**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | |
| Title | **IEEE 802.15.8 PAC Primitives** | |
| Date Submitted | 1 March, 2016 | |
| Source | [] [] [address] | Voice: [ ] Fax: [ ] E-mail: [ ] |
| Re: |  | |
| Abstract | Re-naming of PAC primitives to make follow a consistent convention. | |
| Purpose | Approval | |
| 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. | |

***IEEE 802.15.8 PAC Primitive names***

The names of primitives in PAC draft follow a number of different conventions and thus lack consistency. This document make changes to primitive names following the convention agreed on by TG8 members, which is also consistent with other standards in IEEE 802.15 WG.

The following are texts from PAC draft 0.18.0, and primitive name changes are made with change tracking feature turned on. To reduce the file size, irrelevant texts are removed when there is no danger of confusion.

**<<<< Beginning of the proposed text changes >>>>**

1. Overview
2. Normative references
3. Definitions
4. General description
   1. General
   2. Components of IEEE 802.15.8 PAC
   3. Network Topologies

Several topologies, such as mesh topology or star topology, may be used to support interactions among PDs for various services. One-to-one, one-to-many, and many-to-many topologies shall be supported.

IEEE 802.15.8 shall support a PD having simultaneous communication sessions for same or different applications. IEEE 802.15.8 shall support a PD participation in at least two independent communication sessions with different peers at the same time.



Figure —An example of concurrent communication

* 1. Architecture
  2. Functional Overview

1. MAC protocol
   1. MAC functional description

The MAC sublayer handles all access to the physical radio channel and is responsible for the following tasks:

— Network synchronization

— Supporting peering and de-peering

— Supporting device security

— Employing mechanisms for channel access

— Providing a reliable link between two peer MAC entities

Throughout this subclause, the receipt of a frame is defined as the successful receipt of the frame by the PHY and the successful verification of the frame check sequence (FCS) by the MAC sublayer.

Constants and PAC information base (PIB) attributes that are specified and maintained by the MAC sublayer or PHY layer are written in the text in italics. Constants have a general prefix of “a”, e.g., *aBaseSuperframeDuration*. MAC PIB attributes have a general prefix of “mac”, e.g., *macAckWaitDuration*. PHY PIB attributes have a general prefix of “phy”, e.g., *phyCurrentChannel*.

The next higher layer accesses the services provided by the MAC through the MAC sublayer management entity (MLME) service access point (SAP), as described in 6.1, and the MAC common part sublayer (MCPS) SAP, as described in 6.2. The primitives for the MLME SAP are written in all capital letters prefixed with MLME, e.g., MLME-SCAN.confirm. The primitives for the MCPS SAP are written in all capital letters prefixed with MCPS, e.g., MCPS-DATA.request.

* + 1. Channel access
       1. Superframe
       2. Cyclic-Superframe

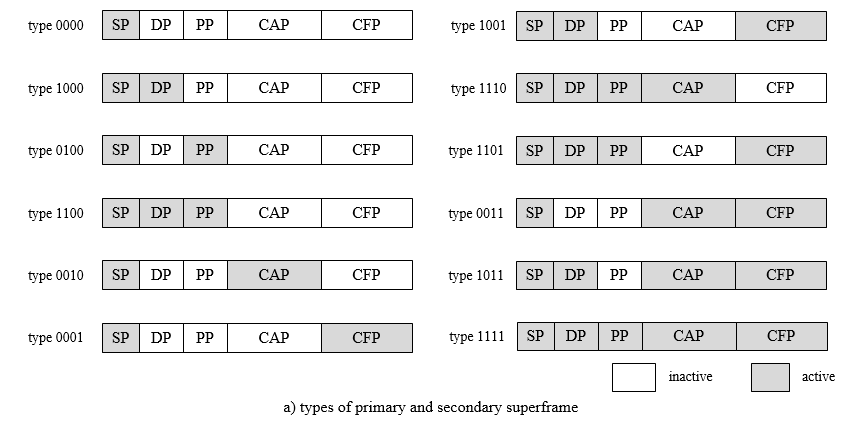
The The cyclic-superframe structure is used for a PD to adjust transceiver wakeup timing for energy efficient QoS supported transmission and for avoiding interference among PAC groups.

A cyclic-superframe is a cycle of a set of consecutive superframes. A superframe consists of a synchronization period (SP), a discovery period (DP), a peering period (PP), a CAP, and a CFP. Each period of a superframe can be set to active or inactive state to form different types of superframe. The value of superframe type is described by the period state descriptor, as defined in 5.2.4.3.1. Figure 4 a) illustrates example types out of 32 possible superframes.

A cyclic-superframe consists of superframes of two categories: primary superframe and secondary superframe. Depending on the order of position in the cyclic-superframe, each superframe can be named either “primary superframe” or “secondary superframe”. Primary or secondary superframes can take any superframe type.

The cyclic superframe structure is described by the duration of cyclic-superframe (DCS), the number of primary superframe (NPS), the number of secondary superframe (NSS), the type of primary superframe, and the type of secondary superframe.

Primary and secondary superframes shall be allocated in a cyclic-superframe. After all primary superframes are placed, secondary superframes are placed contiguously. The primary and secondary superframe can be allocated evenly over a cyclic-superframe by selecting equal number of primary and secondary superframe, as shown in Figure 4 b). When the number of primary superframe is less than secondary superframe, the primary and secondary superframe can be allocated in a cyclic-superframe, as shown in Figure 4 c).



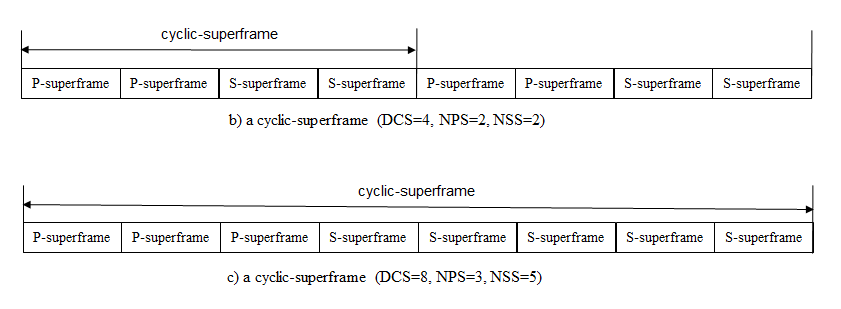


Figure —The cyclic-superframe structure for PAC

The format of cyclic-superframe shall be set by the higher layer. The higher layer designs a cyclic-superframe in the consideration of following questions:

* how many discovery command frames will be sent in a certain interval
* how many peering command frames will be sent in a certain interval
* on which period and how many data frames will be transmitted in a certain interval
* how to minimize the wake-up time
* how to avoid interference with other PAC groups

Several examples of a cyclic-superframe structure are shown in Figure 5.

In synchronization phase, to power transceiver be on only for the SP of all the superframes, the type of primary superframe and the type of secondary superframe are set to b4’0000 as shown in Figure 5 a). The synchronization phase is ended when the format of cyclic-superframe is changed to enter the discovery phase. In discovery phase, to let discovery and peering be performed for every four superframes, the values of DCS, NPS, the type of primary superframe, and the type of secondary superframe are set to 4, 3, b4’0000, and b4’1100 respectively, as shown in Figure 5 b).

In data transmission phase, according to the required data rate and QoS for a peer link or a peer group, the cyclic-superframe structure can support QoS guaranteed energy efficient data transmission by selecting a transmission period and configuring transmission interval. As an example, for medium rate best effort data transmission, the higher layer may choose a CAP for transmitting a data frame and select six superframes long cyclic-superframe, five superframes long primary superframe, SP and DP active for the primary superframe, and CAP active for the secondary superframe, as shown in Figure 5 c). With this configuration, one data frame is transmitted over CAP for every five inactive superframes. For burst reliable data transmission, three consecutive data transmission before sleeping for six inactive superframes are carried over CFP by specifying the cyclic-superframe structure with DCS set to 9, NPS set to 3, the type of primary superframe set to b4’1101, and the type of secondary superframe set to b4’0000, as shown in Figure 5 d).

