The following equations define the MAC slot boundaries, using attributes defined in the MIB, which are such that they compensate for implementation timing variations. The starting reference of these slot boundaries is again the end of the last symbol of the previous frame on the medium.

TxSIFS = SIFS – aRxTxTurnaroundTime

TxPIFS = TxSIFS + aSlotTime

TxDIFS = TxSIFS + 2 x aSlotTime.

The tolerances are specified in the PHY MIB, and shall only apply to the SIFS specification, so that tolerances shall not accumulate.

9.3 PCF

The PCF provides contention-free frame transfer. The PC shall reside in the AP. It is an option for an AP to be able to become the PC. All STAs inherently obey the medium access rules of the PCF, because these rules are based on the DCF, and they set their NAV at the beginning of each CFP. The operating characteristics of the PCF are such that all STAs are able to operate properly in the presence of a BSS in which a PC is operating, and, if associated with a point-coordinated BSS, are able to receive all frames sent under PCF control. It is also an option for a STA to be able to respond to a contention free poll (CF-Poll) received from a PC. A STA that is able to respond to CF-Polls is referred to as being CF-Pollable, and may request to be polled by an active PC. CF-Pollable STAs and the PC do not use RTS/CTS in the CFP. When polled by the PC, a CF-Pollable STA may transmit only one MPDU, which can be to any destination (not just to the PC), and may “piggyback” the acknowledgment of a frame received from the PC using particular data frame subtypes for this transmission. If the data frame is not in turn acknowledged, the CF-Pollable STA shall not retransmit the frame unless it is polled again by the PC, or it decides to retransmit during the CP. If the addressed recipient of a CF transmission is not CF-Pollable, that STA acknowledges the transmission using the DCF acknowledgment rules, and the PC retains control of the medium. A PC may use contention-free frame transfer solely for delivery of frames to STAs, and never to poll non-CF-Pollable STAs.

A PC may perform a backoff on retransmission of an unacknowledged frame during the CFP. A PC that is maintaining a polling list may retry the unacknowledged frame the next time the particular AID is at the top of the polling list.

A PC may retransmit an unacknowledged frame during the CFP after a PIFS time.

When more than one point-coordinated BSS is operating on the same PHY channel in overlapping space, the potential exists for collisions between PCF transfer activities by the independent PCs. The rules under which multiple, overlapping point-coordinated BSSs may coexist are presented in 9.3.3.2. As shown in Figure 47, the PCF is built on top of the CSMA/CA-based DCF, by utilizing the access priority provisions provided by this scheme. An active PC shall be located at an AP, which restricts PCF operation to infrastructure networks. PCF is activated at a PC-capable AP by setting the CFPMaxDuration parameter in the CF Parameter Set of the MLMEStart.request to a non-zero value.

Data frames sent during under the DCF shall use the data subtypes Data or Null Function. Data frames sent by, or in response to polling by, the PC during the CFP shall use the appropriate data subtypes based upon the usage rules:

— Data+CF-Poll, Data+CF-Ack+CF-Poll, CF-Poll, and CF-Ack+CF-Poll shall only be sent by a PC.
— Data, Data+CF-Ack, Null Function, and CF-Ack may be sent by a PC or by any CF-Pollable STA.

STAs receiving Data type frames shall only consider the frame body as the basis of a possible indication to LLC, if the frame is of subtype Data, Data+CF-Ack, Data+CF-Poll, or Data+CF-Ack+CF-Poll. CF-Pollable STAs shall interpret all subtype bits of received Data type frames for CF purposes, but shall only inspect the frame body if the frame is of subtype Data, Data+CF-Ack, Data+CF-Poll, or Data+CF-Ack+CF-Poll.

9.3.1 CFP structure and timing

The PCF controls frame transfers during a CFP. The CFP shall alternate with a CP, when the DCF controls frame transfers, as shown in Figure 59. Each CFP shall begin with a Beacon frame that contains a DTIM element (hereafter referred to as a "DTIM"). The CFPs shall occur at a defined repetition rate, which shall be synchronized with the beacon interval as specified in the following paragraphs.

The PC generates CFPs at the *contention-free repetition rate* (CFPRate), which is defined as a number of DTIM intervals. The PC shall determine the CFPRate (depicted as a repetition interval in the illustrations in Figure 59 and Figure 60) to use from the CFPRate parameter in the CF Parameter Set. This value, in units of DTIM intervals, shall be communicated to other STAs in the BSS in the CFPPeriod field of the CF Parameter Set element of Beacon frames. The CF Parameter Set element shall only be present in Beacon and Probe Response frames transmitted by STAs containing an active PC.

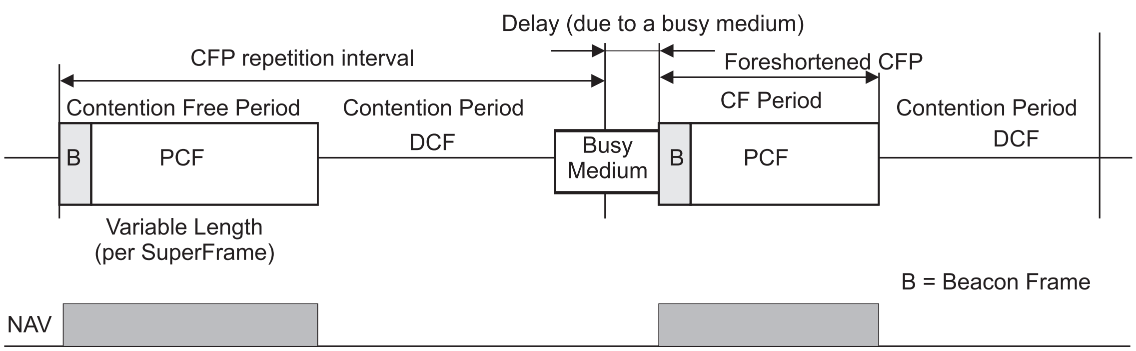


Figure 59—CFP/CP alternation

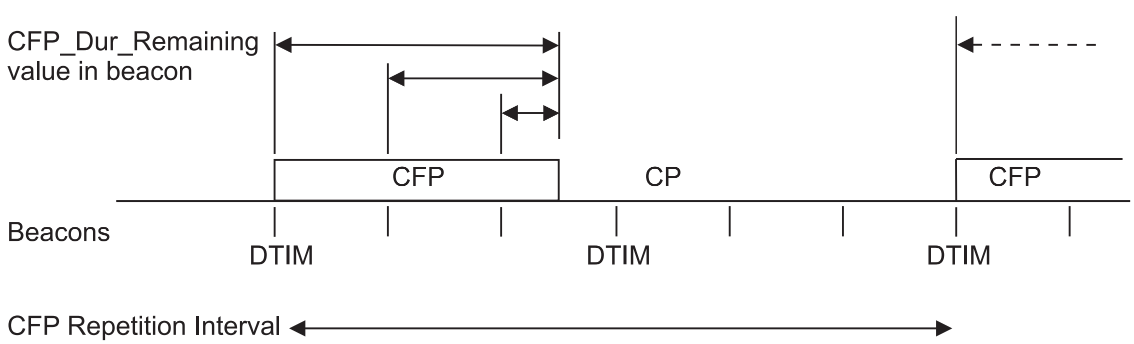


Figure 60—Beacons and CFPs

The length of the CFP is controlled by the PC, with maximum duration specified by the value of the CFP-MaxDuration Parameter in the CF Parameter Set at the PC. Neither the maximum duration nor the actual duration (signaled by transmission of a Control frame of subtype CF-End or CF-End+ACK by the PC) is constrained to be a multiple of the beacon interval. If the CFP duration is greater than the beacon interval, the PC shall transmit beacons at the appropriate times during the CFP (subject to delay due to traffic at the nominal times, as with all beacons). The CF Parameter Set element in all beacons at the start of, or within, a CFP shall contain a nonzero value in the CFPDurRemaining field. This value, in units of TU, shall specify the maximum time from the transmission of this beacon to the end of this CFP. The value of the CFPDurRemaining field shall be zero in beacons sent during the CP. An example of these relationships is illustrated in Figure 60, which shows a case where the CFP is two DTIM intervals, the DTIM interval is three beacon intervals, and the aCFPMaxDuration value is approximately 2.5 beacon intervals.

The PC may terminate any CFP at or before the aCFPMaxDuration, based on available traffic and size of the polling list. Because the transmission of any beacon may be delayed due to a medium busy condition at the nominal beacon transmission time, a CFP may be foreshortened by the amount of the delay. In the case of a busy medium due to DCF traffic, the beacon shall be delayed for the time required to complete the current DCF frame exchange. In cases where the beacon transmission is delayed, the CFPDurRemaining value in the beacon at the beginning of the CFP shall specify a time that causes the CFP to end no later than TBTT plus the value of aCFPMaxDuration. This is illustrated in Figure 61.

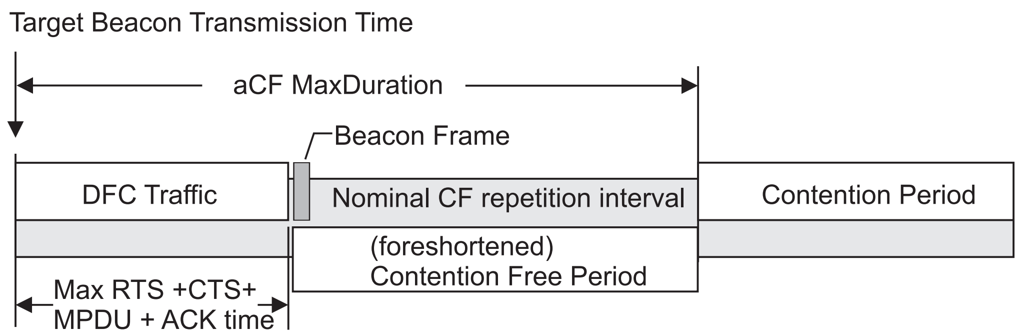


Figure 61—Example of delayed beacon and foreshortened CFP

9.3.2 PCF access procedure

The contention-free transfer protocol is based on a polling scheme controlled by a PC operating at the AP of the BSS. The PC gains control of the medium at the beginning of the CFP and attempts to maintain control for the entire CFP by waiting a shorter time between transmissions than the STAs using the DCF access procedure. All STAs in the BSS (other than the PC) set their NAVs to the CFPMaxDuration value at the nominal start time of each CFP. This prevents most contention by preventing non-polled transmissions by STAs whether or not they are CF-Pollable. Acknowledgment of frames sent during the CFP may be accomplished using Data+CF-ACK, CF-ACK, Data+CF-ACK+CF-Poll (only on frames transmitted by the PC), or CF-ACK+CF-Poll (only on frames transmitted by the PC) frames in cases where a Data (or Null) frame immediately follows the frame being acknowledged, thereby avoiding the overhead of separate ACK Control frames. Non-CF-Pollable or unpolled CF-Pollable STAs acknowledge frames during the CFP using the DCF ACK procedure.

9.3.2.1 Fundamental access

At the nominal beginning of each CFP, the PC shall sense the medium. When the medium is determined to be idle for one PIFS period, the PC shall transmit a Beacon frame containing the CF Parameter Set element and a DTIM element.

After the initial beacon frame, the PC shall wait for at least one SIFS period then transmit one of the following: a data frame, a CF-Poll frame, a Data+CF-Poll frame, or a CF-End frame. If the CFP is null, i.e., there is no traffic buffered and no polls to send at the PC, a CF-End frame shall be transmitted immediately after the initial beacon.

STAs receiving directed, error-free frames from the PC are expected to respond after a SIFS period, in accordance with the transfer procedures defined in 9.3.3. If the recipient STA is not CF-Pollable, the response to receipt of an error-free data frame shall always be an ACK frame.

9.3.2.2 NAV operation during the CFP

The mechanism for handling the NAV during the CFP is designed to facilitate the operation of overlapping CFP coordinated infrastructure BSSs. The mechanism by which infrastructure BSSs coordinate their CFPs is beyond the scope of this standard.

Each STA, except the STA with the PC, shall preset its NAV to the CFPMaxDuration value (obtained from the CF Parameter Set element in beacons from this PC) at each target beacon transmission time (TBTT) (see Clause 11) at which a CFP is scheduled to start (based on the CFPPeriod field in the CF Parameter Set element of the Beacon frames from this PC). Each non-PC STA shall update its NAV using the CFDurRemaining value in any error-free CF Parameter Set element of the Beacon frame that the STA receives. This includes CFDurRemaining values in CF Parameter Set elements from Beacon frames received from other (overlapping) BSSs.

These actions prevent STAs from taking control of the medium during the CFP, which is especially important in cases where the CFP spans multiple medium-occupancy intervals, such as dwell periods of an FH PHY. This setting of the NAV also reduces the risk of hidden STAs determining the medium to be idle for a DIFS period during the CFP and possibly corrupting a transmission in progress.

A STA joining a BSS operating with a PC shall use the information in the CFDurRemaining element of the CF parameter set of any received Beacon or Probe Response frames to update its NAV prior to initiating any transmissions.

The PC shall transmit a CF-End or CF-End+ACK frame at the end of each CFP. A STA that receives either of these frames, from any BSS, shall reset its NAV.

9.3.3 PCF transfer procedure

Frame transfers under the PCF typically consist of frames alternately sent from the AP/PC and sent to the AP/PC. During the CFP, the ordering of these transmissions, and the STA allowed to transmit frames to the PC at any given point in time, shall be controlled by the PC. Figure 62 depicts a frame transfer during a typical CFP. The rules under which this frame transfer takes place are detailed in the following subclauses.

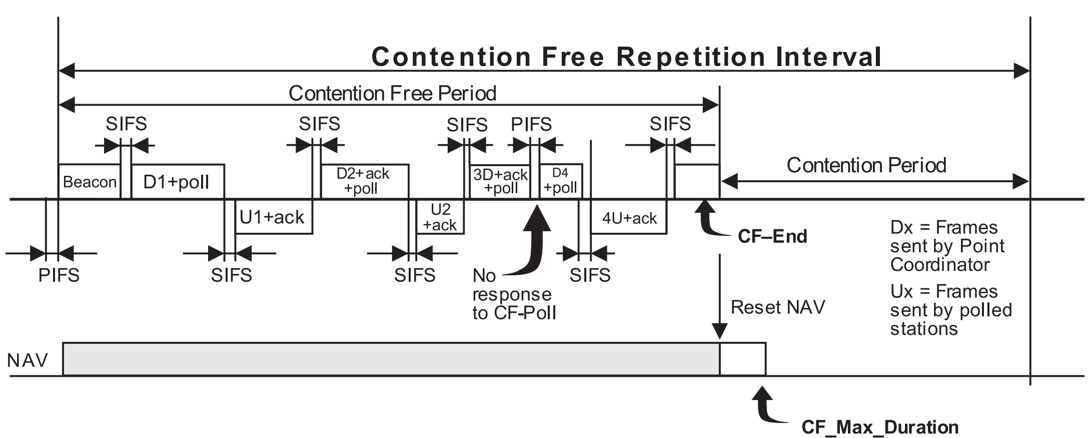


Figure 62—Example of PCF frame transfer

In a STA having an FH PHY, control of the channel is lost at a dwell time boundary. It is required that the current MPDU transmission and the accompanying acknowledgment of the MPDU be transmitted before the dwell time boundary. After having been polled by the PC, if there is not enough time remaining in the dwell to allow transmission of the MPDU plus the acknowledgment, the STA shall defer the transmission of the MPDU and shall transmit a Null frame or CF-ACK frame. The short retry counter and long retry counter for the MSDU shall not be affected.

In a STA having an FH PHY, the PC shall not transmit a CF-Poll to a STA if there is insufficient time remaining before the dwell boundary for the STA to respond with a Null frame or CF-ACK frame.

9.3.3.1 PCF transfers when the PCF STA is transmitter or recipient

The PC shall transmit frames between the Beacon that starts the CFP and the CF-End using the SIFS except in cases where a transmission by another STA is expected by the PC and a SIFS period elapses without the receipt of the expected transmission. In such cases the PC may send its next pending transmission as soon as one PIFS after the end of its last transmission. This permits the PC to retain control of the medium in the presence of an overlapping BSS. The PC may transmit any of the following frame types to CF-Pollable STAs:

— Data, used to send data from the PC when the addressed recipient is not being polled and there is no previous frame to acknowledge;

— Data+CF-ACK, used to send data from the PC when the addressed recipient is not being polled and the PC needs to acknowledge the receipt of a frame received from a CF-Pollable STA a SIFS period before starting this transmission;

— Data+CF-Poll, used to send data from the PC when the addressed recipient is the next STA to be permitted to transmit during this CFP and there is no previous frame to acknowledge;

— Data+CF-ACK+CF-Poll, used to send data from the PC when the addressed recipient is the next STA to be permitted to transmit during this CFP and the PC needs to acknowledge the receipt of a frame received from a CF-Pollable STA a SIFS period before starting this transmission;

— CF-Poll, used when the PC is not sending data to the addressed recipient, but the addressed recipient is the next STA to be permitted to transmit during this CFP and there is no previous frame to acknowledge;

— CF-ACK+CF-Poll, used when the PC is not sending data to the addressed recipient but the addressed recipient is the next STA to be permitted to transmit during this CFP and the PC needs to acknowledge the receipt of a frame from a CF-Pollable STA a SIFS period before starting this transmission;

— CF-ACK, used when the PC is not sending data to, or polling, the addressed recipient, but the PC needs to acknowledge receipt of a frame from a CF-Pollable STA a SIFS period before starting this transmission (useful when the next transmission by the PC is a management frame, such as a beacon); or

— Any management frame that is appropriate for the AP to send under the rules for that frame type.

The PC may transmit data or management frames to non-CF-Pollable, non-power-save STAs during the CFP. These STAs shall acknowledge receipt with ACK frames after a SIFS, as with the DCF. The PC may also transmit broadcast or multicast frames during the CFP. Because the Beacon frame that initiates the CFP contains a DTIM element, if there are associated STAs using power-save mode, the broadcasts and multicasts buffered shall be sent immediately after any beacon containing a TIM element with a DTIM count field with a value of 0.

