0.9	-0.7	-0.9	-0.7
		4/15	-0.2	0.2	0.3	0.6	0.7	0.6	0.7
		5/15	0.9	1.2	1.3	1.6	1.8	1.6	1.8
		6/15	1.8	2.2	2.3	2.5	2.7	2.5	2.7
		7/15	2.7	3.1	3.2	3.4	3.6	3.4	3.6
		8/15	3.5	3.9	4.0	4.3	4.4	4.3	4.4
		9/15	4.4	4.7	4.8	5.1	5.3	5.1	5.3
		10/15	5.1	5.4	5.6	5.8	6.0	5.8	6.0
		11/15	6.0	6.3	6.5	6.7	6.9	6.7	6.9
		12/15	6.9	7.2	7.4	7.6	7.8	7.6	7.8
		13/15	7.9	8.3	8.4	8.6	8.8	8.6	8.8

Table B.6.2 Gaussian Channel 8K FFT, 16QAM Expected C/N Values

FFT Size: 8K			GI: 192 samples (D_x=16)	GI: 384 samples (D_x=8)	GI: 512 samples (D_x=6)	GI: 768 samples (D_x=4)	GI:1024 samples (D_x=3)	GI: 1536 samples (D_x=4)	GI: 2048 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
16QAM	64800	2/15	-0.6	-0.2	-0.1	0.2	0.3	0.2	0.3
		3/15	1.9	2.2	2.4	2.6	2.8	2.6	2.8
		4/15	3.6	4.0	4.1	4.3	4.5	4.3	4.5
		5/15	5.0	5.3	5.5	5.7	5.9	5.7	5.9
		6/15	6.4	6.7	6.9	7.1	7.3	7.1	7.3
		7/15	7.4	7.7	7.9	8.1	8.3	8.1	8.3
		8/15	8.5	8.8	9.0	9.2	9.4	9.2	9.4
		9/15	9.5	9.8	10.0	10.2	10.4	10.2	10.4
		10/15	10.5	10.9	11.0	11.3	11.5	11.3	11.5
		11/15	11.7	12.0	12.2	12.4	12.6	12.4	12.6
		12/15	12.8	13.1	13.3	13.5	13.7	13.5	13.7
		13/15	14.0	14.4	14.5	14.8	15.0	14.8	15.0
16QAM	16200	2/15	0.0	0.3	0.5	0.7	0.9	0.7	0.9
		3/15	2.5	2.8	3.0	3.2	3.4	3.2	3.4
		4/15	4.1	4.5	4.6	4.9	5.1	4.9	5.1
		5/15	5.3	5.7	5.8	6.0	6.2	6.0	6.2
		6/15	6.6	6.9	7.1	7.3	7.5	7.3	7.5
		7/15	7.7	8.0	8.2	8.4	8.6	8.4	8.6
		8/15	8.7	9.0	9.2	9.4	9.6	9.4	9.6
		9/15	9.8	10.1	10.2	10.5	10.7	10.5	10.7
		10/15	10.8	11.1	11.3	11.5	11.7	11.5	11.7
		11/15	11.9	12.3	12.4	12.7	12.9	12.7	12.9
		12/15	13.0	13.4	13.5	13.7	13.9	13.7	13.9
		13/15	14.3	14.6	14.8	15.0	15.2	15.0	15.2

Table B.6.3 Gaussian Channel 8K FFT, 64QAM Expected C/N Values

FFT Size: 8K			GI: 192 samples (D_x=16)	GI: 384 samples (D_x=8)	GI: 512 samples (D_x=6)	GI: 768 samples (D_x=4)	GI:1024 samples (D_x=3)	GI: 1536 samples (D_x=4)	GI: 2048 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
64QAM	64800	2/15	1.9	2.2	2.4	2.6	2.8	2.6	2.8
		3/15	4.4	4.8	4.9	5.2	5.3	5.2	5.3
		4/15	6.3	6.6	6.8	7.0	7.2	7.0	7.2
		5/15	8.1	8.5	8.6	8.9	9.0	8.9	9.0
		6/15	9.8	10.2	10.3	10.6	10.8	10.6	10.8
		7/15	11.1	11.4	11.6	11.8	12.0	11.8	12.0
		8/15	12.5	12.8	13.0	13.2	13.4	13.2	13.4
		9/15	13.8	14.1	14.3	14.5	14.7	14.5	14.7
		10/15	15.1	15.4	15.6	15.8	16.0	15.8	16.0
		11/15	16.5	16.9	17.0	17.3	17.5	17.3	17.5
		12/15	17.9	18.2	18.4	18.6	18.8	18.6	18.8
		13/15	19.4	19.7	19.9	20.1	20.3	20.1	20.3
64QAM	16200	2/15	2.5	2.8	3.0	3.2	3.4	3.2	3.4
		3/15	5.0	5.3	5.5	5.7	5.9	5.7	5.9
		4/15	6.8	7.1	7.3	7.5	7.7	7.5	7.7
		5/15	8.5	8.8	9.0	9.2	9.4	9.2	9.4
		6/15	10.1	10.4	10.6	10.8	11.0	10.8	11.0
		7/15	11.5	11.8	12.0	12.2	12.4	12.2	12.4
		8/15	12.8	13.1	13.2	13.5	13.7	13.5	13.7
		9/15	14.0	14.4	14.5	14.8	15.0	14.8	15.0
		10/15	15.4	15.7	15.9	16.1	16.3	16.1	16.3
		11/15	16.8	17.1	17.3	17.5	17.7	17.5	17.7
		12/15	18.2	18.5	18.7	18.9	19.1	18.9	19.1
		13/15	19.7	20.0	20.2	20.4	20.6	20.4	20.6

Table B.6.4 Gaussian Channel 8K FFT, 256QAM Expected C/N Values

FFT Size: 8K			GI: 192 samples (D_x=16)	GI: 384 samples (D_x=8)	GI: 512 samples (D_x=6)	GI: 768 samples (D_x=4)	GI:1024 samples (D_x=3)	GI: 1536 samples (D_x=4)	GI: 2048 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
256QAM	64800	2/15	3.8	4.1	4.2	4.5	4.7	4.5	4.7
		3/15	6.5	6.8	6.9	7.2	7.4	7.2	7.4
		4/15	8.7	9.1	9.2	9.5	9.7	9.5	9.7
		5/15	10.7	11.0	11.2	11.4	11.6	11.4	11.6
		6/15	12.8	13.1	13.3	13.5	13.7	13.5	13.7
		7/15	14.3	14.7	14.8	15.0	15.2	15.0	15.2
		8/15	16.2	16.5	16.7	16.9	17.1	16.9	17.1
		9/15	17.8	18.2	18.3	18.6	18.8	18.6	18.8
		10/15	19.5	19.8	20.0	20.2	20.4	20.2	20.4
		11/15	21.2	21.6	21.7	22.0	22.2	22.0	22.2
		12/15	23.0	23.4	23.6	23.8	24.0	23.8	24.0
		13/15	25.0	25.4	25.6	25.9	26.1	25.9	26.1
256QAM	16200	2/15	4.4	4.8	4.9	5.2	5.3	5.2	5.3
		3/15	6.9	7.3	7.4	7.7	7.9	7.7	7.9
		4/15	9.4	9.7	9.8	10.1	10.3	10.1	10.3
		5/15	11.1	11.5	11.6	11.8	12.0	11.8	12.0
		6/15	13.1	13.4	13.6	13.8	14.0	13.8	14.0
		7/15	14.8	15.1	15.3	15.5	15.7	15.5	15.7
		8/15	16.5	16.8	17.0	17.2	17.4	17.2	17.4
		9/15	18.1	18.4	18.6	18.8	19.0	18.8	19.0
		10/15	19.8	20.2	20.3	20.6	20.8	20.6	20.8
		11/15	21.5	21.9	22.1	22.3	22.5	22.3	22.5
		12/15	23.4	23.8	23.9	24.2	24.4	24.2	24.4
		13/15	25.4	25.8	26.0	26.3	26.5	26.3	26.5

