**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | |
| Title | **Removed parts for D6.0** | |
| Date Submitted | 20 December 2021 | |
| Source | Kai Lennert Bober | Voice: [ ] Fax: [ ] E-mail: [ ] |
| Re: |  | |
| Abstract | Text parts that were removed by the CRG decision | |
| Purpose |  | |
| Notice | This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. | |
| Release | The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. | |

***Remove acronym “LB-PHY low bandwidth PHY”***

***Remove acronym “CQI channel quality indicator”***

***Remove acronym “eU-OFDM enhanced unipolar OFDM”***

***Remove definition “poll frame: A frame that is sent by the coordinators to poll devices on downlink.”***

***Remove “or polled” from the list in 5.1***

***Remove the following subclause***

5.7.3 LB-PHY introduction

The LB-PHY, specified in 11, is intended for low date rate applications with data rates in the tens of Mb/s using bit-interleaved coded modulation based on OFDM. It supports efficient utilization of the low-bandwidth resources (up to 32 MHz of single-sided bandwidth) of high-power LEDs as well as low-complexity, high energy efficiency and enhanced reliability.

A DC-biased OFDM is the default waveform. Furthermore, enhanced unipolar OFDM (eU-OFDM) is supported. For modulation of the LED, multiple clock rates are used. The LB-PHY supports MIMO and relaying.

The LB-PHY requires use of the non-beacon-enabled channel access, defined in 6.4.

***Remove the following subclause:***

6.4 Non-beacon-enabled channel access

**Overview**

The non-beacon-enabled channel access only works with the LB-PHY, which supports full duplex transmissions. The MAC for non-beacon-enabled OWPAN starts with the transmission of a *Random Access* frame from the coordinator, which carries the information required by new devices to initiate an association procedure.

Following the control frame, the contention access polling period starts. During the contention access polling period, devices compete on the uplink channel sending *Poll Request* frames to the coordinator to enable the coordinator to poll the device in the following contention-free polling period. In the contention-free polling period, the coordinator may poll one or multiple devices within one frame transmission. Devices are polled by the coordinator and, as soon as they recognize the beginning of the polling frame (polling information is contained in the frame header at the beginning of the frame), they are allowed to transmit on uplink channels. The flow chart in Figure 13 illustrates the logic for the non-beacon-enabled OWPAN MAC.

fig-clause-6-Flow chart of the non-beacon-enabled OWPAN MAC

**Flow chart of the non-beacon-enabled OWPAN MAC**

The polling mechanism enables very low-complexity synchronization of the transmission slots and enables the coordinator to dynamically adapt and assign the slots for transmissions. The specific polling algorithm and order is outside the scope of this specification and is left as a system designer's choice. The concept is presented in the example in Figure 14. Polled devices shall send an acknowledgment for the *Random Access* frame to the coordinator in the *High-reliability Control* header of the LB-PHY PPDU or in the *Poll ACK* field of the MPDU.

fig-clause-6-Non-beacon-enabled channel access

**Non-beacon-enabled channel access**

* + 1. **Contention-based polling period**

Figure 15 presents a flow chart for the contention-based random access period process. The coordinator sends a *Random Access* frame periodically at the beginning of a polling cycle. The first valid downlink frame (i.e., DATA, MAC Management or MAC Control frames) after the *Random Access* frame marks the beginning of the contention period, i.e., the first valid frame after the control frame is interpreted as a poll to all stations that are not connected to transmit their *Poll Request* frame. All stations that have already been associated with the coordinator may ignore any polling during the contention period. If no stations have been connected to the coordinator, the first valid frame would be the subsequent control frame. After receiving the control frame, the device may send a *Poll Request* frame. The coordinator starts polling a station in the next polling round after receiving the *Poll Request* frame.

The *Poll Request* frame is a MAC frame, which is sent by the devices to establish the connection with the coordinator via the uplink. The *Poll Request* frame does not have any field except the common management information. All devices operating in non-beacon enable mode shall be capable of transmitting this frame, although a device is not required to be capable of receiving it. All coordinators operating in non-beacon enable mode shall be capable of receiving this frame.

fig-clause-6-Flow chart for contention-based random access period

**Flow chart for contention-based random access period**

In case several devices want to connect at the same time, a collision may occur. If it is not polled during the next polling round, a device assumes a collision has occurred and proceeds accordingly with a backoff and an attempt to reconnect. Then, the random backoff is introduced. For example, one device retries to establish a connection after two *Random Access* frames, another device retries after five *Random Access* frames. A device transmits a *Poll Request* frame immediately after it decodes a valid MAC frame header. Hence, as depicted in Figure 14, it will not wait for the entire frame on the downlink from the coordinator to be transmitted. Upon a successful polling request (the coordinator has successfully received the station's request to be polled and has indicated this by polling the station within the next transmission round), a station exchanges the necessary association and authentication information with the coordinator via association and authentication control frames. Upon successful authentication and association, the device is assigned a AID, which will identify the device among all other devices connected to the same coordinator.

* + 1. **Contention-free polling period**

During the contention-free polling period, each device is required to be able to respond to a Poll frame received from a coordinator. A device shall also be able to request to be polled by an active coordinator.

When polled by the coordinator, a device may transmit only one MPDU. The ACKs for any data frame may "piggyback" on the transmission of any management or data frame (including Null frames). If a frame is not acknowledged, retransmission of the frame will be rescheduled by the coordinator. If a polled device does not have any data or ACK to transmit, it simply ignores the polling request. The coordinator will stop polling a device provided it does not receive any response from the device for more than *aMacPollTimeout*. The concept is given in the flow chart in Figure 16.

fig-clause-6-Flow chart for contention-free polling period

**Flow chart for contention-free polling period**

The *Poll frame* is a MAC frame, which is sent by the coordinators to poll devices on downlink. The *Poll frame* does not have any field except the common management information. All coordinators operating in non-beacon enable mode shall be capable of transmitting this frame, although a coordinator is not required to be capable of receiving it. All devices operating in non-beacon enable mode shall be capable of receiving this frame.

Any frames transmitted from the coordinator to the device are treated as polling frames, except for the *Random Access* frame. This can be assumed because every frame transmission contains a polled device number, which enables the stations to keep track of the polling order.

Furthermore, the coordinator will not start another polling request before the current polled device finishes its transmission on uplink. Hence, no collisions in the uplink are possible and each downlink transmission can function as a polling frame.

