IEEE P802.11bb   
Wireless LANs

|  |  |  |  |
| --- | --- | --- | --- |
| Draft text for LC-optimized PHY for TGbb D0.1 | | | |
| Date: 2019-11-10 | | | |
| Author(s): | | | |
| Name | Company | Address | email |
| Volker Jungnickel | Fraunhofer HHI | Einsteinufer 37, 10587 Berlin, Germany | [volker.jungnickel@hhi.fraunhofer.de](mailto:volker.jungnickel@hhi.fraunhofer.de) |
| Bober, Kai Lennert | [kai.lennert.bober@hhi.fraunhofer.de](mailto:kai.lennert.bober@hhi.fraunhofer.de) |
| Andreas Bluschke | Signify |  | [andreas.bluschke@signify.com](mailto:andreas.bluschke@signify.com) |
| Matthias Wendt | [matthias.wendt@signify.com](mailto:matthias.wendt@signify.com) |
| Marcos Martinez Vazquez | Maxlinear |  | [mmartinez@maxlinear.com](mailto:mmartinez@maxlinear.com) |
| Santi Vicent Colonques | [svicent@maxlinear.com](mailto:svicent@maxlinear.com) |
| Bastien Béchadergue | OLEDCOMM |  | [bastien.bechadergue@oledcomm.net](mailto:bastien.bechadergue@oledcomm.net) |

**Abstract**

This contribution contains draft text for the LC-optimized (LCO) PHY for inclusion into D0.1 of TGbb.

Revision history:

R0: Initial revision

R1: Changed formatting and PPDU Header field

# 32. Light Communication (LC) PHY specification

## 32.1. LC PHY Introduction

### **32.1.1. Introduction to LC PHY**

### **32.1.2. LC PHY functions**

### **32.1.3. PPDU formats**

## 32.2. LC PHY Service interface

### **32.2.1. Introduction**

### **32.2.2. TXVECTOR and RXVECTOR parameters**

### **32.2.3. TRIGVECTOR parameters**

### **32.2.4. PHYCONFIG\_VECTOR parameters**

## 32.3. LC PHY

### **32.3.1. General information**

This sub-clause provides the procedure by which PSDUs are converted to and from transmissions on the light communication wireless medium.

During transmission, a PSDU (in the SU case) or one or multiple PSDUs (in the MU-MIMO downlink) are processed (i.e., scrambled and coded) and appended to the PHY preamble to create the PPDU. At the receiver, the PHY preamble is processed to aid in the detection, demodulation, and delivery of the PSDU.

The LC PHY can be operated in tree principal modes, referred to as LC common mode, LC HE mode and LC optimized mode.

The **LC common-mode** is transmitted in the wavelength range between 800 and 1000 nm in single-input single-output (SISO) operation. In the LC common mode, data subcarriers are modulated using BPSK and forward error correction (FEC) is based on convolutional coding with code rate ½. The LC common mode provides support for 20 MHz bandwidth only.

The **LC HE mode** is transmitted in the wavelength range between 800 and 1000 nm. In the LC HE mode, data subcarriers are modulated using BPSK, BPSK DCM, QPSK, QPSK DCM, 16-QAM, 16- QAM DCM, 64-QAM and 256-QAM. Forward error correction (FEC) coding (convolutional or LDPC coding) is used with coding rates of 1/2, 2/3, 3/4 and 5/6. The LC HE mode provides support for 20 MHz and 40 MHz, 80 MHz and 160 MHz contiguous channel widths, 80+80 MHz non-contiguous channel width.

The **LC optimized mode** is transmitted between 380 nm and 1000 nm. In the LC optimized mode, data subcarriers are modulated using 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 bits per symbol and forward error correction (FEC) coding is used based on LDPC with code rates of 1/2, 2/3, 5/6, 16/18 and 20/21. The LC optimized mode provides support for 50 MHz, 100 MHz and 200 MHz contiguous channel width. The LC optimized mode supports adaptive bitloading and distributed MIMO natively.

While the LC common mode is mandatory, LC HE mode and LC optimized mode are both optional. All LC devices shall support the LC common mode and should support one optional mode.

### **32.3.2. LC common mode**

**…**

### **32.3.3. LC HE mode**

**…**

### **32.3.4. LC optimized mode**

This sub-clause describes the LC optimized PHY mode. The LC optimized mode is based on a DC-biased orthogonal frequency-division multiplexing (OFDM) optimized to achieve Gigabit data rates over wireless light communication media by adding features such as adaptive bit-loading, high bandwidth as well as distributed multiple-input multiple-output (MIMO). The main characteristics of the LC optimized mode is taken over from ITU-T recommendations G.9991 (03/19) and G.9960 Corrigendum 1 (09/19).

The LC optimized mode offers data rates between 20 Mbit/s and 2 Gbit/s per data stream at 200 MHz bandwidth. The main approach is to combine high clock rate with high spectral efficiency and distributed MIMO. For error protection, low-density parity-check codes (LDPC) are used. Quadrature amplitude modulation (QAM) with variable size M of the constellation alphabet on each sub-carrier is used. Controlled by higher layers, the LC optimized PHY has means to adapt the data rate to varying channel conditions by modifying i) the QAM alphabet size M per OFDM subcarrier / subcarrier group, ii) the code rate and iii) choosing the best set of transmitters based on feedback from the STA. The main parameters are summarized in Table 32-1.

The general structure of the LC optimized transmitter is shown in Figure 32-1.



**Figure 32-1 Structure of the LC optimized transmitter**

Table 32-1 OFDM Parameters of LC optimized mode[[1]](#footnote-1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modulation** | | DC biased OFDM | | |
| **Subcarrier Spacing *FSC*** | | 195.3125 kHz | | |
| **OFDM Symbol Duration** | | 5120 ns | | |
| **Cyclic Prefix length for header (payload)** | | 1280 (k\*160) ns, where k=1,2,3,…7 | | |
| **Number of bits/subcarrier** | | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | | |
| **FEC: Information Block Size** | | LDPC: 21, 120 or 540 bytes | | |
| **Code Rates** | | 1/2, 2/3, 5/6, 16/18, 20/21 | | |
| **clock rate (CR)**  **/ MHz** | **Frequency up-shift**  ***FUS* / MHz** | ***N*(*Nused*) /**  **clock cycles[[2]](#footnote-2)** | **Gross data rate / Mbit/**s | |
| **Min.** | **Max.** |
| 50 | 25 | 256 (245) | 23 | 530 |
| 100 | 50 | 512 (501) | 47 | 1084 |
| 200 | 100 | 1024 (1013) | 96 | 2192 |

#### 32.3.4.1. Forward error correction

The structure of the forward error correction (FEC) is shown in Figure 32-2. Header and payload bits of the incoming frame are first scrambled and then encoded using a low-density parity-check (LDPC) encoder. After the LDPC encoder, the header and payload are each segmented into an integer number of symbol frames that can be handed over to adaptive bitloading.



**Figure 32-2 Structure of the error correction in LC optimized mode**

##### **32.3.4.1.1. Scrambling**

All data starting from the first bit of the PHY-frame header and ending by the last bit of the payload shall be scrambled with a pseudorandom sequence generated by the linear feedback shift register (LFSR) with the polynomial *p*(*x*) *= x*23 + *x*18 + 1, as shown in Figure 32-3.



**Figure 32-3 Scrambler for the header and payload data in LC optimized mode**

For the scrambling of the header data, the LFSR generator shall be initialized at the first bit of the header with the initialization vector 2AAAAA16 (where the LSB corresponds to C1).

For the scrambling of the payload data, if the scrambler initialization (SI) field in the LC optimized PHY frame header is not equal to zero, a second initialization is performed.[[3]](#footnote-3) For a second initialization, the first four bits of the LFSR (C1 to C4) may be set to the value of SI=C4C3C2C1, while all other bits C5 to C23 are set to 1. The first bit to be scrambled is XOR'ed with the first bit generated by the LFSR after initialization (i.e., C18 ⊕ C23 of the initialization vector). The special value 016 for SI indicates that the scrambler is not re-initialized between the header and payload. The initialization of the SI field to values other than the special value is optional.



**Figure 32-4 LDPC encoder in LC optimized PHY**

##### **32.3.4.1.2. LDPC encoder**

The FEC encoder is shown in Figure 32-4. It consists of a systematic Quasi-Cyclic Low-Density Parity-Check Block-Code (QC-LDPC-BC) encoder and a puncturing mechanism. The parameters of the FEC encoder are the number of incoming information bits, *K* (information block of bits), the number of coded bits, *NM* (coded block of bits), the number of parity-check bits, *NM* – *K,* the number of output bits, *NFEC* ≤ *NM*, (FEC codeword, whose size depends on the puncturing pattern), the mother code rate, *RM* = *K*/*NM*, defined as the code rate before puncturing and the code rate, *R*=*K*/*NFEC*, defined as the code rate after puncturing. The information block size shall be one of the values specified in Table 32-3.