Table B.6.5 Gaussian Channel 8K FFT, 1024 and 4096QAM Expected C/N Values

FFT Size: 8K			GI: 192 samples (D_x=16)	GI: 384 samples (D_x=8)	GI: 512 samples (D_x=6)	GI: 768 samples (D_x=4)	GI:1024 samples (D_x=3)	GI: 1536 samples (D_x=4)	GI: 2048 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
1024QAM	64800	2/15	5.4	5.7	5.9	6.1	6.3	6.1	6.3
		3/15	8.3	8.7	8.8	9.0	9.2	9.0	9.2
		4/15	10.9	11.3	11.4	11.7	11.8	11.7	11.8
		5/15	13.2	13.6	13.7	14.0	14.2	14.0	14.2
		6/15	15.6	16.0	16.1	16.4	16.6	16.4	16.6
		7/15	17.5	17.8	18.0	18.2	18.4	18.2	18.4
		8/15	19.7	20.0	20.2	20.4	20.6	20.4	20.6
		9/15	21.7	22.1	22.2	22.4	22.6	22.4	22.6
		10/15	23.7	24.0	24.2	24.4	24.6	24.4	24.6
		11/15	25.9	26.2	26.4	26.6	26.8	26.6	26.8
		12/15	28.1	28.5	28.7	28.9	29.2	28.9	29.2
		13/15	30.5	30.9	31.1	31.4	31.6	31.4	31.6
4096QAM	64800	2/15	6.7	7.1	7.2	7.5	7.6	7.5	7.6
		3/15	10.0	10.3	10.5	10.7	10.9	10.7	10.9
		4/15	12.9	13.2	13.4	13.6	13.8	13.6	13.8
		5/15	15.6	16.0	16.1	16.4	16.6	16.4	16.6
		6/15	18.2	18.6	18.7	19.0	19.2	19.0	19.2
		7/15	20.4	20.8	20.9	21.2	21.4	21.2	21.4
		8/15	23.0	23.3	23.5	23.7	23.9	23.7	23.9
		9/15	25.5	25.8	26.0	26.2	26.4	26.2	26.4
		10/15	28.2	28.5	28.7	29.0	29.2	29.0	29.2
		11/15	31.1	31.5	31.7	32.0	32.3	32.0	32.3
		12/15	34.0	34.5	34.7	35.1	35.4	35.1	35.4
		13/15	38.3	39.0	39.4	40.0	40.5	40.0	40.5

Table B.6.6 Gaussian Channel 16K FFT, QPSK Expected C/N Values

FFT Size: 16K			GI: 192 samples (D_x=32)	GI: 384 samples (D_x=16)	GI: 512 samples (D_x=12)	GI: 768 samples (D_x=8)	GI:1024 samples (D_x=6)	GI: 1536 samples (D_x=4)	GI: 2048 samples (D_x=3)	GI: 2432 samples (D_x=3)	GI: 3072 samples (D_x=4)	GI: 3648 samples (D_x=4)	GI: 4096 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
QPSK	64800	2/15	-4.3	-4.1	-3.9	-3.7	-3.6	-3.4	-3.2	-3.2	-3.4	-3.4	-3.2
		3/15	-2.4	-2.2	-2.0	-1.8	-1.7	-1.4	-1.3	-1.3	-1.4	-1.4	-1.3
		4/15	-1.0	-0.7	-0.6	-0.4	-0.3	0.0	0.2	0.2	0.0	0.0	0.2
		5/15	0.2	0.5	0.6	0.8	0.9	1.2	1.4	1.4	1.2	1.2	1.4
		6/15	1.4	1.6	1.7	1.9	2.1	2.3	2.5	2.5	2.3	2.3	2.5
		7/15	2.2	2.5	2.6	2.8	2.9	3.2	3.4	3.4	3.2	3.2	3.4
		8/15	3.1	3.3	3.4	3.7	3.8	4.0	4.2	4.2	4.0	4.0	4.2
		9/15	3.9	4.1	4.3	4.5	4.6	4.9	5.0	5.0	4.9	4.9	5.0
		10/15	4.7	4.9	5.1	5.3	5.4	5.7	5.8	5.8	5.7	5.7	5.8
		11/15	5.5	5.8	5.9	6.1	6.3	6.5	6.7	6.7	6.5	6.5	6.7
		12/15	6.4	6.6	6.8	7.0	7.1	7.4	7.6	7.6	7.4	7.4	7.6
		13/15	7.4	7.7	7.8	8.0	8.2	8.4	8.6	8.6	8.4	8.4	8.6
QPSK	16200	2/15	-3.7	-3.4	-3.3	-3.1	-2.9	-2.7	-2.5	-2.5	-2.7	-2.7	-2.5
		3/15	-1.8	-1.6	-1.4	-1.2	-1.1	-0.9	-0.7	-0.7	-0.9	-0.9	-0.7
		4/15	-0.4	-0.2	0.0	0.2	0.3	0.6	0.7	0.7	0.6	0.6	0.7
		5/15	0.6	0.9	1.0	1.2	1.3	1.6	1.8	1.8	1.6	1.6	1.8
		6/15	1.6	1.8	2.0	2.2	2.3	2.5	2.7	2.7	2.5	2.5	2.7
		7/15	2.5	2.7	2.9	3.1	3.2	3.4	3.6	3.6	3.4	3.4	3.6
		8/15	3.3	3.5	3.7	3.9	4.0	4.3	4.4	4.4	4.3	4.3	4.4
		9/15	4.1	4.4	4.5	4.7	4.8	5.1	5.3	5.3	5.1	5.1	5.3
		10/15	4.8	5.1	5.2	5.4	5.6	5.8	6.0	6.0	5.8	5.8	6.0
		11/15	5.7	6.0	6.1	6.3	6.5	6.7	6.9	6.9	6.7	6.7	6.9
		12/15	6.6	6.9	7.0	7.2	7.4	7.6	7.8	7.8	7.6	7.6	7.8
		13/15	7.7	7.9	8.1	8.3	8.4	8.6	8.8	8.8	8.6	8.6	8.8