Once a device receives a *Poll frame* sent to itself from the coordinator, it replies with a *Poll Request* frame to the coordinator to announce its availability and append data frames if any are queued. The *Poll Response* frame is a MAC frame, which is sent by the devices to respond to the *Poll frame* from coordinators. The *Poll Response* frame does not have any field except the common management information. All devices operating in non-beacon enable mode shall be capable of transmitting this frame, although a device is not required to be capable of receiving it. All coordinators operating in non-beacon enable mode shall be capable of receiving this frame.

In case the response to a *Poll Request* frame is not detected by the coordinator, the coordinator will attempt to poll the next device in the queue. Upon detection of a subsequent downlink frame by the device whose response was not detected, the device will stop transmission immediately in order to avoid any collisions in the uplink as illustrated in Figure 17. This functionality is necessary because the coordinator does not necessarily expect to receive a response to its poll.

If a device has no ACK or information to transmit, it will simply ignore its possibility to transmit. However, a device shall send a response even though it does not have an ACK or information to send after a while. The purpose of this is to keep the connection alive so that the coordinator shall not consider the link is invalid. If there is no information to be transmitted to a device, which is next in the queue, a Data Null frame is used as a polling frame.

fig-clause-6-Consequences of the coordinator failing to recognize a poll response

**Consequences of the coordinator failing to recognize a poll response**

* + 1. **Acknowledgment and retransmission**

The reception of every frame at the MAC sublayer level has to be acknowledged by the receiving side (coordinator or device). The flow chart in Figure 18 shows the logic of acknowledgment and retransmission processes.

fig-clause-6-Flow chart for acknowledgement and retransmission processes

**Flow chart for acknowledgment and retransmission processes**

If the ACK bit in the *Frame Control* field is set, then following ACK and retransmission procedure is applied. If a frame is not acknowledged by the receiver (either the device or the coordinator) at the next polling round (i.e., when the receiver is polled next time), it is deemed that a retransmission is needed. Frames may arrive out of order. Hence, even if a device or a coordinator does not receive the expected an ACK during a given polling round, it may proceed with the transmission of the next frame. If frames are not acknowledged after *macMaxFrameRetries* retransmissions, they are considered lost and are dropped. Figure 19 illustrates the acknowledgment procedure. If the ACK bit is not set, the packet is transmitted only one time and the absence of an ACK is ignored.

fig-clause-6-Downlink transmission example without loss

**Downlink transmission example without loss**

Figure 20 shows an example for acknowledgment and retransmission. In this example, two devices are connected to the coordinator. The coordinator has a data frame (d1\_1) to transmit to the device 1, hence it polls the device and sends the data. Device 1 responds to the poll with a payload frame or a management frame as soon as it receives the pseudo noise sequence of the received frame.

fig-clause-5-nbe-downlink-transmission-loss-retransmission-figure-22b

**Downlink transmission example with loss and retransmission**

After downlink and uplink transmissions are complete, the coordinator sends a data frame to Device 2. Device 2 replies to the poll with a payload frame as soon as the pseudo noise sequence is received, since the information of the current polled device was available from the high-reliability control header of the previous Poll. After completion of the transmission in both directions, the coordinator attempts to send a second data frame (d1\_2) to device 1. Device 1 is polled again; hence it takes the opportunity to send the ACK for the first data frame d1\_1 to the coordinator. Device 2 acknowledges his reception once polled in another round. The Figure 19 shows the above process when no frame loss exists.

Figure 20 illustrates a more complex example when three types of errors are considered.

1. Payload CRC error:

The error does not affect the decoding of the high-reliability control header.

The device, which is supposed to be polled responds to the coordinator.

The device cannot acknowledge the coordinator for this frame due to the failure of decoding the DATA frame.

The lack of ACK results in a retransmission from the coordinator.

Header error:

The high-reliability control header cannot be decoded.

The device can still respond to the coordinator although the header is not decodable since each high-reliability header contains the information for the current polled device and the next device to poll.

Leads to no response for the next *Poll* frame as well as no reception of the frame d2\_3.

The PPDU is not detected successfully:

It has the lowest probability of occurrence.

It also affects the next *Poll* frame and receives no *Poll Response* frame from the current polled device.

ACK information is available in the *Poll ACK* field of the MAC header (7.2.3) and the *High-reliability Control Header* field in the payload of the LB-PHY PPDU (11.2.6.4). The difference between the two lies in the functions and purposes of the ACKs. The *Poll ACK* field is to add an extra ACK information to another previously received frame, while the ACK in the *High-reliability Control Header* enhances the reliability of ACK even if the whole PPDU payload cannot be received and decoded correctly. The ACK information in the *Poll ACK* field and *High-reliability Control Header* of the LB-PHY may acknowledge different frames.

***Change 5.6.2.2 as follows:***

The transfer from a device to the coordinator happens in a coordinated way through the channel access mechanism described in 6.3. Each device listens for the beacon before starting data transmission to a coordinator. When the beacon is found, the device synchronizes to the superframe structure. At the appropriate time, the device transmits its data frame to the coordinator without contention. The coordinator may acknowledge the successful reception of the data by transmitting an ACK frame.

***Change 5.6.2.3 as follows:***

The transfer from the coordinator to a device is controlled by the coordinator. Devices only receive while they are not transmitting, i.e., outside their granted time slice (GTS) in beacon-enabled channel access mode.

The device may acknowledge the successful reception of the data by transmitting an ACK to the coordinator.

***Remove the following sentence from 5.6.2.4:***

Relaying functionality is supported only in the beacon enabled channel access mode.

***Replace the text in 5.6.5 with the following text:***

IEEE Std 802.15.13 OWPANs make use of a scheduled medium access. The beacon-enabled medium access uses a Beacon frame to synchronize devices, a slotted random channel access mechanism in the contention access period (CAP) and a deterministic channel access mechanism in the contention-free period (CFP).

NOTE—The CAP is only used for certain purposes such as association and requesting resources in the CFP.

***Remove the following text in 5.7.1:***

b) Low bandwidth PHY (LB-PHY) introduced in section 5.7.3.

***Remove the following text from 6.2.1:***

An OWPAN may operate in beacon-enabled or non-beaconed enabled mode. Depending on which mode is used by the coordinator, channel access is performed in different ways. However, the remaining transmit and receive process are the same, regardless of the applied channel access mechanism.

***Remove the following text from 6.2.1:***

Compressed addresses consist of a number of the lowest bits of the AID. Coordinators maintaining an OWPAN that supports use of routines relying on compressed addresses shall limit the number of associated devices to the maximum number that is representable with the given number of bits in the compressed address.