The configuration of DCS and NPS for a cyclic-superframe depends on frequency of message transmission or data rate. If a discovery request message needs to be transmitted every *X ms*, DCS will be set as nearest integer of *X* divided by *aMacSuperframeLength* and NPS will be set as 1, and the discovery period of primary superframe is active. If one-to-one peer needs a best effort *Y bps* data rate link with data inter-arrival time *Z ms* and the average transmission rate in a CAP of superframe is *AvgRateCAP*, DCS will be set as nearest integer of *Z* divided by *aMacSuperframeLength* and NPS will be set as nearest integer of *Y* divided by *AvgRateCAP*, and the CAP of primary superframe is active.

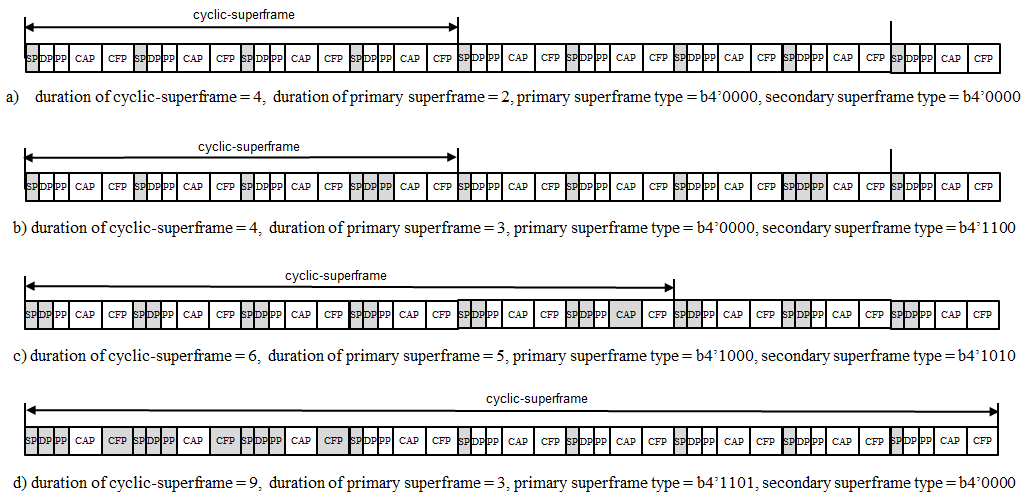


Figure —Example cyclic-superframe structures

The cyclic-superframe structure is specified by the initiator of synchronization phase, discovery phase, peering phase, and data transmission phase. A PD is instructed to set the cyclic-superframe structure through MLME-SYNCHRONIZATION.request primitive, MLME-DISCOVERY.request primitive, MLME-GROUP-START.request primitive, MLME-PEERING.request primitive, and MCPS-DATA.request primitive. The cyclic-superframe shall be repeated between one-to-one peers, one-to-many peers, or many-to-many peers until the cyclic-superframe structure is changed. The initiator shall select the time to start a new cyclic-superframe structure that minimizes the interference caused by this new cyclic-superframe to neighbor peer groups.

The specification of a cyclic-superframe structure shall be advertised with Cyclic-superframe descriptor IE, which is four bytes long header IE with the duration of cyclic-superframe, the number of primary superframe, the type of primary superframe, the type of secondary superframe, and the sequential order of superframe sending Cyclic-superframe descriptor IE. The Cyclic-superframe descriptor IE can be contained in a discovery request command frame, peering request command frame, or data frame.

The start of a cyclic-superframe is detected by the sequential order of superframe sending Cyclic-superframe descriptor IE.

* + - 1. Interframe spacing
    1. Starting and maintaining PACs
       1. Channel Scanning
          1. General description
          2. ED channel scan
          3. Active and passive channel scan

An active or passive channel scan allows a PD to locate any other PD within its radio communications range. A PD in an active scan transmits to extract any other intended PD. A PD in a passive scan detects the intended channels. A message sequence chart for active scan is illustrated in Figure 7 and for passive scan in Figure 8.



Figure —Active scan message sequence chart

During an active or a passive scan, the MAC sublayer shall discard all frames received over the PHY data service that are not related to scan. If a scan related frame is received that contains the address of the scanning PD in its list of pending addresses, the scanning PD shall not attempt to extract the pending data.



Figure —Passive scan message sequence chart

An active or a passive scan over a specified set of channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate an active or a passive scan. For each channel, the PD shall first switch to the channel, by setting *phyCurrentChannel* and *phyCurrentPage* accordingly. For an active scan, the PD shall send a scan request command. For UWB PHYs, the scan process shall be repeated for each mandatory preamble code, setting the *phyCurrentCode* appropriately. Upon successful transmission of the scan request command for an active scan or after switching to the channel for a passive scan, the PD shall enable its receiver for at most [*aBaseSuperframeDuration* × (2*n* + 1)], where *n* is the value of the ScanDuration parameter. During this time, the PD shall reject all non-scan-related frames and record the information contained in the received scan frames in a group descriptor structure, as as described in subclause 6.1.9, including the channel information and, if required, the preamble code.

If a scan frame is received when *macAutoRequest* is set to TRUE, the list of group descriptor structures shall be stored by the MAC sublayer until the scan is complete; at this time, the list shall be sent to the next higher layer in the GroupDescriptorList parameter of the MLME-SCAN.confirm primitive. A PD shall be able to store at least one group descriptor. A scan frame shall be assumed to be unique if it contains both a group ID and a source address that has not been seen before during the scan of the current channel.

If a scan frame is received when *macAutoRequest* is set to FALSE, each recorded group descriptor is sent to the next higher layer in a separate MLME-GROUP-NOTIFY.indication primitive as described in subclause 6.1.9. A received scan frame containing one or more octets of payload shall also cause the group descriptor to be sent to the next higher layer via the MLME-GROUP-NOTIFY.indication primitive. Once the scan with *macAutoRequest* set to FALSE is complete, the MLME-SCAN.confirm shall be issued to the next higher layer with a null GroupDescriptorList.

For UWB PHYs, the scan request is repeated for each preamble code.

If a protected scan frame is received, i.e., the Security Enabled field is set to one, the PD shall attempt to unsecure the scan frame using the unsecuring process described in clause 15.

The security-related elements of the group descriptor, as described in subclause 6.1.9, shall be set to the corresponding parameters returned by the unsecuring process. The SecurityStatus element of the group descriptor shall be set to SUCCESS if the status from the unsecuring process is SUCCESS and set to one of the other status codes indicating an error in the security processing otherwise.

The information from the unsecured frame shall be recorded in the group descriptor even if the status from the unsecuring process indicated an error.

If *macAutoRequest* is set to TRUE, the active scan on a particular channel shall terminate when the number of PDs found equals the implementation-specified limit or the channel has been scanned for the full time. If *macAutoRequest* is set to FALSE, the active scan on a particular channel shall terminate when the channel has been scanned for the full time. If a channel was not scanned for the full time, it shall be considered to be unscanned.

If *macAutoRequest* is set to TRUE, the entire scan procedure shall terminate when the number of group descriptors stored equals the implementation-specified maximum or every channel in the set of available channels has been scanned. If *macAutoRequest* is set to FALSE, the entire scan procedure shall only terminate when every channel in the set of available channels has been scanned.

* + - 1. Device discovery
    1. Common mode and channel access order
    2. Peering and de-peering
    3. Managing dynamic preamble selection (DPS)

It is anticipated that typical ranging traffic will take place using the normal preamble codes. Therefore, even if the messages are encrypted, a hostile device can monitor traffic, and turn on its transmitter to generally disrupt the ranging. To defeat this attack, this standard offers the DPS option, allowing the PDs participating in the ranging exchange to change preamble codes during the ranging exchange.

The coordination of DPS is beyond the scope of this standard; however it may be achieved by encrypted messages so that any hostile devices are denied knowledge of the preambles that will be used. To defeat a “jam and spoof the retry” attack the preambles used should be changed for each ranging attempt.

The *DPSIndexDuration* parameter ensures that the PDs are returned to an interoperable data state in the event of not receiving an expected message. PDs that do not implement DPS do not give up any other ranging capabilities.

Figure 11 shows a suggested message sequence employing DPS for the three-message double-sided two-way ranging exchange. The messages represented in the dot boxes are simple suggestions showing how the DPS capability might be selected. The MLME-DPS.request primitive is described in 6.1.12.1, and the MLME-DPS.confirm, as described in 6.1.12.2.