Table B.6.7 Gaussian Channel 16K FFT, 16QAM Expected C/N Values

FFT Size:16K			GI: 192 samples (D_x=32)	GI: 384 samples (D_x=16)	GI: 512 samples (D_x=12)	GI: 768 samples (D_x=8)	GI:1024 samples (D_x=6)	GI: 1536 samples (D_x=4)	GI: 2048 samples (D_x=3)	GI: 2432 samples (D_x=3)	GI: 3072 samples (D_x=4)	GI: 3648 samples (D_x=4)	GI: 4096 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
16QAM	64800	2/15	-0.8	-0.6	-0.4	-0.2	-0.1	0.2	0.3	0.3	0.2	0.2	0.3
		3/15	1.7	1.9	2.0	2.2	2.4	2.6	2.8	2.8	2.6	2.6	2.8
		4/15	3.4	3.6	3.8	4.0	4.1	4.3	4.5	4.5	4.3	4.3	4.5
		5/15	4.7	5.0	5.1	5.3	5.5	5.7	5.9	5.9	5.7	5.7	5.9
		6/15	6.1	6.4	6.5	6.7	6.9	7.1	7.3	7.3	7.1	7.1	7.3
		7/15	7.1	7.4	7.5	7.7	7.9	8.1	8.3	8.3	8.1	8.1	8.3
		8/15	8.2	8.5	8.6	8.8	9.0	9.2	9.4	9.4	9.2	9.2	9.4
		9/15	9.2	9.5	9.6	9.8	10.0	10.2	10.4	10.4	10.2	10.2	10.4
		10/15	10.3	10.5	10.7	10.9	11.0	11.3	11.5	11.5	11.3	11.3	11.5
		11/15	11.4	11.7	11.8	12.0	12.2	12.4	12.6	12.6	12.4	12.4	12.6
		12/15	12.5	12.8	12.9	13.1	13.3	13.5	13.7	13.7	13.5	13.5	13.7
		13/15	13.8	14.0	14.2	14.4	14.5	14.8	15.0	15.0	14.8	14.8	15.0
16QAM	16200	2/15	-0.3	0.0	0.1	0.3	0.5	0.7	0.9	0.9	0.7	0.7	0.9
		3/15	2.3	2.5	2.6	2.8	3.0	3.2	3.4	3.4	3.2	3.2	3.4
		4/15	3.9	4.1	4.3	4.5	4.6	4.9	5.1	5.1	4.9	4.9	5.1
		5/15	5.1	5.3	5.5	5.7	5.8	6.0	6.2	6.2	6.0	6.0	6.2
		6/15	6.4	6.6	6.7	6.9	7.1	7.3	7.5	7.5	7.3	7.3	7.5
		7/15	7.4	7.7	7.8	8.0	8.2	8.4	8.6	8.6	8.4	8.4	8.6
		8/15	8.4	8.7	8.8	9.0	9.2	9.4	9.6	9.6	9.4	9.4	9.6
		9/15	9.5	9.7	9.9	10.1	10.2	10.5	10.7	10.7	10.5	10.5	10.7
		10/15	10.5	10.8	10.9	11.1	11.3	11.5	11.7	11.7	11.5	11.5	11.7
		11/15	11.7	11.9	12.1	12.3	12.4	12.7	12.9	12.9	12.7	12.7	12.9
		12/15	12.8	13.0	13.1	13.3	13.5	13.7	13.9	13.9	13.7	13.7	13.9
		13/15	14.0	14.3	14.4	14.6	14.8	15.0	15.2	15.2	15.0	15.0	15.2

Table B.6.8 Gaussian Channel 16K FFT, 64QAM Expected C/N Values

FFT Size:16K			GI: 192 samples (D_x=32)	GI: 384 samples (D_x=16)	GI: 512 samples (D_x=12)	GI: 768 samples (D_x=8)	GI:1024 samples (D_x=6)	GI: 1536 samples (D_x=4)	GI: 2048 samples (D_x=3)	GI: 2432 samples (D_x=3)	GI: 3072 samples (D_x=4)	GI: 3648 samples (D_x=4)	GI: 4096 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
64QAM	64800	2/15	1.6	1.9	2.0	2.2	2.4	2.6	2.8	2.8	2.6	2.6	2.8
		3/15	4.2	4.4	4.6	4.8	4.9	5.2	5.3	5.3	5.2	5.2	5.3
		4/15	6.1	6.3	6.4	6.6	6.8	7.0	7.2	7.2	7.0	7.0	7.2
		5/15	7.9	8.1	8.3	8.5	8.6	8.9	9.0	9.0	8.9	8.9	9.0
		6/15	9.6	9.8	10.0	10.2	10.3	10.6	10.8	10.8	10.6	10.6	10.8
		7/15	10.8	11.1	11.2	11.4	11.6	11.8	12.0	12.0	11.8	11.8	12.0
		8/15	12.2	12.5	12.6	12.8	13.0	13.2	13.4	13.4	13.2	13.2	13.4
		9/15	13.5	13.8	13.9	14.1	14.3	14.5	14.7	14.7	14.5	14.5	14.7
		10/15	14.8	15.1	15.2	15.4	15.6	15.8	16.0	16.0	15.8	15.8	16.0
		11/15	16.3	16.5	16.7	16.9	17.0	17.3	17.5	17.5	17.3	17.3	17.5
		12/15	17.6	17.9	18.0	18.2	18.4	18.6	18.8	18.8	18.6	18.6	18.8
		13/15	19.1	19.4	19.5	19.7	19.9	20.1	20.3	20.3	20.1	20.1	20.3
64QAM	16200	2/15	2.3	2.5	2.6	2.8	3.0	3.2	3.4	3.4	3.2	3.2	3.4
		3/15	4.8	5.0	5.1	5.3	5.5	5.7	5.9	5.9	5.7	5.7	5.9
		4/15	6.6	6.8	6.9	7.1	7.3	7.5	7.7	7.7	7.5	7.5	7.7
		5/15	8.2	8.5	8.6	8.8	9.0	9.2	9.4	9.4	9.2	9.2	9.4
		6/15	9.8	10.1	10.2	10.4	10.6	10.8	11.0	11.0	10.8	10.8	11.0
		7/15	11.2	11.5	11.6	11.8	12.0	12.2	12.4	12.4	12.2	12.2	12.4
		8/15	12.5	12.7	12.9	13.1	13.2	13.5	13.7	13.7	13.5	13.5	13.7
		9/15	13.8	14.0	14.2	14.4	14.5	14.8	15.0	15.0	14.8	14.8	15.0
		10/15	15.1	15.4	15.5	15.7	15.9	16.1	16.3	16.3	16.1	16.1	16.3
		11/15	16.5	16.8	16.9	17.1	17.3	17.5	17.7	17.7	17.5	17.5	17.7
		12/15	17.9	18.1	18.3	18.5	18.7	18.9	19.1	19.1	18.9	18.9	19.1
		13/15	19.4	19.7	19.8	20.0	20.2	20.4	20.6	20.6	20.4	20.4	20.6