***Remove “and 6.4” in 6.2.3***

***Remove “or Random Access” in two locations in 6.4.2***

***Remove “or Random Access” in 6.4.3***

***Remove “Random Access frames” from 6.8.1***

***Remove “or Random Access” from 6.4.5***

***Remove “or Random Access frames” from table 21.***

***Remove “Random Access” control frame from table 4:***

Random Access 0b1001 Random Access element 7.6.22

***Remove “or Random Access” from table 34***

***Remove the Poll ACK field in figure 23 and its description:***

* + 1. **Poll ACK field**

The *Poll ACK* field, depicted in Figure 32, contains acknowledgment information specific to the non-beacon-enabled mode. It is only present in frames originating from devices operating in the non-beacon-enabled channel access mode. This is indicated through the *Non-beacon-enabled* bit in the *Frame Control* field, defined in 7.2.2.

|  |  |  |
| --- | --- | --- |
| **Bit 0-6** | **Bit 7-13** | **Bit 14-15** |
| 1. Device 2. Compressed 3. Address | 1. Compressed 2. Sequence 3. Number | 1. ACK |

1. **Poll ACK field**

**Device Compressed Address:** Bits 0 to 6 contain the compressed address of the device, which transmitted the frame. The seven bits of the compressed address number correspond to the lowest seven bits of the AID. The device with such address is to be acknowledged by this *Poll ACK* field.

**Compressed Sequence Number:** Bits 7 to 13 identify the sequence number of the packet, which is being acknowledged. The seven bits of the compressed sequence number correspond to the lowest seven bits of the full sequence number.

**ACK:** Bit 14 is set to one when the frame with the given sequence number is acknowledged, and set to zero otherwise. Bit 15 is set to one when the last *Random Access* frame reception is being acknowledged, and set to zero otherwise.

***Remove the “Non-beacon-enabled” field in figure 24 and remove its description:***

Non-beacon-enabled: Specifies whether the transmitting device operates in the non-beacon-enabled mode and hence whether the Poll ACK field is present in the remaining MAC header. If the transmitting device operates in non-beacon-enabled mode, this field shall be set to one. Otherwise, it shall be set to zero.

***Remove the following rows from table 3:***

|  |  |  |  |
| --- | --- | --- | --- |
| Poll | 0b0100 | none | - |
| Poll Request | 0b0101 | none | - |
| Poll Response | 0b0110 | none | - |

***Remove the “Same MAC Mode” field from Figure 46 and its description:***

Same MAC Mode: This bit shall be set to one if the received frame originates from an IEEE Std 802.15.13 OWPAN that uses the same channel access mode as defined in 6.3 and 6.4 respectively.

***Remove entry for LB-PHY from table 11:***

1 LB-PHY

(defined in Clause 11)

LB-PHY MCS element

(defined in 7.6.20)

***Remove 7.6.20:***

1.1.1 LB-PHY MCS element

The LB-PHY MCS element, depicted in Figure 57, holds a subset of supported MCS for the LB-PHY.

1 Octet 1 Octet

Bandwidths MCS

Figure 1 LB-PHY MCS element

Bandwidths: A bitmap indicating the set of supported bandwidths. A one in the bitmap indicates that the given bandwidth is supported. A zero indicates that the bandwidth is not supported. Figure 58 shows the bitmap structure.

Octet 1, LSB left

Bit in the bitmap: 0 1 2 3 4 5 6 7

Bandwidth 1 MHz 2 MHz 4 MHz 8 MHz 16 MHz 20 MHz 25 MHz 32 MHz

Figure 2 Bandwidth bitmap

MCS: A bitmap indicating a set of supported MCSs. A one in the bitmap indicates that the given MCS is supported. A zero indicates that the MCS is not supported. Figure 59 shows the bitmap structure.

Octet 1, LSB left

Bit in the bitmap: 0 1 2 3 4 5 6 7

MCS 0 1 2 3 4 5 6 7

MCS bitmap

***Remove 7.6.21:***

1.1.1 Random Access element

The Random Access element in depicted in Figure 62.

8 Octets 2 Octets 2 Octets 6 Octets 2 Octets

Timestamp Random

Access

Interval Capability

Information OWPAN

ID LB-PHY

MCS

element

Figure 1 Random Access element

Timestamp: The Timestamp field allows synchronization between the devices in an OWPAN. Each device shall maintain a timing synchronization function timer with modulus 264 counting in increments of microseconds. When coordinators prepare to transmit a Random Access frame, the coordinator timer is copied into the frame’s Timestamp field. Devices associated with a coordinator accept the timing value in any received Random Access frames, but they may add a small offset to the received timing value to account for local processing by the transceiver.

Random Access Interval: Each OWPAN may transmit Random Access frames at its own specific interval. The interval provides the duration between two adjacent random access periods, which lasts from the beginning of the current Random Access control frame to the beginning of the next Random Access control frame.

Capability Information: The 16-bit Capability Information field is used to advertise the network’s capabilities. In this field, each bit is used as a flag to advertise a particular function of the network.

OWPAN ID: OWPAN ID field gives the ID for the OWPAN.

LB-PHY MCS element: The LB-PHY MCS element, defined in 7.6.20, containing the supported clock rates and MCS.

***Remove PIB attribute from table 39:***

|  |  |  |  |
| --- | --- | --- | --- |
| *aMacPollTimeout* | The time after which a coordinator stops polling a device if it receives no response from that device. | 5 | seconds |

***Remove LB-PHY from table 41***

***Remove Clause 11:***

1. **LB-PHY specifications**
   1. **General information**

**Overview**

The LB-PHY is intended for low date rate applications with data rates in the tens of Mb/s using OFDM modulation. OFDM specified by LB-PHY enables a highly adaptive modular implementation, which supports efficient utilization of the low-bandwidth resources (up to 32 MHz of single-sided bandwidth) as well as a low-complexity PHY designed to enable high energy efficiency and enhanced transmission reliability.

A DC-biased OFDM is the default waveform. Furthermore, the eU-OFDM waveform is supported. For modulation of the LED, multiple clock rates are used. The LB-PHY supports MIMO and relaying. Table 46 provides an overview over the LB-PHY parameters.