The encoder shall support mother codes with rates *RM* = 1/2, *RM* = 2/3 and *RM* = 5/6. From these mother codes, codes with higher code rates shall be obtained through puncturing, as described in Clause 32.3.4.1.3. The puncturing block shall support patterns providing all code rates presented in Table 32-3.

The codeword at the output of the puncturing block is of size *NFEC* ≤ *NM*. The bits shall be output in the ascending order of codeword indices determined by vector **v**' (see below), with this order the first information bit input to the encoder will be the first at the output of the puncturing.

The code rate of the mother code, *RM* = *K/NM*, is determined by a (*NM – K*) × *NM* size parity-check matrix composed by an array of *c* × *t* circulant *b* × *b* sub-matrices **A***i,j*

The parameters *c*,*t* (0 < *c* ≤ *t*) imply a rate *RM* = (*t* – *c*)/*t*. By selecting different sets of *c*,*t*, different rates can be obtained.

The sub-matrices **A***i,j* are either a rotated identity or a zero matrix and have a size of *b* × *b*, where parameter *b* = *NM/t* is called the expansion factor of **H** and controls the code block size, *NM*.

The parity-check matrix, **H**, is described in its compact form as

A zero sub-matrix in position (*i*,*j*) is labelled with *ai,j* = –1, and a rotated identity sub-matrix is labelled with a positive integer number *ai,j* defining the number of right column shifts of the identity matrix.

This Standard defines one matrix for each mother code rate and block size.

The compact form Hc of parity-check matrix (1/2)H corresponding to mother code with rate *RM*= 1/2 (*t* = 24, *c* = 12) and number of coded bits *NM* = 336 shall be

-1 -1 -1 6 -1 -1 9 6 -1 -1 2 -1 -1 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

-1 0 -1 -1 -1 3 -1 12 1 -1 -1 3 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1

-1 9 11 -1 -1 13 -1 -1 2 12 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1

1 -1 -1 11 -1 -1 7 -1 -1 -1 11 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1 4 8 -1 -1 -1 -1 -1 2 5 4 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1

-1 3 0 -1 -1 8 -1 -1 1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1

-1 -1 -1 0 6 -1 -1 -1 -1 5 13 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1

-1 -1 -1 9 -1 -1 -1 3 -1 -1 3 1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1

9 0 13 -1 -1 12 -1 -1 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1

-1 5 -1 -1 1 4 -1 -1 5 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1

-1 -1 -1 8 -1 -1 8 -1 -1 9 0 -1 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0

10 11 -1 -1 -1 3 -1 -1 0 -1 -1 -1 4 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 0

The compact form Hc of parity-check matrix (1/2)S corresponding to mother code with rate *RM*=1/2 (*t* = 24, *c* = 12) and number of coded bits *NM* = 1920 shall be

27 -1 -1 -1 55 19 -1 30 -1 -1 -1 -1 -1 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

-1 -1 0 -1 1 -1 70 -1 47 -1 62 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1

-1 -1 41 -1 -1 -1 44 -1 -1 59 60 25 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1

16 77 -1 -1 -1 5 -1 48 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1 45 -1 27 -1 46 19 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1

-1 -1 63 -1 -1 -1 55 -1 -1 -1 48 26 10 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1

-1 -1 -1 42 -1 21 -1 58 -1 41 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1

-1 -1 -1 -1 78 0 -1 7 52 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1

-1 29 9 -1 -1 -1 37 -1 -1 -1 35 21 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1

-1 -1 22 72 -1 -1 47 -1 -1 -1 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1

35 -1 -1 -1 -1 13 -1 35 -1 70 -1 -1 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0

-1 46 28 -1 -1 -1 38 -1 -1 -1 8 -1 10 58 -1 -1 -1 -1 -1 -1 -1 -1 -1 0

The compact form Hc of parity-check matrix (1/2)L corresponding to mother code with rate *RM*= 1/2 (*t*= 24, *c*= 12) and number of coded bits *NM* = 8640 shall be

-1 34 -1 95 -1 279 -1 -1 -1 -1 248 -1 -1 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

-1 -1 0 -1 0 -1 -1 -1 -1 134 356 275 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1

51 -1 27 -1 -1 -1 -1 -1 22 152 -1 57 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1

-1 124 -1 290 -1 281 15 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1

-1 340 -1 99 336 -1 -1 1 -1 -1 -1 -1 33 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1

163 -1 46 -1 -1 -1 -1 -1 -1 306 -1 86 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1

-1 185 -1 24 -1 -1 -1 94 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1

-1 223 -1 225 325 -1 -1 -1 -1 -1 297 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1

46 -1 314 -1 -1 -1 59 -1 -1 67 -1 120 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1

-1 -1 121 -1 -1 -1 -1 161 -1 303 -1 264 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1

-1 303 -1 8 -1 185 -1 -1 138 -1 -1 -1 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0

-1 -1 312 -1 -1 -1 100 -1 -1 144 -1 307 33 166 -1 -1 -1 -1 -1 -1 -1 -1 -1 0

The compact form Hc of parity-check matrix (2/3)S corresponding to mother code with rate *RM* = 2/3 (*t*= 24, *c*= 8) and number of coded bits *NM* = 1440 shall be

49 -1 -1 21 31 -1 57 -1 -1 19 -1 29 2 -1 19 -1 -1 0 -1 -1 -1 -1 -1 -1

-1 7 22 -1 -1 37 -1 32 10 -1 26 -1 -1 59 -1 48 -1 0 0 -1 -1 -1 -1 -1

53 -1 -1 20 50 -1 -1 3 16 -1 49 -1 -1 28 14 -1 -1 -1 0 0 -1 -1 -1 -1

-1 58 23 -1 -1 15 54 -1 -1 5 -1 18 49 -1 -1 13 -1 -1 -1 0 0 -1 -1 -1

55 -1 -1 58 -1 9 -1 26 57 -1 41 -1 31 -1 21 -1 -1 -1 -1 -1 0 0 -1 -1

-1 10 49 -1 59 -1 7 -1 -1 30 -1 18 -1 48 -1 7 59 -1 -1 -1 -1 0 0 -1

48 -1 -1 50 18 -1 -1 11 52 -1 59 -1 -1 37 -1 10 0 -1 -1 -1 -1 -1 0 0

-1 24 16 -1 -1 0 53 -1 -1 41 -1 38 51 -1 58 -1 59 8 -1 -1 -1 -1 -1 0

The compact form Hc of parity-check matrix (2/3)L corresponding to mother code with rate *RM* = 2/3 (*t*= 24, *c*= 8) and number of coded bits *NM* = 6480 shall be

78 -1 -1 167 237 -1 3 -1 266 -1 -1 102 153 -1 -1 212 -1 0 -1 -1 -1 -1 -1 -1

-1 83 189 -1 -1 68 -1 178 -1 90 205 -1 -1 13 4 -1 -1 0 0 -1 -1 -1 -1 -1

-1 226 147 -1 46 -1 -1 76 -1 116 -1 211 -1 112 -1 118 -1 -1 0 0 -1 -1 -1 -1

92 -1 -1 214 -1 236 241 -1 157 -1 143 -1 214 -1 207 -1 -1 -1 -1 0 0 -1 -1 -1

144 -1 -1 258 264 -1 53 -1 114 -1 172 -1 -1 82 262 -1 62 -1 -1 -1 0 0 -1 -1

-1 153 120 -1 -1 199 -1 126 -1 61 -1 183 15 -1 -1 134 -1 -1 -1 -1 -1 0 0 -1

-1 100 -1 141 -1 36 -1 17 -1 156 -1 124 162 -1 -1 57 0 -1 -1 -1 -1 -1 0 0

196 -1 187 -1 73 -1 80 -1 139 -1 57 -1 -1 236 267 -1 62 256 -1 -1 -1 -1 -1 0

The compact form Hc of parity-check matrix (5/6)S corresponding to mother code with rate *RM* = 5/6 (*t*= 24, *c*= 4) and number of coded bits *NM* = 1152 shall be

-1 13 32 47 41 24 -1 25 22 40 1 31 8 15 20 15 42 30 13 3 -1 0 -1 -1

25 46 15 43 45 29 39 47 23 38 39 12 -1 21 -1 38 33 0 0 -1 39 0 0 -1

35 45 45 38 14 16 6 11 -1 18 7 41 35 17 32 45 41 -1 18 17 0 -1 0 0

9 32 6 22 26 31 9 8 22 32 40 4 18 40 36 -1 -1 23 31 41 39 20 -1 0

The compact form Hc of parity-check matrix (5/6)L corresponding to mother code with rate *RM* = 5/6 (*t*= 24, *c*= 4) and number of coded bits *NM* = 5184 shall be

-1 47 146 203 184 112 -1 116 103 181 3 140 38 68 91 70 191 138 62 14 -1 0 -1 -1

117 203 67 194 206 133 174 212 104 171 176 56 -1 96 -1 167 149 4 1 -1 177 0 0 -1

153 206 198 173 55 72 28 53 -1 82 34 186 161 80 144 204 187 -1 84 77 0 -1 0 0

44 147 27 83 118 130 41 38 100 146 183 19 85 180 163 -1 -1 106 140 185 177 94 -1 0

In the following, the encoding operation is described. The encoder shall support the coded block sizes and rates presented in Table 32-3. The parity-check matrix  used to encode a block of information bits is selected according to the mother code indicated in Table 32-3.