Table B.6.9 Gaussian Channel 16K FFT, 256QAM Expected C/N Values

FFT Size:16K			GI: 192 samples (D_x=32)	GI: 384 samples (D_x=16)	GI: 512 samples (D_x=12)	GI: 768 samples (D_x=8)	GI:1024 samples (D_x=6)	GI: 1536 samples (D_x=4)	GI: 2048 samples (D_x=3)	GI: 2432 samples (D_x=3)	GI: 3072 samples (D_x=4)	GI: 3648 samples (D_x=4)	GI: 4096 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
256QAM	64800	2/15	3.5	3.8	3.9	4.1	4.2	4.5	4.7	4.7	4.5	4.5	4.7
		3/15	6.2	6.5	6.6	6.8	6.9	7.2	7.4	7.4	7.2	7.2	7.4
		4/15	8.5	8.7	8.9	9.1	9.2	9.5	9.7	9.7	9.5	9.5	9.7
		5/15	10.5	10.7	10.8	11.0	11.2	11.4	11.6	11.6	11.4	11.4	11.6
		6/15	12.5	12.8	12.9	13.1	13.3	13.5	13.7	13.7	13.5	13.5	13.7
		7/15	14.1	14.3	14.4	14.7	14.8	15.0	15.2	15.2	15.0	15.0	15.2
		8/15	15.9	16.2	16.3	16.5	16.7	16.9	17.1	17.1	16.9	16.9	17.1
		9/15	17.6	17.8	18.0	18.2	18.3	18.6	18.8	18.8	18.6	18.6	18.8
		10/15	19.2	19.5	19.6	19.8	20.0	20.2	20.4	20.4	20.2	20.2	20.4
		11/15	20.9	21.2	21.3	21.6	21.7	22.0	22.2	22.2	22.0	22.0	22.2
		12/15	22.7	23.0	23.2	23.4	23.6	23.8	24.0	24.0	23.8	23.8	24.0
		13/15	24.7	25.0	25.2	25.4	25.6	25.9	26.1	26.1	25.9	25.9	26.1
256QAM	16200	2/15	4.2	4.4	4.6	4.8	4.9	5.2	5.3	5.3	5.2	5.2	5.3
		3/15	6.7	6.9	7.1	7.3	7.4	7.7	7.9	7.9	7.7	7.7	7.9
		4/15	9.1	9.4	9.5	9.7	9.8	10.1	10.3	10.3	10.1	10.1	10.3
		5/15	10.9	11.1	11.2	11.4	11.6	11.8	12.0	12.0	11.8	11.8	12.0
		6/15	12.8	13.1	13.2	13.4	13.6	13.8	14.0	14.0	13.8	13.8	14.0
		7/15	14.5	14.8	14.9	15.1	15.3	15.5	15.7	15.7	15.5	15.5	15.7
		8/15	16.2	16.5	16.6	16.8	17.0	17.2	17.4	17.4	17.2	17.2	17.4
		9/15	17.8	18.1	18.2	18.4	18.6	18.8	19.0	19.0	18.8	18.8	19.0
		10/15	19.5	19.8	20.0	20.2	20.3	20.6	20.8	20.8	20.6	20.6	20.8
		11/15	21.3	21.5	21.7	21.9	22.1	22.3	22.5	22.5	22.3	22.3	22.5
		12/15	23.1	23.4	23.5	23.8	23.9	24.2	24.4	24.4	24.2	24.2	24.4
		13/15	25.1	25.4	25.6	25.8	26.0	26.3	26.5	26.5	26.3	26.3	26.5

Table B.6.10 Gaussian Channel 16K FFT, 1024QAM and 4096QAM Expected C/N Values

FFT Size:16K			GI: 192 samples (D_x=32)	GI: 384 samples (D_x=16)	GI: 512 samples (D_x=12)	GI: 768 samples (D_x=8)	GI:1024 samples (D_x=6)	GI: 1536 samples (D_x=4)	GI: 2048 samples (D_x=3)	GI: 2432 samples (D_x=3)	GI: 3072 samples (D_x=4)	GI: 3648 samples (D_x=4)	GI: 4096 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
1024QAM	64800	2/15	5.1	5.4	5.5	5.7	5.9	6.1	6.3	6.3	6.1	6.1	6.3
		3/15	8.1	8.3	8.5	8.7	8.8	9.0	9.2	9.2	9.0	9.0	9.2
		4/15	10.7	10.9	11.1	11.3	11.4	11.7	11.8	11.8	11.7	11.7	11.8
		5/15	13.0	13.2	13.4	13.6	13.7	14.0	14.2	14.2	14.0	14.0	14.2
		6/15	15.4	15.6	15.8	16.0	16.1	16.4	16.6	16.6	16.4	16.4	16.6
		7/15	17.2	17.5	17.6	17.8	18.0	18.2	18.4	18.4	18.2	18.2	18.4
		8/15	19.4	19.7	19.8	20.0	20.2	20.4	20.6	20.6	20.4	20.4	20.6
		9/15	21.4	21.7	21.8	22.0	22.2	22.4	22.6	22.6	22.4	22.4	22.6
		10/15	23.4	23.7	23.8	24.0	24.2	24.4	24.6	24.6	24.4	24.4	24.6
		11/15	25.6	25.8	26.0	26.2	26.4	26.6	26.8	26.8	26.6	26.6	26.8
		12/15	27.8	28.1	28.3	28.5	28.7	28.9	29.1	29.1	28.9	28.9	29.1
		13/15	30.2	30.5	30.7	30.9	31.1	31.4	31.6	31.6	31.4	31.4	31.6
4096QAM	64800	2/15	6.5	6.7	6.9	7.1	7.2	7.5	7.6	7.6	7.5	7.5	7.6
		3/15	9.8	10.0	10.1	10.3	10.5	10.7	10.9	10.9	10.7	10.7	10.9
		4/15	12.6	12.9	13.0	13.2	13.4	13.6	13.8	13.8	13.6	13.6	13.8
		5/15	15.4	15.6	15.8	16.0	16.1	16.4	16.6	16.6	16.4	16.4	16.6
		6/15	18.0	18.2	18.4	18.6	18.7	19.0	19.2	19.2	19.0	19.0	19.2
		7/15	20.2	20.4	20.6	20.8	20.9	21.2	21.4	21.4	21.2	21.2	21.4
		8/15	22.7	23.0	23.1	23.3	23.5	23.7	23.9	23.9	23.7	23.7	23.9
		9/15	25.2	25.5	25.6	25.8	26.0	26.2	26.4	26.4	26.2	26.2	26.4
		10/15	27.9	28.1	28.3	28.5	28.7	29.0	29.2	29.2	29.0	29.0	29.2
		11/15	30.8	31.1	31.3	31.5	31.7	32.0	32.3	32.3	32.0	32.0	32.3
		12/15	33.7	34.0	34.2	34.5	34.7	35.1	35.4	35.4	35.1	35.1	35.4
		13/15	37.7	38.3	38.6	39.0	39.4	40.0	40.5	40.5	40.0	40.0	40.5