**Summary of the LB-PHY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Modulation** | | | DC-biased OFDM, eU-OFDM | | |
| **FEC** | | | Convolutional coding | | |
| **Code rates** | | | 1/2, 2/3, 3/4 | | |
| **Subcarrier spacing *FSC*** | | | Clock rate / 32 | | |
| **Cyclic prefix** | | | 16 samples | | |
| **MIMO** | | | Up to 16 by 16 | | |
| **Modulation** | **FEC** | **Clock rate** | | **Data rate** | |
| **Min.** | **Max.** |
| BPSK | Inner convolutional code (1/2) | 1-32 MHz | | 0.3 Mb/s | 9.6 Mb/s |
| BPSK | Inner convolutional code (3/4) | 0.45 Mb/s | 14.4 Mb/s |
| QPSK | Inner convolutional code (1/2) | 0.6 Mb/s | 19.2 Mb/s |
| QPSK | Inner convolutional code (3/4) | 0.9 Mb/s | 28.8 Mb/s |
| 16-QAM | Inner convolutional code (1/2) | 1.2 Mb/s | 38.4 Mb/s |
| 16-QAM | Inner convolutional code (3/4) | 1.8 Mb/s | 57.6 Mb/s |
| 64-QAM | Inner convolutional code (2/3) | 2.4 Mb/s | 76.8 Mb/s |
| 64-QAM | Inner convolutional code (3/4) | 2.7 Mb/s | 86.4 Mb/s |

Clock rates between 1 MHz and 32 MHz are defined, as listed in Table 48. The diagram of the DC-biased OFDM system is illustrated in Figure 75.

fig71

**Diagram of the DC-biased OFDM system**

**Base MCS**

The base MCS for the LB-PHY shall be binary phase shift keying (BPSK) with code rate 1/2, i.e., MCS ID 000 as defined in Table 49.

**PHY constants**

Table 47 lists values for the LB-PHY constants.

**LB-PHY constants**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Value** | **Unit** |
| *aPhyMaxPsduSize* | The maximum supported PSDU size. This attribute is PHY-specific. | 2036 | octets |
| *aPhyMifsDuration* | The duration of the MIFS for transmissions using the HB-PHY. | 3 | µs |
| *aPhyMinPsduSize* | The minimum supported PSDU size. This attribute is PHY-specific. | 20 | octets |
| *aPhyTurnaroundTime* | Not applicable. No turnaround is required, as the LB-PHY always performs full duplex transmission. | N/A | N/A |
| *aPhyClockAccuracy* | The minimum accuracy of the PHY reference clock. | +/- 20 | ppm |

**PHY PIB attributes**

Table 48 lists the PHY PIB attributes for the LB-PHY.

**PHY PIB attributes for the LB-PHY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Description** | **get/set** | **Range** | **Unit** |
| *phyClockRate* | The used clock rate | get / set | 1, 2, 4, 8, 16, 20, 25, 32 | Enumeration of MHz values |

* 1. **PPDU format**

**Overview**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Preamble** | **Channel Estimation** | **PHY header** | **Optional Fields** | **Payload** |

**PPDU format for LB-PHY**

The LB-PHY uses the PPDU format shown in Figure 76.

* + 1. **Bit order**

The PSDU consists of an ordered sequence of octets. The octets of the PSDU shall be transmitted in the order they were received through the PD-SAP. Within each octet of the PSDU, the LSB of each octet shall be transmitted first.

Header fields that contain numbers shall be transmitted starting with the LSB first to the MSB last.

An exception to this rule is the high reliability control header in 11.2.6.4. This header is treated as a sequence of octets with the LSB of each octet transmitted first.

**Preamble**

The PHY Preamble field is used for PPDU detection and synchronization. Preamble enables the identification of the existence of a transmission, as well as automatic gain control. It consists of pseudo noise training sequence, which lasts for the duration equivalent of two OFDM symbols. The sequence in preamble field is a time domain sequence and does not have any channel coding or line coding.

The following demonstrates the generation of pseudo noise training sequences.

Step 1: Select two 20-bit pseudo noise sequences:

Other pseudo noise sequences may be used as an option.

*pn\_seq*0 = 20'*b*01001011000001110111

*pn\_seq*1 = 20'*b*01101100111101010000

Step 2: Up-sample the above sequences by 8

Step 3: Pulse shape with the following pulse {-479, 416, -10, -409, 67, -409, -10, 416}

Step 4: Flip the sequence at the end of Step 3, e.g., (*x1*, ..., *xn*, *xn*, …, *x1*), in order to get two sequences for each original pseudo noise sequences (i.e., *pn\_seq0* and *pn\_seq1*).

**Channel estimation**

*Channel Estimation* field consists of two repetitions of a “Hermitian symmetric long training sequence” preceded by a cyclic prefix. The sequence in *Channel Estimation* field is a time domain sequence and does not have any channel coding or line coding.

A sequence of two identical OFDM training symbols is used to estimate the channel impulse response, as well as for additional fine-timing synchronization. The channel estimation sequence contains the following values modulated on the subcarriers of two identical inverse fast Fourier transforms (IFFTs) (index 0 corresponds to the DC subcarrier modulation value):



**Timing parameters for the PHY control information (specification for 20 MHz)**

The sequence is Hermitian symmetric in order to obtain a real time-domain signal after the inverse discrete Fourier transformation (IDFT). The sequence also has very good auto-correlation properties. In Figure 77, the channel estimation OFDM symbol is transmitted twice in two identical copies of the time-equivalent signal of the frequency-domain modulation sequence. The cyclic prefix is a cyclic extension of this same time-domain signal and has a duration of 32 samples (twice the length of the typical cyclic extension for this PHY mode specification).

**PHY header**

The PHY header contains all information necessary for demodulating the subsequent frame payload. It is encoded in 1/2 FEC rate (no puncturing is used BPSK modulation using DC-biased OFDM.

**Basic LB-PHY header**

Basic PHY header is compulsory for all frames. The Advanced Modulation Header may be added after the basic PHY header, which is indicated by the basic PHY header.

The basic header contains the minimum information required for demodulating the subsequent payload. In LB-PHY, the header includes information such as the constellation size, the FEC rate and the payload size. The basic header contains 24 bits and fits within two OFDM symbols of the current PHY mode. The basic PHY header defines the fields given in Figure 78.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bit 0-2** | **Bit 3** | **Bit 4-14** | **Bit 15** | **Bit 16** | **Bit 17** | **Bit 18-23** |
| MCS  ID | Reserved | PSDU  Length | Advanced  Modulation  Header | High  Reliability  Control | Parity | Tail |

**Fields in the basic LB-PHY header**

The individual fields of the basic header are described as follows.

**MCS ID:** This field consists of three bits and indicates the QAM constellation size and the FEC rate (achieved with the use of a convolutional encoder and puncturing) used for the subsequent payload. The values specified in Table 49 are valid for the *MCS ID* field. Data rates for different MCSs depend the clock rate. Table 49 lists example common selections of clock rates. The range of the available clock rates is [1, 32] MHz. The clock rate shall be obtained by dividing the available frequency by eight.