A CF-Pollable STA that receives a directed data frame with any of subtype that includes CF-Poll may transmit one data frame a SIFS period after receiving the CF-Poll. CF-Pollable STAs shall ignore, but not reset, their NAV when performing transmissions in response to a CF-Poll.

Non-CF-Pollable STAs that receive a directed frame during the CFP shall transmit an ACK, but shall not reset their NAV.

For frames that require MAC-level acknowledgment, CF-Pollable STAs that received a CF-Poll (of any type) may perform this acknowledgment using the Data+CF-ACK subtype in the response to the CF-Poll. For example, the U1 frame in Figure 62 contains the acknowledgment to the preceding D1 frame. Also the D2 frame contains the acknowledgment to the preceding U1 frame. The PC may use the CF-ACK subtypes to acknowledge a received frame even if the data frame sent with the CF-ACK subtype is addressed to a different STA than the one being acknowledged. CF-Pollable STAs that are expecting an acknowledgment shall interpret the subtype of the frame (if any) sent by the PC a SIFS period after that STA's transmission to the PC. If a frame that requires MAC level acknowledgment is received by a non-CF-Pollable STA, that STA shall not interpret the CF-Poll indication (if any), and shall acknowledge the frame by sending an ACK Control frame after a SIFS period.

The lengths of the frames may be variable, only bounded by the frame and/or fragment length limitations that apply for the BSS. If a CF-Pollable STA does not respond to a CF-Poll (of any type) within the SIFS period following a transmission from the PC, or a non-CF-Pollable STA does not return the ACK frame within a SIFS period following a transmission from the PC that requires acknowledgment, then the PC shall resume control and may transmit its next frame after a PIFS period from the end of the PC's last transmission.

A CF-Pollable STA shall always respond to a CF-Poll directed to its MAC address and received without error. If the STA has no frame to send when polled, the response shall be a Null frame. If the STA has no frame to send when polled, but an acknowledgment is required for the frame that conveyed the CF-Poll, the response shall be a CF-ACK (no data) frame. The null response is required to permit a "no-traffic" situation to be distinguished from a collision between overlapping PCs.

The CFP shall end when the CFPDurRemaining time has elapsed since the Beacon frame originating the CFP or when the PC has no further frames to transmit nor STAs to poll. In either case, the end of the CFP shall be signaled by the transmission of a CF-End by the PC. If there is a received frame that requires acknowledgment at the time the CF-End is to be transmitted, the PC shall transmit a CF-End+ACK frame instead. All STAs of the BSS receiving a CF-End or CF-End+ACK shall reset their NAVs so they may attempt to transmit during the CP.

9.3.3.2 Operation with overlapping point-coordinated BSSs

Because the PCF operates without the CSMA/CA contention window randomization and backoff of the DCF, there is a risk of repeated collisions if multiple, overlapping, point-coordinated BSSs are operating on the same PHY channel, and their CFP-Rates and beacon intervals are approximately equal. To minimize the risk of significant frame loss due to CF collisions, the PC shall use a DIFS plus a random backoff delay (with CW in the range of 1 to aCWmin) to start a CFP when the initial beacon is delayed because of deferral due to a busy medium. The PC may optionally use this backoff during the CFP prior to retransmitting an unacknowledged, directed data or management frame.

To further reduce the susceptibility to inter-PC collisions, the PC shall require the medium be determined as being idle for a DIFS period plus a random (over a range of 1 to aCWmin) number of slot times once every aMediumOccupancyLimit TU during the CFP. This results in loss of control of the medium to overlapping BSS or hidden STA traffic, because the STAs in this BSS are prevented from transmitting by their NAV setting to CFPMaxDuration or CFPDurRemaining. For operation of the PCF in conjunction with an FH PHY, aMediumOccupancyLimit shall be set equal to the dwell time. For operation in conjunction with other PHY types, aMediumOccupancyLimit may be set equal to CFPMaxDuration, unless extra protection against PCF collisions is desired. The aMediumOccupancyLimit is also useful for compliance in regulatory domains that impose limits on continuous transmission time by a single STA as part of a spectrum etiquette.

9.3.3.3 CFPMaxDuration limit

The value of CFPMaxDuration shall be limited to allow coexistence between contention and contention-free traffic.

The minimum value for CFPMaxDuration is two times MaxMPDUTime plus the time required to send the initial Beacon frame and the CF-End frame of the CFP. This may allow sufficient time for the AP to send one data frame to a STA, while polling that STA, and for the polled STA to respond with one data frame.

The maximum value for CFPMaxDuration is the duration of (BeaconPeriod × DTIMPeriod × CFPRate) minus (MaxMPDUTime plus (2 × aSIFSTime) plus (2 × aSlotTime) plus (8 × ACKSize)), expressed in microseconds, when operating with a contention window of aCWmin. This allows sufficient time to send at least one data frame during the CP.

9.3.3.4 Contention-Free usage rules

A PC may send broadcast or multicast frames, and directed data or management frames to any active STA, as well as to CF-Pollable Power Save STAs. During the CFP, CF-Pollable STAs shall acknowledge after a SIFS period, the receipt of each Data+CF-Poll frame or Data+CF-ACK+CF-Poll frame using Data+CF-Ack or CF-Ack (no data) frames, the receipt of each CF_Poll (no data) using Data or Null (no data), and the receipt of all other data and management frames using ACK Control frames. Non-CF-Pollable STAs shall acknowledge receipt of data and management frames using ACK Control frames sent after a SIFS period. This non-CF-Pollable operation is the same as that already employed by such STAs for DCF operation.

When polled by the PCF (Data+CF-Poll, Data+CF-ACK+CF-Poll, CF-Poll, or CF-ACK+CF-Poll) a CF-Pollable STA may send one data frame to any destination. Such a frame directed to or through the PC STA shall be acknowledged by the PC, using the CF-ACK indication (Data+CF-ACK, Data+CF-ACK+CF-Poll, CF-ACK, CF-ACK+CF-Poll, or CF-End+ACK) sent after a SIFS. Such a frame directed to a non-CF-Pollable STA shall be acknowledged using an ACK Control frame sent after a SIFS period. A polled CF-Pollable STA with neither a data frame nor an acknowledgment to send shall respond by transmitting a Null frame after a SIFS period. A polled CF-Pollable STA with insufficient time before the end of the CFP or current medium occupancy limit, to send its queued MPDU and receive an acknowledgment, shall respond by transmitting a Null frame, or a CF-ACK frame if polled using Data+CF-Poll or Data+CF-ACK+CF-Poll, after a SIFS period. The CF-Pollable STA may set the More Data bit in its response to permit the PC to distinguish between an empty STA queue and a response due to insufficient time to transfer an MPDU.

The PC shall not issue frames with a subtype that includes CF-Polls if insufficient time remains in the current CFP to permit the polled STA to transmit a data frame containing a minimum length MPDU.

9.3.4 Contention-Free polling list

If the PC supports use of the CFP for inbound frame transfer as well as for frame delivery, the PC shall maintain a "polling list" for use in selecting STAs that are eligible to receive CF-Polls during CFPs. The polling list functional characteristics are defined below. If the PC supports the use of the CFP solely for frame delivery, the PC does not require a polling list, and shall never generate data frames with a subtype that includes CF-Poll. The form of contention-free support provided by the PC is identified in the Capability Information field of Beacon, Association Response, Reassociation Response, and Probe Response management frames, which are sent from APs. Any such frames sent by STAs, as in noninfrastructure networks, shall always have these bits set to zero.

The polling list is used to force the polling of CF-Pollable STAs, whether or not the PC has pending traffic to transmit to those STAs. The polling list may be used to control the use of Data+CF-Poll and Data+CF-ACK+CF-Poll types for transmission of data frames being sent to CF-Pollable STAs by the PC. The polling list is a *logical* construct, which is not exposed outside of the PC. A minimum set of polling list maintenance techniques are required to ensure interoperability of arbitrary CF-Pollable STAs in BSSs controlled by arbitrary access points with active PCs. APs may also implement additional polling list maintenance techniques that are outside the scope of this standard.

9.3.4.1 Polling list processing

The PC shall send a CF-Poll to at least one STA during each CFP when there are entries in the polling list. During each CFP, the PC shall issue polls to a subset of the STAs on the polling list in order by ascending AID value.

While time remains in the CFP, the delivery of all CF frames has been completed and all STAs on the polling list have been polled, the PC may generate one or more CF-Polls to *any* STAs on the polling list. While time remains in the CFP, the delivery of all CF frames has been completed and all STAs on the polling list have been polled, the PC *may* send data or management frames to *any* STAs.

In order to gain maximum efficiency from the CFP, and the ability to piggyback acknowledgments on successor data frames in the opposite direction, the PC should generally use Data+CF-Poll and Data+CF-ACK+CF-Poll types for each data frame transmitted while sufficient time for the potential response to the CF-Poll remains in the CFP.

9.3.4.2 Polling list update procedure

A STA indicates its CF-Pollability using the CF-Pollable subfield of the Capability Information field of Association Request and Reassociation Request frames. If a STA desires to change the PC's record of CF-Pollability, that STA shall perform a reassociation. During association, a CF-Pollable STA may also request to be placed on the polling list for the duration of its association, or by setting the CF-Poll Request subfield in the Capability Information field. If a CF-Pollable STA desires never to be placed on the polling list, that STA shall perform Association with both the CF-Pollable subfield false and CF-Poll Request subfield true. Never being polled is useful for CF-Pollable STAs that normally use power-save mode, permitting them to receive buffered traffic during the CFP (since they have to be awake to receive the DTIM that initiated the CFP), but not requiring them to stay awake to receive CF-Polls when they have no traffic to send. If a STA desires to be removed from the polling list, that STA shall perform a reassociation.

CF-Pollable STAs that are not on the polling list, but did not request never to be polled during their most recent association, may be dynamically placed on the polling list by the PC to handle bursts of frame transfer activity by that STA.

9.4 Fragmentation

The MAC may fragment and reassemble directed MSDUs or MMPDUs. The fragmentation and defragmentation mechanisms allow for fragment retransmission.

The length of a fragment MPDU shall be an equal number of octets for all fragments except the last, which may be smaller. The length of a fragment MPDU shall always be an even number of octets, except for the last fragment of an MSDU or MMPDU, which may be either an even or an odd number of octets. The length of a fragment shall never be larger than aFragmentationThreshold unless WEP is invoked for the MPDU. If WEP is active for the MPDU, then the MPDU shall be expanded by IV and ICV (see 8.2.5); this may result in a fragment larger than aFragmentationThreshold.

When data is to be transmitted, the number of octets in the fragment (before WEP processing) shall be determined by aFragmentationThreshold and the number of octets in the MPDU that have yet been assigned to a fragment at the instant the fragment is constructed for the first time. Once a fragment is transmitted for the first time, its frame body content and length shall be fixed until it is successfully delivered to the immediate receiving STA. A STA shall be capable of receiving fragments of arbitrary length.

If a fragment requires retransmission, its frame body content and length shall remain fixed for the lifetime of the MSDU or MMPDU at that STA. After a fragment is transmitted once, contents and length of that fragment are not allowed to fluctuate to accommodate the dwell time boundaries. Each fragment shall contain a Sequence Control field, which is comprised of a sequence number and fragment number. When a STA is transmitting an MSDU or MMPDU, the sequence number shall remain the same for all fragments of that MSDU or MMPDU. The fragments shall be sent in order of lowest fragment number to highest fragment number, where the fragment number value starts at zero, and increases by one for each successive fragment. The Frame Control field also contains a bit, the More Fragments bit, that is equal to zero to indicate the last (or only) fragment of the MSDU or MMPDU.

The source STA shall maintain a transmit MSDU timer for each MSDU being transmitted. The attribute aMaxTransmitMSDULifetime specifies the maximum amount of time allowed to transmit an MSDU. The timer starts on the attempt to transmit the first fragment of the MSDU. If the timer exceeds aMaxTransmit-MSDULifetime, then all remaining fragments are discarded by the source STA and no attempt is made to complete transmission of the MSDU.

9.5 Defragmentation

Each fragment contains information to allow the complete MSDU or MMPDU to be reassembled from its constituent fragments. The header of each fragment contains the following information that is used by the destination STA to reassemble the MSDU or MMPDU:

- Frame type
- Address of the sender, obtained from the Address2 field
- Destination address
- *Sequence Control field:* This field allows the destination STA to check that all incoming fragments belong to the same MSDU or MMPDU, and the sequence in which the fragments should be reassembled. The sequence number within the Sequence Control field remains the same for all fragments of an MSDU or MMPDU, while the fragment number within the Sequence Control field increments for each fragment.
- *More Fragments indicator:* Indicates to the destination STA that this is not the last fragment of the MSDU or MMPDU. Only the last or sole fragment of the MSDU or MMPDU shall have this bit set to zero. All other fragments of the MSDU or MMPDU shall have this bit set to one.

The destination STA shall reconstruct the MSDU or MMPDU by combining the fragments in order of fragment number portion of the Sequence Control field. If WEP has been applied to the fragment, it shall be decrypted before the fragment is used for defragmentation of the MSDU or MMPDU. If the fragment with the More Fragments bit set to zero has not yet been received, then the destination STA knows that the MSDU or MMPDU is not yet complete. As soon as the STA receives the fragment with the More Fragments bit set to zero, the STA knows that no more fragments may be received for the MSDU or MMPDU.

All STAs shall support the concurrent reception of fragments of at least three MSDUs or MMPDUs. Note that a STA receiving more than three fragmented MSDUs or MMPDUs concurrently may experience a significant increase in the number of frames discarded.

The destination STA shall maintain a Receive Timer for each MSDU or MMPDU being received, for a minimum of three MSDUs or MMPDUs. The STA may implement additional timers to be able to receive additional concurrent MSDUs or MMPDUs. The receiving STA shall discard all fragments that are part of an MSDU or MMPDU for which a timer is not maintained. There is also an attribute, aMaxReceiveLifetime, that specifies the maximum amount of time allowed to receive an MSDU. The receive MSDU or MMPDU timer starts on the reception of the first fragment of the MSDU or MMPDU. If the receive MSDU timer exceeds aMaxReceiveLifetime, then all received fragments of this MSDU or MMPDU are discarded by the destination STA. If additional fragments of a directed MSDU or MMPDU are received after its aMaxReceiveLifetime is exceeded, those fragments shall be acknowledged and discarded.

To properly reassemble MPDUs into an MSDU or MMPDU, a destination STA shall discard any duplicated fragments received. A STA shall discard duplicate fragments as described in 9.2.9, duplicate detection. However, an acknowledgment shall be sent in response to a duplicate fragment of a directed MSDU.

9.6 Multirate support

Some PHYs have multiple data transfer rate capabilities that allow implementations to perform dynamic rate switching with the objective of improving performance. The algorithm for performing rate switching is beyond the scope of this standard, but in order to ensure coexistence and interoperability on multirate-capable PHYs, this standard defines a set of rules that shall be followed by all STAs.

All Control frames shall be transmitted at one of the rates in the BSSBasicRateSet (see 10.3.10.1), or at one of the rates in the PHY mandatory rate set so they will be understood by all STAs.

All frames with multicast and broadcast RA shall be transmitted at one of the rates included in the BSSBasicRateSet, regardless of their type.

Data and/or management MPDUs with a unicast immediate address shall be sent on any supported data rate selected by the rate switching mechanism (whose output is an internal MAC variable called MACCurrentRate, defined in units of 500 kbit/s, which is used for calculating the Duration/ID field of each frame). A STA shall not transmit at a rate that is known not to be supported by the destination STA, as reported in the supported rates element in the management frames. For frames of type Data+CF-ACK, Data+CF-Poll+CF-ACK and CF-Poll+CF-ACK, the rate chosen to transmit the frame must be supported by both the addressed recipient STA and the STA to which the ACK is intended.

Under no circumstances shall a STA initiate transmission of a data or management frame at a data rate higher than the greatest rate in the OperationalRateSet, a parameter of the MLME-JOIN.request primitive.

In order to allow the transmitting STA to calculate the contents of the Duration/ID field, the responding STA shall transmit its Control Response frame (either CTS or ACK) at the same rate as the immediately previous frame in the frame exchange sequence (as defined in 9.7), if this rate belongs to the PHY mandatory rates, or else at the highest possible rate belonging to the PHY rates in the BSSBasicRateSet.

9.7 Frame exchange sequences

The allowable frame exchange sequences are summarized in Table 21 and Table 22. A legend applicable to both tables follows Table 22.

Table 21—Frame sequences

Sequence	Frames in sequence	Usage
Data(bc/mc)	1	Broadcast or multicast MSDU
Mgmt(bc)	1	Broadcast MMPDU
{RTS – CTS –} [Frag – ACK –] Last – ACK	2	Directed MSDU or MMPDU
PS-Poll – ACK	2	Deferred PS-POLL response
PS-Poll – [Frag – ACK –] Last – ACK	3	Immediate PS-POLL response
DTIM(CF) – [≤CF-Sequence> –] {CF-End}	2 or more	Start of CFP
[<CF-Sequence> –] {CF-End}	2 or more	Continuation of CFP after missing ACK or medium occupancy boundary

Table 22—CF frame sequences

CF frame sequence	Frames in sequence	Usage
Beacon(CF)	1	Beacon during CFP
Data(bc/mc)	1	Broadcast or multicast MSDU
Mgmt(bc)	1 or 2	Broadcast MMPDU
Mgmt(dir) – ACK	2 or 3	Directed MMPDU
Data(dir)+CF-Poll{+CF-Ack} – Data(dir)+CF-Ack – {CF-Ack(no data)}	2	Poll and ACK sent with MPDUs
Data(dir)+CF-Poll{+CF-Ack} – CF-Ack(no data)	2	Poll of STA with empty queue, insufficient time for queued MPDU, or too little time remaining before a dwell or medium occupancy boundary to send a queued frame
CF-Poll(no data){+CF-Ack} – Data(dir) – {CF-Ack(no data)}	2	Separate poll, ACK sent with MPDU
CF-Poll(no data){+CF-Ack} – Data(dir) – ACK	3	Polled STA sends to STA in BSS
CF-Poll(no data){+CF-Ack} – Null(no data)	2	Separate poll, STA queue empty, or insufficient time for queued MPDU or too little time remaining before a dwell or medium occupancy boundary to send a queued frame
Data(dir){+CF-Ack} – ACK	2	ACK if not CF-Pollable or not polled

LEGEND (For Table 21 and Table 22)

1—Items enclosed in brackets "[…]" may occur zero or more times in the sequence.