Upon the generation of the MLME-DPS.confirm primitive, as illustrated in Figure 11, both of the PHYs have switched to use the TxDPSIndex and RxDPSIndex codes for their preamble symbols and MAC sublayers on both sides have started timers of the *DPSIndexDuration* value. The changing of preamble code is intended to help protect against attack, but as a side effect neither PHYs can communicate with the rest of its peers. The *DPSIndexDuration* value serves to prevent this situation persisting any longer than is necessary. If this timer duration is exceeded before the MAC sublayer issues the MCPS-DATA.confirm (for the originator) or the MCPS-DATA.indication primitive (for the recipient), then the MAC sublayer shall initiate the MLME-DPS.indication to the next higher layer as described in 6.1.12.3 .



Figure —message sequence chart for the three message DS-TWR using DPS

* 1. MAC frame formats
     1. Device addresses
     2. General MAC frame format
     3. Format of individual frame types
     4. Information Elements (IEs)
        1. General IE format
        2. IE list termination
        3. Header IEs
        4. Payload IEs

The Payload IEs are defined in Table 16, Table 17 and Table 18, one table for each of the IE classes. These tables are ordered by IE ID number with a description for each and reference to the sub-clause that defines the IE and its usage.

**Table 16—Class 0 Payload IEs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class 0  Payload IE ID** | **IE Description** | **Acronym** | **Sub-clause** |
| 0 | Payload IE list terminator. | - | 5.2.4.2 |
| 1 | Ranging Request Reply Time IE | RRRT IE | 13.2.3.1 |
| 2 | Ranging Reply Time Instantaneous IE | RRTI IE | 13.2.3.2 |
| 3 | Ranging Reply Time Deferred IE | RRTD IE | 13.2.3.3 |
| 4 | Ranging Preferred Reply Time IE | RPRT IE | 13.2.3.4 |
| 5 | Ranging Control Double-sided TWR IE | RCDT IE | 13.2.3.5 |
| 6 | Ranging Round Trip Measurement IE | RRTM IE | 13.2.3.6 |
| 7 | Ranging Time-of-Flight IE | RTOF IE | 13.2.3.7 |
| 8 to 15 | Reserved |  |  |

**Table 17—Class 1 Payload IEs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class 1  Payload IE ID** | **IE Description** | **Acronym** | **Sub-clause** |
| 0 to 511 | Reserved | - | - |

**Table 18—Class 3 Payload IEs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class 3  Payload IE ID** | **IE Description** | **Acronym** | **Sub-clause** |
| 0 to 15 | Reserved | - | - |

* + 1. Transmission, reception, and acknowledgment
  1. Synchronization procedure
  2. Discovery
  3. Peering
  4. Communication period
  5. MAC commands

1. MAC services
   1. MLME-SAP primitives

The MLME-SAP allows to access management commands between the MAC sublayer and MAC user. Table 27 summarizes the primitives supported by the MLME-SAP interface.

Table —MLME-SAP primitives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Request** | **Confirm** | **Indication** | **Response** |
| MLME-GET | 6.1.1.1 | 6.1.1.2 | − | − |
| MLME-SET | 6.1.1.3 | 6.1.1.4 | − | − |
| MLME-DISCOVERY | 6.1.2.1 | 6.1.2.2 | − | − |
| MLME-PEERING | 6.1.3.1 | 6.1.3.2 | 6.1.3.3 | 6.1.3.4 |
| MLME-DE-PEERING | 6.1.4.1 | 6.1.4.2 | 6.1.4.3 | − |
| MLME-FRAME-ERROR-NOTIFICATION | − | − | 6.1.5.1 | − |
| MLME-RESET | 6.1.6.1 | 6.1.6.2 | − | − |
| MLME-RECEIVER-ENABLE | 6.1.7.1 | 6.1.7.2 | − | − |
| MLME-SCAN | 6.1.8.1 | 6.1.8.2 | − | − |
| MLME-GROUP-START | 6.1.9.1 | 6.1.9.2 | 6.1.9.3 | 6.1.9.4 |
| MLME-SYNCHRONIZATION | 6.1.10.1 | 6.1.10.2 | − | − |
| MLME-SYNCHRONIZATION-LOSS | − | − | 6.1.10.3 | − |
| MLME-CHANNEL-SOUNDING | 6.1.11.1 | 6.1.11.2 | − | − |
| MLME-DPS | 6.1.12.1 | 6.1.12.2 | 6.1.12.3 | − |
| MLME-CALIBRATE | 6.1.13.1 | 6.1.13.2 | − | − |

* + 1. PIB access

These primitives are used to read or write PIB values.

* + - 1. MLME-GET.request

The primitive reads a given PIB field. The properties of this primitive are:

MLME-GET.request{

PIBName;

}

The primitive parameter is defined in Table 28.

Table —MLME-GET.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| PIBName | String | As defined in Table 65 | Name of the PIB field to read. |

* + - 1. MLME-GET.confirm

This primitive reports the result requested by MLME-GET.request. The properties of this primitive are:

MLME-GET.confirm{

Status;

PIBName;

PIBValue;

}

The primitive parameters are defined in Table 29.

Table —MLME-GET.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Status | Enumeration | SUCCESS, FAIL, UNSUPPORTED | The status of the request for PIB field information. |
| PIBName | String | As defined in Table 65 | Name of the PIB field to read. |
| PIBValue | As defined in Table 65 | As defined in Table 65 | The value of the indicated PIB field to read. |

The property PIBValue is set to 0 when Status is UNSUPPORTED.

* + - 1. MLME-SET.request

The primitive allows writing a given value to the indicated PIB field. The properties of this primitive are:

MLME-SET.request{

PIBName;

PIBValue;

}

The primitive parameters are defined in Table 30.

Table —MLME-SET.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| PIBName | String | As defined in Table 65 | Name of the PIB field to write. |
| PIBValue | Various types | As defined in Table 65 | The value of the indicated PIB field to write. |

* + - 1. MLME-SET.confirm

This primitive reports the result requested by MLME-SET.request. The properties of this primitive are:

MLME-SET.confirm{

Status;

PIBName;

}

The primitive parameters are defined in Table 31.

Table —MLME-SET.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Status | Enumeration | SUCCESS, INVALID, READ\_ONLY, UNSUPPORTED. | INVALID means PIBValue is out of range. UNSUPPORTED means PIBName is not defined in the PIB. |
| PIBName | String | As defined in Table 65 | Name of the written PIB field. |

* + 1. Discovry primitives

These primitives are used for the discovery of neighboring PDs.

* + - 1. MLME-DISCOVERY.request

The primitive requests a PD to perform discovery. The properties of this primitive are:

MLME-DISCOVERY.request{

DiscoveryMode;

DiscoveryInfo;

ChannelPage;

ScanChannels;

CyclicSuperframeStructure;

}

The primitive parameters are defined in Table 32.

Table —MLME-DISCOVERY.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| DiscoveryMode | Enumeration | TX, RX | Indicates transmission or reception of discovery information. |
| DiscoveryInfo | Array of PAC IDs descriptors | As defined in Table 33 | Indicates the discovery information. |
| ChannelPage | Integer | Any valid channel page as defined in Table TBD | The channel page on which to attempt discovery. |
| ChannelNumber | Integer | Any valid channel number as defined in Table TBD | The channel number on which to attempt discovery. |
| ScanChannels | Array of integers | Any valid channel number as defined in Table TBD | The channel numbers to be used for discovery. |
| CyclicSuperframeStructure | Cyclic-superframe structure descriptor | As defined in Table 34 | Indicates the structure of cyclic-superframe |

In RX mode, channelNumber is zero.

Discovery descriptors are defined in Table 33.

Table —PAC ID description

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Device\_ID | MAC address | PD specific | MAC address of the PD. |
| MulticastGroup\_ID | Integer | 0 to 216−1 | Multicast Group addresss of PAC network |
| Application\_ID | char | 14 octets | Application name, user name, login. |

Cyclic-superframe structure descriptors are defined in Table 34.