**Valid MCS ID values**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **MCS ID b0-b2** | **Modulation** | **FEC rate** | **Data rate at 1 MHz clock rate** | **Data rate at 16 MHz clock rate** | **Data rate at 20 MHz clock rate** | **Data rate at 32 MHz clock rate** |
| 000 | BPSK | Inner convolutional code (1/2) | 0.3 Mb/s | 4.8 Mb/s | 6 Mb/s | 9.6 Mb/s |
| 001 | Inner convolutional code (3/4) | 0.45 Mb/s | 7.2 Mb/s | 9 Mb/s | 14.4 Mb/s |
| 010 | QPSK | Inner convolutional code (1/2) | 0.6 Mb/s | 9.6 Mb/s | 12 Mb/s | 19.2 Mb/s |
| 011 | Inner convolutional code (3/4) | 0.9 Mb/s | 14.4 Mb/s | 18 Mb/s | 28.8 Mb/s |
| 100 | 16-QAM | Inner convolutional code (1/2) | 1.2 Mb/s | 19.2 Mb/s | 24 Mb/s | 38.4 Mb/s |
| 101 | Inner convolutional code (3/4) | 1.8 Mb/s | 28.8 Mb/s | 36 Mb/s | 57.6 Mb/s |
| 110 | 64-QAM | Inner convolutional code (2/3) | 2.4 Mb/s | 38.4 Mb/s | 48 Mb/s | 76.8 Mb/s |
| 111 | Inner convolutional code (3/4) | 2.7 Mb/s | 43.2 Mb/s | 54 Mb/s | 86.4 Mb/s |

**Reserved:** Thisbit is reserved for introducing additional transmission rates in future modifications of the standard.

**PSDU Length:** This value scales from zero up to *aPhyMaxPsduSize*. The 11-bit field indicates the size of the payload in octets.

The LB-PHY supports only 32 bit aligned data in order to simplify the implementation. Bit 0 and 1 of this field have to be set to zero. If the high reliability control header, as defined in Figure 82, is included, the 32 bit aligned length of the PSDU plus eight is written to this field. If the high reliability control header is not included, only the 32 bit aligned length of the PSDU is written to this field. The length of the FCS, which is four octets, is included in the value of this field.

If the PSDU is an MPDU like in Figure 30, the length of this MPDU is written to this field. The value is incremented by eight depending on if the high reliability control header is present or not. The last four octets of the MPDU have to be cleared before writing them to the PHY in order to allow correct FCS calculation.

**Advanced Modulation Header:** This bit indicates whether an Advanced Modulation Header follows the basic PHY header. One indicates: Advanced Modulation Header follows the basic PHY header. Zero indicates: Advanced Modulation Header will not appear after the basic PHY header.

**High Reliability Header**: This bit indicates whether a high-reliability control header is added at the beginning of the payload. One indicates: The payload starts with the high-reliability control header. Zero indicates: The payload does not start with the high-reliability control header.

**Parity** bit does an even parity check for the information in bits 0 - 16.

**Tail** consists of six bits, which are set to zero to complete the basic PHY header.

**The Advanced Modulation Header**

The Advanced Modulation Header is encoded in separate OFDM symbols from the basic PHY header. The advanced modulation header is also encoded using 1/2 FEC rate BPSK. The advanced modulation header is an optional field, which contains the information necessary for demodulating the subsequent waveform.

The advanced modulation header defines the fields given in Figure 79.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit 0** | **Bit 1** | **Bit 2** | **Bit 3** | **Bit 4-9** | **Bit 10** | **Bit 11-12** | **Bit 13** | **Bit 14** | **Bit 15** | **Bit 16** | **Bit 17** | **Bit 18-23** |
| reserved | CQI | eU-OFDM | STR | reserved | Relaying  Enabled | Relaying  Mode | MIMO  Enabled | MIMO  Pilot  Symbol | MIMO  Mode | reserved | Parity | Tail |

**Fields in the advanced modulation header**

The individual fields of the basic header are described as follows.

**CQI** bit indicates whether the CQIs should be calculated in the PHY for the current transmission frame.

0b1 indicates that the CQIs should be estimated.

0b0 indicates that no CQIs should be estimated.

The channel estimation symbols preceding the PHY header are used for the estimation of the CQIs if the MIMO mode is not enabled. If MIMO mode is enabled, the MIMO pilot symbols used for CQI estimation is further defined in *MIMO Pilot Symbol* field in the advanced modulation header.

Upon estimation of the CQIs, the PHY conveys the results to the MAC using a predefined PHY service primitive. The calculation of the bit and power allocation scheme as well as the necessary exchange for updating the bit and power allocation scheme at the transmitter and receiver are handled at the MAC sublayer.

**eU-OFDM** indicates whether the payload field is encoded using eU-OFDM.

0b1 indicates that the payload is encoded using eU-OFDM.

0b0 indicates that the payload is not encoded using eU-OFDM.

The use of this waveform may be negotiated in advance using control/management frames.

**STR** bit indicates the number of eU-OFDM streams superimposed in the signal encoding procedure.

0b0 indicates: the number of eU-OFDM streams superimposed in the signal encoding procedure is one.

0b1 indicates: the number of eU-OFDM streams superimposed in the signal encoding procedure is four.

**Reserved** bits are reserved for future use.

**Relaying mode** specifies the type of relaying mode that should be performed.

0b00 indicates that the relaying and duplexing mode is FD-AF.

0b01 indicates that the relaying and duplexing mode is FD-DF.

0b10 indicates that the relaying and duplexing mode is HD-AF.

0b11 indicates that the relaying and duplexing mode is HD-DF.

**MIMO Enabled** bit indicates whether a MIMO mode is enabled for the current PHY frame.

0b1 indicates that the MIMO mode is enabled and the subsequent payload is encoded using the MIMO scheme already negotiated between the transmitter and the receiver.

0b0 indicates that the MIMO mode is not enabled and the subsequent payload is encoded using the SISO scheme specified by the parameters in the basic PHY header and the advanced modulation header.

**MIMO Pilot Symbols** Format bit indicates the format of the pilot symbols used for CQI estimation.

0b0 indicates that the MIMO pilot symbols format I is used.

0b1 indicates that the MIMO pilot symbols format II is used.

The two MIMO pilot symbols formats are specified in 11.2.6.