Table B.6.11 Gaussian Channel 32K FFT, QPSK Expected C/N Values

FFT Size:32K			GI: 192 samples (D_x=32)	GI: 384 samples (D_x=32)	GI: 512 samples (D_x=24)	GI: 768 samples (D_x=16)	GI:1024 samples (D_x=12)	GI: 1536 samples (D_x=8)	GI: 2048 samples (D_x=6)	GI: 2432 samples (D_x=6)	GI: 3072 samples (D_x=8)	GI: 3648 samples (D_x=8)	GI: 4096 samples (D_x=3)	GI: 4864 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
QPSK	64800	2/15	-4.0	-4.0	-3.9	-3.7	-3.5	-3.3	-3.1	-3.1	-3.3	-3.3	-2.6	-2.6
		3/15	-2.1	-2.1	-2.0	-1.8	-1.6	-1.4	-1.2	-1.2	-1.4	-1.4	-0.7	-0.7
		4/15	-0.7	-0.7	-0.6	-0.3	-0.2	0.1	0.3	0.3	0.1	0.1	0.7	0.7
		5/15	0.5	0.5	0.6	0.8	1.0	1.3	1.4	1.4	1.3	1.3	1.9	1.9
		6/15	1.6	1.6	1.8	2.0	2.2	2.4	2.6	2.6	2.4	2.4	3.1	3.1
		7/15	2.5	2.5	2.6	2.8	3.0	3.3	3.5	3.5	3.3	3.3	3.9	3.9
		8/15	3.4	3.4	3.5	3.7	3.9	4.1	4.3	4.3	4.1	4.1	4.8	4.8
		9/15	4.2	4.2	4.3	4.5	4.7	4.9	5.1	5.1	4.9	4.9	5.6	5.6
		10/15	5.0	5.0	5.1	5.3	5.5	5.7	5.9	5.9	5.7	5.7	6.4	6.4
		11/15	5.8	5.8	5.9	6.2	6.3	6.6	6.8	6.8	6.6	6.6	7.2	7.2
		12/15	6.7	6.7	6.8	7.0	7.2	7.5	7.6	7.6	7.5	7.5	8.1	8.1
		13/15	7.7	7.7	7.9	8.1	8.2	8.5	8.7	8.7	8.5	8.5	9.2	9.2
QPSK	16200	2/15	-3.4	-3.4	-3.2	-3.0	-2.9	-2.6	-2.4	-2.4	-2.6	-2.6	-1.9	-1.9
		3/15	-1.5	-1.5	-1.4	-1.2	-1.0	-0.8	-0.6	-0.6	-0.8	-0.8	-0.1	-0.1
		4/15	-0.1	-0.1	0.0	0.2	0.4	0.6	0.8	0.8	0.6	0.6	1.3	1.3
		5/15	0.9	0.9	1.0	1.2	1.4	1.7	1.9	1.9	1.7	1.7	2.3	2.3
		6/15	1.9	1.9	2.0	2.2	2.4	2.6	2.8	2.8	2.6	2.6	3.3	3.3
		7/15	2.8	2.8	2.9	3.1	3.3	3.5	3.7	3.7	3.5	3.5	4.2	4.2
		8/15	3.6	3.6	3.7	3.9	4.1	4.3	4.5	4.5	4.3	4.3	5.0	5.0
		9/15	4.4	4.4	4.5	4.7	4.9	5.2	5.4	5.4	5.2	5.2	5.8	5.8
		10/15	5.1	5.1	5.3	5.5	5.6	5.9	6.1	6.1	5.9	5.9	6.6	6.6
		11/15	6.0	6.0	6.2	6.4	6.5	6.8	7.0	7.0	6.8	6.8	7.4	7.4
		12/15	6.9	6.9	7.0	7.3	7.4	7.7	7.9	7.9	7.7	7.7	8.3	8.3
		13/15	8.0	8.0	8.1	8.3	8.5	8.7	8.9	8.9	8.7	8.7	9.4	9.4

Table B.6.12 Gaussian Channel 32K FFT, 16QAM Expected C/N Values

FFT Size:32K			GI: 192 samples (D_x=32)	GI: 384 samples (D_x=32)	GI: 512 samples (D_x=24)	GI: 768 samples (D_x=16)	GI:1024 samples (D_x=12)	GI: 1536 samples (D_x=8)	GI: 2048 samples (D_x=6)	GI: 2432 samples (D_x=6)	GI: 3072 samples (D_x=8)	GI: 3648 samples (D_x=8)	GI: 4096 samples (D_x=3)	GI: 4864 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
16QAM	64800	2/15	-0.5	-0.5	-0.4	-0.2	0.0	0.2	0.4	0.4	0.2	0.2	0.9	0.9
		3/15	1.9	1.9	2.1	2.3	2.5	2.7	2.9	2.9	2.7	2.7	3.4	3.4
		4/15	3.7	3.7	3.8	4.0	4.2	4.4	4.6	4.6	4.4	4.4	5.1	5.1
		5/15	5.0	5.0	5.2	5.4	5.5	5.8	6.0	6.0	5.8	5.8	6.4	6.4
		6/15	6.4	6.4	6.6	6.8	6.9	7.2	7.4	7.4	7.2	7.2	7.8	7.8
		7/15	7.4	7.4	7.6	7.8	7.9	8.2	8.4	8.4	8.2	8.2	8.8	8.8
		8/15	8.5	8.5	8.6	8.9	9.0	9.3	9.5	9.5	9.3	9.3	9.9	9.9
		9/15	9.5	9.5	9.7	9.9	10.0	10.3	10.5	10.5	10.3	10.3	11.0	11.0
		10/15	10.6	10.6	10.7	10.9	11.1	11.4	11.5	11.5	11.4	11.4	12.0	12.0
		11/15	11.7	11.7	11.9	12.1	12.2	12.5	12.7	12.7	12.5	12.5	13.2	13.2
		12/15	12.8	12.8	12.9	13.2	13.3	13.6	13.8	13.8	13.6	13.6	14.2	14.2
		13/15	14.1	14.1	14.2	14.4	14.6	14.9	15.1	15.1	14.9	14.9	15.5	15.5
16QAM	16200	2/15	0.0	0.0	0.2	0.4	0.5	0.8	1.0	1.0	0.8	0.8	1.5	1.5
		3/15	2.5	2.5	2.7	2.9	3.1	3.3	3.5	3.5	3.3	3.3	4.0	4.0
		4/15	4.2	4.2	4.3	4.5	4.7	5.0	5.1	5.1	5.0	5.0	5.6	5.6
		5/15	5.4	5.4	5.5	5.7	5.9	6.1	6.3	6.3	6.1	6.1	6.8	6.8
		6/15	6.6	6.6	6.8	7.0	7.2	7.4	7.6	7.6	7.4	7.4	8.1	8.1
		7/15	7.7	7.7	7.9	8.1	8.2	8.5	8.7	8.7	8.5	8.5	9.1	9.1
		8/15	8.7	8.7	8.9	9.1	9.2	9.5	9.7	9.7	9.5	9.5	10.1	10.1
		9/15	9.8	9.8	9.9	10.1	10.3	10.6	10.8	10.8	10.6	10.6	11.2	11.2
		10/15	10.8	10.8	10.9	11.2	11.3	11.6	11.8	11.8	11.6	11.6	12.2	12.2
		11/15	12.0	12.0	12.1	12.3	12.5	12.8	12.9	12.9	12.8	12.8	13.4	13.4
		12/15	13.0	13.0	13.2	13.4	13.6	13.8	14.0	14.0	13.8	13.8	14.5	14.5
		13/15	14.3	14.3	14.5	14.7	14.9	15.1	15.3	15.3	15.1	15.1	15.8	15.8