Table —Cyclic-superframe structure descriptor

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Cyclic-superframe duration | Integer | 1 octet | Number of superframes in a cyclic-superframe. |
| Primary superframe number | Integer | 1 octet | Number of primary superframe. |
| Primary superframe type | Integer | As defined in Figure 21  (5.2.4.3.1) | Type of primary superframe. |
| Secondary superframe type | Integer | As defined in Figure 21  (5.2.4.3.1) | Type of secondary superframe. |

* + - 1. MLME-DISCOVERY.confirm

This primitive requests association with given PD. The properties of this primitive are:

MLME-DISCOVERY.confirm{

DiscoveryMode;

DiscoveryInfo;

ChannelPage;

ScanChannels;

CyclicSuperframeStructure;

Status;

}

The primitive parameters are defined in Table 35.

Table —MLME-DISCOVERY.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| DiscoveryMode | Enumeration | TX, RX | Indicates transmission or reception of discovery information. |
| DiscoveryInfo | Array of PAC IDs descriptors | As defined in Table 33 | Indicates discovery information. |
| ChannelPage | Integer | Any valid channel page as defined in Table TBD | The channel page on which discovery was attempted. |
| ChannelNumber | Integer | Any valid channel number as defined in Table TBD | The channel number on which discovery was attempted |
| ScanChannels | Array of integers | Any valid channel number as defined in Table TBD | The channel numbers used for discovery. |
| CyclicSuperframeStructure | Cyclic-superframe structure descriptor | As defined in Table 34 | Indicates the structure of cyclic-superframe |
| Status | Enumeration | SUCCESS, FAIL, DISCOVERY\_IN\_PROGRESS | Status of the discovery request |

* + 1. Peering primitives

These primitives are used when a PD becomes associated with another PD.

* + - 1. MLME-PEERING.request

This primitive requests association with given PD. The properties of this primitive are:

MLME-PEERING.request{

ChannelNumber;

ChannelPage;

GroupMode

MulticastGroup\_ID;

DestinationAddress;

CyclicSuperframeStructure;

}

The primitive parameters are defined in Table 36.

Table —MLME-PEERING.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| ChannelNumber | Integer | As defined in Table TBD | The channel number on which to attempt peering. |
| ChannelPage | Integer | As defined in Table TBD | The channel page on which to attempt peering. |
| GroupMode | Integer | As defined in Table TBD | Group mode determines the type of peering procedure |
| MulticastGroup\_ID | Integer | 0 to 216 −1 | Group ID provided by the application layer. |
| DestinationAddress | MAC address | IEEE 48 bit addressing | Address of the PD with which to peer for one-to-one peering. |
| CyclicSuperframeStructure | Cyclic-superframe structure descriptor | As defined in Table 34 | Indicates the structure of cyclic-superframe |

Cyclic-superframe structure descriptors are defined in Table 34.

* + - 1. MLME-PEERING.confirm

The primitive reports the result requested by MLME-PEERING.request of the initiating PD. The properties of this primitive are:

MLME-PEERING.confirm{

DestinationAddress;

GroupMode;

MulticastGroup\_ID;

Status;

}

The primitive parameters are defined in Table 37.

Table —MLME-PEERING.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| DestinationAddress | MAC address | PD specific | Address of the peered PD. |
| GroupMode | Integer | As defined in Table TBD | Group mode determines the type of peering procedure |
| MulticastGroup\_ID | Integer | 0 to 216 −1 | Group ID of the established group. |
| Status | Enumeration | SUCCESS, CHANNEL\_ACCESS\_FAILURE, NO\_ACK, ACCESS\_DENIED, | The status of the peering attempt. |

If the peering request was successful, then the Status parameter will be set to SUCCESS. Otherwise, the Status parameter will be set to indicate the type of failure.

* + - 1. MLME-PEERING.indication

The primitive is used to indicate the reception of a peering request command. The properties of this primitive are:

MLME-PEERING.indication{

SourceID;

GroupMode

MulticastGroupID

PHYcapability;

CyclicSuperfameStructure;

}

The primitive parameters are defined in Table 38.

Table —MLME-PEERING.indication parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| SourceID | MAC address | PD specific | Address of the PD requesting peering. |
| GroupMode | Integer | As defined in Table TBD | Group mode determines the type of peering procedure |
| MulticastGroup\_ID | Integer | 0 to 216 −1 | Group ID of the requested group. |
| PHYcapability | Enumeration | LOW\_MOBILITY, HIGH\_MOBILITY, GFSK, UWB\_BPM\_BPSK, UWB\_OOK | Operational capability of the PD requesting peering. |
| CyclicSuperframeStructure | Cyclic-superframe structure descriptor | As defined in Table 34 | Indicates the structure of cyclic-superframe |

Cyclic-superframe structure descriptors are defined in Table 34.

* + - 1. MLME-PEERING.response

The primitive is used to initiate a response to an MLME-PEERING.indication primitive. The properties of this primitive are:

MLME-PEERING.response{

SourceID;

GroupMode;

MulticastGroup\_ID;

Status;

}

The primitive parameters are defined in Table 39.

Table —MLME-PEERING.response parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| SourceID | MAC address | PD specific | Address of the PD requesting peering. |
| GroupMode | Integer | As defined in Table TBD | Group mode determines the type of peering procedure |
| MulticastGroup ID | Integer | 0 to 216 −1 | ID of the MulticastGroup |
| Status | Enumeration | SUCCESFUL, OUT\_OF\_CAPACITY, ACCESS\_DENIED, | Status of the peering attempt. |

* + 1. De-peering primitives

These primitives are used when a PD becomes de-peered from another PD.

* + - 1. MLME-DE-PEERING.request

This primitive requests de-peering of given PD. The properties of this primitive are:

MLME-DE-PEERING.request{

DestinationAddress;

SourceAddress;

GroupMode;

MulticastGroup\_ID;

Reason;

}

The primitive parameters are defined in Table 40.

Table —MLME-DE-PEERING.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| DestinationAddress | MAC address | PD specific | Address of the PD with which to peer. |
| SourceAddresss | MAC address | PD specific | The address of the PD requesting peering. |
| GroupMode | Integer | As defined in Table TBD | Group mode determines the type of peering procedure |
| MulticastGroup\_ID | Integer | 0 to 216 −1 | Group ID of destination PD. |
| Reason | Integer | 0 to 1 | Reasons for dissociation:    0 - Source wants to leave.    1 - Source requests destination to leave. |

* + - 1. MLME-DE-PEERING.confirm

This primitive reports the result requested by MLME-DE-PEERING.request. The properties of this primitive are:

MLME-DE-PEERING.confirm{

DestinationAddress;

SourceAddress;

GroupMode;

MulticastGroup\_ID;

Status;

}

The primitive parameters are defined in Table 41.

Table —MLME-DE-PEERING.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| DestinationAddresss | MAC address | PD specific | Address of the PD with which to de-peer. |
| SourceAddresss | MAC address | PD specific | The address of the PD requesting de-peering. |
| GroupMode | Integer | As defined in Table TBD | Group mode determines the type of peering procedure |
| MulticastGroup\_ID | Integer | 0 to 216 −1 | MulticastGroup ID of destination PD. |
| Status | Enumeration | SUCCESS, NO\_ACK, CHANNEL\_ACCESS\_FAILURE, | The status of the de-peeringattempt. |

If the de-peering request was successful, then the Status parameter will be set to SUCCESS. Otherwise, the Status parameter will be set to indicate the type of failure.

* + - 1. MLME-DE-PEERING.indication

The primitive is used to indicate the reception of a de-peering request command.

MLME-DE-PEERING.indication{

SourceID;

GroupMode;

MulticastGroup ID

Reason;

}

The primitive parameters are defined in Table 42.

Table —MLME-DE-PEERING.indication

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| SourceID | MAC address | PD specific | The address of the PD requesting de-peering. |
| GroupMode | Integer | As defined in Table TBD | Group mode determines the type of peering procedure |
| MulticastGroup\_ID | Integer | 0 to 216 −1 | ID of MulticastGroup |
| Reason | Integer | 0 − 1 | 0 – source wants to leave  1 – source requests destination to leave |

* + 1. Frame error notification primitives

This primitive is used to notify the next higher layer that an error has occurred during the processing of a frame.