**MIMO Mode** bit indicates which MIMO mode will be used.

0b1indicates that the MIMO mode is enabled and the subsequent payload is spatially multiplexed.

0b0 indicates that the MIMO mode is enabled for repetition coding and the subsequent payload is encoded using the SISO scheme specified by the parameters in the basic PHY header and the advanced modulation header.

**Reserved** bit is reserved for future use.

**Parity** bit does an even parity check for the information in bits 0 - 16.

**Tail** consists of six bits, which are set to zero to complete the advanced modulation header.

**Optional fields**

**Overview**

Optional fields contain pilot symbols for MIMO channel estimation. For MIMO pilot symbols, repetitions, FEC and line coding do not apply. The optional fields include *N*PS MIMO pilot symbols.

MIMO pilot symbols constitute *N*MIMO OFDM symbols (or the time-frame equivalent of *N*MIMO OFDM symbols), where *N*MIMO is the number of MIMO channels. The MIMO pilot symbol formats are described as follows.

**MIMO Pilot Symbols Format I**

For each OFE, only one OFDM symbol interval is set to the desired channel estimation sequence. All other intervals are set to zero. Thus, the channel estimation sequence transmission intervals never coincide with any other transmitters. Hence, the MIMO pilot symbols for the different transmitters are orthogonal to each other. The format is presented in Figure 80.



**MIMO pilot symbols format I**

**MIMO Pilot Symbols Format II**

For each OFE, every frame interval is set to the desired channel estimation sequence. In addition, the channel estimation sequences for each transmitter are modified by adjusting the polarity of the individual channel estimation sequence sequences according to a pre-determined set of Walsh sequences, where a value of one in the Walsh sequence corresponds to an unmodified channel estimation sequence while a value of minus one corresponds to a channel estimation sequence with reverse polarity. The format is presented in Figure 81. The channel estimation sequences in white are left unmodified, while the channel estimation sequences in gray are multiplied by -1.



**MIMO pilot symbols format II**

The Walsh sequences for a MIMO configuration with two transmitters (*N*MIMO = 2) correspond to the rows of the matrix *W*2 MIMO:

The Walsh sequences for a MIMO configuration with four transmitters (*N*MIMO= 4) correspond to the rows of the matrix *W*4 MIMO:

The Walsh sequences for a MIMO configuration with eight transmitters (*N*MIMO = 8) correspond to the rows of the matrix W8 MIMO:

As a general rule, the Walsh sequences for a MIMO configuration with transmitters () correspond to the rows of the matrix :

**High-reliability Control header**

Robust transmission of the polling and acknowledgment information helps avoiding unnecessary retransmissions. Furthermore, when this information is encoded separately from the rest of the payload, errors in the payload (especially for long payloads) which cause the PPDU to be discarded (and retransmitted) do not affect the polling and acknowledgment mechanism. The MAC header is encoded using the lowest data-rate (most robust) modulation format 1/2 FEC rate BPSK separately from the data payload. Unlike the other headers, this header is treated as a sequence of octets which are transmitted LSB first.

The high-reliability control defines the subfields given in Figure 82.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit 0-1** | **Bit 2-6** | **Bit 7-11** | **Bit 12-16** | **Bit 17** | **Bit 18-29** | **Bit 30** | **Bit 31** | **Bit 32 – 39** | **Bit 40-41** | **Bit 42-47** |
| reserved | Polled  Device | Next  Device  To Poll | Device  To Acknowledge | Buffer  Status Reporting | Sequence  Number | Payload  ACK | Random  Access  ACK | CRC | reserved | Tail |

**Fields in high-reliability control**

The individual subfields of the high-reliability control header are described as follows.

**Polled Device** subfield specifies the device, which is being polled by the coordinator in the downlink. On the uplink, these bits are left unspecified.

**Next Device To Poll** subfield specifies the next device, which is being polled by the coordinator in the downlink. On the uplink, these bits are left unspecified.

**Device To Acknowledge** subfield specifies the device, which is being acknowledged by the PPDU.

**Buffer Status Reporting** subfield indicates that the current device has more date in the queue waiting for transmission. The bit is only used by devices. For coordinators, the bit is unspecified.

**Sequence Number** contains the 12-bit sequence number associated with the MPDU in the payload. Hence, it is equal to the *Sequence Number* field in the MPDU's *Sequence Control* field.

**Payload ACK** indicates that this ACK is for a payload.

**Random Access** **ACK** indicates that this ACK is for a *Random Access* frame.

**CRC** consists of eight bits CRC for previous sub-fields.

**Reserved** subfield includes two bits reserved for future use.

**Tail** subfield consists of six bits of zeros to complete the high-reliability control.

* + 1. **Payload**

**General**

Payload is transmitted at one of the supported data rates. Payload consists of service, PSDU Length, PSDU, tail, and pad fields.

The payload defines the fields given in Figure 83.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit 0-15** | **Bit 16-47** | **Variable** | **Variable** | **variable** |
| Service | PSDU  Length | PSDU | Tail | Pad |

**Fields in the payload**

The individual fields of the payload are described as follows.

**Service:** Service bits for scrambler initialization.

**PSDU Length:** Indicates the octet aligned length of the subsequent PSDU as opposed to the 32 bit aligned PSDU length in 11.2.5.1.

**PSDU:** Indicates the length of the PHY frame.

**Tail:** Six zero bits.

**Pad:** Pad bits are appended to payload field as to ensure the number of bits in the payload field to be a multiple of *N*CBPS.

* + - 1. **Service subfield**

The *Service* subfield has 16 bits, which shall be denoted as bits 0-15. The bit zero shall be transmitted first in time. The bits from zero to six of the *Service* subfield, which are transmitted first, are set to zeros and are used estimate the initial state of the transmitter scrambler and to synchronize the descrambler in the receiver. The remaining nine bits (7-15) of the *Service* subfield shall be reserved for future use. All reserved bits shall be set to zero. The bit allocation is demonstrated in Figure 84.



**Service field bit allocation**

**PSDU Length subfield**

The field specifies the octet aligned length of the PSDU in the payload. For a simple hardware implementation it is preferable to have all data 32 bit aligned. However, MPDUs submitted to the PHY may have any length. Therefore, this field is added before the PSDU in order to recover the exact length at the receiver. The *PSDU length aligned 32* field gives the 32 bit aligned PSDU length.

* + - 1. **PSDU field**

The PSDU field has a variable length and carries the data of the PHY frame. The last four octets of the PSDU are for the FCS in order to detect transmission errors in the PSDU

* + - 1. **Tail field**

The PPDU Tail field shall be six bits of zeros, which are required to complete the payload.