2—Items enclosed in braces "{…}" may occur zero or one time in the sequence.

3—An isolated hyphen "-" represents a SIFS interval separating the pair of frames.

4—"Data(bc/mc)" represents any frame of type Data with a broadcast or multicast address in the Address1 field.

5—"Mgmt(bc)" represents any Management type frame with a broadcast address in the DA field.

6—"RTS" represents a Control frame of subtype RTS.

7—"CTS" represents a Control frame of subtype CTS.

8—"ACK" represents a Control frame of subtype ACK.

9—"Frag" represents an MPDU of type Data or an MMPDU of type Management with an individual address in the Address1 field that has the More Fragments field set to "1."

10—"Last" represents an MDPU of type Data or an MMPDU of type Management with an individual address in the Address1 field that has the More Fragments field set to "0."

11—"PS-Poll" represents a Control frame of subtype PS-Poll.

12—"DTIM(CF)" represents a management frame of subtype Beacon and that contains a DTIM information element with a nonzero value in the CFDurRemaining field of its Parameter Set element.

13—"CF-End" represents a Control frame of type CF-End, or (if the final frame of the immediately preceding <CF-Sequence> was a directed data or management frame requiring acknowledgment by the AP) of type CF-End+Ack.

14—"Beacon(CF)" represents a management frame of subtype Beacon with a nonzero value in the CFDurRemaining field of its CF Parameter Set element.

15—"Data(dir)" represents any MPDU of type Data with an individual address in the Address1 field.

16—"Mgmt(dir)" represents any MMPDU of type Management with an individual address in the Address1 field.

17—"CF-Ack(no data)" represents a data frame of subtype CF-ACK (no data).

18—"CF-Poll(no data)" represents a data frame of subtype CF-Poll (no data).

19—"Null(no data)" represents a data frame of subtype Null Function (no data).

20—The notation "{+CF-Ack}" indicates that the frame may or may not include a contention-free acknowledgment.

21—The notation "+CF-Ack" indicates that the frame includes a contention-free acknowledgment.

22—The notation "+CF-Poll" indicates that the frame includes a contention-free poll.

23—<CF-Sequence> represents a sequence of one or more frames sent during a CFP. A valid <CF-Sequence> shall consist of one of the frame sequences shown in Table 22. The collection of sequences of frame exchanges corresponding to the [<CF-Sequence>] from may occur in any order within the CFP.

Individual frames within each of these sequences are separated by a SIFS.

9.8 MSDU transmission restrictions

To avoid reordering MSDUs between pairs of LLC entities and/or unnecessarily discarding MSDUs, the following restrictions shall be observed by any STA that is able to concurrently process multiple outstanding MSDUs for transmission. Note that here the term "outstanding" refers to an MSDU or MMPDU that is eligible to be transmitted at a particular time. A STA may have any number (greater than or equal to one) of eligible MSDUs outstanding concurrently, subject to the restrictions below.

The STA shall ensure that no more than one MSDU or MMPDU from a particular SA to a particular individual RA is outstanding at a time. Note that a simpler, more restrictive invariant to maintain is that no more than one MSDU with a particular individual RA may be outstanding at a time.

In a STA where the optional StrictlyOrdered service class has been implemented, that STA shall ensure that there is no group-addressed (multidestination) MSDU of the StrictlyOrdered service class outstanding from the SA of any other outstanding MSDU (either directed or group-addressed). This is because a group-addressed MSDU is implicitly addressed to a collection of peer STAs that could include any individual RA.

It is recommended that the STA select a value of aMaxMSDUTransmitLifetime that is sufficiently large that the STA does not discard MSDUs due to excessive Transmit MSDU timeouts under normal operating conditions.

10. Layer management

10.1 Overview of management model

Both MAC and physical layers conceptually include management entities, called MAC sublayer management and PHY layer management entities (MLME and PLME, respectively). These entities provide the layer management service interfaces through which layer management functions may be invoked.

In order to provide correct MAC operation, a station management entity (SME) shall be present within each STA. The SME is a layer-independent entity that may be viewed as residing in a separate management plane or as residing "off to the side." The exact functions of the SME are not specified in this standard, but in general this entity may be viewed as responsible for such functions as the gathering of layer-dependent status from the various layer management entities, and similarly setting the value of layer-specific parameters. SME would typically perform such functions on behalf of general system management entities and would implement standard management protocols. Figure 11 depicts the relationship among management entities.

The various entities within this model interact in various ways. Certain of these interactions are defined explicitly within this standard, via a service access point (SAP) across which defined primitives are exchanged. Other interactions are not defined explicitly within this standard, such as the interfaces between MAC and MLME and between the PLCP and PLME, represented as double arrows within the figure. The specific manner in which these MAC and PHY management entities are integrated into the overall MAC and PHY layers is not specified within this standard.

The management SAPs within this model are the following:

— SME-MLME SAP
— SME-PLME SAP
— MLME-PLME SAP

The latter two SAPs support identical primitives, and in fact may be viewed as a single SAP (called the PLME SAP) that may be used either directly by MLME or by SME. In this fashion, the model reflects what is anticipated to be a common implementation approach in which PLME functions are controlled by the MLME (on behalf of SME). In particular, PHY implementations are not required to have separate interfaces defined other than their interfaces with the MAC and MLME.

10.2 Generic management primitives

The management information specific to each layer is represented as a management information base (MIB) for that layer. The MAC and PHY layer management entities are viewed as "containing" the MIB for that layer. The generic model of MIB-related management primitives exchanged across the management SAPs is to allow the SAP user-entity to either GET the value of a MIB attribute, or to SET the value of a MIB attribute. The invocation of a SET.request primitive may require that the layer entity perform certain defined actions.

Figure 63 depicts these generic primitives.

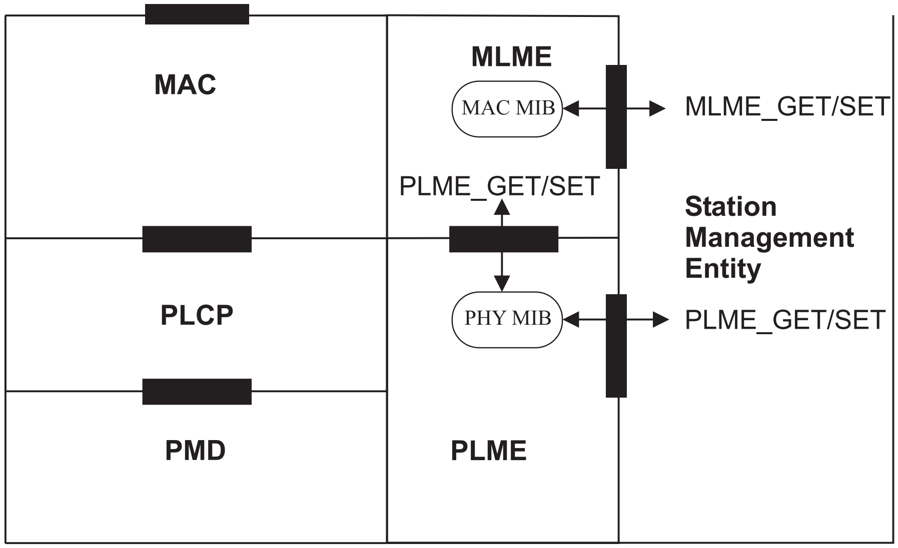


Figure 63—GET and SET operations

The GET and SET primitives in fact are represented as REQUESTs with associated CONFIRM primitives. These primitives are prefixed by MLME or PLME depending upon whether the MAC or PHY layer management SAP is involved. In the following, XX denotes MLME or PLME:

XX-GET.request (MIBattribute)
Requests the value of the given MIBattribute.

XX-GET.confirm (status, MIBattribute, MIBattributevalue)
Returns the appropriate MIB attribute value if status = "success," otherwise returns an error indication in the Status field. Possible error status values include "invalid MIB attribute" and "attempt to get write-only MIB attribute."

XX-SET.request (MIBattribute, MIBattributevalue)
Requests that the indicated MIB attribute be set to the given value. If this MIBattribute implies a specific action, then this request that the action be performed.

XX-SET.confirm (status, MIBattribute)
If status = "success," this confirms that the indicated MIB attribute was set to the requested value, otherwise returns an error condition in status field. If this MIBattribute implies a specific action, then this confirms that the action was performed. Possible error status values include "invalid MIB attribute" and "attempt to set read-only MIB attribute."

Additionally, there are certain requests (with associated confirms) that may be invoked across a given SAP which do not involve the setting or getting of a specific MIB attribute. One of these is supported by each SAP, as follows:

— XX-RESET.request: where XX is MLME or PLME as appropriate.
— XX-RESET.confirm

This service is used to initialize the management entities, the MIBs, and the datapath entities. It may include a list of attributes for items to be initialized to non-default values. The corresponding .confirm indicates success or failure of the request.

Other SAP-specific primitives are identified in 10.3.

10.3 MLME SAP interface

The services provided by the MLME to the SME are specified in this subclause. These services are described in an abstract way and do not imply any particular implementation or exposed interface. MLME SAP primitives are of the general form ACTION.request followed by ACTION.confirm. The SME uses the services provided by the MLME through the MLME SAP.

10.3.1 Power management

This mechanism supports the process of establishment and maintenance of the power management mode of a STA.

10.3.1.1 MLME-POWERMGT.request

10.3.1.1.1 Function

This primitive requests a change in the power management mode.

10.3.1.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-POWERMGT.request          (
                               PowerManagementMode,
                               WakeUp,
                               ReceiveDTIMs
                               )
```

Name	Type	Valid range	Description
PowerManagementMode	Enumeration	ACTIVE, POWER_SAVE	An enumerated type that describes the desired power management mode of the STA.
WakeUp	Boolean	True, false	When true, the MAC is forced immediately into the Awake state. This parameter has no effect if the current power management mode is ACTIVE.
ReceiveDTIMs	Boolean	True, false	When true, this parameter causes the STA to awaken to receive all DTIM frames. When false, the STA is not required to awaken for every DTIM frame.

10.3.1.1.3 When generated

This primitive is generated by the SME to implement the power savings strategy of an implementation.

10.3.1.1.4 Effect of receipt

This request sets the STA's power management parameters. The MLME subsequently issues a MLME-POWERMGT.confirm that reflects the results of the power management change request.

10.3.1.2 MLME-POWERMGT.confirm

10.3.1.2.1 Function

This primitive confirms the change in power management mode.

10.3.1.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-POWERMGT.confirm      (
                           ResultCode
                           )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, NOT_SUPPORTED	Indicates the result of the MLME-POWERMGT.request

10.3.1.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-POWERMGT.request to establish a new power management mode. It is not generated until the change has completed.

10.3.1.2.4 Effect of receipt

The SME is notified of the change of power management mode.

10.3.2 Scan

This mechanism supports the process of determining the characteristics of the available BSSs.

10.3.2.1 MLME-SCAN.request

10.3.2.1.1 Function

This primitive requests a survey of potential BSSs that the STA may later elect to try to join.

10.3.2.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SCAN.request          (
                           BSSType,
                           BSSID,
                           SSID,
                           ScanType,
                           ProbeDelay,
                           ChannelList,
                           MinChannelTime,
                           MaxChannelTime
                           )
```

Name	Type	Valid range	Description
BSSType	Enumeration	INFRASTRUCTURE, INDEPENDENT, ANY_BSS	Determines whether Infrastructure BSS, Independent BSS, or both are included in the scan.
BSSID	MACAddress	Any valid individual or broadcast MAC address	Identifies a specific or broadcast BSSID
SSID	Octet string	0–32 octets	Specifies the desired SSID or the broadcast SSID
ScanType	Enumeration	ACTIVE, PASSIVE	Indicates either active or passive scanning
ProbeDelay	Integer	N/A	Delay (in μs) to be used prior to transmitting a Probe frame during active scanning
ChannelList	Ordered set of integers	Each channel will be selected from the valid channel range for the appropriate PHY and carrier set.	Specifies a list of channels that are examined when scanning for a BSS
MinChannelTime	Integer	≥ ProbeDelay	The minimum time (in TU) to spend on each channel when scanning
MaxChannelTime	Integer	≥ MinChannelTime	The maximum time (in TU) to spend on each channel when scanning

10.3.2.1.3 When generated

This primitive is generated by the SME for a STA to determine if there are other BSSs that it may join.

10.3.2.1.4 Effect of receipt

This request initiates the scan process when the current frame exchange sequence is completed.

10.3.2.2 MLME-SCAN.confirm

10.3.2.2.1 Function

This primitive returns the descriptions of the set of BSSs detected by the scan process.

10.3.2.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SCAN.confirm          (
                           BSSDescriptionSet,
                           ResultCode
                           )
```

Name	Type	Valid range	Description
BSSDescriptionSet	Set of BSSDescriptions	N/A	The BSSDescriptionSet is returned to indicate the results of the scan request. It is a set containing zero or more instances of a BSSDescription.
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS	Indicates the result of the MLME-SCAN.confirm

Each BSSDescription consists of the following elements:

Name	Type	Valid range	Description
BSSID	MACAddress	N/A	The BSSID of the found BSS
SSID	Octet string	1–32 octets	The SSID of the found BSS
BSSType	Enumeration	INFRASTRUCTURE, INDEPENDENT	The type of the found BSS
Beacon Period	Integer	N/A	The Beacon period of the found BSS (in TU)
DTIM Period	Integer	As defined in frame format	The DTIM period of the BSS (in beacon periods)
Timestamp	Integer	N/A	The timestamp of the received frame (probe response/beacon) from the found BSS
Local Time	Integer	N/A	The value of the STA's TSF timer at the start of reception of the first octet of the timestamp field of the received frame (probe response or beacon) from the found BSS
PHY parameter set	As defined in frame format	As defined in frame format	The parameter set relevant to the PHY
CF parameter set	As defined in frame format	As defined in frame format	The parameter set for the CF periods, if found BSS supports CF mode
IBSS parameter set	As defined in frame format	As defined in frame format	The parameter set for the IBSS, if found BSS is an IBSS
CapabilityInformation	As defined in frame format	As defined in frame format	The advertised capabilities of the BSS
BSSBasicRateSet	Set of integers	1 through 127 inclusive (for each integer in the set)	The set of data rates (in units of 500 kb/s) that must be supported by all STAs that desire to join this BSS. The STAs must be able to receive at each of the data rates listed in the set.

10.3.2.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-SCAN.request to ascertain the operating environment of the STA.

10.3.2.2.4 Effect of receipt

The SME is notified of the results of the scan procedure.

10.3.3 Synchronization

This mechanism supports the process of selection of a peer in the authentication process.

10.3.3.1 MLME-JOIN.request

10.3.3.1.1 Function

This primitive requests synchronization with a BSS.

10.3.3.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-JOIN.request (
BSSDescription,
JoinFailureTimeout,
ProbeDelay.
OperationalRateSet
)

Name	Type	Valid range	Description
BSSDescription	BSSDescription	N/A	The BSSDescription of the BSS to join. The BSSDescription is a member of the set of descriptions that was returned as a result of a MLME-SCAN.request.
JoinFailureTimeout	Integer	≥1	The time limit, in units of beacon intervals, after which the join procedure will be terminated
ProbeDelay	Integer	N/A	Delay (in μs) to be used prior to transmitting a Probe frame during active scanning
OperationalRateSet	Set of integers	1 through 127 inclusive (for each integer in the set)	The set of data rates (in units of 500 kbit/s) that the STA may use for communication within the BSS. The STA must be able to receive at each of the data rates listed in the set. The OperationalRateSet is a superset of the BSSBasicRateSet advertised by the BSS.

10.3.3.1.3 When generated

This primitive is generated by the SME for a STA to establish synchronization with a BSS.

10.3.3.1.4 Effect of receipt

This primitive initiates a synchronization procedure once the current frame exchange sequence is complete. The MLME synchronizes its timing with the specified BSS based on the elements provided in the BSSDescription parameter. The MLME subsequently issues a MLME-JOIN.confirm that reflects the results.

10.3.3.2 MLME-JOIN.confirm

10.3.3.2.1 Function

This primitive confirms synchronization with a BSS.

10.3.3.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-JOIN.confirm (
ResultCode
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TIMEOUT	Indicates the result of the MLME-JOIN.request

10.3.3.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-JOIN.request to establish synchronization with a BSS.

10.3.3.2.4 Effect of receipt

The SME is notified of the results of the synchronization procedure.

10.3.4 Authenticate

This mechanism supports the process of establishing an authentication relationship with a peer MAC entity.

10.3.4.1 MLME-AUTHENTICATE.request

10.3.4.1.1 Function

This primitive requests authentication with a specified peer MAC entity.

10.3.4.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-AUTHENTICATE.request    (
                             PeerSTAAddress,
                             AuthenticationType,
                             AuthenticateFailureTimeout
                             )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the authentication process
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY	Specifies the type of authentication algorithm to use during the authentication process
AuthenticationFailure-Timeout	Integer	≥1	Specifies a time limit (in TU) after which the authentication procedure will be terminated

10.3.4.1.3 When generated

This primitive is generated by the SME for a STA to establish authentication with a specified peer MAC entity in order to permit Class 2 frames to be exchanged between the two STAs. During the authentication procedure, the SME may generate additional MLME-AUTHENTICATE.request primitives.

10.3.4.1.4 Effect of receipt

This primitive initiates an authentication procedure. The MLME subsequently issues a MLME-AUTHENTICATE.confirm that reflects the results.

10.3.4.2 MLME-AUTHENTICATE.confirm

10.3.4.2.1 Function

This primitive reports the results of an authentication attempt with a specified peer MAC entity.

10.3.4.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-AUTHENTICATE.confirm (
PeerSTAAddress,
AuthenticationType,
ResultCode
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the authentication process was attempted. This value must match the peer-STAAddress parameter specified in the corresponding MLME-AUTHENTICATE.request.
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY	Specifies the type of authentication algorithm that was used during the authentication process. This value must match the authenticationType parameter specified in the corresponding MLME-AUTHENTICATE.request.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TIMEOUT, TOO_MANY_SIMULTANEOUS_REQUESTS, REFUSED	Indicates the result of the MLME-AUTHENTICATE.request.

10.3.4.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-AUTHENTICATE.request to authenticate with a specified peer MAC entity.

10.3.4.2.4 Effect of receipt

The SME is notified of the results of the authentication procedure.

10.3.4.3 MLME-AUTHENTICATE.indication

10.3.4.3.1 Function

This primitive reports the establishment of an authentication relationship with a specific peer MAC entity.

10.3.4.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-AUTHENTICATE.indication (
PeerSTAAddress,
AuthenticationType
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the authentication relationship was established
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY	Specifies the type of authentication algorithm that was used during the authentication process

10.3.4.3.3 When generated

This primitive is generated by the MLME as a result of the establishment of an authentication relationship with a specific peer MAC entity that resulted from an authentication procedure that was initiated by that specific peer MAC entity.

10.3.4.3.4 Effect of receipt

The SME is notified of the establishment of the authentication relationship.

10.3.5 De-authenticate

This mechanism supports the process of invalidating an authentication relationship with a peer MAC entity.

10.3.5.1 MLME-DEAUTHENTICATE.request

10.3.5.1.1 Function

This primitive requests that the authentication relationship with a specified peer MAC entity be invalidated.

10.3.5.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.request    (
                                PeerSTAAddress,
                                ReasonCode
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the deauthentication process
ReasonCode	As defined in frame format.	As defined in frame format.	Specifies the reason for initiating the deauthentication procedure

10.3.5.1.3 When generated

This primitive is generated by the SME for a STA to invalidate authentication with a specified peer MAC entity in order to prevent the exchange of Class 2 frames between the two STAs. During the deauthentication procedure, the SME may generate additional MLME-DEAUTHENTICATE.request primitives.

10.3.5.1.4 Effect of receipt

This primitive initiates a deauthentication procedure. The MLME subsequently issues a MLME-DEAUTHENTICATE.confirm that reflects the results.

10.3.5.2 MLME-DEAUTHENTICATE.confirm

10.3.5.2.1 Function

This primitive reports the results of a deauthentication attempt with a specified peer MAC entity.

10.3.5.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.confirm    (
                               PeerSTAAddress,
                               ResultCode
                               )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the deauthentication process was attempted
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TOO_MANY_SIMULTANEOUS_REQUESTS	Indicates the result of the MLME-DEAUTHENTICATE.request

10.3.5.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-DEAUTHENTICATE.request to invalidate the authentication relationship with a specified peer MAC entity.

10.3.5.2.4 Effect of receipt

The SME is notified of the results of the deauthentication procedure.

10.3.5.3 MLME-DEAUTHENTICATE.indication

10.3.5.3.1 Function

This primitive reports the invalidation of an authentication relationship with a specific peer MAC entity.

10.3.5.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.indication (
                               PeerSTAAddress,
                               ReasonCode
                               )
```


Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the authentication relationship was invalidated
ReasonCode	As defined in frame format.	As defined in frame format.	Specifies the reason the deauthentication procedure was initiated

10.3.5.3.3 When generated

This primitive is generated by the MLME as a result of the invalidation of an authentication relationship with a specific peer MAC entity.

10.3.5.3.4 Effect of receipt

The SME is notified of the invalidation of the specific authentication relationship.

10.3.6 Associate

The following primitives describe how a STA becomes associated with an access point (AP).

10.3.6.1 MLME-ASSOCIATE.request

10.3.6.1.1 Function

This primitive requests association with a specified peer MAC entity that is acting as an AP.

10.3.6.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ASSOCIATE.request    (
                          PeerSTAAddress,
                          AssociateFailureTimeout,
                          CapabilityInformation,
                          ListenInterval
                          )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the association process
AssociateFailureTimeout	Integer	≥1	Specifies a time limit (in TU) after which the associate procedure will be terminated
CapabilityInformation	As defined in frame format	As defined in frame format	The operational capability definitions to be used by the MAC entity
ListenInterval	Integer	≥0	Specifies the number of beacon intervals that may pass before the STA awakens and listens for the next beacon

10.3.6.1.3 When generated

This primitive is generated by the SME when a STA wishes to establish association with an AP.

10.3.6.1.4 Effect of receipt

This primitive initiates an association procedure. The MLME subsequently issues a MLME-ASSOCIATE.confirm that reflects the results.

10.3.6.2 MLME-ASSOCIATE.confirm

10.3.6.2.1 Function

This primitive reports the results of an association attempt with a specified peer MAC entity that is acting as an AP.

10.3.6.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ASSOCIATE.confirm (
ResultCode
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, TIMEOUT, REFUSED	Indicates the result of the MLME-ASSOCIATE.request

10.3.6.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-ASSOCIATE.request to associate with a specified peer MAC entity that is acting as an AP.

10.3.6.2.4 Effect of receipt

The SME is notified of the results of the association procedure.

10.3.6.3 MLME-ASSOCIATE.indication

10.3.6.3.1 Function

This primitive reports the establishment of an association with a specific peer MAC entity.

10.3.6.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ASSOCIATE.indication (
PeerSTAAddress
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the association was established

10.3.6.3.3 When generated

This primitive is generated by the MLME as a result of the establishment of an association with a specific peer MAC entity that resulted from an association procedure that was initiated by that specific peer MAC entity.

10.3.6.3.4 Effect of receipt

The SME is notified of the establishment of the association.

10.3.7 Reassociate

The following primitives describe how a STA becomes associated with another AP.

10.3.7.1 MLME-REASSOCIATE.request

10.3.7.1.1 Function

This primitive requests a change in association to a specified new peer MAC entity that is acting as an AP.

10.3.7.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-REASSOCIATE.request        (
                                NewAPAddress,
                                ReassociateFailureTimeout,
                                CapabilityInformation,
                                ListenInterval
                                )
```

Name	Type	Valid range	Description
NewAPAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the reassociation process
ReassociateFailureTimeout	Integer	≥1	Specifies a time limit (in TU) after which the reassociate procedure will be terminated
CapabilityInformation	As defined in frame format	As defined in frame format	The operational capability definitions to be used by the MAC entity
LlistenInterval	Integer	≥0	Specifies the number of beacon intervals that may pass before the STA awakens and listens for the next beacon.

10.3.7.1.3 When generated

This primitive is generated by the SME for a STA to change association to a specified new peer MAC entity that is acting as an AP.

10.3.7.1.4 Effect of receipt

This primitive initiates a reassociation procedure. The MLME subsequently issues a MLME-REASSOCIATE.confirm that reflects the results.

10.3.7.2 MLME-REASSOCIATE.confirm

10.3.7.2.1 Function

This primitive reports the results of a reassociation attempt with a specified peer MAC entity that is acting as an AP.

10.3.7.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-REASSOCIATE.confirm      (
                              ResultCode
                              )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, TIMEOUT, REFUSED	Indicates the result of the MLME-REASSOCI-ATE.request

10.3.7.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-REASSOCIATE.request to reassociate with a specified peer MAC entity that is acting as an AP.

10.3.7.2.4 Effect of receipt

The SME is notified of the results of the reassociation procedure.

10.3.7.3 MLME-REASSOCIATE.indication

10.3.7.3.1 Function

This primitive reports the establishment of a reassociation with a specified peer MAC entity.

10.3.7.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-REASSOCIATE.indication   (
                              PeerSTAAddress
                              )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the reassociation was established

10.3.7.3.3 When generated

This primitive is generated by the MLME as a result of the establishment of a reassociation with a specific peer MAC entity that resulted from a reassociation procedure that was initiated by that specific peer MAC entity.

10.3.7.3.4 Effect of receipt

The SME is notified of the establishment of the reassociation.

10.3.8 Disassociate

10.3.8.1 MLME-DISASSOCIATE.request

10.3.8.1.1 Function

This primitive requests disassociation with a specified peer MAC entity that is acting as an AP.

10.3.8.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DISASSOCIATE.request    (
                             PeerSTAAddress,
                             ReasonCode
                             )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the disassociation process
ReasonCode	As defined in frame format.	As defined in frame format.	Specifies the reason for initiating the disassociation procedure

10.3.8.1.3 When generated

This primitive is generated by the SME for a STA to to establish disassociation with an AP.

10.3.8.1.4 Effect of receipt

This primitive initiates a disassociation procedure. The MLME subsequently issues a MLME-DISASSOCIATE.confirm that reflects the results.

10.3.8.2 MLME-DISASSOCIATE.confirm

10.3.8.2.1 Function

This primitive reports the results of a disassociation procedure with a specific peer MAC entity that is acting as an AP.

10.3.8.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DISASSOCIATE.confirm    (
                             ResultCode
                             )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, TIMEOUT, REFUSED	Indicates the result of the MLME-DISASSOCIATE.request

10.3.8.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-DISASSOCIATE.request to disassociate with a specified peer MAC entity that is acting as an AP.

10.3.8.2.4 Effect of receipt

The SME is notified of the results of the disassociation procedure.

10.3.8.3 MLME-DISASSOCIATE.indication

10.3.8.3.1 Function

This primitive reports disassociation with a specific peer MAC entity.

10.3.8.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DISASSOCIATE.indication    (
                                PeerSTAAddress,
                                ReasonCode
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the association relationship was invalidated
ReasonCode	As defined in frame format.	As defined in frame format.	Specifies the reason the disassociation procedure was initiated

10.3.8.3.3 When generated

This primitive is generated by the MLME as a result of the invalidation of an association relationship with a specific peer MAC entity.

10.3.8.3.4 Effect of receipt

The SME is notified of the invalidation of the specific association relationship.

10.3.9 Reset

This mechanism supports the process of resetting the MAC.

10.3.9.1 MLME-RESET.request

10.3.9.1.1 Function

This primitive requests that the MAC entity be reset.

10.3.9.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESET.request          (
                            STAAddress,
                            SetDefaultMIB
                            )
```

Name	Type	Valid range	Description
STAAddress	MACAddress	Any valid MAC address	Specifies the MAC address that is to be used by the MAC entity that is being reset. This value may be used to provide a locally administered STA address.
SetDefaultMIB	Boolean	True, false	If true, all MIB attributes are set to their default values. The default values are implementation dependent. If false, the MAC is reset, but all MIB attributes retain the values that were in place prior to the generation of the MLME-RESET.request primitive.

10.3.9.1.3 When generated

This primitive is generated by the SME to reset the MAC to initial conditions. The MLME-RESET.request primitive must be used prior to use of MLME-START.request primitive.

10.3.9.1.4 Effect of receipt

This primitive sets the MAC to initial conditions, clearing all internal variables to the default values. MIB attributes may be reset to their implementation-dependent default values by setting the SetDefaultMIB flag to true. The MLME subsequently issues a MLME-RESET.confirm that reflects the results.

10.3.9.2 MLME-RESET.confirm

10.3.9.2.1 Function

This primitive reports the results of a reset procedure.

10.3.9.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESET.confirm          (
                            ResultCode
                            )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS	Indicates the result of the MLME-RESET.request

10.3.9.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-RESET.request to reset the MAC entity.

10.3.9.2.4 Effect of receipt

The SME is notified of the results of the reset procedure.

10.3.10 Start

This mechanism supports the process of creating a new BSS.

10.3.10.1 MLME-START.request

10.3.10.1.1 Function

This primitive requests that the MAC entity start a new BSS.

10.3.10.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-START.request        (
                          SSID,
                          BSSType,
                          BeaconPeriod,
                          DTIMPeriod,
                          CF parameter set,
                          PHY parameter set,
                          IBSS parameter set,
                          ProbeDelay.
                          CapabilityInformation,
                          BBSBasicRateSet,
                          OperationalRateSet
                          )
```

Name	Type	Valid range	Description
SSID	Octet string	1–32 octets	The SSID of the BSS
BSSType	Enumeration	INFRA-STRUCTURE, INDEPEN-DENT	The type of the BSS
Beacon Period	Integer	≥1	The Beacon period of the BSS (in TU)
DTIM Period	Integer	As defined in frame format	The DTIM Period of the BSS (in beacon periods)
CF parameter set	As defined in frame format	As defined in frame format	The parameter set for CF periods, if the BSS supports CF mode. aCFPPeriod is modified as a side effect of the issuance of a MLME-START.request primitive.
PHY parameter set	As defined in frame format	As defined in frame format	The parameter set relevant to the PHY
IBSS parameter set	As defined in frame format	As defined in frame format	The parameter set for the IBSS, if BSS is an IBSS

Name	Type	Valid range	Description
ProbeDelay	Integer	N/A	Delay (in μs) to be used prior to transmitting a Probe frame during active scanning
CapabilityInformation	As defined in frame format	As defined in frame format	The capabilities to be advertised for the BSS
BSSBasicRateSet	Set of integers	1 through 127 inclusive (for each integer in the set)	The set of data rates (in units of 500 kbit/s) that must be supported by all STAs to join this BSS. The STA that is creating the BSS must be able to receive at each of the data rates listed in the set.
OperationalRateSet	Set of integers	1 through 127 inclusive (for each integer in the set)	The set of data rates (in units of 500 kbit/s) that the STA may use for communication within the BSS. The STA must be able to receive at each of the data rates listed in the set. The OperationalRateSet is a superset of the BSSBasicRateSet advertised by the BSS.

10.3.10.1.3 When generated

This primitive is generated by the SME to start either an infrastructure BSS (with the MAC entity acting as an AP), or start an independent BSS (with the MAC entity acting as the first STA in the IBSS).

The MLME-START.request primitive must be generated after a MLME-RESET.request primitive has been used to reset the MAC entity and before an MLME-JOIN.request primitive has been used to successfully join an existing infrastructure BSS or independent BSS.

The MLME-START.request primitive must not be used after successful use of the MLME-START.request primitive or successful use of the MLME-JOIN.request without generating an intervening MLME-RESET.request primitive.

10.3.10.1.4 Effect of receipt

This primitive initiates the BSS initialization procedure once the current frame exchange sequence is complete. The MLME subsequently issues a MLME-START.confirm that reflects the results of the creation procedure.

10.3.10.2 MLME-START.confirm

10.3.10.2.1 Function

This primitive reports the results of a BSS creation procedure.

10.3.10.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-START.confirm          (
                            ResultCode
                            )
```

Name	Type	Valid range	resetDescription
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, BSS_ALREADY_STARTED_OR_JOINED	Indicates the result of the MLME-START.request

10.3.10.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-START.request to create a new BSS.

10.3.10.2.4 Effect of receipt

The SME is notified of the results of the BSS creation procedure.

10.4 PLME SAP interface

The PHY management service interface consists of the generic PLMEGET and PLMESET primitives on PHY MIB attributes, as described previously, together with the PLME-RESET primitive and the following specific primitives.

10.4.1 PLME-RESET.request

10.4.1.1 Function

This primitive shall be a request by the LME to reset the PHY. The PHY shall be always reset to the receive state to avoid accidental data transmission.

10.4.1.2 Semantics of the service primitive

The primitive shall provide the following parameters:

RESET.request ( )

There are no parameters associated with this primitive.

10.4.1.3 When generated

This primitive shall be generated at any time to reset the PHY.

10.4.1.4 Effect of receipt

Receipt of this primitive by the PHY sublayer shall cause the PHY entity to reset both the transmit and the receive state machines and place the PHY into the receive state.

10.4.2 PLME-DSSSTESTMODE.request

10.4.2.1 Function

This primitive requests that the DSSS PHY entity enter a test mode operation. The parameters associated with this primitive are considered as recommendations and are optional in any particular implementation.