* + - 1. MLME-FRAME-ERROR-NOTIFICATION.indication

This primitive indicates a communications status. The properties of this primitive are:

MLME-FRAME-ERROR-NOTIFICATION.indication{

SourceAddress;

DestinationAddress;

MulticastGroup\_ID;

Status;

}

The primitive parameters are defined in Table 43.

Table —MLME-FRAME-ERROR-NOTIFICATION.indication parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| SourceAddress | MAC address | PD specific | Address of the PD from which the frame causing an error originated. |
| DestinationAddress | MAC address | PD specific | Address of the PD from which the frame causing an error originated. |
| MulticastGroup\_ID | Integer | 0 to 216 −1 | Group ID of destination PD. |
| Status | Enumeration | TRANSACTION\_OVERFLOW, TRANSACTION\_EXPIRED, CHANNEL\_ACCESS\_FAILURE, NO\_ACK, TBD | The status of the communication error frame. |

TRANSACTION\_OVERFLOW – The MLME does not have the capacity to store the superframe that was to be sent.

TRANSACTION\_EXPIRED – The transaction was not handled within macTransactionPersistenceTime.

CHANNEL\_ACCESS\_FAILURE – There was a failure in the CAP, CFP while attempting to send the frame.

NO\_ACK – An acknowledgment was expected but not received.

* + 1. Reset MAC sublayer

These primitives are used to reset the MAC sublayer. The execution is implementation specific.

* + - 1. MLME-RESET.request

This primitive is used by the next higher layer to request a reset operation. The properties of this primitive are:

MLME-RESET.request{

ResetMode;

}

The primitive parameters are defined in Table 44.

Table —MLME-RESET.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| ResetMode | Boolean | TRUE, FALSE | Reset the MAC sublayer. |

If TRUE, the MAC sublayer is reset and all MAC PIB fields are set to their default values. If FALSE, the MAC sublayer is reset, but all MAC PIB fields retain their values prior to the generation of the MLME-RESET.request primitive.

* + - 1. MLME-RESET.confirm

This primitive reports the results of the reset operation. The properties of this primitive are:

MLME-RESET.confirm {

Status;

}

The primitive parameters are defined in Table 45.

Table —MLME-RESET.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Status | Enumeration | SUCCESS, FAIL | The result of the reset operation. |

* + 1. Receiver enable

These primitives are used to enable or disable a receiver of a PD.

* + - 1. MLME-RECEIVER-ENABLE.request

This primitive allows the next higher layer to request that the receiver is either enabled for a finite period of time or disabled. The properties of this primitive are:

MLME-RECEIVER-ENABLE.request {

Defer;

RxOnTime;

RxOnDuration;

}

The primitive parameters are defined in Table 46.

Table —MLME-RECEIVER-ENABLE.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Defer | Boolean | TRUE, FALSE. | TRUE if the requested operation can be deferred until the next superframe, if the requested time has already passed. FALSE if the requested operation is only to be attempted in the current superframe. |
| RxOnTime | Integer | 0 to (224 −1) | The number of symbols measured from the start of the superframe before the receiver is to be enabled or disabled. |
| RxOnDuration | Integer | 0 to (224 −1) | The number of symbols for which the receiver is to be enabled. This property is equal to 0, if the receiver is to be disabled. |

The primitive enables the radio to receiver mode for a fixed duration, at a time relative to the start of the current or next superframe, or immediately during the CAP, or according to the TDD receiver mode, and without conflict with another higher priority operation of the radio transceiver.

In case of conflicting operation with a higher priority task of the radio transceiver, the PD will interrupt the receive operation, until the completion of the superseded operation. The property RxOnDuration will be checked to determine whether the time has expired. If so, the ReceiverEnable primitive operation is complete. If not, the receiver is re-enabled until either the PD has another superseded task, or the time specified by the property RxOnDuration has expired. When the primitive is issued to disable the receiver, the PD will disable its receiver unless the radio transceiver has a higher priority task.

This primitive may also be generated to cancel a previously generated request to enable the receiver. The receiver is enabled or disabled exactly once per primitive request.

If the RxOnDuration parameter is equal to zero, the MLME requests the PHY to disable its receiver.

* + - 1. MLME-RECEIVER-ENABLE.confirm

This primitive reports the results of the attempt to enable or disable the receiver. The properties of this primitive are:

MLME-RECEIVER-ENABLE.confirm {

Status;

}

The primitive parameters are defined in Table 47.

Table —MLME-RECEIVER-ENABLE.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Status | Enumeration | SUCCESS, PAST\_TIME, ON\_TIME\_TOO\_LONG, INVALID\_PARAMETER | The result of the request to enable or disable the receiver. |

The primitive is generated by the MLME and issued to its next higher layer in response to an MLME-RECEIVER-ENABLE.request primitive. This primitive returns a status of either SUCCESS, if the request to enable or disable the receiver was successful or the appropriate error code.

Before attempting to enable the receiver, the MLME first determines whether (RxOnTime + RxOnDuration) is less than the synchronization period interval. If (RxOnTime + RxOnDuration) is not less than the synchronization period interval, the MLME issues the MLME-RECEIVER-ENABLE.confirm primitive with the property Status=ON\_TIME\_TOO\_LONG.

The MLME then determines whether the receiver can be enabled in the current superframe. If the current time measured from the start of the superframe is less than (RxOnTime – macSIFSPeriod), the MLME attempts to enable the receiver in the current superframe. If the current time measured from the start of the superframe is greater than or equal to (RxOnTime – macSIFSPeriod) and DeferPermit is equal to TRUE, the MLME defers until the next superframe and attempts to enable the receiver in that superframe. Otherwise, if the MLME cannot enable the receiver in the current superframe and is not permitted to defer the receive operation until the next superframe, the MLME issues the MLME-RECEIVER-ENABLE.confirm with the property Status=PAST\_TIME.

* + 1. Channel scanning

This primitive is used to initiate a channel scan over a given list of channels. The properties of this primitive are:

* + - 1. MLME-SCAN.request

The MLME-SCAN.request primitive is used to initiate a channel scan over a given list of channels. The properties of this primitive are:

MLME-SCAN.request{

ScanType;

ChannelPage;

ScanChannels;

ScanDuration;

}

The primitive parameters are defined in Table 48.

Table —MLME-SCAN.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| ScanType | Enumeration | ED, ACTIVE, PASSIVE. | Indicates the type of scan performed. |
| ChannelPage | Integer | Any valid channel page as indicated in Table TBD | The channel page on which to perform the scan. |
| ScanChannels | Array of integers | Any valid channel number as indicated in Table TBD | The channel numbers to be scanned |
| ScanDuration | Integer | TBD | A value used to calculate the length of time to spend scanning each channel for ED, active, and passive scans. |

* + - 1. MLME-SCAN.confirm

This primitive reports the result of the channel scan request. The properties of this primitive are:

MLME-SCAN.confirm{

Status,

ScanType;

ChannelPage;

UnscannedChannels;

ResultListSize;

EnergyDetectList;

DetectedCategory;

}

The primitive parameters are defined in Table 49.

Table —MLME-SCAN.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Status | Enumeration | SUCCESS, LIMIT\_REACHED, SCAN\_IN\_PROGRESS, FRAME\_TOO\_LONG, INVALID\_PARAMETER | The status of the scan request. |
| ScanType | Enumeration | ED, ACTIVE, PASSIVE | Indicates the type of scan performed |
| UnscannedChannels | Array of integers | Any valid channel number as defined in Table TBD | A list of the channels given in the request which were not scanned. This parameter is not valid for ED scans |
| ResultListSize | Integer | Implementation specific | The number of elements returned in EnergyDetectList. |
| EnergyDetectList | Array of integers | 0 to 255 per element in the array. | The list of energy measurements, one for each channel searched during an ED scan. This parameter is null for active, passive scans. |
| DetectedCategory | Integer | 0 to 2 | Categorization of energy detection with the following values:  0: Category detection is not supported  1: UWB PHY detected  2: Non-UWB PHY signal source detected. |

If the MLME receives a MLME-SCAN.request primitive while performing a previously initiated scan operation, the MLME will not perform the scan and the Status property will be set to SCAN\_IN\_PROGRESS. If the MLME-SCAN.request primitive requested an active or passive scan, the EnergyDetectList property will be NULL.