Table B.6.13 Gaussian Channel 32K FFT, 64QAM Expected C/N Values

FFT Size:32K			GI: 192 samples (D_x=32)	GI: 384 samples (D_x=32)	GI: 512 samples (D_x=24)	GI: 768 samples (D_x=16)	GI:1024 samples (D_x=12)	GI: 1536 samples (D_x=8)	GI: 2048 samples (D_x=6)	GI: 2432 samples (D_x=6)	GI: 3072 samples (D_x=8)	GI: 3648 samples (D_x=8)	GI: 4096 samples (D_x=3)	GI: 4864 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
64QAM	64800	2/15	1.9	1.9	2.1	2.3	2.4	2.7	2.9	2.9	2.7	2.7	3.4	3.4
		3/15	4.5	4.5	4.6	4.8	5.0	5.2	5.4	5.4	5.2	5.2	5.9	5.9
		4/15	6.3	6.3	6.5	6.7	6.9	7.1	7.3	7.3	7.1	7.1	7.8	7.8
		5/15	8.2	8.2	8.3	8.5	8.7	8.9	9.1	9.1	8.9	8.9	9.6	9.6
		6/15	9.9	9.9	10.0	10.2	10.4	10.7	10.8	10.8	10.7	10.7	11.3	11.3
		7/15	11.1	11.1	11.3	11.5	11.7	11.9	12.1	12.1	11.9	11.9	12.6	12.6
		8/15	12.5	12.5	12.7	12.9	13.0	13.3	13.5	13.5	13.3	13.3	14.0	14.0
		9/15	13.8	13.8	13.9	14.2	14.3	14.6	14.8	14.8	14.6	14.6	15.2	15.2
		10/15	15.1	15.1	15.3	15.5	15.7	15.9	16.1	16.1	15.9	15.9	16.6	16.6
		11/15	16.6	16.6	16.7	16.9	17.1	17.4	17.6	17.6	17.4	17.4	18.0	18.0
		12/15	17.9	17.9	18.0	18.3	18.4	18.7	18.9	18.9	18.7	18.7	19.4	19.4
		13/15	19.4	19.4	19.6	19.8	19.9	20.2	20.4	20.4	20.2	20.2	20.9	20.9
64QAM	16200	2/15	2.5	2.5	2.7	2.9	3.1	3.3	3.5	3.5	3.3	3.3	4.0	4.0
		3/15	5.0	5.0	5.2	5.4	5.6	5.8	6.0	6.0	5.8	5.8	6.5	6.5
		4/15	6.8	6.8	7.0	7.2	7.4	7.6	7.8	7.8	7.6	7.6	8.3	8.3
		5/15	8.5	8.5	8.7	8.9	9.0	9.3	9.5	9.5	9.3	9.3	9.9	9.9
		6/15	10.1	10.1	10.3	10.5	10.7	10.9	11.1	11.1	10.9	10.9	11.6	11.6
		7/15	11.5	11.5	11.7	11.9	12.0	12.3	12.5	12.5	12.3	12.3	12.9	12.9
		8/15	12.8	12.8	12.9	13.1	13.3	13.6	13.8	13.8	13.6	13.6	14.2	14.2
		9/15	14.1	14.1	14.2	14.4	14.6	14.9	15.0	15.0	14.9	14.9	15.5	15.5
		10/15	15.4	15.4	15.5	15.8	15.9	16.2	16.4	16.4	16.2	16.2	16.8	16.8
		11/15	16.8	16.8	17.0	17.2	17.3	17.6	17.8	17.8	17.6	17.6	18.3	18.3
		12/15	18.2	18.2	18.3	18.6	18.7	19.0	19.2	19.2	19.0	19.0	19.7	19.7
		13/15	19.7	19.7	19.9	20.1	20.3	20.5	20.7	20.7	20.5	20.5	21.2	21.2

Table B.6.14 Gaussian Channel 32K FFT, 256QAM Expected C/N Values

FFT Size:32K			GI: 192 samples (D_x=32)	GI: 384 samples (D_x=32)	GI: 512 samples (D_x=24)	GI: 768 samples (D_x=16)	GI:1024 samples (D_x=12)	GI: 1536 samples (D_x=8)	GI: 2048 samples (D_x=6)	GI: 2432 samples (D_x=6)	GI: 3072 samples (D_x=8)	GI: 3648 samples (D_x=8)	GI: 4096 samples (D_x=3)	GI: 4864 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
256QAM	64800	2/15	3.8	3.8	3.9	4.1	4.3	4.6	4.8	4.8	4.6	4.6	5.2	5.2
		3/15	6.5	6.5	6.6	6.8	7.0	7.3	7.5	7.5	7.3	7.3	7.9	7.9
		4/15	8.8	8.8	8.9	9.1	9.3	9.6	9.7	9.7	9.6	9.6	10.2	10.2
		5/15	10.7	10.7	10.9	11.1	11.3	11.5	11.7	11.7	11.5	11.5	12.2	12.2
		6/15	12.8	12.8	13.0	13.2	13.4	13.6	13.8	13.8	13.6	13.6	14.3	14.3
		7/15	14.4	14.4	14.5	14.7	14.9	15.1	15.3	15.3	15.1	15.1	15.8	15.8
		8/15	16.2	16.2	16.3	16.6	16.7	17.0	17.2	17.2	17.0	17.0	17.7	17.7
		9/15	17.9	17.9	18.0	18.2	18.4	18.7	18.9	18.9	18.7	18.7	19.3	19.3
		10/15	19.5	19.5	19.7	19.9	20.0	20.3	20.5	20.5	20.3	20.3	21.0	21.0
		11/15	21.2	21.2	21.4	21.6	21.8	22.1	22.3	22.3	22.1	22.1	22.8	22.8
		12/15	23.1	23.1	23.2	23.4	23.6	23.9	24.1	24.1	23.9	23.9	24.7	24.7
		13/15	25.1	25.1	25.2	25.5	25.7	26.0	26.2	26.2	26.0	26.0	26.8	26.8
256QAM	16200	2/15	4.5	4.5	4.6	4.8	5.0	5.2	5.4	5.4	5.2	5.2	5.9	5.9
		3/15	7.0	7.0	7.1	7.3	7.5	7.8	7.9	7.9	7.8	7.8	8.4	8.4
		4/15	9.4	9.4	9.5	9.8	9.9	10.2	10.4	10.4	10.2	10.2	10.8	10.8
		5/15	11.2	11.2	11.3	11.5	11.7	11.9	12.1	12.1	11.9	11.9	12.6	12.6
		6/15	13.1	13.1	13.3	13.5	13.7	13.9	14.1	14.1	13.9	13.9	14.6	14.6
		7/15	14.8	14.8	15.0	15.2	15.3	15.6	15.8	15.8	15.6	15.6	16.3	16.3
		8/15	16.5	16.5	16.7	16.9	17.1	17.3	17.5	17.5	17.3	17.3	18.0	18.0
		9/15	18.1	18.1	18.3	18.5	18.7	18.9	19.1	19.1	18.9	18.9	19.6	19.6
		10/15	19.9	19.9	20.0	20.2	20.4	20.7	20.9	20.9	20.7	20.7	21.4	21.4
		11/15	21.6	21.6	21.7	22.0	22.1	22.4	22.6	22.6	22.4	22.4	23.1	23.1
		12/15	23.4	23.4	23.6	23.8	24.0	24.3	24.5	24.5	24.3	24.3	25.0	25.0
		13/15	25.5	25.5	25.6	25.9	26.1	26.4	26.6	26.6	26.4	26.4	27.2	27.2