The encoding process shall be as follows[[4]](#footnote-4):

The encoder supports the coded block sizes and rates presented in Table 32-3. The parity-check matrix **H** used to encode a block of information bits is selected according to the mother code indicated in Table32-3. The encoding process shall be as follows:

1) A group of incoming *K* information bits **u**=[*u*0, *u*1,...,*u*K-1] are collected and copied to the output of the encoder to form a block of systematic code bits.

2) *NM–K* parity-check bits, **p**=[*p*0, *p*1,...,*p*NM-K-1] are computed using the parity-check matrix **H** and the information block **u**. The resulting coded block **v**= [**u** | **p**] satisfies the parity check equations **vHT**=**0**. Here **0** is a zero row vector of dimension *NM*-*K*.

3) The *NM*-*K* parity check bits **p** are copied to the output of the encoder as a block of parity check bits **p**=[*p*0, *p*1,...,*p*NM-K-1] to form the output coded block **v** = [**u** | **p**] = [*v*0, *v*1,...,*v* NM -1].

4) The output of the encoder **v** is the input to the puncturing block (see Figure 32-4).

##### **32.3.4.1.3. Puncturing**

Puncturing shall discard some of the coded block bits to achieve a higher code rate (*R*). Puncturing is applied to both information and parity-check bits. The puncturing block uses the puncturing patterns specified in Table 32-2. The puncturing patterns are denoted as , where *T* is the length of the puncturing pattern and *i* is the number of zeros in the pattern.[[5]](#footnote-5)

Table 32- Puncturing patterns

|  |  |
| --- | --- |
|  | **Puncturing pattern** |
|  | [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1] |
|  |  |
|  |  |
|  |  |
|  |  |

The coded block **v** input to the puncturing block shall be processed using the puncturing pattern as follows. For the pattern , the puncturing block shall omit all incoming coded bits *vt*,t=0,…,NM-1 for which . The resulting output FEC codeword will be **v**’ = [*v*0, *v*1,..., *v* NFEC -1] with *NFEC* ≤ NM.

##### **32.3.4.1.4. FEC encoding parameters**

The FEC encoding scheme shall support the encoding parameters specified in Table 32-3.

**Table 32-3 FEC Encoding parameters**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Code rate, *R*** | **Information block size, *K*** | **Puncturing pattern, PP** | **Mother code matrix** | **FEC codeword size, *NFEC*** |
| For header | 1/2 | PHYH = 168 |  | (1/2)H | 336 |
| For payload | 1/2 | 960 |  | (1/2)S | 1920 |
| 1/2 | 4320 |  | (1/2)L | 8640 |
| 2/3 | 960 |  | (2/3)S | 1440 |
| 2/3 | 4320 |  | (2/3)L | 6480 |
| 5/6 | 960 |  | (5/6)S | 1152 |
| 5/6 | 4320 |  | (5/6)L | 5184 |
| 16/18 | 960 |  | (5/6)S | 1080 |
| 16/18 | 4320 |  | (5/6)L | 4860 |
| 20/21 | 960 |  | (5/6)S | 1008 |
| 20/21 | 4320 |  | (5/6)L | 4536 |

#### 32.3.4.2. Adaptive Bitloading

After FEC, codewords map onto OFDM subcarriers (tones). Tone mapping is also performed on unused subcarriers which are stuffed with random data. Data are mapped onto each tone depending on a bit allocation table (BAT) which can be either predefined or modified at runtime based on feedback obtained over the reverse link direction. According to the BAT, bits are arranged in symbols which are then mapped onto complex-valued IQ constellation points the modulation alphabet size of which depends on the number of bits mapped onto each tone.

In the following, the OFDM modulator characteristics is described.

##### **32.3.4.2.1 Subcarrier spacing and indexing**

The subcarrier spacing *FSC* is the frequency spacing between any two adjacent subcarriers.

The physical index *i* corresponds to the order of subcarriers in ascending frequency. The subcarrier with physical index *i* shall be centered at frequency *f* = *FUS* – (*N*/2 – *i*) × *FSC*. The index *i* goes from 0 to *N* – 1.

The logical index indicates the order in which data is loaded on subcarriers. In the LCO PHY, the physical index and the logical index shall be the same, i.e., the subcarriers are loaded in order of ascending frequency.

##### **32.3.4.2.2. Tone mapping**

The tone mapper divides the incoming symbol frames of the header and payload into groups of bits (according to the BATs and subcarrier grouping being used) and associates each group of bits with specific subcarriers on to which these groups shall be loaded. This information along with subcarrier-specific gain scaling values, see transmit spectrum shaping in Clause **32.3.4.2.3.**, are passed to the constellation encoder.

Not all subcarriers may always be used for data transmission. As LC does not cause interference to radio waves, the LCO PHY shall use a simplified subcarrier mapping scheme, compared to G.9960 Cor. 1 (09/19). Nonetheless, some subcarriers may be permanently masked or dynamically switched off, depending on the channel characteristics.

For the purpose of tone mapping, the LCO PHY shall distinguish the following types of subcarriers.

1. Supported subcarriers (SSCs) are those on which transmission is allowed.[[6]](#footnote-6)
   1. Active subcarriers (ASCs) have loaded bits (*b* ≥ 1) for data transmission, shall be subject to constellation point mapping, constellation scaling and constellation scrambling. [Editors note: reference the corresponding clauses]
   2. Inactive subcarriers (ISCs) not having data bits loaded (e.g., because SNR is low), may be used for measurement or other purposes and shall be subject to transmit power shaping. [Editors note: reference the corresponding clauses]
2. Masked subcarriers (MSCs) are those on which transmission is not allowed, i.e., the gain on these subcarrier shall be set to zero. In this Standard, only permanently masked subcarriers (PMSCs) are considered which are never allowed for transmission.

Tone mapping shall be defined by a bit allocation table (BAT). It associates subcarrier indices with the number of bits to be loaded on a subcarrier. The BAT can be predefined or defined at runtime.[[7]](#footnote-7) The used BAT is indicated to the receiving node in the BAT\_ID field in the LCO PHY header. Up to 32 BAT\_IDs can be defined as given in Table 32-4.

**Table 32-4 Bit allocation tables used in the LC optimized PHY**

|  |  |  |
| --- | --- | --- |
| **BAT\_ID** | **Type** | **Content** |
| 0 | Predefined | uniform 1-bit loading on all subcarriers, except MSC |
| 1 | Predefined | uniform 2-bit loading on all subcarriers, except MSC |
| 2 to 7 | Predefined | Reserved by ITU-T |
| 8 to 31 | Runtime | see below |

A runtime BAT associates indices of SSCs with the number of bits to be loaded on each subcarrier. The subset of indices in the BAT with the number of loaded bits *b* > 0 identifies the ASC.

The number of bits loaded on any subcarrier shall not exceed the maximum number of bits allowed (see Clause **32.3.4.2.3.**). The number of bits shall also meet the bit loading capabilities of the communicating nodes, as advertised by them prior to communication. [Editors note: Include an example BAT]

A runtime BAT can be defined by the receiving node (receiver-defined BAT) or selected by the transmitting node (transmitter-determined BAT). Runtime BATs shall be signaled from the STA that generates the BAT to the communicating STA prior to sending any data by using a specific BAT feedback protocol specified in Clause 31.X [Editors note: reference the corresponding clauses]

For runtime BATs the number of bits is variable in general and depends on the signal-to-noise ratio (SNR) on each subcarrier. The runtime BAT is normally suggested by the receiver and communicated to the transmitter using the BAT feedback protocol. [Editors note: reference the corresponding clauses]

A runtime BAT uses subcarrier grouping of *G* = 1 (no grouping as default), 2, 4, 8, and 16 subcarriers on consecutive frequencies where all subcarriers of the same group use the same bit loading. If subcarriers in a group are masked, bit loading is applied to the SSCs only.