* + 1. Group configuration

These primitives are used to initiate a PAC Group or to begin using a new superframe configuration.

* + - 1. MLME-GROUP-START.request

This primitive is also used by a PD already peered with an existing PAC Group to begin using a new superframe configuration. The properties of this primitive are:

MLME-GROUP-START.request{

GroupID;

InitiatorID;

ChannelNumber;

ChannelPage;

StartTime;

SuperframeOrder;

CyclicSuperframeStructure;

}

The primitive parameters are defined in Table 50.

Table —MLME-GROUP-START.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Initiator\_ID | MAC address | PD specific | The address of the initiator PD |
| MulticastGroup\_ID | Integer | 0 to 216 −1 | Group ID of initiator PD. |
| ChannelNumber | Integer | Any valid channel number a defined in Table TBD | The channel number to use. |
| ChannelPage | Integer | Any valid channel page as defined in Table TBD | The channel page to use. |
| StartTime | Unsigned integer | 0 to (224 −1) | The time at which to begin transmitting the synchronization period. If this parameter is equal to 0, the synchronization period will begin immediately. Otherwise, the specified time is relative to the received synchronization period of the current initiator PD. |
| CyclicSuperframeStructure | Cyclic-superframe structure descriptor | As defined in Table 34 | Indicates the structure of cyclic-superframe |

Cyclic-superframe structure descriptors are defined in Table 34.

* + - 1. MLME-GROUP-START.confirm

This primitive reports the results of the attempt to start using a new superframe configuration. The properties of this primitive are:

MLME-GROUP-START.confirm{

Status;

}

The primitive parameters are defined in Table 51.

Table —MLME-GROUP-START.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Status | Enumeration | SUCCESS, SUPERFRAME\_OVERLAP, INVALID\_PARAMETER, FRAME\_TOO\_LONG, CHANNEL\_ACCESS\_FAILURE | The result of the attempt to start a PAC Group or an updated superframe configuration. |

The MLME-GROUP-START.confirm primitive returns a status of either SUCCESS, indicating that the MAC sublayer has started a new PAC Group or using the new superframe configuration. Otherwise, an appropriate error code is return as follows:

CHANNEL\_ACCESS\_FAILURE – The transmission lost synchronization.

FRAME\_TOO\_LONG – The length of the MPDU exceeds MaxPHYPacketSize.

SUPERFRAME\_OVERLAP – The outgoing superframe overlaps the incoming superframe.

* + - 1. MLME-GROUP-START.indication

The primitive is used to indicate the reception of GroupStart request command. The properties of this primitive are:

MLME-GROUP-START.indication{

PD\_ID;

PHYcapability;

CyclicSuperframeStructure;

}

The primitive parameters are defined in Table 52.

Table —MLME-GROUP-START.indication parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| PD\_ID | MAC address | PD specific | Address of the PD requesting GroupStart. |
| PHYcapability | Enumeration | LOW\_MOBILITY, HIGH\_MOBILITY, GFSK, UWB\_BPM\_BPSK, UWB\_OOK | Operational capability of the PD requesting GroupStart. |
| CyclicSuperframeStructure | Cyclic-superframe structure descriptor | As defined in Table 34 | Indicates the structure of cyclic-superframe |

Cyclic-superframe structure descriptors are defined in Table 34.

* + - 1. MLME-GROUP-START.response

The primitive is used to initiate a response to an MLME-GROUP-START.indication primitive. The properties of this primitive are:

MLME-GROUP-START.response{

PD\_ID;

MulticastGroup\_ID;

Status;

}

The primitive parameters are defined in Table 53.

Table —MLME-GROUP-START.response parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| PD\_ID | MAC address | PD specific | Address of the PD requesting GroupStart. |
| MulticastGroup ID | Integer | 0 to 216 −1 | MulticastGroup address. This parameter is set to 0 if there is no multicastGroup address. |
| Status | Enumeration | SUCCESFUL, OUT\_OF\_CAPACITY, ACCESS\_DENIED, | Status of the GroupStart attempt. |

* + 1. Synchronization

These primitives are used to synchronize with other PDs and to communicate loss of synchronization to the next higher layer.

* + - 1. MLME-SYNCHRONIZATIOHN.request

The primitive requests to synchronize with other PDs by acquiring and tracking their synchronization signal. The properties of this primitive are:

MLME-SYNCHRONIZATION.request{

ChannelNumber;

ChannelPage;

TrackMode;

CyclicSuperframeStructure;

}

The primitive parameters are defined in Table 54.

Table —MLME-SYNCHRONIZATION.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| ChannelNumber | Integer | As defined in Table TBD | The channel number on which to attempt synchronization. |
| ChannelPage | Integer | As defined in Table TBD | The channel page on which to attempt synchronization. |
| TrackMode | Boolean | TRUE, FALSE | TRUE if the PD is to synchronize from the next synchronization period. FALSE if the PD is to synchronize with only the next synchronization period. |
| CyclicSuperframeStructure | Cyclic-superframe structure descriptor | As defined in Table 34 | Indicates the structure of cyclic-superframe |

When the primitive is received by the MLME, it will set phyCurrentPage and phyCurrentChannel equal to the values of the ChannelPage and ChannelNumber, respectively. If the TrackMode parameter is equal to TRUE, the MLME will track the synchronization period, i.e., enable its receiver just before the expected time of synchronization period. If the TrackMode parameter is equal to FALSE, the MLME will locate the synchronization period, but not continue to track it.

If this primitive is received by the MLME while it is currently tracking the synchronization period, the MLME will not discard the primitive but will treat it as a new synchronization request.

Cyclic-superframe structure descriptors are defied in Table 34.

* + - 1. MLME-SYNCHRONIZATION.confirm
      2. MLME-SYNCHRONIZATION-LOSS.indication

The primitive indicates the loss of synchronization. The properties of this primitive are:

MLME-SYNCHRONIZATION-LOSS.indication{

LossReason;

GroupID;

PD\_ID;

ChannelNumber;

ChannelPage;

}

The primitive parameters are defined in Table 55.

Table —MLME-SYNCHRONIZATION-LOSS.indication parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| LossReason | Enumeration | SYNC\_LOST, SUPERFRAME\_OVERLAP, TBD | The reason synchronization was lost. |
| MulticastGroup\_ID | Integer | 0 to 255 | Group ID of the PD that has lost synchronization |
| PD\_ID | MAC address | PD specific | The MAC address of the PD that has lost synchronization |
| ChannelNumber | Integer | Any valid channel number as defined in Table TBD | The channel number on which the PD that has lost synchronization operated. |
| ChannelPage | Integer | Any valid channel page as defined in Table TBD | The channel page on which the PD that has lost synchronization operated. |

The primitive is generated by the MLME of a PD and issued to its next higher layer in the event of a loss of synchronization. The reason for synchronization loss is expressed by the parameter of LossReason:

SYNC\_LOST – The PD has missed several synchronization periods.

SUPERFRAME\_OVERLAP – If a change in the superframe configuration causes the outgoing superframe overlaps the incoming superframe.

* + 1. Channel sounding

These primitives are used to obtain the results of a channel sounding from PDs that supports the channel sounding capability.

* + - 1. MLME-CHANNEL-SOUNDING.request

The primitive is used by the next higher layer to request that the PHY performs channel sounding by PDs with that capability. The properties of this primitive are:

MLME-CHANNEL-SOUNDING.request{

ChannelNumber;

ChannelPage;

}

The primitive parameters are defined in Table 56.

Table —MLME-CHANNEL-SOUNDING.request

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| ChannelNumber | Integer | Any valid channel number as defined in Table TBD | The channel number on which to attempt channel sounding. |
| ChannelPage | Integer | Any valid channel page as defined in Table TBD | The channel page on which to attempt channel sounding. |

* + - 1. MLME-CHANNEL-SOUNDING.confirm

The primitive reports the result of a request to the PHY to provide channel sounding information. Such information is an estimate of the SINR at the input antenna of the requestee PD. The properties of this primitive are:

MLME-CHANNEL-SOUNDING.confirm{

Status;

SINR;

CQI;

}

The primitive parameters are defined in Table 57.