Table B.6.15 Gaussian Channel 32K FFT, 1024QAM and 4096QAM Expected C/N Values

FFT Size:32K			GI: 192 samples (D_x=32)	GI: 384 samples (D_x=32)	GI: 512 samples (D_x=24)	GI: 768 samples (D_x=16)	GI:1024 samples (D_x=12)	GI: 1536 samples (D_x=8)	GI: 2048 samples (D_x=6)	GI: 2432 samples (D_x=6)	GI: 3072 samples (D_x=8)	GI: 3648 samples (D_x=8)	GI: 4096 samples (D_x=3)	GI: 4864 samples (D_x=3)
Mod	**Code-length**	**Code rate**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**	**Threshold C/N**
1024QAM	64800	2/15	5.4	5.4	5.6	5.8	5.9	6.2	6.4	6.4	6.2	6.2	6.8	6.8
		3/15	8.4	8.4	8.5	8.7	8.9	9.1	9.3	9.3	9.1	9.1	9.8	9.8
		4/15	11.0	11.0	11.1	11.3	11.5	11.7	11.9	11.9	11.7	11.7	12.4	12.4
		5/15	13.3	13.3	13.4	13.6	13.8	14.1	14.2	14.2	14.1	14.1	14.7	14.7
		6/15	15.7	15.7	15.8	16.0	16.2	16.5	16.6	16.6	16.5	16.5	17.1	17.1
		7/15	17.5	17.5	17.7	17.9	18.0	18.3	18.5	18.5	18.3	18.3	19.0	19.0
		8/15	19.7	19.7	19.9	20.1	20.2	20.5	20.7	20.7	20.5	20.5	21.2	21.2
		9/15	21.7	21.7	21.9	22.1	22.3	22.5	22.7	22.7	22.5	22.5	23.2	23.2
		10/15	23.7	23.7	23.9	24.1	24.2	24.5	24.7	24.7	24.5	24.5	25.2	25.2
		11/15	25.9	25.9	26.0	26.3	26.4	26.7	26.9	26.9	26.7	26.7	27.4	27.4
		12/15	28.2	28.2	28.3	28.6	28.7	29.0	29.2	29.2	29.0	29.0	29.8	29.8
		13/15	30.6	30.6	30.7	31.0	31.2	31.5	31.7	31.7	31.5	31.5	32.3	32.3
4096QAM	64800	2/15	6.8	6.8	6.9	7.1	7.3	7.5	7.7	7.7	7.5	7.5	8.2	8.2
		3/15	10.0	10.0	10.2	10.4	10.6	10.8	11.0	11.0	10.8	10.8	11.5	11.5
		4/15	12.9	12.9	13.1	13.3	13.4	13.7	13.9	13.9	13.7	13.7	14.4	14.4
		5/15	15.7	15.7	15.8	16.0	16.2	16.5	16.6	16.6	16.5	16.5	17.1	17.1
		6/15	18.3	18.3	18.4	18.6	18.8	19.1	19.3	19.3	19.1	19.1	19.7	19.7
		7/15	20.5	20.5	20.6	20.8	21.0	21.3	21.5	21.5	21.3	21.3	21.9	21.9
		8/15	23.0	23.0	23.2	23.4	23.5	23.8	24.0	24.0	23.8	23.8	24.5	24.5
		9/15	25.5	25.5	25.6	25.9	26.0	26.3	26.5	26.5	26.3	26.3	27.0	27.0
		10/15	28.2	28.2	28.4	28.6	28.8	29.1	29.3	29.3	29.1	29.1	29.8	29.8
		11/15	31.2	31.2	31.3	31.6	31.8	32.1	32.4	32.4	32.1	32.1	33.0	33.0
		12/15	34.1	34.1	34.3	34.6	34.8	35.2	35.5	35.5	35.2	35.2	36.3	36.3
		13/15	38.4	38.4	38.7	39.1	39.5	40.2	40.8	40.8	40.2	40.2	42.3	42.3

Annex C: ATSC 3.0 Service Examples

C.1 SINGLE PLP SERVICE

This example provides possible physical layer parameter choices for a single PLP service. The intended service is a fixed service that has a similar coverage as ATSC 1.0 (i.e., 15 dB required SNR). Given this example of physical layer parameters, the achievable data rate in the physical layer is around 25 Mbps. Therefore, a high quality 4K UHD service or multiple HD services intended for fixed reception are feasible. Note that the required SNRs for each configuration were obtained from Annex A.

Table C.1.1 Example of Physical Layer Parameters for Single-PLP Service

Frame Length		245.185 ms (Including Bootstrap)
Bandwidth		6 MHz
Preamble Parameters	FFT Size	32K
	Guard Interval	GI6_1536
	Pilot Pattern	SP D_x=8
	Signaling Protection	L1-Basic / Detail Mode 1
	# of Preamble Symbols	1
	Reduced Carriers	None
Payload OFDM Parameters	FFT Size	32K
	Guard Interval	GI6_1536
	Pilot Pattern	SP D_x=16, D_y=2
	Pilot Boosting	No Pilot Boosting
	# of Payload Symbols	48
	Time Interleaver	CTI (1024 depth)
	Frequency Interleaver	On
	First / Last SBS	Off / On
Payload BICM Parameters	Outer Code	BCH
	Inner Code	9/15 LDPC (64800)
	Constellation	256QAM
PHY Data Rate		25.156 Mbps
Required SNR (AWGN)	Simulation	15.7 dB
	Lab. Test	16.5 dB
	Field Test	16.8 dB
Required SNR (RC20)	Simulation	16.1 dB
	Lab. Test	17.1 dB
	Field Test	18.1 dB
Required SNR (RL20)	Simulation	18.1 dB
	Lab. Test	19.3 dB
	Field Test	18.0 dB

C.2 MULTIPLE PLP SUBFRAME SERVICE

This example provides possible physical layer parameter choices for a 2-PLP service, where each PLP is contained in a separate subframe. This multiple subframe configuration is useful when there are different intended services (e.g., mobile and fixed services) that require different waveform parameters such as FFT size, guard interval, or pilot patterns. In this example, PLP 0 is contained in the first subframe that may be suitable for mobile or indoor services using 8K FFT size and a denser pilot pattern. Conversely, PLP 1 in the second subframe may be suitable for a fixed service using 32K FFT size and a sparser pilot pattern.