The header shall use a uniform loading of two bits per subcarrier on all subcarriers.

Two types of probe symbols are specified: silent symbols and channel estimation probe symbols. Tone mapping shall apply to these symbols according to the following:

• For silent symbols, all subcarriers shall be considered as MSCs (masked subcarriers).

• Channel estimation probe symbols shall be modulated using a uniform loading of two bits per subcarrier on all SSC sets. For these probe symbols, the ISC set shall be equal to the SSC set. All ISC subcarriers shall be modulated by a pseudorandom sequence of bits, as described below.

Additional channel estimation symbols (ACE) shall be modulated using a uniform loading of two bits per subcarrier on all SSC sets. For the ACE, the ISC shall be equal to the SSC set. All ISC subcarriers shall be modulated by a pseudorandom sequence of bits, as described below.

**Table 32-5 Example LFSR seeds for an initial seed of 7FFFFF16**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| k | Sk | k | Sk | k | Sk | k | Sk |
| 1 | 7FFFFF16 | 17 | 07628716 | 33 | 03714416 | 49 | 1BEDE616 |
| 2 | 26B48916 | 18 | 3E1A3116 | 34 | 27858716 | 50 | 608D6B16 |
| 3 | 278A9116 | 19 | 05DE6D16 | 35 | 2CF7F716 | 51 | 4B75D316 |
| 4 | 15F4ED16 | 20 | 5C5B4E16 | 36 | 027D4616 | 52 | 22BA6416 |
| 5 | 5B4CB116 | 21 | 59641316 | 37 | 70A7EB16 | 53 | 7D064616 |
| 6 | 2F021F16 | 22 | 0613D916 | 38 | 4C622C16 | 54 | 7F56E616 |
| 7 | 7A64C116 | 23 | 19504A16 | 39 | 54DC6816 | 55 | 61433316 |
| 8 | 414CD716 | 24 | 50FDE016 | 40 | 01715E16 | 56 | 4F136816 |
| 9 | 649D5E16 | 25 | 5CD04816 | 41 | 274A7B16 | 57 | 7359EF16 |
| 10 | 13482616 | 26 | 66C64616 | 42 | 55238D16 | 58 | 2D86A916 |
| 11 | 2A3DFC1 | 27 | 7169B316 | 43 | 008B0616 | 59 | 25373D16 |
| 12 | 2B957016 | 28 | 48049716 | 44 | 3FA25516 | 60 | 25846616 |
| 13 | 3C677716 | 29 | 053FE316 | 45 | 777A6A16 | 61 | 4CE92A16 |
| 14 | 75798616 | 30 | 51F1B116 | 46 | 5154DD16 | 62 | 6B7E3D16 |
| 15 | 10396216 | 31 | 7D2BA016 | 47 | 55C20316 | 63 | 760B3416 |
| 16 | 0DB87B16 | 32 | 11E4D816 | 48 | 0D21F916 | 64 | 761EA616 |

Unloaded SSC shall be loaded with a pseudorandom binary sequence defined by the linear feedback shift register (LFSR) generator with the polynomial *p*(*x*) *= x*23 *+ x*18 *+* 1 shown in Figure 32-5.



Figure 32-5 LFSR for modulation of unloaded supported subcarriers.

The LFSR generator shall be initialized at the beginning of each OFDM symbol by using an initial seed assigned by the MAC layer (see 31.X). [Editors note: reference the corresponding clauses] The *i*th payload symbol uses the modified seed *Sk* where *k* is equal to (*i*–1, modulo 64) + 1, where *i* = 1, 2, 3, 4,.... *Sk* is generated by advancing the LFSR by 8192\*(*k*-1) from the original seed. An example of LFSR seeds for an initial seed of 7FFFFF16 is provided in Table 32-5.

Note that seeds *S*1 to *S*64 are used to initialize the LFSR for payload symbols 1-64, 65-128 and so on. The LSB of the seed S*k* corresponds to *c*1.

The initial seed shall be chosen among the pool of allowed seeds described in Clause 31.X. [Editors note: reference the corresponding clauses]

The LFSR shall be advanced by two bits for each subcarrier (for both SSC and MSC) of each symbol of the payload. Two LFSR bits corresponding to the subcarrier index 0 are (*c*1, *c*2) of the initialization seed. Two LFSR bits corresponding to the subcarrier index 1 are (*c*1, *c*2) after two shifts, and so on. For modulation of unloaded subcarriers, ACE symbols shall be treated in the same manner as payload symbols.

The modulation of inactive subcarriers (ISC, not loaded with encoded payload bits) shall be as follows

1) Starting at the beginning of the first payload OFDM symbol, each subcarrier from the ISC set is modulated with the two bits which are the LSBs of the LFSR, *c*1, and *c*2 using the 2-bit constellation mapping defined in Clause 32.3.4.3.3. (*c*1 is transmitted first). [Editors note: check the consistency of claus numbers]

2) In every OFDM symbol of payload, if the number of bits in the symbol frame does not fill the entire symbol, the bits from the LFSR shall be used to fill the remainder of the symbol frame, by taking the sequential groups of *m* LSBs of the LFSR and mapping them on to the remaining subcarriers so that LSB of LFSR is transmitted first and in the order defined by the current BAT, where *m* is the number of bits allocated for that subcarrier by the BAT. For the first padded subcarrier, if *n* bits of the *m* loaded bits are data bits (*n* < *m*), these *n* data bits shall be loaded as the LSBs of the group of bits mapped on the constellation point, and the *m-n* bits of the LFSR shall be used as the MSBs of the group of bits mapped on the constellation point starting from LSB of LFSR.

3) In the case of a PROBE frame, starting at the beginning of the first payload OFDM symbol, each subcarrier from the ISC set shall be modulated with the two bits which are the LSBs of the LFSR, *c*1 and *c*2, using 2-bit constellation mapping defined in 1.1.3 (*c*1 is transmitted first).

The bits from LFSR are loaded on subcarriers in the order of logical indices (i.e., in the same way as data is loaded over payload symbols), according to subcarrier indexing defined in **32.3.4.1.**. Modulation of unloaded subcarriers shall start from the unloaded SSC with the lowest logical index of the first payload symbol, continue in ascending order of logical indices until the unloaded SSC with the highest logical index of the first payload symbol, continue with the unloaded SSC with the lowest logical index of the second payload symbol, continue in ascending order of logical indices until the unloaded SSC with the highest logical index of the second payload symbol, and continue until the unloaded SSC with the highest logical index of the last payload symbol. The ASCs are loaded according to the corresponding BAT as defined in **32.3.4.2.1**.

##### **32.3.4.2.3. Constellation encoder**

Constellation mapping associates every group of bits loaded on to a subcarrier, with the values of *I* (in-phase component) and *Q* (quadrature-phase component) of a constellation diagram. Each incoming group of *b* bits {*d*b–1, *d*b–2, … *d*0} shall be associated with a specific value of *I* and *Q* computed described in this Clause.

Each group of bits {*d*b–1, *d*b–2, … *d*0} shall be mapped on to the constellation mapper with the LSB bit, *d*0, first.

For **even number numbers of bits**, *b, i.e.* (2, 4, 6, 8, 10, 12), square-shaped constellations are used. Support of all even order constellations is mandatory at both the transmitter and the receiver. With square-shaped constellations, 2*b* constellation points are set as a square.

For *b*=2, bit *d0* is mapped to I branch as {0, 1}🡪{-1, 1} and same way d1 is mapped to Q branch.

For *b*=4, bits [d1 d0]aremapped to the I branch as {00, 10, 11, 01}🡪{-3, -1, 1, 3} and same way bits [d3 d2]aremapped to the Q branch.

For *b* ≥ 4 constellations are derived as follows:

1) Divide the incoming group of *b* bits into two equal subgroups, so that *b*/2 LSBs form the *I*-subgroup and *b*/2 MSBs form the *Q*-subgroup; both subgroups are incoming LSBs (which are *d*0 and *d*b/2, respectively) first.