Table —MLME-CHANNEL-SOUNDING.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| Status | Enumeration | SUCCESS, FAIL, UNSUPPORTED | The status of the attempt to return channel sounding data. |
| SINR | Integer | −40 to 40 | SINR estimate at the input antenna in dBm. |
| CQI | Integer | 0 to 6 | CQI value measured during reception of the PSDU.  0 – very good  1 – good  2 – medium  3– bad |

* + 1. Dynamic preamble selection

The dynamic preamble selection (DPS) primitives apply to the UWB PHY allowing the temporary use of different preamble codes as a means to thwart a hostile third party attack on the ranging procedures.

* + - 1. MLME-DPS.request

This requests that the UWB PHY temporarily use the specified preamble code in transmitter and/or receiver for a single use. The properties of this primitive are:

MLME-DPS.request {

TxDPSIndex;  
RxDPSIndex;  
DPSIndexDuration;

}

The primitive parameters are defined in Table 58.

Table —MLME-DPS.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Value range** | **Description** |
| TxDPSIndex | Integer | 0 to 48 | The code index value for the UWB PHY transmitter to use in UWB SHR generation. A value of 0 indicates that the *phyPreambleCode* value is used (Table 65). See note below. |
| RxDPSIndex | Integer | 0 to 48 | The code index value for the UWB PHY receiver to use in SHR detection and reception. A value of 0 indicates that the *phyPreambleCode* value is used (Table 65). See note below. |
| DPSIndexDuration | Integer | 0 to (224 −1) | This specifies a time in µs after which the DPS selection will revert to use the *phyPreambleCode* value in both transmitter and receiver, assuming it is not ended earlier by the issuing of either an MCP-DATA.request or an MCP-DATA.indication primitive (Table 65). |

Note: The TxDPSIndex and RxDPSIndex values select codes as per the “Preamble Sequence ID” column of Table 105 for the BPM-BPSK modulation mode, and as per parameter *i* of Table 109 for the OOK modulation mode where the DPS index specified here is *i*+ 1.

The MLME responds with an MLME-DPS.confirm primitive with the appropriate status parameter, and assuming the MLME-DPS.request has been successful, the resulting altered preamble codes persist until one of the follow events:

* The DPS is cancelled by issuing a new MLME-DPS.request with zero values for both TxDPSIndex and RxDPSIndex parameters,
* A frame is received and the MCP-DATA.indication primitive is generated,
* A frame is transmitted via an MCP-DATA.request primitive and the MCP-DATA.confirm primitive is generated,
* The period specified by the DPSIndexDuration parameter expires and the MLME-DPS.indication primitive is issued.

The use of DPS is further explained in 5.1.5.

* + - 1. MLME-DPS.confirm

The MLME-DPS.confirm primitive reports the results of the attempt to enable or disable DPS.

The properties of this primitive are:

MLME-DPS.confirm {

Status;

}

The primitive parameters are defined in Table 59.

Table —MLME-DPS.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Value range** | **Description** |
| Status | Enumeration | SUCCESS, DPS\_NOT\_SUPPORTED | The result of the request to enable or disable DPS |

The MLME-DPS.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-DPS.request primitive.

If any parameter value in the MLME-DPS.request primitive is out of range is not supported by the PD in its currentPHY operating mode as selected by *phyModeSelection*, the status of DPS\_NOT\_SUPPORTED is returned. If the request to enable or disable the DPS is successful, the MLME issues the MLME-DPS.confirm primitive with a status of SUCCESS.

* + - 1. MLME-DPS.indication

The MLME-DPS.indication primitive informs the next higher layer that the DPSIndexDuration period, specified in the MLME-DPS.request, has expired and the preamble codes in both the transmitter and the receiver have reverted to use the code specified by the *phyPreambleCode* value.

The properties of this primitive are:

MLME-DPS.indication { <empty> }

There are no parameters for this primitive.

* + 1. Ranging calibration

A PD with a UWB PHY that supports ranging may optionally include the ability to dynamically characterize the internal delays that give rise to the *phyTxRmarkerOffset* and *phyRxRmarkerOffset* quantities. These quantities represent the propagation time from the internal transmit timestamp to the transmit antenna, and the propagation time from the receive antenna to the internal receiver timestamp. The MLME-CALIBRATE primitives are defined to support the characterization of these delays.

* + - 1. MLME-CALIBRATE.request

The MLME-CALBRATE.request primitive attempts to have the PD respond with ranging marker offset information. There are no parameters for this primitive.

MLME-CALBRATE.request { <empty> }

In response to the MLME-CALIBRATE.request the MLME issues an MLME-CALIBRATE.confirm primitive.

* + - 1. MLME-CALIBRATE.confirm

The MLME-CALIBRATE.confirm primitive reports the result of a previous MLME-CALIBRATE.request,

The properties of this primitive are:

MLME-CALIBRATE.confirm {

Status;   
CalTxRMARKEROffset;  
CalRxRMARKEROffset;

}

The primitive parameters are defined in Table 60.

Table —MLME-CALIBRATE.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Value range** | **Description** |
| Status | Enumeration | SUCCESS, UNSUPPORTED | The status result of the attempt to return the delay calibration data. |
| CalTxRMARKEROffset | Unsigned  integer | 0 to (232 −1) | A count of the propagation time from the transmit ranging counter to the transmit antenna. The units of this are defined in 10.4.1. |
| CalRxRMARKEROffset | Unsigned  integer | 0 to (232 −1) | A count of the propagation time from the receive antenna to the receive ranging counter. The units of this are defined in 10.4.1. |

The MLME-CALIBRATE.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-CALIBRATE.request primitive.

In some implementations the MLME-CALIBRATE.confirm may return values calibrated into the PD during its manufacture, in other cases the DP may perform a physical measurement of the delays. Either way, since system delays may depend on the PHY configuration, it is recommended to configure the required *phyModeSelection*, *phyPreambleCode*, etc. before initiating the MLME-CALIBRATE.request.

If the MLME-CALIBRATE feature is supported, the MLME issues the MLME-CALIBRATE.confirm primitive with a Status of SUCCESS, indicating that the CalTxRMARKEROffset and CalRxRMARKEROffset values are valid, otherwise the returned Status the status parameter will be set to UNSUPPORTED.

* 1. MAC data service

The MCPS-SAP supports the transport of data information. Table 61 lists the primitives maintained by the MCPS-SAP.

Table —MCPS-SAP primitives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Request** | **Confirm** | **Indication** | **Response** |
| MCPS.Data |  |  |  | − |

* + 1. MCPS-DATA.request

The primitive requests the transfer of data to another PD. The properties of this primitive are:

MCPS-DATA.request {

phyModeSelection;

SourceAddress;

DestinationAddress;

MulticastGroup\_ID;

MSDUlength;

MSDUHandle;

TransmitTimeOut;

DataType;

DataRate;

PurgeMode;

AckTX;

Ranging;

IeList;

RequestRriTx;

CyclicSuperframeStructure;

}

The primitive parameters are defined in Table 62.

Table —MCPS-DATA.request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| SourceAddress | MAC address | PD specific | The address of the PD from which the frame originated. |
| phyModeSelection | Enumeration | LOW-MOBILITY, HIGH\_MOBILITY, GFSK, UWB\_BPM\_BPSK, UWB\_OOK | PHY selection |
| DestinationAddress | MAC address | PD specific | Address of the PD for which the frame is intended. |
| MulticastGroup\_ID | Integer | 0 to 216 −1 | Group ID of a PAC network. |
| MSDUlength | Integer | TBD | MDSU length in octets. |
| MSDUhandle | Integer | TBD | Integer number identifying the MSDU. |
| TransmitTimeOut | Integer | TBD | The maximum allowed delay in microseconds from when the data is presented to the SAP until the frame has finished transmission and the acknowledgment, if required, is received. |
| DataType | Enumeration | AUDIO, VIDEO, DATA | Indicates the type of data that is sent in the stream. |
| DataRate | Integer | TBD | Indicates the data rate |
| PurgeMode | Boolean | TRUE, FALSE | If TRUE, the MSDU in the transaction buffer corresponding to MSDUhandle, is discarded from the transaction buffer. |
| AckTX | Boolean | FALSE, TRUE | This parameter is present only when phyModeSelection is UWB.  TRUE if acknowledged transmission is used,  FALSE otherwise. |
| Ranging | Boolean | FALSE, TRUE | This parameter is present only when phyModeSelection is UWB.  TRUE if ranging bit in PHR is to be set,  FALSE otherwise. |
| IeList | Set if IEs as described in  Array of integers (IEs) | As described in Table TBD | This parameter is present only when phyModeSelection is UWB.  Determines/supplies the IEs to be sent, including:  RRRT IE, RRTD IE, RPRT IE, RRTM IE, RCDT IE and RTOF IE |
| RequestRrtiTx | Boolean | TRUE, FALSE | This parameter is present only when phyModeSelection is UWB.  This parameter requests that the PD inserts an RRTI IE in the sent data frame. If the MCPS-DATA.request is early enough the data frame shall be sent at the configured *macRngRrtiTime* otherwise shall be sent as soon as possible thereafter. |
| CyclicSuperframeStructure | Cyclic-superframe structure descriptor | As defined in Table 34 | Indicates the structure of cyclic-superframe |