10.4.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
PLME-DSSSTESTMODE.request    (
                             TEST_ENABLE,
                             TEST_MODE,
                             SCRAMBLE_STATE,
```


SPREADING_STATE,
DATA_TYPE,
DATA_RATE;
)

Name	Type	Valid range	Description
TEST_ENABLE	Boolean	True, false	If true, enables the PHY test mode according to the remaining parameters
TEST_MODE	Integer	1, 2, 3	TEST_MODE selects one of three operational states: 01 = transparent receive 02 = continuous transmit 03 = 50% duty cycle
SCRAMBLE_STATE	Boolean	True, false	If true, sets the operational state of the scrambler to ON
SPREADING_STATE	Boolean	True, false	If true, selects the operational state of the chipping
DATA_TYPE	Integer	1, 2, 3	Selects one of three data patterns to be used for the transmit portions of the tests
DATA_RATE	Integer	2, 4	Selects between 1 and 2 Mbit/s operation 02 = 1 Mbit/s 04 = 2 Mbit/s

10.4.2.3 When generated

This primitive shall be generated at any time to enter the DSSS PHY test mode.

10.4.2.4 Effect of receipt

Receipt of this primitive by the PHY sublayer shall cause the DSSS PHY entity to enter the test mode of operation.

10.4.3 PLME-DSSSTESTOUTPUT.request

10.4.3.1 Function

This optional primitive shall be a request by the LME to enable selected test signals from the PHY. The parameters associated with this primitive are considered as recommendations and are optional in any particular implementation.

10.4.3.2 Semantics of the service primitive

The primitive parameters are as follows:

PLME-DSSSTESTOUTPUT.request (
TEST_OUTPUT,
)

Name	Type	Valid range	Description
TEST_OUTPUT	Boolean	True, false	If true, enable the selected test signals for testing DS PHY

TEST_OUTPUT enables and disables selected signals for debugging and testing the PHY. Some signals that may be available for output are PHY-TXSTART.request, PHY-RXSTART.indicate(RXVECTOR), PHY-CCA.indicate, the chipping clock, the data clock, the symbol clock, TX data, and RX data.

10.4.3.3 When generated

This primitive shall be generated at any time to enable the test outputs when in the DSSS PHY test mode.

10.4.3.4 Effect of receipt

Receipt of this primitive by the DSSS PHY sublayer shall cause the DSSS PHY entity to enabled the test outputs using the modes set by the most recent PLME-DSSSTESTMODE.request primitive.

11. MAC sublayer management entity

11.1 Synchronization

All STAs within a single BSS shall be synchronized to a common clock using the mechanisms defined herein.

11.1.1 Basic approach

A timing synchronization function (TSF) keeps the timers for all STAs in the same BSS synchronized. All STAs shall maintain a local TSF timer.

11.1.1.1 TSF for infrastructure networks

In an infrastructure network, the AP shall be the timing master and shall perform the TSF. The AP shall initialize its TSF timer independently of any simultaneously started APs in an effort to minimize the synchronization of the TSF timers of multiple APs. The AP shall periodically transmit special frames called *beacons* that contain a copy of its TSF timer to synchronize the other STAs in a BSS. A receiving STA shall always accept the timing information in beacons sent from the AP servicing its BSS. If a STA's TSF timer is different from the timestamp in the received beacon, the receiving STA shall set its local timer to the received timestamp value.

Beacons shall be generated for transmission by the AP once every BeaconPeriod time units.

11.1.1.2 TSF for an independent BSS (IBSS)

The TSF in an IBSS shall be implemented via a distributed algorithm that shall be performed by all of the members of the BSS. Each STA in the BSS shall transmit beacons according to the algorithm described in this clause. Each STA in an IBSS shall adopt the timing received from any beacon or probe response that has a TSF value later than its own TSF timer.

11.1.2 Maintaining synchronization

Each STA shall maintain a TSF timer with modulus 2^{64} counting in increments of microseconds. STAs expect to receive beacons at a nominal rate. The interval between beacons is defined by the aBeaconPeriod parameter of the STA. A STA sending a beacon shall set the value of the beacon's timestamp so that it equals the value of the STA's TSF timer at the time that the first bit of the timestamp is transmitted to the PHY plus the transmitting STA's delays through its local PHY from the MAC-PHY interface to its interface with the wireless medium (antenna, LED emission surface, etc.). The algorithms in this clause define a mechanism that maintains the synchronization of the TSF timers in a BSS to within 4 µs plus the maximum propagation delay of the PHY for PHYs of 1 Mb/s, or greater.

11.1.2.1 Beacon generation in infrastructure networks

The AP shall define the timing for the entire BSS by transmitting beacons according to the aBeaconPeriod attribute within the AP. This defines a series of TBTTs exactly aBeaconPeriod time units apart. Time zero is defined to be a TBTT with the beacon being a DTIM and transmitted at the beginning of a CFP. At each TBTT, the AP shall schedule a beacon as the next frame for transmission. If the medium is determined by the carrier-sense mechanism (see 9.2.1) to be unavailable, the AP shall delay the actual transmission of a beacon according to the basic medium access rules specified in Clause 9. The beacon period is included in Beacon and Probe Response frames, and STAs shall adopt that beacon period when joining the BSS.

NOTE—Though the transmission of a beacon may be delayed because of CSMA deferrals, subsequent beacons shall be scheduled at the nominal beacon interval. This is shown in Figure 64.

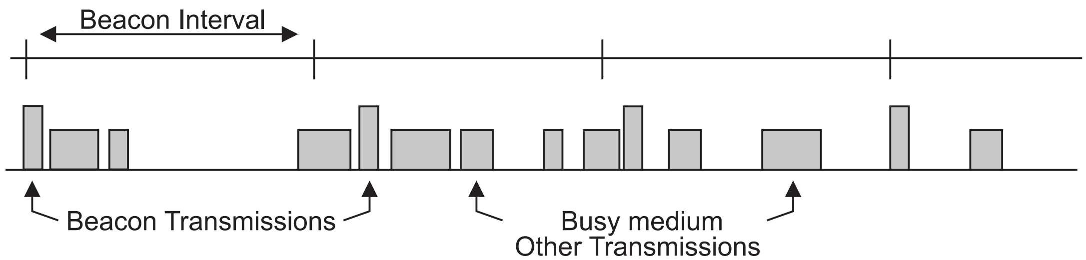


Figure 64—Beacon transmission on a busy network

11.1.2.2 Beacon generation in an IBSS

Beacon generation in an IBSS is distributed. The beacon period is included in Beacon and Probe Response frames, and STAs shall adopt that beacon period when joining the IBSS. All members of the IBSS participate in beacon generation. Each STA shall maintain its own TSF timer that is used for aBeaconPeriod timing. The beacon interval within an IBSS is established by the STA that instantiates the IBSS. This defines a series of TBTTs exactly aBeaconPeriod time units apart. Time zero is defined to be a TBTT. At each TBTT the STA shall

a) Suspend the decrementing of the backoff timer for any pending non-beacon or non-ad hoc traffic indication (ATIM) transmission,
b) Calculate a random delay uniformly distributed in the range between zero and twice aCWmin × aSlotTime,
c) Wait for the period of the random delay, decrementing the random delay timer using the same algorithm as for backoff,
d) If a beacon arrives before the random delay timer has expired, then the remaining random delay is canceled, the pending beacon transmission is canceled, and the ATIM backoff timer shall resume decrementing,
e) If the random delay has expired and no beacon has arrived during the delay period, send a beacon.

(See Figure 65.)

The beacon transmission shall always occur during the Awake Period of STAs that are operating in a low-power mode. This is described in more detail in 11.2.

11.1.2.3 Beacon reception

STAs shall use information from the CF Parameter Set element of all received Beacon frames to update their NAV as specified in 9.3.2.2.

STAs in an infrastructure network shall only use other information in received Beacon frames, if the BSSID field are equal to the MAC address currently in use by the STA contained in the AP of the BSS.

STAs in an IBSS shall use other information in any received Beacon frame for which the IBSS subfield of the Capability field is set to 1 and the content of the SSID element is equal to the SSID of the IBSS. Use of this information is specified in 11.1.4.

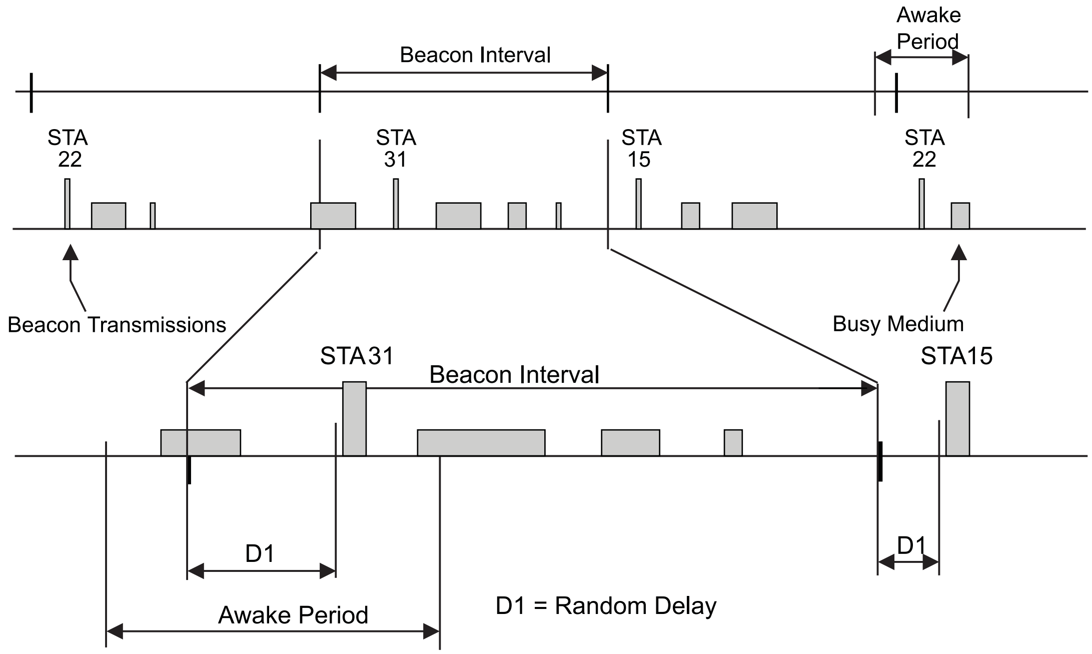


Figure 65—Beacon transmission in an IBSS

11.1.2.4 TSF timer accuracy

Upon receiving a Beacon frame with a valid FCS and BSSID or SSID, as described in 11.1.2.3, a STA shall update its TSF timer according to the following algorithm: The received timestamp value shall be adjusted by adding an amount equal to the receiving STA's delay through its local PHY components plus the time since the first bit of the timestamp was received at the MAC/PHY interface. In the case of an infrastructure BSS, the STA's TSF timer shall then be set to the adjusted value of the timestamp. In the case of an IBSS, the STA's TSF timer shall be set to the adjusted value of the received timestamp, if the adjusted value of the timestamp is later than the value of the STA's TSF timer. The accuracy of the TSF timer shall be ±0.01%.

11.1.3 Acquiring synchronization, scanning

A STA shall operate in either a Passive Scanning mode or an Active Scanning mode depending on the current value of the ScanMode parameter of the MLME-SCAN.request primitive.

Upon receipt of the MLME-SCAN.request primitive, a STA shall perform scanning. The SSID parameter indicates the SSID for which to scan. To become a member of a particular ESS using passive scanning, a STA shall scan for Beacon frames containing that ESS's SSID, returning all Beacon frames matching the desired SSID in the BSSDescriptionSet parameter of the corresponding MLME-SCAN.confirm primitive with the appropriate bits in the Capabilities Information field indicating whether the beacon came from an Infrastructure BSS or IBSS. To actively scan, the STA shall transmit Probe frames containing the desired SSID. Upon completion of scanning, an MLME-SCAN.confirm is issued by the MLME indicating all of the BSS information received.

Upon receipt of an MLME-JOIN.request, the STA will join a BSS by adopting the BSSID, TSF timer value, PHY parameters, and the beacon period specified in the request.

Upon receipt of an MLME-SCAN.request with the broadcast SSID, the STA shall passively scan for any Beacon frames, or actively transmit Probe frames containing the broadcast SSID, as appropriate depending upon the value of ScanMode. Upon completion of scanning, an MLME-SCAN.confirm is issued by the MLME indicating all of the BSS information received.

If a STA's scanning does not result in finding a BSS with the desired SSID and of the desired type, or does not result in finding any BSS, the STA may start an IBSS upon receipt of the MLME-START.request.

A STA may start its own BSS without first scanning for a BSS to join.

When a STA starts a BSS, that STA shall determine the BSSID of the BSS. If the BSSType indicates an infrastructure BSS, then the STA shall start an infrastructure BSS and the BSSID shall be equal to the STA's aStationID. The value of the BSSID shall remain unchanged, even if the value of aStationID is changed after the completion of the MLME-Start.request. If the BSSType indicates an IBSS, the STA shall start an IBSS, and the BSSID shall be an individual locally administered IEEE MAC address as defined in 5.2 of IEEE Std 802-1990. The remaining 46 bits of that MAC address shall be a number selected in a manner that minimizes the probability of STAs generating the same number, even when those STAs are subjected to the same initial conditions. The value SSID parameter shall be used as the SSID of the new BSS. It is important that designers recognize the need for statistical independence among the random number streams among STAs.

11.1.3.1 Passive scanning

If a ScanType is passive, the STA shall listen to each channel scanned for no longer than a maximum duration defined by the ChannelTime parameter.

11.1.3.2 Active scanning

Active scanning involves the generation of Probe frames and the subsequent processing of received Probe Response frames. The details of the active scanning procedures are as specified in the following subclauses.

11.1.3.2.1 Sending a probe response

STAs, subject to criteria below, receiving Probe Request frames shall respond with a probe response only if the SSID in the probe request is the broadcast SSID or matches the specific SSID of the STA. Probe Response frames shall be sent as directed frames to the address of the STA that generated the probe request. The probe response shall be sent using normal frame transmission rules. An AP shall respond to all probe requests meeting the above criteria. In an IBSS, the STA that generated the last beacon shall be the STA that responds to a probe request.

In each BSS there shall be at least one STA that is awake at any given time to respond to probe requests. A STA that sent a beacon shall remain in the Awake state and shall respond to probe requests until a Beacon frame with the current BSS ID is received. If the STA is an AP, it shall always remain in the Awake state and always respond to probe requests. There may be more than one STA in an IBSS that responds to any given probe request, particularly in cases where more than one STA transmitted a Beacon frame following the most recent TBTT, either due to not receiving successfully a previous beacon or due to collisions between beacon transmissions.

11.1.3.2.2 Active scanning procedure

Upon receipt of the MLME-SCAN.request with ScanType indicating an active scan, a STA shall use the following procedure:

For each channel to be scanned,

a) Wait until the ProbeDelay time has expired or a PHYRxStart.indication has been received;
b) Perform the Basic Access procedure as defined in 9.2.5.1;

c) Send a probe with the broadcast destination, SSID, and broadcast BSSID;
d) Clear and start a ProbeTimer;
e) If PHYCCA.indication (busy) has not been detected before the ProbeTimer reaches MinChannelTime, then clear NAV and scan the next channel, else when ProbeTimer reaches MaxChannelTime, process all received probe responses;
f) Clear NAV and scan the next channel.

See Figure 66.

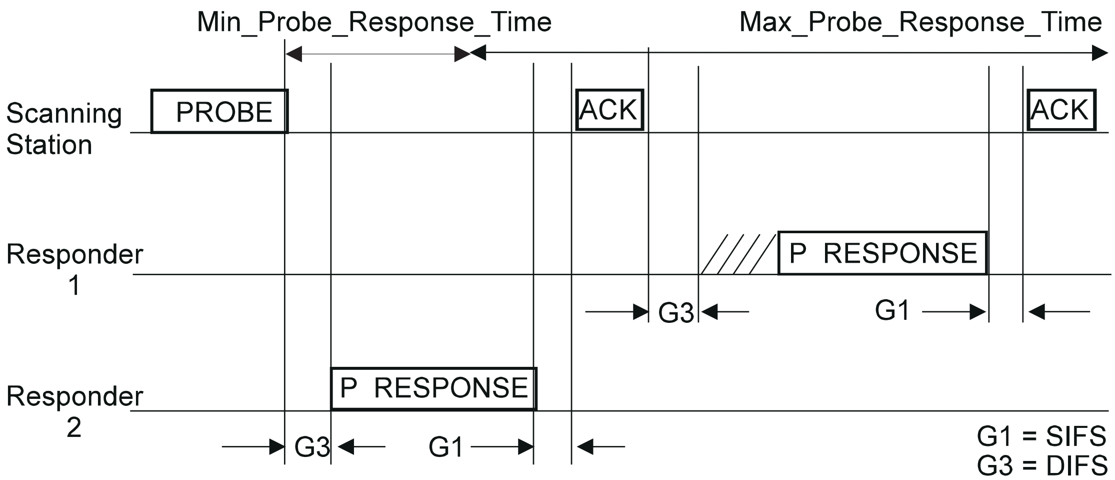


Figure 66—Probe response

When all channels in the ChannelList have been scanned, the MLME shall issue an MLME-Scan.confirm with the BSSDescriptionSet containing all of the information gathered during the scan.

11.1.3.3 Initializing a BSS

Upon receipt of an MLME-Start.request, a STA shall determine the BSS's BSSID (as described in 11.1.3), select channel synchronization information, select a beacon period, initialize and start its TSF timer, and begin transmitting beacons.

11.1.3.4 Synchronizing with a BSS

Upon receipt of an MLME-Join.request, a STA shall adopt the BSSID, channel synchronization information, and TSF timer value of the parameters in the request. Upon receipt of a Beacon frame from the BSS, the MLME shall issue an MLME-Join.confirm indicating the operation was successful. If the JoinFailureTimeout expires prior to the receipt of a Beacon frame from the BSS, the MLME shall issue an MLME-Join.confirm indicating the operation was unsuccessful.

11.1.4 Adjusting STA timers

In the infrastructure network, STAs shall always adopt the timer in a beacon or probe response coming from the AP in their BSS.

In an IBSS, a STA shall always adopt the information in the contents of a Beacon or Probe Response frame when that frame contains a matching SSID and the value of the time stamp is later than the STA's TSF timer. In response to an MLME-Join.request, a STA shall initialize its TSF timer to 0 and shall not transmit a beacon or probe response until it hears a beacon or probe response from a member of the IBSS with a matching SSID.

All Beacon and Probe Response frames carry a Timestamp field. A STA receiving such a frame from another STA in an IBSS with the same SSID shall compare the Timestamp field with its own TSF time. If the Timestamp field of the received frame is later than its own TSF time, the STA shall adopt all parameters contained in the Beacon frame.

11.1.5 Timing synchronization for frequency-hopping (FH) PHYs

NOTE—This subclause pertains only to STAs using an FH PHY.

The TSF described here provides a mechanism for STAs in an FH system to synchronize their transitions from one channel to another (their "hops"). Every STA shall maintain a table of all of the hopping sequences that are used in the system. All of the STAs in a BSS shall use the same hopping sequence. Each beacon and probe response includes the channel synchronization information necessary to determine the hop pattern and timing for the BSS.

STAs shall use their TSF timer to time the aCurrentDwellTime. The aCurrentDwellTime is the length of time that STAs shall stay on each frequency in their hopping sequence. Once STAs are synchronized, they have the same TSF timer value.

STAs in the BSS shall issue an appropriate PLME service primitive for the PHY in use to tune to the next frequency in the hopping sequence whenever

TSF timer MOD aCurrentDwellTime = 0

11.2 Power management

11.2.1 Power management in an infrastructure network

STAs changing Power Management mode shall inform the AP of this fact using the Power Management bits within the Frame Control field of transmitted frames. The AP shall not arbitrarily transmit MSDUs to STAs operating in a power-saving mode, but shall buffer MSDUs and only transmit them at designated times.

The STAs that currently have buffered MSDUs within the AP are identified in a *traffic indication map* (TIM), which shall be included as an element within all beacons generated by the AP. A STA shall determine that an MSDU is buffered for it by receiving and interpreting a TIM.

STAs operating in Power-Save (PS) modes shall periodically listen for beacons, as determined by the STA's ListenInterval and ReceiveDTIMs parameters of the MLME-Power-Mgt.request primitive.

In a BSS operating under the DCF, or during the contention period of a BSS using the PCF, upon determining that an MSDU is currently buffered in the AP, a STA operating in the *PS mode* shall transmit a short PS-Poll frame to the AP, which shall respond with the corresponding buffered MSDU immediately, or acknowledge the PS-Poll and respond with the corresponding MSDU at a later time. If the TIM indicating the buffered MSDU is sent during a contention-free period (CFP), a CF-Pollable STA operating in the PS mode does not send a PS-Poll frame, but remains active until the buffered MSDU is received (or the CFP ends). If any STA in its BSS is in PS mode, the AP shall buffer all broadcast and multicast MSDUs and deliver them to all STAs immediately following the next Beacon frame containing a *delivery TIM* (DTIM) transmission.

A STA shall remain in its current Power Management mode until it informs the AP of a Power Management mode change via a successful frame exchange. Power Management mode shall not change during any single frame exchange sequence, as described in 9.7.

11.2.1.1 STA Power Management modes

A STA may be in one of two different power states:

— *Awake:* STA is fully powered.
— *Doze:* STA is not able to transmit or receive and consumes very low power.

The manner in which a STA transitions between these two power states shall be determined by the STA's Power Management mode. These modes are summarized in Table 23.

The Power Management mode of a STA is selected by the PowerManagementMode parameter of the MLME-POWERMGT.request. Once the STA updates its Power Management mode, the MLME shall issue an MLME-POWERMGT.confirm indicating the success of the operation.

Table 23—Power Management modes

Active mode or AM	STA may receive frames at any time. In Active mode, a STA shall be in the Awake state. A STA on the polling list of a PCF shall be in Active mode for the duration of the CFP.
Power Save or PS	STA listens to selected beacons (based upon its aListenInterval) and sends PS-Poll frames to the AP if the TIM element in the most recent beacon indicates a directed MSDU buffered for that STA. The AP shall transmit buffered directed MSDUs to a PS STA only in response to a PS-Poll from that STA, or during the CFP in the case of a CF-Pollable PS STA. In PS mode, a STA shall be in the Doze state and shall enter the Awake state to receive selected beacons, to receive broadcast and multicast transmissions following certain received beacons, to transmit, and to await responses to transmitted PS-Poll frames or (for CF-Pollable STAs) to receive contention-free transmissions of buffered MSDUs.

To change Power Management modes, a STA shall inform the AP through a successful frame exchange initiated by the STA. The Power Management bit in the Frame Control field of the frame sent by the STA in this exchange indicates the Power Management mode that the STA shall adopt upon successful completion of the entire frame exchange.

A STA that is changing from Doze to Awake in order to transmit shall perform clear channel assessment (CCA) until a frame sequence is detected by which it can correctly set its NAV, or until a period of time equal to the ProbeDelay has transpired.

11.2.1.2 AP TIM transmissions

The TIM shall identify the STAs for which traffic is pending and buffered in the AP. This information is coded in a *partial virtual bitmap*, as described in 7.3.2.6. In addition, the TIM contains an indication whether broadcast/multicast traffic is pending. Every STA is assigned a Association ID code (AID) by the AP as part of the association process. AID 0 (zero) is reserved to indicate the presence of buffered broadcast/multicast MSDUs. The AP shall identify those STAs for which it is prepared to deliver buffered MSDUs by setting bits in the TIM's partial virtual bitmap that correspond to the appropriate SIDs.

11.2.1.3 TIM types

Two different TIM types are distinguished: TIM and DTIM. After a DTIM, the AP shall send out the buffered broadcast/multicast MSDUs using normal frame transmission rules, before transmitting any unicast frames.

The AP shall transmit a TIM with every beacon. Every DTIMPeriod, a TIM of type “DTIM” is transmitted within a beacon, rather than an ordinary TIM.

Figure 1 illustrates the AP and STA activity under the assumption that a DTIM is transmitted once every three TIMs. The top line in Figure 1 represents the time axis, with the beacon interval shown together with a DTIM Interval of three beacon intervals. The second line depicts AP activity. The AP schedules beacons for transmission every beacon interval, but the beacons may be delayed if there is traffic at the TBTT. This is indicated as “busy medium” on the second line. For the purposes of this figure, the important fact about beacons is that they contain TIMs, some of which may be DTIMs. Note that the second STA with ReceiveDTIMs set to false does not power up its receiver for all DTIMs.

The third and fourth lines in Figure 1 depict the activity of two STAs operating with different power management requirements. Both STAs power-on their receivers whenever they need to listen for a TIM. This is indicated in as a ramp-up of the receiver power prior to the TBTT. The first STA, for example, powers up its receiver and receives a TIM in the first beacon; that TIM indicates the presence of a buffered MSDU for the receiving STA. The receiving STA then generates a PS-Poll frame, which elicits the transmission of the buffered data MSDU from the AP. Broadcast and multicast MSDUs are sent by the AP subsequent to the transmission of a beacon containing a DTIM. The DTIM is indicated by the DTIM count field of the TIM element having a value of 0.

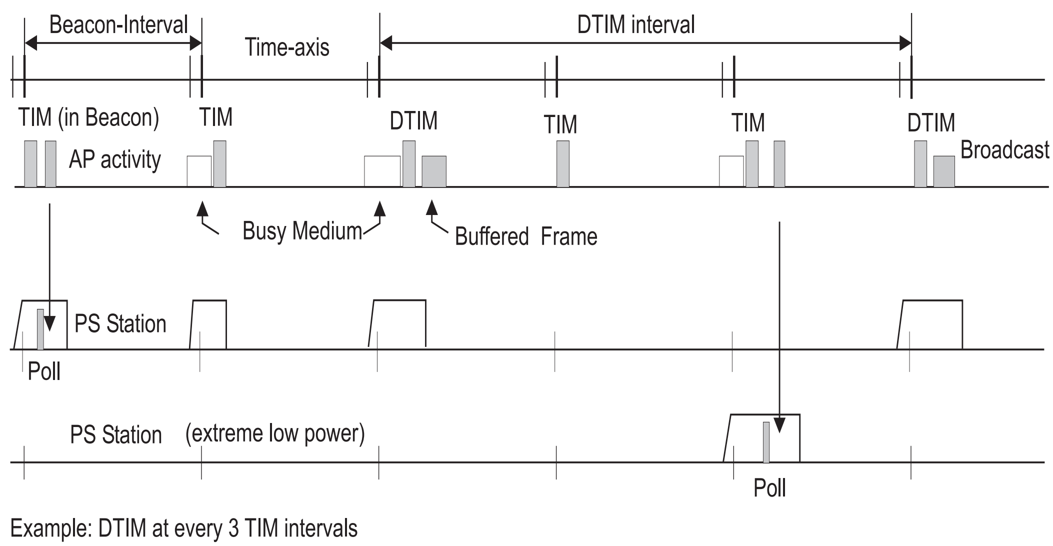


Figure 1—Infrastructure power management operation (no PCF operating)

11.2.1.4 AP operation during the contention period

APs shall maintain a Power Management status for each currently associated STA that indicates in which Power Management mode the STA is currently operating. An AP shall, depending on the Power Management mode of the STA, temporarily buffer the MSDU or management frame destined to the STA. No MSDUs or management frames received for STAs operating in the Active mode shall be buffered for power management reasons.

a) MSDUs, or management frames destined for PS STAs, shall be temporarily buffered in the AP. The algorithm to manage this buffering is beyond the scope of this standard.
b) MSDUs, or management frames destined for STAs in the Active mode, shall be directly transmitted.

c) At every beacon interval, the AP shall assemble the partial virtual bitmap containing the buffer status per destination for STAs in the PS mode, and shall send this out in the TIM field of the beacon. The bit for AID 0 (zero) shall be set whenever broadcast or multicast traffic is buffered.
d) All broadcast/multicast MSDUs, with the Order bit in the Frame Control field clear, shall be buffered if any associated STAs are in PS mode.
e) Immediately after every DTIM, the AP shall transmit all buffered broadcast/multicast MSDUs. The More Data field of each broadcast/multicast frame shall be set to indicate the presence of further buffered broadcast/multicast MSDUs. If the AP is unable to transmit all of the buffered broadcast/multicast MSDUs before the TBTT following the DTIM, the AP shall indicate that it will continue to deliver the broadcast/multicast MSDUs by setting the broadcast/multicast bit in the partial virtual bitmap of the TIM element of every Beacon frame, until all buffered broadcast/multicast frames have been transmitted.
f) A single buffered MSDU or management frame for a STA in the PS mode shall be forwarded to the STA after a PS-Poll has been received from that STA. The More Data field shall be set to indicate the presence of further buffered MSDUs or management frames for the polling STA. Further PS-Poll frames from the same STA shall be acknowledged and ignored until the MSDU or management frame has either been successfully delivered, or presumed failed due to maximum retries being exceeded. This prevents a retried PS-Poll from being treated as a new request to deliver a buffered frame.
g) An AP shall have an aging function to delete pending traffic when it is buffered for an excessive time period.
h) Whenever an AP is informed that a STA changes to the Active mode, then the AP shall send buffered MSDUs and management frames (if any exist) to that STA without waiting for a PS-Poll.

11.2.1.5 AP operation during the CFP

APs shall maintain a Power Management status for each currently associated CF-Pollable STA that indicates in which Power Management mode the STA is currently operating. An AP shall, for STAs in PS mode, temporarily buffer the MSDU destined to the STA.

a) MSDUs destined for PS STAs shall be temporarily buffered in the AP. The algorithm to manage this buffering is beyond the scope of this standard.
b) MSDUs destined to STAs in the Active mode shall be transmitted as defined in Clause 9.
c) Prior to every CFP, and at each beacon interval within the CFP, the AP shall assemble the partial virtual bitmap containing the buffer status per destination for STAs in the PS mode, set the bits in the partial virtual bitmap for STAs the point coordinator (PC) is intending to poll during this CFP, and shall send this out in the TIM field of the DTIM. The bit for AID 0 (zero) shall be set whenever broadcast or multicast traffic is buffered.
d) All broadcast and multicast MSDUs, with the Order bit in the Frame Control field clear, shall be buffered if any associated STAs are in the PS mode, whether or not those STAs are CF-Pollable.
e) Immediately after every DTIM (Beacon frame with DTIM Count field of the TIM element equal to zero), the AP shall transmit all buffered broadcast and multicast frames. The More Data field shall be set to indicate the presence of further buffered broadcast/multicast MSDUs. If the AP is unable to transmit all of the buffered broadcast/multicast MSDUs before the TBTT following the DTIM, the AP shall indicate that it will continue to deliver the broadcast/multicast MSDUs by setting the broadcast/multicast bit in the partial virtual bitmap of the TIM element of every Beacon frame, until all buffered broadcast/multicast frames have been transmitted.
f) Buffered MSDUs or management frames for STAs in the PS mode shall be forwarded to the CF-Pollable STAs under control of the PC. Transmission of these buffered MSDUs or management frames shall begin immediately after transmission of buffered broadcast and multicast frames (if any), and shall occur in order by increasing AID of CF-Pollable STAs. A CF-Pollable STA for which the TIM element of the most recent beacon indicated buffered MSDUs or management frames shall be in the Awake state at least until the receipt of a directed frame from the AP in which the Frame Control field does not indicate the existence of more buffered MSDUs or management frames. After

acknowledging the last of the buffered MSDUs or management frames, the CF-Pollable STA operating in the PS mode may enter the Doze state until the next DTIM is expected.

g) An AP shall have an aging function to delete pending traffic buffered for an excessive time period. The exact specification of the aging function is beyond the scope of this standard.
h) Whenever an AP detects that a CF-Pollable STA has changed from the PS mode to the Active mode, then the AP shall queue any buffered frames addressed to that STA for transmission to that CF-Pollable STA as directed by the AP's PC function (PCF).

11.2.1.6 Receive operation for STAs in PS mode during the contention period

STAs in PS mode shall operate as follows to receive an MSDU or management frame from the AP when no PC is operating and during the contention period when a PC is operating.

a) STAs shall wake up early enough to be able to receive the next scheduled beacon after ListenInterval from the last TBTT.
b) When a STA detects that the bit corresponding to its AID is set in the TIM, the STA shall issue a PS-Poll to retrieve the buffered MSDU or management frame. If more than one bit is set in the TIM, the PS-Poll shall be transmitted after a random delay uniformly distributed between zero and aCWmin.
c) The STA shall remain in the Awake state until it receives the response to its poll, or it receives another beacon whose TIM indicates that the AP does not have any MSDUs or management frames buffered for this STA. If the bit corresponding to the STA's AID is set in the subsequent TIM, the STA shall issue another PS-Poll to retrieve the buffered MSDU or management frame(s).
d) If the More Data field in the received MSDU or management frame indicates that more traffic for that STA is buffered, the STA, at its convenience, shall Poll until no more MSDUs or management frames are buffered for that STA.
e) When ReceiveDTIMs is true, the STA shall wake up early enough to be able to receive every DTIM. A STA receiving broadcast/multicast MSDUs shall remain awake until the More Data field of the broadcast/multicast MSDUs indicate there are no further buffered broadcast/multicast MSDUs, or until a TIM is received indicating there are no more buffered broadcast/multicast MSDUs.

11.2.1.7 Receive operation for STAs in PS mode during the CFP

STAs in PS mode that are associated as CF-Pollable shall operate as follows in a BSS with an active PC to receive MSDUs or management frames from the AP during the CFP:

a) STAs shall enter the Awake state so as to receive the Beacon frame (which contains a DTIM) at the start of each CFP.
b) To receive broadcast/multicast MSDUs, the STA shall wake up early enough to be able to receive every DTIM that may be sent during the CFP. A STA receiving broadcast/multicast MSDUs shall remain awake until the More Data field of the broadcast/multicast MSDUs indicate there are no further buffered broadcast/multicast MSDUs, or until a TIM is received indicating there are no more buffered broadcast/multicast MSDUs buffered.
c) When a STA detects that the bit corresponding to its AID is set in the DTIM at the start of the CFP (or in a subsequent TIM during the CFP), the STA shall remain in the Awake state for at least that portion of the CFP through the time that the STA receives a directed MSDU or management frame from the AP with the More Data field in the Frame Control field indicating that no further traffic is buffered.
d) If the More Data field in the Frame Control field of the last MSDU or management frame received from the AP indicates that more traffic for the STA is buffered, then, when the CFP ends, the STA may remain in the Awake state and transmit PS-Poll frames during the contention period to request the delivery of additional buffered MSDU or management frames, or may enter the Doze state during the contention period (except at TBTTs for DTIMs expected during the contention period), awaiting the start of the next CFP.

11.2.1.8 STAs operating in the Active mode

A STA operating in this mode shall have its receiver activated continuously; it does not need to interpret the traffic announcement part of the beacons.

11.2.1.9 AP aging function

The AP shall have an aging function to delete buffered traffic when it has been buffered for an excessive period of time. That function shall be based on the aListenInterval of the STA for which the traffic is buffered. The AP aging function shall not cause the buffered traffic to be discarded after any period that is shorter than the aListenInterval of the STA for which the traffic is buffered. The exact specification of the aging function is beyond the scope of this standard.

11.2.2 Power management in an IBSS

This subclause specifies the power management mechanism for use within an IBSS.

11.2.2.1 Basic approach

The basic approach is similar to the infrastructure case in that the STAs are synchronized, and multicast MSDUs and those MSDUs that are to be transmitted to a power-conserving STA are first announced during a period when all STAs are awake. The announcement is done via an ad hoc traffic indication message (ATIM). A STA in the PS mode shall listen for these announcements to determine if it needs to remain in the awake state.

When an MSDU is to be transmitted to a destination STA that is in a PS mode, the transmitting STA first transmits an ATIM frame during the ATIM Window, in which all the STAs including those operating in a PS mode are awake. The ATIM Window is defined as a specific period of time, defined by aATIMWindow, following a TBTT, during which only beacon or ATIM frames shall be transmitted. ATIM transmission times are randomized, after a Beacon frame is either transmitted or received by the STA, using the backoff procedure with the contention window equal to aCWminx. Directed ATIMs shall be acknowledged. If a STA transmitting a directed ATIM does not receive an acknowledgment, the STA shall execute the backoff procedure for retransmission of the ATIM. Multicast ATIMs shall not be acknowledged.

If a STA receives a directed ATIM frame during the ATIM Window, it shall acknowledge the directed ATIM and stay awake for the entire beacon interval waiting for the announced MSDU(s) to be received. If a STA does not receive an ATIM, it may enter the Doze state at the end of the ATIM Window. Transmissions of MSDUs announced by ATIMs are randomized after the ATIM Window, using the backoff procedure described in Clause 9.

It is possible that an ATIM may be received from more than one STA, and that a STA that receives an ATIM may receive more than a single MSDU from the transmitting STA. ATIM frames are only addressed to the destination STA of the MSDU.

An ATIM for a broadcast or multicast MSDU shall have a destination address identical to that of the MSDU.

After the ATIM interval, only those directed MSDUs that have been successfully announced with an acknowledged ATIM, and broadcast/multicast MSDUs that have been announced with an ATIM, shall be transmitted to STAs in the PS mode. Transmission of these frames shall be done using the normal DCF access procedure.

Figure 67 illustrates the basic power-save operation.

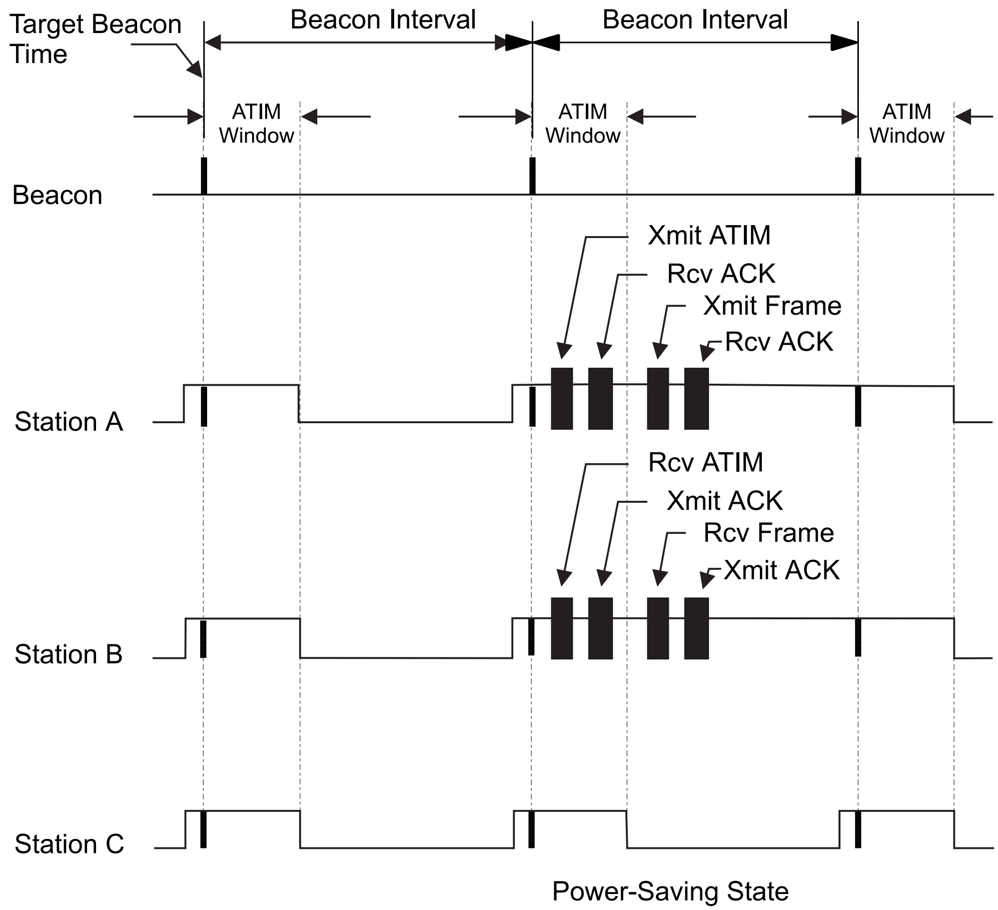


Figure 67—Power management in an IBSS—Basic operation

The estimated power-saving state of another STA may be based on the power management information transmitted by that STA and on additional information available locally, such as a history of failed transmission attempts. The use of RTS/CTS in an IBSS may reduce the number of transmissions to a STA that is in PS mode. If an RTS is sent and a CTS is not received, the transmitting STA may assume that the destination STA is in PS mode. The method of estimating the power management state of other STAs in the IBSS is outside the scope of this standard.

11.2.2.2 Initialization of power management within an IBSS

The following procedure shall be used to initialize power management within a new IBSS, or to learn about the power management being used within an existing IBSS.

a) A STA joining an existing IBSS by the procedure in 11.1.3.3 shall update its ATIM Window with the value contained in the ATIM Window field of the IBSS Parameter Set element within the Beacon or Probe Response management frame received during the scan procedure.

b) A STA creating a new IBSS by the procedure in 11.1.3.3 shall set the value of the ATIM Window field of the IBSS Parameter Set element within the Beacon management frames transmitted to the value of its ATIM Window.

c) The start of the ATIM Window shall be the TBTT, defined in 11.1.2.2. The end of the ATIM Window shall be defined as

   TSF timer MOD BeaconInterval = ATIMWindow.

d) The ATIM Window period shall be static during the lifetime of the IBSS.

e) An ATIM Window value of zero shall indicate that power management is not in use within the IBSS.

11.2.2.3 STA power state transitions

A STA may enter PS mode if and only if the value of the ATIM Window in use within the IBSS is greater than zero. A STA shall set the Power Management subfield in the Frame Control field of MSDUs that it transmits according to the procedure in 7.1.3.1.7.

A STA in PS mode shall transition between Awake and Doze states according to the following rules:

a) If a STA is operating in PS mode, it shall enter the Awake state prior to each TBTT.
b) If a STA receives a directed ATIM management frame containing its individual address, or a multicast ATIM management frame during the ATIM Window it shall remain in the Awake state until the end of the next ATIM Window.
c) If a STA transmits a Beacon or an ATIM management frame, it shall remain in the Awake state until the end of the next ATIM Window regardless of whether an acknowledgment is received for the ATIM.
d) If the STA has not transmitted an ATIM and does not receive either a directed ATIM management frame containing its individual address, or a multicast ATIM management frame during the ATIM Window, it may return to the Doze state following the end of the current ATIM Window.

11.2.2.4 ATIM and frame transmission

If power management is in use within an IBSS, all STAs shall buffer MSDUs for STAs that are known to be in PS mode. The algorithm used for the estimation of the power management state of STAs within the IBSS is outside the scope of this standard. MSDUs to STAs in Active mode may be sent at any valid time.

a) Following the reception or transmission of the beacon, during the ATIM Window, the STA shall transmit a directed ATIM management frame to each STA for which it has one or more buffered unicast MSDUs. If the STA has one or more buffered multicast MSDUs, with the Strictly Ordered bit clear, it shall transmit an appropriately addressed multicast ATIM frame. A STA transmitting an ATIM management frame shall remain awake for the entire current beacon interval.
b) All STAs shall use the backoff procedure defined in 9.2.5.2 for transmission of the first ATIM following the beacon. All remaining ATIMs shall be transmitted using the conventional DCF access procedure.
c) ATIM management frames shall only be transmitted during the ATIM Window.
d) A STA shall transmit no frame types other than RTS, CTS, and ACK Control frames and Beacon and ATIM management frames during the ATIM Window.
e) Directed ATIM management frames shall be acknowledged. If no acknowledgment is received, the ATIM shall be retransmitted using the conventional DCF access procedure. Multicast ATIM management frames shall not be acknowledged.
f) If a STA is unable to transmit an ATIM during the ATIM Window, for example due to contention with other STAs, the STA shall retain the buffered MSDU(s) and attempt to transmit the ATIM during the next ATIM Window.
g) Immediately following the ATIM Window, a STA shall begin transmission of buffered broadcast/multicast frames for which an ATIM was previously transmitted. Following the transmission of any broadcast/multicast frames, any MSDUs and management frames addressed to STAs for which an acknowledgment for a previously transmitted ATIM frame was received shall be transmitted. All STAs shall use the backoff procedure defined in 9.2.5.2 for transmission of the first frame following the ATIM Window. All remaining frames shall be transmitted using the conventional DCF access procedure.

h) A buffered MSDU may be transmitted using fragmentation. If an MSDU has been partially transmitted when the next beacon frame is sent, the STA shall retain the buffered MSDU and announce the remaining fragments by transmitting an ATIM during the next ATIM Window.
i) If an STA is unable to transmit a buffered MSDU during the beacon interval in which it was announced, for example due to contention with other STAs, the STA shall retain the buffered MSDU and announce the MSDU again by transmitting an ATIM during the next ATIM Window.
j) Following the transmission of all buffered MSDUs, a STA may transmit MSDUs without announcement to STAs that are known to be in the Awake state for the current beacon interval due to an appropriate ATIM management or Beacon frame having been transmitted or received.
k) A STA may discard frames buffered for later transmission to power-saving STAs if the STA determines that the frame has been buffered for an excessive amount of time or if other conditions internal to the STA implementation make it desirable to discard buffered frames (for example, buffer starvation). In no case shall a frame be discarded that has been buffered for less than aBeaconPeriod. The algorithm to manage this buffering is beyond the scope of this standard.

11.3 Association and reassociation

This subclause defines how a STA associates and reassociates with an AP.

11.3.1 STA association procedures

Upon the receipt of an MLME-ASSOCIATE.request, a STA shall associate with an AP via the following procedure:

a) The STA shall transmit an association request to an AP with which that STA is authenticated.
b) If an Association Response frame is received with a status value of “successful,” the STA is now associated with the AP and the MLME shall issue an MLME-ASSOCIATE.confirm indicating the successful completion of the operation.
c) If an Association Response frame is received with a status value other than “successful” or the AssociateFailureTimeout expires, the STA is not associated with the AP and the MLME shall issue an MLME-ASSOCIATE.confirm indicating the failure of the operation.

11.3.2 AP association procedures

An AP shall operate as follows in order to support the association of STAs.

a) Whenever an Association Request frame is received from a STA and the STA is authenticated, the AP shall transmit an association response with a status code as defined in 7.3.1.9. If the status value is “successful,” the Association ID assigned to the STA shall be included in the response. If the STA is not authenticated, the AP shall transmit a Deauthentication frame to the STA.
b) When the association response with a status value of “successful” is acknowledged by the STA, the STA is considered to be associated with this AP.
c) The AP shall inform the distribution system (DS) of the association and the MLME shall issue an MLME-ASSOCIATE.indication.

11.3.3 STA reassociation procedures

Upon receipt of an MLME-REASSOCIATE.request, a STA shall reassociate with an AP via the following procedure:

a) The STA shall transmit a Reassociation Request frame to an AP.

b) If a Reassociation Response frame is received with a status value of "successful," the STA is now associated with the AP and the MLME shall issue an MLME-REASSOCIATE.confirm indicating the successful completion of the operation.
c) If a Reassociation Response frame is received with a status value other than "successful" or the ReassociateFailureTimeout expires, the STA is not associated with the AP and the MLME shall issue an MLME-REASSOCIATE.confirm indicating the failure of the operation.

11.3.4 AP reassociation procedures

An AP shall operate as follows in order to support the reassociation of STAs.

a) Whenever a Reassociation Request frame is received from a STA and the STA is authenticated, the AP shall transmit a reassociation response with a status value as defined in 7.3.1.9. If the status value is "successful," the Association ID assigned to the STA shall be included in the response. If the STA is not authenticated, the AP shall transmit a Deauthentication frame to the STA.
b) When the reassociation response with a status value of "successful" is acknowledged by the STA, the STA is considered to be associated with this AP.
c) The AP shall inform the DS of the reassociation and the MLME shall issue an MLME-REASSOCIATE.indication.

11.4 Management information base (MIB) definitions

The description of the MIB in this subclause defines the function of the various managed objects, attributes, actions, and notifications. The ASN.1 encoding of the MIB is presented in Annex D. In case of discrepancy between the definition in this subclause and that in Annex D, the definition in the annex shall take precedence.

11.4.1 MIB summary

The following summarizes the IEEE 802.11 MIB. Each group, attribute, action and notification is listed. This summary is for information purposes only. If any errors exist, the formal definitions have precedence.

11.4.1.1 STA management attributes

11.4.1.1.1 agStationConfiggrp

aStationID,
aMediumOccupancyLimit,
aCFPollable,
aAuthenticationType,
aAuthenticationAlgorithms
aCFPPeriod,
aCFPMaxDuration,
aAuthenticationResponseTimeout,
aWEPUndecryptableCount,
aReceiveDTIMs;

11.4.1.1.2 agPrivacygrp

aPrivacyOptionImplemented,
aPrivacyInvoked,
aWEPDefaultKeys,
aWEPDefaultKeyID,
aWEPKeyMappings

aWEPKeyMappingLength,
aExcludeUnencrypted,
aWEPICVErrorCount,
aWEPExcludedCount;

11.4.1.2 MAC attributes

11.4.1.2.1 agOperationgrp

aMACAddress,
aGroupAddresses,
aRTSThreshold,
aShortRetryLimit,
aLongRetryLimit,
aFragmentationThreshold,
aMaxTransmitMSDULifetime,
aMaxReceiveLifetime,
aManufacturerID,
aProductID;

11.4.1.2.2 agCountersgrp

aTransmittedFragmentCount,
aMulticastTransmittedFrameCount,
aFailedCount,
aRetryCount,
aMultipleRetryCount,
aFrameDuplicateCount,
aRTSSuccessCount,
aRTSFailureCount,
aACKFailureCount,
aReceivedFragmentCount,
aMulticastReceivedFrameCount,
aFCSErrorCount;

11.4.1.3 ResourceTypeID Attributes

11.4.1.3.1 Not grouped

aResourceTypeIDName,
aResourceInfo;

11.4.1.4 Notifications

11.4.1.4.1 SMT notifications

nDisassociate

11.4.2 Managed object class templates

11.4.2.1 SMT object class

11.4.2.1.1 oSMT