2) Compute values of *I* and *Q* for the incoming group {*db*–1, *db*–2, … *d*0} as:

*I* = *sgnI* × *valI*

*Q* = *sgnQ* × *valQ* where

*sgnI = 2 × d0 – 1,*

*sgnQ* = 2 × db/2 –1,

*valI* = *|Ib*–2 – 2*b*/2–1| and *valQ* = *|Qb*–2 – 2*b*/2–1|

where |.| is the absolute value. *Ib–*2 and *Qb*–2 are the values of *I* and *Q* for the incoming (*b*-2)-bit group {*db*-–1, *db*–2, … *db*/2+1, *db*/2–1, … *d*1}, i.e., with *d*0 and *d*b/2 being removed.

For **odd numbers of bits**, (1, 3, 5, 7, 9, 11), specific constellations shall be used. Support of all odd constellations shall be mandatory at the transmitter, while with *b* ≥ 5 they shall be optional at the receiver.

For *b*=1, bit *d0* is mapped to I branch as {0, 1}🡪{-1, 1}.

For *b*=3, if *d1=*1, bit *d0* is mapped to I branch as {0, 1}🡪{-1, 1} and same way d2 is mapped to Q branch. If *d1=*0, bit groups [*d2* *d0*] map as [0 1] 🡪 {3, -1}, [0 0] 🡪 {-1, -3}, [1 1] 🡪 {1, 3}, and [1 0] 🡪 {-3, 1}.

For *b* > 3, cross-shaped constellations are used. First, 2*b* constellation points form a rectangle, with *MI* = 2*B*1 columns (*MI* points on the *I*-axis) and *MQ* = 2*B*2 rows (*MQ* points on the *Q*-axis), where *B*1 = *ceiling*(*b*/2) and *B*2 = *floor*(*b*/2). The mapping of points uses following steps

1. Divide the incoming group of bits into two subgroups, so that *B*1 LSBs form the first subgroup (*I*-group) and *B*2 MSBs form the second subgroup (*Q*-group); both subgroups are incoming LSBs (which are *d*0 and *dB*2+1, respectively) first.
2. Compute values of *I* and *Q* for the incoming group {*db*–1, *db*–2, … *d*0} as

*I* = *sgnI* × *valI*

*Q* = *sgnQ* × *valQ*

*sgnI* = 2 × *d*0 – 1

*sgnQ* = 2 × *d*B1 – 1

*valI* = *|I*2*×B*1| and *valQ* = *|Q2×B2*|

*I*2*×B*1 is the value of *I* for (2×*B*1)-bit group {0, *db*–1, *db*–2, … *d*0} and *Q*2*×B*2 is the value of *Q* for (2×*B*2)-bit group {*db*–1, *db*–2, … *d*1} computed as defined for even constellations.

1. Transform *s* = (*MI* – *MQ*)/4 columns of constellation points in each quadrant having highest absolute values of *I* (positive or negative) into rows of *Q* by changing their {*I*, *Q*} coordinates to {*I'*, *Q'*} in the following way:

|*Q'*| = |*I*| - 2*s*, and sign (*Q'*) = sign (*I*)

|*I'*| = *MQ* – |*Q*|, and sign (*I'*) = sign (*Q*).

**Constellation point scaling** shall be done as follows. Each constellation point (*I*, *Q*), corresponding to the complex value *I + jQ* at the output of the constellation mapper, shall be scaled by the power-normalization factor χ(*b*) and the frequency-domain spectrum shaping coefficient *tss*.

The values (*I, Q*) for each constellation point shall be scaled so that all constellations have the same average power. The **power normalization factor**, χ(*b*), for *b*-bit loading is given in Table 32-4.

Table 32-4 Power normalization factor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *b* | 1 | 2 | 3 | 4 | 5 | 6 |
| χ(*b*) | 1 | 1/√2 | 1/√6 | 1/√10 | 1/√20 | 1/√40 |
| *b* | 7 | 8 | 9 | 10 | 11 | 12 |
| χ(*b*) | 1/√82 | 1/√170 | 1/√330 | 1/√682 | 1/√1322 | 1/√2730 |

**Transmit spectrum shaping** is achieved by a scaling factor *tss* defined for each subcarrier. The *tss* values are set by the transmitter and shall be in the range 0 and -30 dB in steps of -0.5 dB. Smaller values of *tss* provide more attenuation. The value *tss* = 0 dB corresponds to no attenuation on the particular subcarrier. If no spectrum shaping is applied, all *tss* values shall be equal to 0 dB. The values of *tssi* are relevant only for SSCs, and shall be ignored for MSCs.



Figure 32-6 LFSR generator for the constellation scrambler.

**Constellation scrambling** shall be done as follows. The phase of constellation points obtained from the constellation mapper shall be shifted according to the pseudorandom bit sequence generated by a linear feedback shift register (LFSR), as shown in Fig. 32-6.

The LFSR generator shall implement the polynomial *g*(*x*) = *x*13 + *x*12 + *x*11 + *x*8 + 1 and shall be advanced by two bits for each subcarrier. Bits shall be assigned to subcarriers in order of logical index. The two LSBs of the register shall be taken to determine the phase shift as shown in Table 32-5.

Table 32-5 Constellation phase shift depending on LFSR output

|  |  |  |
| --- | --- | --- |
| **LFSR output** | | **Phase shift (rad)** |
| s2 | s1 |  |
| 0 | 0 | 0 |
| 0 | 1 | π/2 |
| 1 | 0 | π |
| 1 | 1 | 3 π/2 |

For the header, ACE and payload, the shift of the LFSR for subcarrier index *i* shall be 2*i* (for both SSC and MSC). Two LFSR bits corresponding to the subcarrier index 0 are (*s*1, *s*2) of the initialization seed. Two LFSR bits corresponding to the subcarrier index 1 are (*s*1, *s*2) after two shifts, and so on. For preamble, the shift of the LFSR for subcarrier index (*i·km*) shall be 2*i* where *km* denotes the subcarrier spacing multiplier for preamble section *m* (see clause 32.3.4.3.4.).

The LFSR generator shall be initialized with the seed 1FFF16 for each OFDM symbol. The LSB of the seed corresponds to *s*1. The constellation scrambling shall be applied to the PHY header, ACE and all payload symbols by rotating the originally mapped constellation point Z0*i*,l by the phase shift θ to obtain the complex value for the Z*i*,l for input to the IFFT (see clause 32.3.4.3.4.).



#### 32.3.4.3. OFDM Modulator

The functional diagram of the OFDM modulator is presented in Figure 32-7.

The OFDM modulator consists of the following major parts: IDFT, cyclic prefix and frequency up-shift. The incoming signal to the modulator at the *l*th OFDM symbol in the present frame for a single subcarrier, with index *i,* is the complex value *Zi,l* generated by the constellation encoder (for symbols of the header and the payload) or by the preamble generator (for symbols of the preamble). Time-domain samples generated by the IDFT, after adding the cyclic prefix, are frequency up‑shifted by *F*US. The functional diagram of OFDM modulator is presented in Figure 32-7. All aspects of the signal processing used in the OFDM modulator shall comply with the following equations and text.



Figure 32-7 OFDM modulator for LC optimized PHY

##### **32.3.4.3.1 IDFT**

The IDFTconverts the stream of the *N* complex numbers *Zi,l* at its input into the stream of *N* complex time-domain samples *Xn,l*. The input numbers *Zi,l* represent *N* mapped blocks of data, where the *i*th block of data represents the complex value Z*i,l* of the *i*th modulated subcarrier of the OFDM signal, where *i* = 0, 1, … *N*–1 is the subcarrier index and *l* is the sequential number of the OFDM symbol within the current frame, excluding the preamble. The conversion shall be performed in accordance with the equation:

.

where *MF* denotes the total number of OFDM symbols in the current frame excluding the preamble symbols, and the value of *N* represents the maximum number of possibly modulated subcarriers in the OFDM spectrum and shall be a power of 2: *N =* 2*k*, where *k* shall be an integer. The value of *Zi,l* for all masked subcarriers shall be set to 0.

##### **32.3.4.3.2. Cyclic Prefix**

The Cyclic Prefix (CP) provides a guard interval between adjacent OFDM symbols to protect against inter-symbol interference (ISI). The CP of the *l*th OFDM symbol in the frame shall be implemented by prepending the last *NCP(l)* samples of the IDFT output to its output *N* samples to create an OFDM symbol.

The order of samples shall be as follows. The first sample is the IDFT output sample *N*−*N*CP*(l).* The last sample of the CP is the IDFT output sample *N-*1; the next sample is the IDFT output sample 0. Total OFDM symbol duration is *NW*(*l*) = *N* + *NCP*(*l*) samples.