* + - 1. **Padding Bits**

The number of bits in the payload field shall be a multiple of *N*CBPS, the number of coded bits in an OFDM symbol (24, 48, 96, or 144 bits). To achieve this, the length of the message is extended so that it becomes a multiple of *N*DBPS, the number of data bits per OFDM symbol. At least 6 bits are appended to the message, in order to accommodate the Tail bits. The number of OFDM symbols, *N*SYM; the number of bits in the payload field, *N*PAYLOAD; and the number of pad bits, *N*PAD, are computed from the length of the PSDU (LENGTH) as follows:

The appended pad bits are set to zero and are subsequently scrambled with the rest of the bits in the payload.

* + - 1. **PHY DATA scrambler and descrambler**

The DATA field shall be scrambled with a length-127 PPDU-synchronous scrambler. The octets of the PSDU are placed in the serial bit stream for transmission, bit zero first and bit seven last. The PPDU synchronous scrambler uses the generator polynomial as follows and is illustrated in Figure 85.



**PHY DATA scrambler**

The 127-bit sequence generated repeatedly by the scrambler shall be (leftmost used first), 00001110 11110010 11001001 00000010 00100110 00101110 10110110 00001100 11010100 11100111 10110100 00101010 11111010 01010001 10111000 1111111, when the all 1’s initial state is used. The same scrambler is used to scramble transmit data and to descramble receive data.

**Modulation and coding**

**Interleaving**

OFDM is invariably used in conjunction with [channel coding](https://en.wikipedia.org/wiki/Channel_coding) ([forward error correction](https://en.wikipedia.org/wiki/Forward_error_correction)), and usually applies frequency and/or time [interleaving](https://en.wikipedia.org/wiki/Forward_error_correction#Interleaving).

Frequency (subcarrier) interleaving increases resistance to frequency-selective channel conditions such as fading. For example, when a part of the channel bandwidth fades, frequency-interleaving causes the bit errors that would result from those subcarriers in the faded part of the bandwidth to spread out in the bit-stream rather than being concentrated. Similarly, time interleaving causes bits that are originally close together in the bit-stream to be transmitted far apart in time, thus mitigating against severe fading as would happen when travelling at high speed.

However, time interleaving is of little benefit in slowly fading channels, such as for stationary reception, and frequency interleaving offers little to no benefit for narrowband channels that suffer from flat fading (where the whole channel bandwidth fades at the same time).

The reason why interleaving is used on OFDM is to attempt to spread the errors out in the bit-stream that is presented to the error correction decoder, because when such decoders are presented with a high concentration of errors the decoder is unable to correct all the bit errors, and a burst of uncorrected errors occurs.

All encoded data bits shall be interleaved by a block interleaver with a block size corresponding to the number of bits in two OFDM symbols, 2NCBPS. The interleaver is defined by a two-step permutation. The first permutation ensures that adjacent coded bits are mapped onto nonadjacent subcarriers. The second ensures that adjacent coded bits are mapped alternately onto less and more significant bits of the constellation and, thereby, long runs of low reliability bits, i.e. LSBs, are avoided. The index of the coded bit before the first permutation shall be denoted by ; shall be the index after the first and before the second permutation; and shall be the index after the second permutation, just prior to modulation mapping.

The first permutation is defined by

.

The function floor (.) denotes the largest integer not exceeding the parameter.

The second permutation is defined by the rule,

The value of is determined by the number of coded bits per subcarrier, , according to,

The de-interleaver, which performs the inverse relation, is also defined by two permutations.

Here the index of the original received bit before the first permutation shall be denoted by ; shall be the index after the first and before the second permutation; and shall be the index after the second permutation, just prior to delivering the coded bits to the convolutional (Viterbi) decoder. The first permutation is defined by,

.

The second permutation is defined by,

.

**Forward error correction (FEC)**

The FEC is inner convolutional coding. The PPDU shall be encoded with a convolutional encoder of coding rate R = 1/2, 2/3, or 3/4, corresponding to the desired data rate. The convolutional encoder shall use the industry-standard generator polynomials, g0 = 1338 and g1 = 1718, of rate R = 1/2, as shown in Figure 86. The bit denoted as "A" shall be output from the encoder before the bit denoted as "B". The summation operation presented in Figure 86 is a modulo-2 summation, i.e., an XOR operation. A subscript eight denotes octal values.

Higher rates shall be derived from this encoding mechanism by employing "puncturing." Puncturing is a procedure for omitting some of the encoded bits in the transmitter (thus reducing the number of transmitted bits and increasing the coding rate) and inserting a dummy zero metric into the convolutional decoder on the receive side in place of the omitted bits. The puncturing patterns are illustrated in Figure 87. Decoding by the Viterbi algorithm is recommended.

fig-clause-11-Convolution encoder (1338,1718)

**Convolution encoder (1338,1718)**



**Puncturing (bit stealing) algorithm**

**OFDM modulator**

**DC-biased OFDM modulator**

The real time-domain OFDM signal, generated at the PHY, is used to modulate the light emitting device (an LED or a laser diode), which serves as the transmitter front-end. The modulation is conducted only within the active operational range of the device. In this range, the electrical signal and the light output signal must be positive at all times.

The approach for modulating the LED active range with an OFDM signal is to set a positive operating point, around which the bipolar OFDM signal can be realized. Figure 88 illustrates this principle. The positive bias can be introduced as part of the analog front-end (in the case of AC-coupled LED drivers) or as part of the information signal (in case of DC-coupled drivers). This approach is known as DC-biased OFDM.

fig80

**DC-biased OFDM**

The IDFT size is fixed in the LB-PHY mode to enable lower implementation complexity. The modulation of the different frequency-domain subcarriers is achieved through an IDFT operation, described as follows:

where is the symbol mapped to subcarrier index . Conventionally, the IDFT is implemented with an IFFT algorithm.

The DFT/IDFT size in the current PHY mode is fixed to 64.

Subcarriers with negative indices -28 to -3 are loaded with 24 data symbols and two pilots. The pilots are located at index -21 and -7. Subcarriers with positive indices 3 to 28 are loaded with the conjugate complex of the data and pilot symbols at the negative indices. The pilot symbols have all value one.

Subcarriers with indices -2, -1, -1, 2 are set to zero in order to avoid possible low-frequency distortion in the system due to baseline wandering and background light interference.