* + 1. MCPS-DATA.confirm

The primitive reports the results of a request to transfer data to another PD. The properties of this primitive are:

MCPS-DATA.confirm {

MSDUhandle;

Status;

PurgeMode;

TxRangingCounter;

RxRangingCounter;

IeList;

TxRrtiValue;

AoaAzimuth;

AoaElevation;

AoaPresent;

Rssi;

}

The primitive parameters are defined in Table 63.

Table —MCPS-DATA.confirm parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| MSDUhandle | Integer | TBD | Integer number identifying the MSDU. |
| Status | Enumeration | SUCCESS, TRANSACTION\_EXPIRED, CHANNEL\_ACCESS\_FAILURE, NO\_ACK, COUNTER\_ERROR, INVALID\_PARAMETER, PURGE\_SUCESS, INVALID\_HANDLE | The status of the last MSDU transmission. |
| TxRangingCounter | Unsigned Integer | TBD | This parameter is present only when phyModeSelection is UWB.  The timestamp of the transmitted data frame |
| RxRangingCounter | Unsigned Integer | TBD | This parameter is present only when phyModeSelection is UWB.  For an acknowledged transmission this is the timestamp of the received acknowledgement frame. For a non-acknowledged transmission this parameter is invalid. |
| IeList | IEs received in the ACK | As defined in Table TBD | This parameter is present only when phyModeSelection is UWB.  For an acknowledged transmission this reports the IEs received in the ACK including: RRTI IE |
| TxRrtiValue | Unsigned Integer | 0 to (232 −1) | This parameter is present only when phyModeSelection is UWB.  This reports the value sent in the RRTI IE where this was requested by the RequestRrtiTx parameter of the MCPS-DATA.request. |
| AoaAzimuth | Float | -π to +π | This parameter is present only when phyModeSelection is UWB.  For an acknowledged transmission this is the AOA of the received signal in azimuth measured in radians, relative to some PD specific axis defined by its antenna arrangement or orientation. This parameter is valid only when the AoaPresent parameter is either AZIMUTH or BOTH. |
| AoaElevation | Float | -π to +π | This parameter is present only when phyModeSelection is UWB.  For an acknowledged transmission this is the AOA of the received signal AOA of the received signal in elevation measured in radians. This parameter is valid only when the AoaPresent parameter is either ELEVATION or BOTH. |
| AoaPresent | Enumeration | NONE,  BOTH,  AZIMUTH,  ELEVATION | This parameter is present only when phyModeSelection is UWB.  Indicates validity of AoaAzimuth and AoaElevation parameter. For a non-acknowledged transmission or where AOA is not supported this parameter value shall be NONE. |
| Rssi | Integer | 0 to 255 | This parameter is present only when phyModeSelection is UWB.  For an acknowledged transmission this reports the received signal strength for the ACK frame. This is a measure of the RF power level at the antenna based on the gain setting in the RX chain and the measured signal level in the channel. For the UWB PHY the RSSI value is measured during the frame Preamble and locked when a valid SFD is detected. A value of zero indicates that RSSI measurement is not supported or was not measured for this frame. |

The MCPS-DATA.confirm primitive returns a status of either SUCCESS, indicating that the request to transmit was successful, or the appropriate error code.

If the requested transaction is too large to fit in the CAP or CFP, the MAC sublayer shall discard the frame and issue the MCPS-DATA.confirm primitive with a status of FRAME\_TOO\_LONG.

If the transmission uses CSMA-CA or CFP and those failed due to adverse conditions on the channel, the MAC sublayer will discard the MPDU and the Status property will be set to CHANNEL\_ACCESS\_FAILURE.

If MCPS-DATA.request PurgeMode is set to TRUE, the MCPS-DATA.confirm Status property will be set to either PURGE\_SUCCESS if an MSDU, matching the MSDUhandle property, was removed from the transaction buffer; otherwise INVALID\_HANDLE.

* + 1. MCPS-DATA.indication

The primitive indicates the reception of data from another PD. The properties of this primitive are:

MCPS-DATA.indication{

phyModeSelection;

SourceAddress;

DestinationAddress;

MulticastGroup\_ID;

MSDUlength;

CQI;

DataSequenceNumber;

DataRate;

TxRangingCounter;

RxRangingCounter;

IeList;

AoaAzimuth;

AoaElevation;

AoaPresent;

Rssi;

CyclicSuperframeStructure;

}

The primitive parameters are defined in Table 64.

Table —MCPS-DATA.indication parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Valid range** | **Description** |
| SourceAddress | MAC address | PD specific | The address of the PD from which the frame originated. |
| DestinationAddress | MAC address | PD specific | Address of the PD for which the frame is intended. |
| MulticastGroup\_ID | Integer |  | Group ID of PAC network. |
| MSDUlength | Integer | TBD | MDSU length in octets. |
| CQI | Integer | 0 – 4 | **CQI value measured during reception of the PSDU. TBD** |
| DataSequenceNumber | Integer | TBD | Data number sequence of the received data frame. |
| DataRate | Integer | TBD | Indicates the data rate. |
| RxRangingCounter | Unsigned Integer | 0 to (232 −1)  TBD | This parameter is present only when phyModeSelection is UWB.  The timestamp of the received data frame |
| TxRangingCounter | Unsigned Integer | TBD | This parameter is present only when phyModeSelection is UWB.  If the received frame had the AR bit set, then this is the timestamp of the transmitted acknowledgement frame, otherwise this parameter is invalid. |
| IeList | Set if IEs as described in  Array of integers (IEs) | As described in | This parameter is present only when phyModeSelection is UWB.  Reports the IEs received including:  RRRT IE, RRTD IE, RRTI IE, RPRT IE, RRTM IE, RCDT IE and RTOF IE |
| AoaAzimuth | Float | -π to +π | This parameter is present only when phyModeSelection is UWB.  AOA of the received signal in azimuth measured in radians, relative to some PD specific axis defined by its antenna arrangement or orientation. This parameter is valid only when the AoaPresent parameter is either AZIMUTH or BOTH. |
| AoaElevation | Float | -π to +π | This parameter is present only when phyModeSelection is UWB.  AOA of the received signal in elevation measured in radians. This parameter is valid only when the AoaPresent parameter is either ELEVATION or BOTH. |
| AoaPresent | Enumeration | NONE,  BOTH,  AZIMUTH,  ELEVATION | This parameter is present only when phyModeSelection is UWB.  Indicates validity of AoaAzimuth and AoaElevation parameter. Where AOA is not supported this parameter value shall be NONE. |
| Rssi | Integer | 0 to 255 | This parameter is present only when phyModeSelection is UWB.  The received signal strength for received frame. This is a measure of the RF power level at the antenna based on the gain setting in the RX chain and the measured signal level in the channel. For the UWB PHY the RSSI value is measured during the frame Preamble and locked when a valid SFD is detected. A value of zero indicates that RSSI measurement is not supported or was not measured for this frame. |
| CyclicSuperframeStructure | Cyclic-superframe structure descriptor | As defined in Table 34 | Indicates the structure of cyclic-superframe |

Cyclic-superframe structure descriptors are defined in Table 34.

**<<<< End of the proposed text changes >>>>**