After cyclic extension as described above, time-domain samples shall comply with the following equations



The number of IDFT samples *N* shall be the same for all symbols of the same frame. The value of *N*CP(*l*) (and the duration of the OFDM symbol *N*w(*l*), accordingly) may change during the course of the frame, as follows. All symbols of the header and ACE symbols shall have use the long CP with *NGI-HD*. All the rest of the payload symbols shall use the short CP *NGI*, where *N*GI is selected from the valid values defined in table 32-1 and indicated in the header.

Thesymbol timing is as follows. The PHY frame consists of a preamble followed by an integer number, *MF*, of OFDM symbols. The first symbol following the preamble (the first symbol of the PHY-header) shall have symbol count 0, and the last symbol of the frame shall have symbol count *MF* − 1. The time position of each symbol in the frame is defined by sample count. The first sample of the symbol with symbol count 0 shall have sample count *M*(*0*) *= Npr*, where *Npr* is the number of samples in the preamble. The count of the first sample of the *l*th symbol (*l* = 1, 2, … *MF* −1) in the frame shall be

where *N*S(*k*) *= N* + *N*CP(*k*) and *N*S(*k*) may be different for symbols of the header and payload.

##### **32.3.4.3.3. Frequency up-shift**

The frequency up-shift offsets the spectrum of the transmit signal shifting it by *FUS*.The value of *FUS* shall be a multiple of the subcarrier frequency *FSC*: *F*US = *m*\**F*SC, where *m* is an integer and m=N/2.

The real and imaginary components of the signal after frequency up-shift shall be as follows:



where *un*/*p* is *un* after interpolation with factor *p* which shall be equal to or higher than 2.

##### **32.3.4.3.4. Output signal**

The output signal of the OFDM modulator shall be the real component of *sn*:

Sout-HF = Re(sn)

The OFDM modulator provides a bipolar output signal in general. Unipolar OFDM signals are obtained by

1. converting a drive current for the optical transmitter from the output voltage
2. adding a constant bias current (which may include the dimming in case of visible light),
3. setting the RMS modulation amplitude as a constant fraction of the bias.

The frontend will force below-zero signals to zero, thus clipping any negative parts of the waveform.

##### **32.3.4.3.5. OFDM parameters**

Table 32-6 summarizes valid values of control parameters of the OFDM modulator described in the clauses above. This list is a superset of possible parameters which shall be used over several media; a list of valid values of modulation parameters and their valid combinations for light communication is also explained.

| Table 32-6 – Valid OFDM control parameters | | | |
| --- | --- | --- | --- |
| Notation | Parameter | Valid values or range | Note |
| *N* | Number of subcarriers | 256, 512, 1024, 2048, 4096 | The LC optimized PHY uses *N*=512 (100 MHz) and *N*=1024 (200 MHz). |
| *FSC* | Subcarrier spacing [kHz] | 24.4140625 × *k*, *k* = 1, 2, 4, 8, 16, 32, 64 | The LC optimized PHY uses *k*=8, *FSC*=195.3125 kHz. |
| *NGI* | Guard interval [samples] | *k* × *N*/32, *k* = 1, 2, 3, … 8 |  |
| *NGI-HD* | Guard interval of the header | *N*/4 |  |
| *NGI-DF* | Default guard interval of the payload | *N*/4 | *NGI-DF*≥ *NGI* |
| *FUS* | Up-shift frequency, [kHz] | *m*×*FSC*,  *m* ≥ *N*/2 | *m* is an integer; the LC optimized PHY uses m=*N*/2. |
| NOTE – Guard intervals are expressed in samples at Nyquist rate. | | | |

#### 32.3.4.4. PPDU format

The LC optimized PHY uses the physical protocol data unit (PPDU) format. An overview is given in Figure 32-8. The PPDU comprises a preamble, a channel estimation sequence, a header, a header check sequence (HCS), additional channel estimation symbols (ACE) and the physical layer convergence protocol (PLCP) service data unit (PSDU) which usually carries MAC layer information.

Data

Preamble

Header

MIMO RS

**Figure 32-8 PPDU format for LC optimized mode**

##### **32.3.4.4.1. Preamble**

The preamble is prepended to every PHY frame. It is intended to assist the receiver in detecting, synchronizing to the frame boundaries. The preamble consists of *NI*=3 sections. The first section is intended for fast automatic gain control. The second section is intended for coarse time synchronization. The third section is intended for fine time synchronization and channel estimation needed to decode the header and payload information.

Each section *I* comprises *NI* repetitions of an OFDM symbol (*SI*) employing subcarrier spacing *kI* × *FSC*, where *F*SC denotes the subcarrier spacing of the payload. A zero value for *NI* means that section *I* is not included in the preamble. The values of *kI* shall be selected from the set 1, 2, 4 or 8. The preamble subcarriers of section *I* shall be one in every *kI* subcarriers with respect to the subcarriers used for the payload OFDM symbol starting from subcarrier zero.

Table 32-7 – General structure of the preamble

|  |  |  |  |
| --- | --- | --- | --- |
|  | **1st section** | **2nd section** | **3rd section** |
| Number of symbols (*NI*) (Note 1) | *N*1 | *N*2 | *N*3 |
| Subcarrier spacing (*kI* × *F*SC) | *k*1 | *k*2 = *k*1 (Note 2) | *k*3 |
| OFDM symbol (*SI*) | *S*1 | *S*2 = – *S*1  (Note 3) | *S*3 |
| NOTE 1 – Windowing is not used for LC.  NOTE 2 – The subcarrier spacing of the 2nd section shall be equal to the subcarrier spacing of the 1st section.  NOTE 3 – The OFDM symbol of the 2nd section shall be an inverted waveform of the 1st section. | | | |

The number of repetitions of OFDM symbol *SI* (*NI*) in each of the preamble sections may be a non-integer number to incorporate an optional guard interval between sections provided that a fraction of *NI* is consistent with the guard interval specified in Table 32-6. The specific preamble types and construction methods are defined in clause 7.2.

In the LC optimized PHY, the first section comprises 10 repetitions of an OFDM symbol ***S*** employing subcarrier spacing *4* × *FSC*, where *F*SC denotes the subcarrier spacing.

The second section comprises 4 repetitions of the inverted symbol **-*S***.

The third section comprises 2.5 repetitions of an OFDM symbol ***S*** employing subcarrier spacing 1× *FSC*, where *F*SC denotes the subcarrier spacing. The number of repetitions is a non-integer number to indicate that two OFDM symbols are used so that the CP is the same as in the PHY header, i.e. *NCP=NGI-HD=N/4*.

The first two sections use the subcarriers with indices 4, 8, 12, 16, …, N-4, except those being masked. In the third section, all subcarriers with indices 0, 1, 2, …, N-1 are used, except those being masked.

For the non-masked subcarriers of the preamble, a bit sequence of all ones shall be mapped using the 1-bit constellation as specified in clause **32.3.4.2.2.**. Other bit sequences are for further study.

The constellation scrambler LFSR generator shall be initialized at the beginning of each one of the used preamble sections to a seed that is section dependent.

In the first section, the LFSR generator of the constellation scrambler is initialized as 16E616=1011011100110[[8]](#footnote-8). In the third section, the LFSR generator of the constellation scrambler is initialized as 110516= 1000100000101.

For preamble generation, the output of the mapper shall be subsequently rotated using the two bits that are the LSBs of the LFSR, s1 and s2, as defined in Table 32-5 (constellation scrambler) resulting in constellation point *Zi*. The LFSR shall be advanced by two bits for each preamble's subcarrier (for both SSC and MSC) in the order specified in clause **32.3.4.2.3.**.

##### **32.3.4.4.3. Header**

The PHY header is PHYH bits long. It is transmitted over *D* consecutive OFDM symbols, where *D* may be either 1 or 2. The content of the core part is protected by the 16-bit header check sequence (HCS).