```
SMT MANAGED OBJECT CLASS
DERIVED FROM “ISO/IEC 10165-2”:top;
CHARACTERIZED BY
    pSMTbase                                    PACKAGE
        BEHAVIOUR
            bSMTbase BEHAVIOUR
                DEFINED AS “The SMT object class provides the necessary support at the STA to manage
                the processes in the STA such that the STA may work cooperatively as a part of an IEEE
                802.11 network.”;
            ATTRIBUTES
                aStationID                          GET,
                aAuthenticationAlgorithms           GET,
                aAuthenticationType                 GET-REPLACE,
                aPrivacyOptionImplemented           GET,
                aMediumOccupancyLimit               GET-REPLACE,
                aCFPollable                         GET,
                aCFPPeriod                          GET-REPLACE,
                aCFPMaxDuration                     GET-REPLACE,
                aAuthenticationResponseTimeout      GET-REPLACE
                aReceiveDTIMs                       GET-REPLACE;
            ATTRIBUTE GROUPS
                agStationConfiggrp,
                agPrivacygrp;
            NOTIFICATIONS
                nDisassociate;
CONDITIONAL PACKAGES
    pSMTPrivacy PRESENT IF WEP Supported
        BEHAVIOUR
            bSMTPrivacy BEHAVIOUR
                DEFINED AS “The SMTPrivacy package is a set of attributes that shall be present if WEP
                is implemented in the STA.”
            ATTRIBUTES
                aPrivacyInvoked                     GET-REPLACE,
                aWEPDefaultKeys                     REPLACE,
                aWEPDefaultKeyID                    GET-REPLACE,
                aWEPKeyMappings                     REPLACE,
                aWEPKeyMappingLength                GET,
                aExcludeUnencrypted                 GET-REPLACE,
                aWEPICVErrorCount                   GET-REPLACE,
                aWEPExcludedCount                   GET,
                aWEPUndecryptableCount              GET;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) };
```

11.4.2.2 MAC object class

11.4.2.2.1 oMAC