Subcarriers -31, -30, -29, 29, 30, 31 are set to zero because those are near the band edge of the low pass filters in the system and may get attenuated excessively. Subcarriers with index 0, i.e. DC, and -32 are also set to zero. The mapping is illustrated in Figure 89.



**IDFT realization by means of an IFFT algorithm**

**Enhanced unipolar OFDM (eU-OFDM)**

An optional alternative modulation approach is the eU-OFDM. It constitutes of a digital processing algorithm, which turns the bipolar OFDM signal into a strictly unipolar information signal without the addition of an energy intensive DC component that carries no additional information.

The transmitter signals to the receiver the new transmission PHY mode using the eU and STR bits in the advanced modulation PHY header. For conformance purposes, the preamble and the PHY headers are encoded in a DC-biased OFDM fashion. Following the four BPSK OFDM symbols containing the PHY header, as well as the MIMO pilot symbols when applicable, the data field is encoded in a eU-OFDM fashion, as shown in Figure 90. The eU-OFDM algorithm works as follows.



**Enhanced Unipolar OFDM (eU-OFDM)**

**Data stream mapping with one stream for eU-OFDM**

For a single eU-OFDM stream, specified by STR = '0', two consecutive copies of every OFDM symbol are generated. The polarity of the samples in the second copy is inverted, and finally, all negative samples in the resulting time-domain signal are set to zero. The resulting positive signal is used to modulate the transmitter. The concept is illustrated in Figure 91.



**Unipolar OFDM generation (1 stream)**

**Data stream mapping with two streams for eU-OFDM**

For two eU-OFDM streams cases, every three OFDM symbols are grouped into an eU-OFDM block, where the first two symbols are assigned to data stream 1 (St1) and the remaining one symbol is assigned to data stream 2 (St2). The two symbols in St1 are modulated using the algorithm described for STR='0'. The symbol in St2 is modulated in a similar manner, but instead of two copies, four consecutive copies are created for the OFDM symbol in St2, where the first two copies are kept unchanged, while the polarity of the samples in the next two copies is inverted. Following this procedure, all negative samples in St1 and St2 are removed, and the two signals are summed. Any time domain oversampling and pulse shaping should be done after the removal of the negative samples. If done before the negative samples are removed, the oversampling and pulse shaping should also be performed at the receiver side during the signal re-modulation process required for the data recovery as explained in the RX algorithm. The resulting positive signal is used to modulate the transmitter. The concept is presented in Figure 92.



**Unipolar OFDM generation (2 streams)**

**Data stream mapping with three streams for eU-OFDM**

For three eU-OFDM streams, every seven OFDM symbols are grouped into a eU-OFDM block, where the first four symbols are assigned to data stream 1 (St1), the next two symbols are assigned to data stream 2 (St2) and the last symbol is assigned to data stream 3 (St3). The four symbols in St1 and the two streams in St2 are modulated using the algorithm described for 2-stream case. The symbol in St3 is modulated in a similar manner. However, eight consecutive copies of that symbol are generated, where the first four copies are left unchanged, while the polarity of the samples in the following four copies is reversed. Following this procedure, all negative samples in St1, St2 and St3 are removed. Any time-domain oversampling and pulse shaping should be done after the removal of the negative samples. If done before the negative samples are removed, the oversampling and pulse shaping should also be performed at the receiver side during the signal re-modulation process required for the data recovery as explained in the RX algorithm. The signals in the three streams are summed and the resulting positive signal is used to modulate the transmitter. The concept is presented in Figure 93.



**Unipolar OFDM generation (3 streams)**

**Data stream mapping with four streams for eU-OFDM**

For four eU-OFDM streams, indicated by STR = '1', every fifteen OFDM symbols are grouped into an eU-OFDM block, where the first eight symbols are assigned to data stream 1 (St1), the next four symbols are assigned to data stream 2 (St2), the following two symbols are assigned to data stream 3 (St3) and the last symbol is assigned to data stream 4 (St4).

In the first stream, two consecutive copies of every OFDM symbol are transmitted, where the second copy is multiplied by -1 (the signs of all samples are inverted in the time domain) as described in the cases with one, two, or three streams.

In the second stream, four consecutive copies of every OFDM symbol are transmitted, where the first two copies of the symbol are trans mitted in their original format, while the signs of the time-domain samples of the third and the fourth copy are inverted, i.e., the samples are multiplied by -1 as described in the cases for two streams and for three streams.

In the third stream, eight consecutive copies of every OFDM symbol are transmitted, where the first four copies are conveyed in their original format, while the signs of the time-domain samples of the fifth, sixth, seventh and eighth copy are inverted, i.e., the samples are multiplied by -1 as described in the case for two streams.

In the fourth stream, sixteen consecutive copies of every OFDM symbol are transmitted, where the first eight copies are conveyed in their original format, while the signs of the time-domain samples of the ninth, tenth, eleventh and twelfth, thirteenth, fourteenth, fifteenth and sixteenth copy are inverted, i.e., the samples are multiplied by −1.

At this point, all negative samples in the four streams are removed and the signals from the three streams are added together. Any oversampling and pulse shaping should be done after the removal of the negative samples. If done before the negative samples are removed, the oversampling and pulse shaping should also be performed at the receiver side during the signal re-modulation process required for the data recovery as explained in the RX algorithm. The resulting positive signal is used to modulate the transmitter.

***Remove Table D4:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table D.4—Non-beacon-enabled channel access** | | | | | | |
| * **Item number** | * **Item description** | * **Reference** | * **Status** | * **Support** | | |
| * **N/A** | * **Yes** | * **No** |
| NBECA1 | The entity is able to transmit Random Access frames in order to advertise a maintained OWPAN and initiate the polling cycle. | 6.4.2 | O |  |  |  |
| NBECA2 | The entity is able to transmit Poll Frames in order to initiate channel access of an associated device. | 6.4.3, 7.4 | O |  |  |  |
| NBECA3 | The entity is able to parse Poll frames, handle the included information, and perform transmissions in accordance with the polling-based medium access specification. | 6.4.3, 7.4 | O |  |  |  |

**Remove MCFT10 from table D17**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MCFT10 | The entity is able to parse MDPUs with type control and subtype Random Access. | 7.5 | O |  |  |  |

**Remove the following rows from table D-16:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MMFT4 | The entity is able to parse MDPUs with type management and subtype Poll | 7.4 | O |  |  |  |
| MMFT5 | The entity is able to parse MDPUs with type management and subtype Poll request | 7.4 | O |  |  |  |
| MMFT6 | The entity is able to parse MDPUs with type management and subtype Poll response | 7.4 | O |  |  |  |