Table 32-8 – Header field of the LC PHY PPDU

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Field | Number of Bits | Description |
| B0-B3 | Frame Type (FT) | 4 | The FT field indicates the PHY frame type. For compatibility and coexistence with other standards, the field is reserved to have a value of 0b1111. |
| B4-B25 | ***reserved*** | 20 |  |
| B26 | Extended Header Indication (EHI) | 1 | If EHI=”0”, the PHY header contains PHYH information bits. If EHI=”1”, the PHY header contains 2×PHYH information bits. |
| B27 | Header Segmentation Indication (HSE) |  | The HSE indicates whether the header is spread over two symbols. |
| B28-B31 | ***reserved*** | 4 |  |
| B32-B47 | Message Duration (MSG\_DUR) | 16 | The MSG\_DUR indicates the transmit duration of the frame. |
| B48-B49 | Block Size (BLKSZ) | 2 | The BLKSZ determines the block size of the FEC codeword applied in the data field of the PPDU. |
| B50-B52 | FEC Rate (FEC\_RATE) |  | The FEC\_RATE indicates the FEC coding rate applied in the data field of the PPDU. |
| B53-B55 | Repetition Number (REP) |  | REP indicates the number of payload repetitions as detailed in clause **32.3.4.6.1.**. |
| B56-B58 | FEC concatenation factor (FCF) |  | The FCF is detailed in clause **32.3.4.6.1.**. |
| B59-B62 | Scrambler Initialization (SI) |  | The SI field comprises the initialization bits for the Data scrambler as detailed in **32.3.4.1.1.** . |
| B63 | ***reserved*** |  |  |

Table 32-8 – Header field of the LC PHY PPDU *(continued)*

|  |  |  |  |
| --- | --- | --- | --- |
| B64-B68 | Bit allocation table ID (BAT\_ID) |  | BAT\_ID indicates the BAT used in the data field of the PPDU. |
| B69-B71 | Bandplan and subcarrier grouping identifier (BNDPLN / GRP\_ID) |  | The Bandplan and subcarrier grouping ID. |
| B72-B74 | Guard Interval ID (GI\_ID) | 3 | THE GI\_ID indicates the CP length used for the Data field of the PPDU. |
| B75-B151 | ***reserved*** |  |  |
| B152-B167 | Header Checksum (HCS) | 16 | The HCS uses CRC-16 as defined in [Editors note: Include the correct Clause number here] The HCS bits shall be processed in the transmitted order. The registers shall be initialized to all ones. |

Depending on the value of the extended header indication (EHI) field in the core part of the PHY‑frame header, the PHY-frame header may be extended by additional PHYH bits that are transmitted over an additional *D* consecutive OFDM symbols. If the EHI bit is set to one, additional PHYH bits representing the extended part of the PHY-frame header are appended to the end of the core part of the PHY-frame header. The extended part of the PHY-frame header shall be encoded and segmented exactly the same way as the core part, as described in Clause **32.3.4.5.**. The content of the extended part is protected by the 16-bit extended header check sequence (E\_HCS).The core part and the extended part of the PHY-frame header shall be transmitted over separate OFDM symbols, as illustrated in Figure 32-9.



Figure 32-9 – Allowed cases of PHY-frame header transmissions

##### **32.3.4.4.4. MIMO reference signals**

Optional fields contain reference signals (RS) for multiple-input multiple-output (MIMO) channel estimation. For MIMO RS, forward error correction and HCS do not apply. MIMO RS are defined in the frequency domain. The use of MIMO RS is configurable by the MAC.

MIMO RSs allow orthogonal detection of multiple data streams or signals from multiple transmitters. MIMO RS are orthogonal in the frequency domain. A specific comb of subcarriers identifies a particular stream or transmitter.

Construction of MIMO RS starts from the LFSR also used in the synchronization preamble. MIMO RS for the first transmitter use the subcarriers with indices Δ, 2Δ, 3Δ, 4Δ, …, N-Δ, except those being masked, where Δ is the comb spacing and a power of 2. The value of Δ is defined in Clause [Editor note: Include the correct clause number here] taking the fundamental relation Δ*≤N/NCP* into account. On supported subcarriers, at first, a bit sequence of all ones is mapped using the 1-bit constellation. Next, the LFSR generator of the constellation scrambler is initialized as 16E616=1011011100110[[9]](#footnote-9). The LFSR is advanced by two bits per subcarrier in the order specified in Figure 2. The output of the mapper is rotated using the two LSBs of the LFSR, s1 and s2, as defined in Table 2 resulting in the constellation points *Zi* for subcarrier index *i*.

The resulting constellation sequence is finally fed into the OFDM modulator described in **32.3.4.3.**.

The variable *CS* in the PHY header identifies other transmitters at the same coordinator. For *CS≥1*, a cyclic shift of the comb is performed such that now the subcarriers with indices *CS*, Δ*+CS*, 2Δ*+CS*, 3Δ*+CS*, 4Δ*+CS*, …, N-Δ*+CS* are used*,* except those being masked. By using a single RS, up to *CSmax* ≤Δ-1 streams or transmitters can be identified. The MAC layer will add more RSs for more streams or transmitters as indicated by NRS being a power of 2. In this way, up to NRS\*(Δ-1) streams can be identified. Decompose the identifier of the *i*th stream or transmitter as *i*=*a*\*(Δ-1)+*b* where *b*<Δ-1.

The comb shift is then *CS*=*b* and the original RS is multiplied with the entries in the *a*th row of the MxM Hadamard matrix HK where M=2K. HK is obtained by incrementing k from k=1…K

.

##### **32.3.4.4.5. Payload**

The payload contains MPDUs as defined in Clause 31.

#### 32.3.4.5. Header encoding

The header encoder is shown in Figure 32-10. It contains a header FEC encoder and a header repetition encoder.

##### **32.3.4.5.1. Header FEC encoder**

The bits of the PHY header shall be input into the header FEC encoder in their original order and encoded as described in Clause **32.3.4.1.**. The size of the FEC codeword and the coding rate of the header FEC encoder are described in Table 32-3. Since the coding rate used for header encoding is 1/2, the number of bits in the FEC codeword is always even, and the number of bits in the encoded header block is even.



**Figure 32-10 Header encoder**

##### **32.3.4.5.2. Header repetition encoder**

The FEC codeword enters the HRE operating as follows. The FEC codeword is first copied *M* times, where *M* = ceiling (*kH/NFEC*), *kH* is the number of bits to be loaded on to the OFDM symbol carrying the header. The first encoded header block is formed by concatenation of *M* copies of the header FEC encoder output. The bits (*bi*) within each codeword are cyclically shifted by 2 bits as follows:

• 1st FEC codeword copy is {*b*0, *b*1, …, *bNFEC*–2, *bNFEC*–1}.

• 2nd FEC codeword copy is {*b*2, *b*3, …, *bNFEC* –1, *b*0, *b*1}.

• 3rd FEC codeword copy is {*b*4, *b*5, …, *bNFEC*–1, *b*0, *b*1, *b*2, *b*3}.

• …

• *M*th FEC codeword copy, where *M* > 3, is {*b*(2×*M–*2), *b*(2×*M*–1), …, *bNFEC*–1, *b*0, *b*1, …, *b*(2×*M*–4), *b*(2×*M*–3)}.

The second encoded header block is formed by cyclic shifting of each copy by NFEC/2 bits and concatenation of *M* copies of the shifted FEC codeword. The bits (*bi*) within each codeword is cyclically shifted by 2 bits as follows:

• 1st FEC codeword copy is {*bNFEC/2*, *bNFEC*/2+1, …, *bNFEC*–2, *bNFEC*–1, *b*0, *b*1, …, *bNFEC*/2–2, *bNFEC*/2–1}.

• 2nd FEC codeword copy is {*bNFEC*/2+2, *bNFEC*/2+3,…, *bNFEC*–2, *bNFEC*-1, *b*0, *b*1, …, *bNFEC*/2, *bNFEC*/2+1}.

• 3rd FEC codeword copy is { *bNFEC*/2+4, *bNFEC*/2+5, …, *bNFEC*–2, *bNFEC*-1, *b*0, *b*1, …, *bNFEC*/2+2, *bNFEC*/2+3}.

• …

• *M*th FEC codeword copy, where *M*>3, is {*bNFEC*/2+(2×M-2), *bNFEC*/2+(2×M–1), …, *bNFEC*–1, *b*0, *b*1, …, *bNFEC*/2+(2×M–4), *bNFEC*/2+(2×M–3)}.

##### **32.3.4.5.3. Header segmentation**

The encoded header block from the output of the header encoder is segmented into symbol frames. The maximum number of bits in the symbol frame does not exceed the values of *kH* for header symbol frames. Header symbol frames shall be passed to the adaptive bitloading by using a fixed constellation of 2 bits per subcarrier as shown in Figure 1 and described in **32.3.4.2.**.

The encoded header block is segmented into *D* symbol frames (*D*= 1, 2). The value of *D* is selected by the coordinator and indicated in the PHY header. The first *kH* bits of the first encoded header block is mapped into the first symbol frame, so that *b*0 is transmitted first. If *D*= 2, the first *kH* bits of the second encoded header block is mapped into the second symbol frame, so that *bNFEC*/2 is transmitted first. The rest of the bits of the first and the second encoded header blocks are discarded. Header segmentation is illustrated in Figure 32-11.