Table C.2.1 Example of Physical Layer Parameters for 2-Subframe Service

Frame Length			245.333 ms (Including Bootstrap)
Bandwidth			6 MHz
Preamble Parameters	FFT Size		8K
	Guard Interval		GI6_1536
	Pilot Pattern		SP D_x=4
	Signaling Protection		L1-Basic / Detail Mode 1
	# of Preamble Symbols		2
	Reduced Carriers		None
Payload OFDM Parameters	1st Subframe (PLP 0)	FFT Size	8K
		Guard Interval	GI6_1536
		Pilot Pattern	SP D_x=4, D_y=2
		Pilot Boosting	No Pilot Boosting
		# of Payload Symbols	51
		Time Interleaver	CTI (1024 depth)
		Frequency Interleaver	On
		First / Last SBS	Off / On
	2nd Subframe (PLP 1)	FFT Size	32K
		Guard Interval	GI6_1536
		Pilot Pattern	SP D_x=16, D_y=2
		Pilot Boosting	No Pilot Boosting
		# of Payload Symbols	34
		Time Interleaver	CTI (1024 depth)
		Frequency Interleaver	On
		First / Last SBS	On / On
Payload BICM Parameters	1st Subframe (PLP 0)	Outer Code	BCH
		Inner Code	8/15 LDPC (16200)
		Constellation	16QAM
	2nd Subframe (PLP 1)	Outer Code	BCH
		Inner Code	9/15 LDPC (64800)
		Constellation	256QAM
PHY Data Rate	1st Subframe (PLP 0)		2.620 Mbps
	2nd Subframe (PLP 1)		17.568 Mbps
Required SNR (AWGN)	1st Subframe (PLP 0)	Simulation	6.6 dB
		Lab. Test	7.4 dB
		Field Test	7.6 dB
	2nd Subframe (PLP 1)	Simulation	15.7 dB
		Lab. Test	16.5 dB

		Field Test	16.8 dB
Required SNR (RC20)	1st Subframe (PLP 0)	Simulation	7.0 dB
		Lab. Test	7.9 dB
		Field Test	8.6 dB
	2nd Subframe (PLP 1)	Simulation	16.1 dB
		Lab. Test	17.1 dB
		Field Test	18.1 dB
Required SNR (RL20)	1st Subframe (PLP 0)	Simulation	8.8 dB
		Lab. Test	9.8 dB
		Field Test	8.7 dB
	2nd Subframe (PLP 1)	Simulation	18.1 dB
		Lab. Test	19.3 dB
		Field Test	18.0 dB

C.3 MULTIPLE PLP TDM SERVICE

This example provides possible physical layer parameter choices for a 2-PLP TDM service using a single subframe. This multiple PLP configuration uses the common waveform parameters (FFT size, guard interval and pilot patterns), and hence, 16K FFT and D_x=6, D_y=2 are used for this example to accommodate both mobile and fixed services. Due to the use of a multiple PLP configuration within a subframe, the HTI mode is configured as the time interleaver choice, which requires an integer number of FEC Blocks allowing inevitable dummy modulation values at the end of a subframe.

Table C.3.1 Example of Physical Layer Parameters for 2-PLP TDM Service

Frame Length			245.704 ms (Including Bootstrap)
Bandwidth			6 MHz
Preamble Parameters	FFT Size		16K
	Guard Interval		GI6_1536
	Pilot Pattern		SP D_x=4
	Signaling Protection		L1-Basic / Detail Mode 1
	# of Preamble Symbols		1
	Reduced Carriers		None
Payload OFDM Parameters	FFT Size		16K
	Guard Interval		GI6_1536
	Pilot Pattern		SP D_x=8, D_y=2
	Pilot Boosting		No Pilot Boosting
	# of Payload Symbols		93
	Time Interleaver		HTI (Cell Interleaver: On, CDL: Off)
	Frequency Interleaver		On
	First / Last SBS		Off / On
Payload BICM Parameters	PLP 0	Outer Code	BCH
		Inner Code	8/15 LDPC (16200)
		Constellation	16QAM
	PLP 1	Outer Code	BCH
		Inner Code	9/15 LDPC (64800)
		Constellation	256QAM
PHY Data Rate	PLP 0		2.586 Mbps

	PLP 1		17.320 Mbps
Required SNR (AWGN)	PLP 0	Simulation	6.6 dB
		Lab. Test	7.4 dB
		Field Test	7.6 dB
	PLP 1	Simulation	15.7 dB
		Lab. Test	16.5 dB
		Field Test	16.8 dB
Required SNR (RC20)	PLP 0	Simulation	7.0 dB
		Lab. Test	7.9 dB
		Field Test	8.6 dB
	PLP 1	Simulation	16.1 dB
		Lab. Test	17.1 dB
		Field Test	18.1 dB
Required SNR (RL20)	PLP 0	Simulation	8.8 dB
		Lab. Test	9.8 dB
		Field Test	8.7 dB
	PLP 1	Simulation	18.1 dB
		Lab. Test	19.3 dB
		Field Test	18.0 dB

C.4 MULTIPLE PLP LDM SERVICE

This example provides possible physical layer parameter choices for a 2-PLP service using a simple LDM configuration (i.e., PLP 0 in Core Layer and PLP 1 in Enhanced Layer within one subframe). This LDM configuration provides superior performance (around 5 dB SNR) compared to the TDM service examples described in C.2 and C.3 of Annex C.

Table C.4.1 Example of Physical Layer Parameters for 2-PLP LDM Service

Frame Length			245.704 ms (Including Bootstrap)
Bandwidth			6 MHz
Preamble Parameters	FFT Size		16K
	Guard Interval		GI6_1536
	Pilot Pattern		SP D_x=4
	Signaling Protection		L1-Basic / Detail Mode 1
	# of Preamble Symbols		1
	Reduced Carriers		None
Payload OFDM Parameters	FFT Size		16K
	Guard Interval		GI6_1536
	Pilot Pattern		SP D_x=8, D_y=2
	Pilot Boosting		No Pilot Boosting
	# of Payload Symbols		93
	Time Interleaver		CTI (1024 depth)
	Frequency Interleaver		On
	First / Last SBS		Off / On
Payload BICM Parameters	Core PLP (PLP 0)	Outer Code	BCH
		Inner Code	4/15 LDPC (16200)
		Constellation	QPSK
	Enhanced PLP (PLP 1)	Outer Code	BCH

		Inner Code	9/15 LDPC (64800)
		Constellation	64QAM
	Injection Level		2.0 dB
PHY Data Rate	Core PLP (PLP 0)		2.499 Mbps
	Enhanced PLP (PLP 1)		17.461 Mbps
Required SNR (AWGN)	Core PLP (PLP 0)	Simulation	1.9 dB
		Lab. Test	3.0 dB
		Field Test	3.3 dB
	Enhanced PLP (PLP 1)	Simulation	15.8 dB
		Lab. Test	16.7 dB
		Field Test	16.9 dB
Required SNR (RC20)	Core PLP (PLP 0)	Simulation	2.1 dB
		Lab. Test	3.4 dB
		Field Test	4.3 dB
	Enhanced PLP (PLP 1)	Simulation	16.2 dB
		Lab. Test	17.3 dB
		Field Test	18.8 dB
Required SNR (RL20)	Core PLP (PLP 0)	Simulation	3.3 dB
		Lab. Test	7.0 dB
		Field Test	4.7 dB
	Enhanced PLP (PLP 1)	Simulation	18.3 dB
		Lab. Test	19.5 dB
		Field Test	18.7 dB

End of Document