```
MAC MANAGED OBJECT CLASS
DERIVED FROM “ISO/IEC 10165-2”:top;
CHARACTERIZED BY
    pMACbase                                    PACKAGE
        BEHAVIOUR
```

```
        bMACbase BEHAVIOUR
            DEFINED AS "The MAC object class provides the necessary support for the access con-
            trol, generation, and verification of frame check sequences, and proper delivery of valid
            data to upper layers.";
        ATTRIBUTES
            aMACAddress                         GET,
            aGroupAddresses                     GET-REPLACE,
            aTransmittedFragmentCount           GET-REPLACE,
            aMulticastTransmittedFrameCount     GET-REPLACE,
            aFailedCount                        GET-REPLACE,
            aReceivedFragmentCount              GET-REPLACE,
            aMulticastReceivedFrameCount        GET-REPLACE,
            aFCSErrorCount                      GET-REPLACE,
            aRTSThreshold                       GET-REPLACE,
            aShortRetryLimit                    GET-REPLACE,
            aLongRetryLimit                     GET-REPLACE
            aFragmentationThreshold             GET-REPLACE,
            aMaxTransmitMSDULifetime            GET-REPLACE,
            aMaxReceiveLifetime                 GET-REPLACE,
            aManufacturerID                     GET,
            aProductID                          GET;
        ATTRIBUTE GROUPS
            agOperationgrp,
agCountersgrp;
CONDITIONAL PACKAGES
    pMACStatistics PRESENT IF Supported
        BEHAVIOUR
        bMACStatistics BEHAVIOUR
            DEFINED AS "The MACStatistics package provides extended statistical information on
            the operation of the MAC."
        ATTRIBUTES
            aRetryCount                         GET-REPLACE,
            aMultipleRetryCount                 GET-REPLACE,
            aRTSSuccessCount                    GET-REPLACE,
            aRTSFailureCount                    GET-REPLACE,
            aACKFailureCount                    GET-REPLACE,
            aFrameDuplicateCount                GET-REPLACE;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) };
```

11.4.2.3 Resource type object class

11.4.2.3.1 oResourceTypeID

```
ResourceTypeID MANAGED OBJECT CLASS
DERIVED FROM IEEE802CommonDefinitions.oResourceTypeID;
CHARACTERIZED BY
    pResourceTypeID                     PACKAGE
        ATTRIBUTES
            aResourceTypeIDName                 GET,
            aResourceInfo                       GET;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) ResourceTypeID(3) };
```


11.4.3 Attribute group templates

11.4.3.1 STA management attribute group templates

11.4.3.1.1 agStationConfiggrp

```
StationConfiggrp ATTRIBUTE GROUP
    GROUP ELEMENTS
        aStationID,
        aMediumOccupancyLimit,
        aCFPollable,
        aAuthenticationType,
        aAuthenticationAlgorithms,
        aCFPPeriod,
        aCFPMaxDuration,
        aAuthenticationResponseTimeout,
        aWEPUndecryptableCount
        aReceiveDTIMs;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attributeGroup(8)
StationConfiggrp(1) };
```

11.4.3.1.2 agPrivacygrp

```
Privacygrp ATTRIBUTE GROUP
    GROUP ELEMENTS
        aPrivacyOptionImplemented,
        aPrivacyInvoked,
        aWEPDefaultKeys,
        aWEPDefaultKeyID,
        aExcludeUnencrypted,
        aWEPICVErrorCount,
        aWEPExcludedCount;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attributeGroup(8)
Privacygrp(2) };
```

11.4.3.2 MAC attribute group templates

11.4.3.2.1 agOperationgrp

```
Operationgrp ATTRIBUTE GROUP
    GROUP ELEMENTS
        aMACAddress,
        aGroupAddresses,
        aShortRetryLimit,
        aLongRetryLimit,
        aFragmentationThreshold,
        aMaxTransmitMSDULifetime,
        aMaxReceiveLifetime,
        aManufacturerID,
        aProductID;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(0) attributeGroup(8)
Operationgrp(1) };
```

11.4.3.2.2 agCountersgrp