**Figure 32-11 Header segmentation**

##### **32.3.4.5.4. Tx selection for the header**

Tx selection for the header is the same as for the payload, see Clause **32.3.4.6.2.**

#### 32.3.4.6. Payload encoding

The payload encoder is shown in Figure 32-12. It contains an FEC encoder and a payload repetition encoder (PRE) to support robust communication.



**Figure 32-12 Payload encoder**

The incoming PHY-frame payload is divided into sequential blocks of information bits, *K* bits per block. Each block of information bits is encoded by the FEC, as described in **32.3.4.1.**. The valid values of *K*, the coded block size *NFEC*, and the coding rate *R*, are given in Table 32-3. The bits of each information block are in the same order as they are in the payload; the payload bit to be transmitted first is the first in the corresponding information block.

In normal mode, indicated by REP = 001 in the PHY header, PRE is disabled. The FEC codewords are passed directly to the output of the payload encoder and concatenated into the encoded payload block; their order are the same as the order of corresponding information blocks at the input of the payload encoder. For robust communication, each FEC codeword is further encoded by the PRE where the FEC codewords are repeated and concatenated into the encoded payload block.

For robust communication, each FEC codeword is further encoded by the PRE, as described in clause **32.3.4.6.1.**. The PRE-encoded FEC codewords are concatenated into the encoded payload block as defined in clause **32.3.4.6.1.**.

##### **32.3.4.6.1. Payload repetition encoder**

Payload repetition encoder (PRE) shall support the number of repetitions *NREP*specified in Table 32-10. The used number of repetitions shall be advertised in the REP field in the PHY-frame header.

Table 32-10 Allowed values of REP

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **REP value** (*b*7*b*6*b*5) | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| **Interpretation** (NREP) | reserved | 1 | 2 | 3 | 4 | 6 | 8 | reserved |

The PRE shall operate as follows. Each incoming FEC codeword shall be first copied *NREP* times. Each copy shall be divided into *S* sections, numbered from 0 to *S–*1, with *B* bits in each section, as follows:

– Bits of the FEC codeword shall be mapped into sections in ascending sequential order; the bit of the FEC codeword to be transmitted first shall be the first bit (*b*0) of Section 0.

– If after all bits of the FEC codeword are mapped, the last *q* bit positions of the last section remain empty, these positions shall be filled by the first *q* bits of Section 0 in ascending sequential order.

Mapping of an FEC codeword on to sections is shown in Figure 32-13.



**Figure 32-13 Mapping of a FEC codeword onto sections**

If *floor*(*kP/NREP*) is divisible by 4, the number of bits per section shall be set to *B*= *floor*(*kP/NREP*) − 1; otherwise, it shall be set to *B* = *floor*(*kP/NREP*), where *kP* is the total number of bits that can be loaded on to the payload OFDM symbol according to the current BAT. The number of sections per FEC codeword is *S=ceiling*(*NFEC*/*B*).

If the computed value of *S* is1, *H* consecutive FEC codewords may be concatenated. The number of sections in this case shall be: *S = ceil*(*H*×*NFEC/B*), where *H* is selected to provide *S* > 1 for the given values of *NFEC,* *N*REP and *kP*. Concatenation of codewords may only be applied when an FEC information block size of 960 is used. The total size of the concatenated codewords shall not exceed the maximum FEC codeword size.

PRE parameters NREP and *H* shall be selected such that q < *H* x NFEC.

If the number of FEC codewords in the payload is not a multiple of *H*, the necessary *z* < *H* dummy FEC codewords shall be added. These dummy codewords shall be copies of the last FEC codeword of the same payload. The values of *H* (1, 2 and 4) and *z* (0 to *H*−1) are indicated in the FCF field of the PHY-frame header (see Table 32-8).

The PRE shall output sections sequentially, in groups of *S* sections. Each group carries a copy of the FEC codeword. The number of groups per each FEC codeword is *NREP*. The order of bits in each section shall be the same as these bits appear in the incoming FEC codeword.

The format of the encoded payload block with PRE enabled is presented in Figure 32-14. The total number of sections in the encoded payload block is *NREP*×*S*.



**Figure 32-14 Format of the encoded payload block (payload consists of J FEC codewords)**

The order of sections in the first group shall be ascending, from 0 to *S*–1; the order of sections in all subsequent groups shall be cyclically shifted. The shift is defined by the cyclic section shift (CSS) vector {0 CSS2 CSS3 … CSS*NREP*} with a length of *NREP*, where CSS*L* is the sequential number of the section to be transmitted first in the *L*th group of sections. The value of CSS shall be computed using the following rule:

*NREP*=2: if (*S* mod 2)=0 CSS:={0,1} else CSS:={0,0}

*NREP*=3: if (*S* mod 3) = 0 CSS:= {0,1,2} else CSS:= {0,0,0}

*N*REP=4: if (*S* mod 4)=0 CSS:= {0,1,2,3} else if (*S* mod 2)=0 CSS:={0,0,1,1} else CSS:={0,0,0,0}

*NREP*=6: if (*S* mod 6)= 0 CSS:={0,1,2,3,4,5} else if (*S* mod 3)=0 CSS:={0,0,1,1,2,2} else if (*S*mod 2)=0 CSS:={0,0,0,1,1,1} else CSS:={0,0,0,0,0,0}

*NREP*=8: if (*S* mod 8)=0 CSS:={0,1,2,3,4,5,6,7}else if (*S* mod 4)=0 CSS:= {0,0,1,1,2,2,3,3} else if (*S*mod 2)=0 CSS:={0,0,0,0,1,1,1,1}else CSS:= {0,0,0,0,0,0,0,0}

As an example, with CSS = 3L for a group of *S*=4 sections, sections will be transmitted in the following order: 3, 0, 1, 2. The first group of sections, for comparison, is transmitted: 0, 1, 2, 3.

##### **32.3.4.6.2. Tx selection for the payload**

Transmitter selection can be described by a matrix-vector operation ***T****·****x*** operating subcarrier-wise. It shall be implemented when using distributed MIMO with the RS according to Clause **32.3.4.4.4.**.

Tx selection assigns the best OFEs which were suggested by the receiving STA for joint transmission where a cluster of OFEs transits the same data. This enables resilient communication through spatial diversity. Moreover, Tx selection allows also simultaneous transmissions, i.e. spatial multiplexing of multiple data streams to multiple STAs if interference is considered negligible.

The distributed MIMO transmitter multiplies the nTxx1 vector of payload information symbols ***x***with the *NRS**x* *nTx* precoding matrix ***T*** which contains ones for all selected transmitters and zeros elsewhere.

1. Significant parts of the LCO PHY text are taken over from ITU-T recommendation G. 9960 Cor. 1 09/19 in the coax baseband (CB) mode. Corresponding simplifications have been taken into account in this specification. [↑](#footnote-ref-1)
2. For OCR≤200 MHz, the lowest carrier indexes (0, 10) are all unused. [↑](#footnote-ref-2)
3. The method for generating SI values is out of Scope of this Standard. [↑](#footnote-ref-3)
4. One method of encoding is to determine a systematic generator matrix **G** from **H** such that **GHT**= **0**. A *K*-bit information block **u**=[*u*0, *u*1,...,*u*K-1] can be encoded by the systematic generator matrix **G** via the operation **v**= **uG** to become a *NM*-bit coded block **v** = [**u** | **p**] = [*u*0, *u*1,...,*u* NM -1]**,** where **p**=[*p*0, *p*1,...,*p*NM-K-1]are the parity-check bits. Encoding an LDPC code from **G** can be quite complex. However, the QC-LDPC-BC codes specified here are such that very low complexity encoding directly from **H** is possible. [↑](#footnote-ref-4)
5. The pattern does not result in any code rate changes and is introduced to be consistent with the puncturing notation. [↑](#footnote-ref-5)
6. Note that SSCs may be dynamically switched off depending on the channel characteristics, due to attenuation and noise. [↑](#footnote-ref-6)
7. Predefined BAT may be used when channel characteristics are unknown (i.e., no knowledge is available on whether particular subcarriers could be loaded with bits or not). [↑](#footnote-ref-7)
8. Initial zeros are dropped when converting hex to binary numbers. [↑](#footnote-ref-8)
9. Initial zeros are dropped when converting hex to binary numbers. [↑](#footnote-ref-9)