```
Countersgrp ATTRIBUTE GROUP
    GROUP ELEMENTS
```


aTransmittedFragmentCount,
aMulticastTransmittedFrameCount,
aFailedCount,
aRetryCount,
aMultipleRetryCount,
aRTSSuccessCount,
aRTSFailureCount,
aACKFailureCount,
aFrameDuplicateCount,
aReceivedFragmentCount,
aMulticastReceivedFrameCount,
aFCSErrorCount;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(0) attributeGroup(8) Countersgrp(2) };

11.4.4 Attribute templates

11.4.4.1 SMT attribute templates

11.4.4.1.1 aStationID

StationID ATTRIBUTE
DERIVED FROM
IEEE802CommonDefinitions.MACAddress;
BEHAVIOUR DEFINES AS
"This attribute is a value that has the form of a MAC address. It is used for management purposes only to allow an external management entity to uniquely identify a STA."
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) StationID(1) };

11.4.4.1.2 aAuthenticationAlgorithms

AuthenticationAlgorithms ATTRIBUTE
WITH APPROPRIATE SYNTAX
set-of integer;
BEHAVIOUR DEFINED AS
"This attribute shall be a set of all the authentication algorithms supported by the STAs. The following are the default values and the associated algorithm.
Value = 1: Open System
Value = 2: Shared Key";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) AuthenticationAlgorithms(2) };

11.4.4.1.3 aPrivacyOptionImplemented

PrivacyOptionImplemented ATTRIBUTE
WITH APPROPRIATE SYNTAX
boolean;
BEHAVIOUR DEFINED AS
"When this attribute is true, it shall indicate that the IEEE 802.11 WEP option is implemented. The default value of this attribute shall be false.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) PrivacyOptionImplemented(3) };

11.4.4.1.4 aAuthenticationType

AuthenticationType ATTRIBUTE
WITH APPROPRIATE SYNTAX
set of integer;
BEHAVIOUR DEFINED AS
"This attribute shall indicate the authentication algorithms acceptable to the STA during the authentication sequence. The value of this attribute shall be selected from the set in the aAuthenticationAlgorithms attribute. The default value of this attribute shall be {1}.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) AuthenticationType(4) };

11.4.4.1.5 aPrivacyInvoked

PrivacyInvoke ATTRIBUTE
WITH APPROPRIATE SYNTAX
boolean;
BEHAVIOUR DEFINED AS
"When this attribute is true, it shall indicate that the IEEE 802.11 WEP mechanism is used for transmitting frames of type Data. The default value of this attribute shall be false.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) PrivacyInvoked(5) };

11.4.4.1.6 aWEPDefaultKeys

DefaultWEPKey ATTRIBUTE
WITH APPROPRIATE SYNTAX
set of DefaultWEPKey.type;
BEHAVIOUR DEFINED AS
"This attribute shall contain the four default WEP secret key values corresponding to the four possible aWEPDefaultKeyID values. The default value of each of the keys in this attribute shall be null.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) WEPDefaultKeys(6) };

11.4.4.1.7 aWEPDefaultKeyID

WEPDefault ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"This attribute shall indicate the use of the first, second, third, or fourth element of the aDefaultWEPKey array when set to values of zero, one, two, or three. The default value of this attribute shall be zero.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) WEPDefaultKeyID(7) };

11.4.4.1.8 aWEPKeyMappings

WEPKeyMapping ATTRIBUTE
WITH APPROPRIATE SYNTAX
set of ordered triples of type { MAC address, boolean, defaultWEPKey.type };
BEHAVIOUR DEFINED AS
"See 8.3.2 for detailed behaviour.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) WEPKeyMappings(8) };

11.4.4.1.9 aWEPKeyMappingLength

WEPKeyMapping ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The maximum number of tuples that aWEPKeyMappings can hold.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7)
WEPKeyMappingLength(9) };

11.4.4.1.10 aExcludeUnencrypted

ExcludeUnencrypted ATTRIBUTE
WITH APPROPRIATE SYNTAX
boolean;
BEHAVIOUR DEFINED AS
"When this attribute is true, the STA shall not indicate at the MAC service interface received MSDUs that have the WEP subfield of the Frame Control field equal to zero. When this attribute is false, the STA may accept MSDUs that have the WEP subfield of the Frame Control field equal to zero. The default value of this attribute shall be false.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) ExcludeUnencrypted (10) };

11.4.4.1.11 aMediumOccupancyLimit

MediumOccupancyLimit ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"This attribute shall indicate the maximum amount of time, in TU, that a PC may control the usage of the wireless medium without relinquishing control for long enough to allow at least one instance of DCF access to the medium. The default value of this attribute shall be 100. The maximum value of this attribute shall be 1000.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(0) attribute(7)
MediumOccupancyLimit(11) };

11.4.4.1.12 aCFPollable

CFPollable ATTRIBUTE
WITH APPROPRIATE SYNTAX
boolean;
BEHAVIOUR DEFINED AS
"When this attribute is true, it shall indicate that the STA is able to respond to a CF-Poll with a data frame within a SIFS time. This attribute shall be false if the STA is not able to respond to a CF-Poll with a data frame within a SIFS time.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(0) attribute(7) CFPollable(12) };

11.4.4.1.13 aWEPICVErrorCount

ICVErrorCount ATTRIBUTE
DERIVED FROM
"ISO/IEC 10165-2":counter;
BEHAVIOUR DEFINED AS

"This counter shall increment when a frame is received with the WEP subfield of the Frame Control field set to one and the value of the ICV as received in the frame does not match the ICV value that is calculated for the contents of that frame.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) WEPICVErrorCount(13) };

11.4.4.1.14 aWEPExcludedCount

ICVErrorCount ATTRIBUTE
DERIVED FROM
"ISO/IEC 10165-2":counter;
BEHAVIOUR DEFINED AS
"This counter shall increment when a frame is received with the WEP subfield of the Frame Control field set to zero and the value of aExcludedUnencrypted causes that frame to be discarded.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) WEPExcludedCount(14) };

11.4.4.1.15 aCFPPeriod

CFPPeriod ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The attribute shall describe the number of DTIM intervals between the start of CFPs. It is modified by MLME-START.request primitive.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) CFPPeriod(15) };

11.4.4.1.16 aCFPMaxDuration

CFPMaxDuration ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The attribute shall describe the maximum duration of the CFP in TU that may be generated by the PCF. It is modified by MLME-START.request primitive.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) CFPMaxDuration(16) };

11.4.4.1.17 aAuthenticationResponseTimeout

AuthenticationResponseTimeout ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The attribute shall describe the number of TU that a responding STA should wait for the next frame in the authentication sequence.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7)
AuthenticationResponseTimeout(17) };

11.4.4.2 MAC attribute templates

11.4.4.2.1 aMACAddress

MACAddress ATTRIBUTE
DERVIVED FROM

IEEE802CommonDefinitions.MACAddress;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) MACAddress(1) };

11.4.4.2.2 aGroupAddresses

GroupAddresses ATTRIBUTE
WITH APPROPRIATE SYNTAX
set of IEEE802CommonDefinitions.MACAddress;
BEHAVIOUR DEFINED AS
"A set of MACAddresses identifying the multicast addresses, excluding the broadcast address, for which this STA shall receive frames. The default value of this attribute shall be null."
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) GroupAddresses(2) };

11.4.4.2.3 aTransmittedFragmentCount

TransmittedFragmentCount ATTRIBUTE
DERIVED FROM
"ISO/IEC 10165-2":pdusSentCounter;
BEHAVIOUR DEFINED AS
"This counter shall be incremented for each successfully delivered fragment of type Data or Management."
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) TransmittedFragmentCount(3) };

11.4.4.2.4 aMulticastTransmittedFrameCount

MulticastTransmittedFrameCount ATTRIBUTE
DERIVED FROM
"ISO/IEC 10165-2":pdusSentCounter;
BEHAVIOUR DEFINED AS
"This counter shall increment only when the multicast/broadcast bit is set in the destination MAC address of a transmitted frame.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) MulticastTransmittedFrameCount(4) };

11.4.4.2.5 aFailedCount

FailedCount ATTRIBUTE
DERIVED FROM
"ISO/IEC 10165-2":counter;
BEHAVIOUR DEFINED AS
"This counter shall increment when a frame is not transmitted due to the number of transmit attempts exceeding either the aShortRetryLimit or aLongRetryLimit.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) FailedCount(5) };

11.4.4.2.6 aRetryCount

RetryCount ATTRIBUTE
DERIVED FROM
"ISO/IEC 10165-2":counter;
BEHAVIOUR DEFINED AS
"This counter shall increment when a frame is successfully transmitted after one or more retransmissions.";

REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) RetryCount(6) };

11.4.4.2.7 aMultipleRetryCount

MultipleRetryCount ATTRIBUTE
DERIVED FROM
“ISO/IEC 10165-2”:counter;
BEHAVIOUR DEFINED AS
“This counter shall increment when a frame is successfully transmitted after more than one retransmission.”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) MultipleRetryCount(7) };

11.4.4.2.8 aRTSSuccessCount

RTSSuccessCount ATTRIBUTE
DERIVED FROM;
“ISO/IEC 10165-2”:counter
BEHAVIOUR DEFINED AS
“This counter shall increment when a CTS is received in response to an RTS.”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) RTSSuccessCount(8) };

11.4.4.2.9 aRTSFailureCount

RTSFailureCount ATTRIBUTE
DERIVED FROM
“ISO/IEC 10165-2”:counter;
BEHAVIOUR DEFINED AS
“This counter shall increment when a CTS is not received in response to an RTS.”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) RTSFailureCount(9) };

11.4.4.2.10 aACKFailureCount

ACKFailureCount ATTRIBUTE
DERIVED FROM
“ISO/IEC 10165-2”:counter;
BEHAVIOUR DEFINED AS
“This counter shall increment when an ACK is not received when expected.”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) ACKFailureCount(10) };

11.4.4.2.11 aFrameDuplicateCount

FrameDuplicateCount ATTRIBUTE
DERIVED FROM
“ISO/IEC 10165-2”:counter;
BEHAVIOUR DEFINED AS
“This counter shall increment when a frame is received that the Sequence Control field indicates is a duplicate.”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) FrameDuplicateCount(11) };

11.4.4.2.12 aReceivedFragmentCount

ReceivedFragmentCount ATTRIBUTE
DERIVED FROM
"ISO/IEC 10165-2":pdusReceivedCounter;
BEHAVIOUR DEFINED AS
"This counter shall be incremented for each successfully received fragment of type Data or Management."
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) ReceivedFragmentCount(12) };

11.4.4.2.13 aMulticastReceivedFrameCount

MulticastReceivedFrameCount ATTRIBUTE
DERIVED FROM
"ISO/IEC 10165-2":pdusReceivedCounter;
BEHAVIOUR DEFINED AS
"This counter shall increment when a frame is received with the multicast/broadcast bit set in the destination MAC address.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) MulticastReceivedFrameCount(13) };

11.4.4.2.14 aFCSErrorCount

FCSErrorCount ATTRIBUTE
DERIVED FROM
"ISO/IEC 10165-2": CorruptedPDUsReceivedCounter;
BEHAVIOUR DEFINED AS
"This counter shall increment when an FCS error is detected in a received frame.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) FCSErrorCount(14) };

11.4.4.2.15 aRTSThreshold

RTSThreshold ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"This attribute shall indicate the number of bytes in an MPDU, below which an RTS/CTS handshake shall not be performed. An RTS/CTS handshake shall be performed at the beginning of any frame exchange sequence where the MPDU is of type Data or Management, the MPDU has an individual address in the Address1 field, and the length of the MPDU is equal to or larger than this threshold. (For additional details, refer to Table 21 in 9.7.) Setting this attribute to be larger than the maximum MSDU size shall have the effect of turning off the RTS/CTS handshake for frames of Data or Management type transmitted by this STA. Setting this attribute to zero shall have the effect of turning on the RTS/CTS handshake for all frames of Data or Management type transmitted by this STA. The default value of this attribute shall be 3000.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) RTSThreshold(15) };

11.4.4.2.16 aShortRetryLimit

ShortRetryLimit ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;

BEHAVIOUR DEFINED AS
"This attribute shall indicate the maximum number of transmission attempts of a frame, the length of which is less than or equal to aRTSThreshold, that shall be made before a failure condition is indicated. The default value of this attribute shall be 7.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) ShortRetryLimit(16) };

11.4.4.2.17 aLongRetryLimit

LongRetryLimit ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"This attribute shall indicate the maximum number of transmission attempts of a frame, the length of which is greater than aRTSThreshold, that shall be made before a failure condition is indicated. The default value of this attribute shall be 4.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) LongRetryLimit(17) };

11.4.4.2.18 aFragmentationThreshold

FragmentationThreshold ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR
"This attribute shall specify the current maximum size, in octets, of the MPDU that may be delivered to the PHY. An MSDU shall be broken into fragments if its size exceeds the value of this attribute after adding MAC headers and trailers. The default value for this attribute shall be equal to aMPDUMaxLength of the attached PHY and shall never exceed aMPDUMaxLength of the attached PHY. The value of this attribute shall never be less than 256. The default value of this attribute is 2346.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) FragmentationThreshold(18) };

11.4.4.2.19 aMaxTransmitMSDULifetime

MaxTransmitMSDULifetime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The MaxTransmitMSDULifetime shall be the elapsed time in TU, after the initial transmission of an MSDU, after which further attempts to transmit the MSDU shall be terminated. The default value of this attribute shall be 512.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) MaxTransmitMSDULifetime(19) };

11.4.4.2.20 aMaxReceiveLifetime

MaxReceiveLifetime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The MaxReceiveLifetime shall be the elapsed time in TU, after the initial reception of a fragmented MMPDU or MSDU, after which further attempts to reassemble the MMPDU or MSDU shall be terminated. The default value shall be 512.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) MaxReceiveLifetime(20) };

11.4.4.2.21 aWEPUndecryptableCount

ICVErrorCount ATTRIBUTE
DERIVED FROM
"ISO/IEC 10165-2":counter;
BEHAVIOUR DEFINED AS
"This counter shall increment when a frame is received with the WEP subfield of the Frame Control field set to one and the WEPOn value for the key mapped to the TA's MAC address indicates that the frame should not have been encrypted or that frame is discarded due to the receiving STA not implementing the privacy option.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) WEPUndecryptableCount(21) };

11.4.4.2.22 aManufacturerID

ManufacturerID ATTRIBUTE
WITH APPROPRIATE SYNTAX
octet string;
BEHAVIOUR DEFINED AS
"The ManufacturerID shall include, at a minimum, the name of the manufacturer. It may include additional information at the manufacturer's discretion. The default value of this attribute shall be null.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) ManufacturerID(22) };

11.4.4.2.23 aProductID

ProductID ATTRIBUTE
WITH APPROPRIATE SYNTAX
octet string;
BEHAVIOUR DEFINED AS
"The ProductID shall include, at a minimum, an identifier that is unique to the manufacturer. It may include additional information at the manufacturer's discretion. The default value of this attribute shall be null.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) MAC(2) attribute(7) ProductID(23) };

11.4.4.3 Resource type attribute templates

11.4.4.3.1 aResourceTypeIDName

ResourceTypeIDName ATTRIBUTE
DERIVED FROM
IEEE802CommonDefinitions.ResourceTypeIDName;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) ResourceTypeID(3) attribute(7) ResourceTypeIDName(1) };

11.4.4.3.2 aResourceInfo

ResourceInfo ATTRIBUTE
DERIVED FROM
IEEE802CommonDefinitions.ResourceInfo;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) ResourceTypeID(3) attribute(7) ResourceInfo(2) };

11.4.5 Notification templates

11.4.5.1 SMT notification templates

11.4.5.1.1 nDisassociate

Disassociate NOTIFICATION
WITH APPROPRIATE SYNTAX
IEEE802CommonDefinitions.MACAddress;
BEHAVIOUR DEFINED AS
"The disassociate notification shall be sent when the STA receives a Disassociate frame. The value of the notification shall be the BSSID of the BSS from which the Disassociate frame was received.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) notification(10) Disassociate(1) };

12. Physical layer (PHY) service specification

12.1 Scope

The PHY services provided to the IEEE 802.11 wireless LAN MAC are described in this clause. Different PHYs are defined as part of the IEEE 802.11 standard. Each PHY can consist of two protocol functions as follows:

a) A physical layer convergence function, which adapts the capabilities of the physical medium dependent (PMD) system to the PHY service. This function is supported by the physical layer convergence procedure (PLCP), which defines a method of mapping the IEEE 802.11 MAC sublayer protocol data units (MPDU) into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.
b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a wireless medium (WM) between two or more STAs.

Each PMD sublayer may require the definition of a unique PLCP. If the PMD sublayer already provides the defined PHY services, the physical layer convergence function might be null.

12.2 PHY functions

The protocol reference model for the IEEE 802.11 architecture is shown in Figure 11. Most PHY definitions contain three functional entities: the PMD function, the physical layer convergence function, and the layer management function.

The PHY service is provided to the MAC entity at the STA through a service access point (SAP), called the PHY-SAP, as shown in Figure 11. A set of primitives might also be defined to describe the interface between the physical layer convergence protocol sublayer and the PMD sublayer, called the PMD-SAP.

12.3 Detailed PHY service specifications

12.3.1 Scope and field of application

The services provided by the PHY to the IEEE 802.11 MAC are specified in this subclause. These services are describe in an abstract way and do not imply any particular implementation or exposed interface.

12.3.2 Overview of the service

The PHY function as shown in is separated into two sublayers: the PLCP sublayer and the PMD sublayer. The function of the PLCP sublayer is to provide a mechanism for transferring MAC protocol data units (MPDU) between two or more STAs over the PMD sublayer.

12.3.3 Overview of interactions

The primitives associated with communication between the IEEE 802.11 MAC sublayer and the IEEE 802.11 PHY fall into two basic categories:

a) Service primitives that support MAC peer-to-peer interactions
b) Service primitives that have local significance and support sublayer-to-sublayer interactions.

12.3.4 Basic service and options

All of the service primitives described here are considered mandatory unless otherwise specified.

12.3.4.1 PHY-SAP peer-to-peer service primitives

Table 24 indicates the primitives for peer-to-peer interactions.

Table 24—PHY-SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm	Response
PHY-DATA	X	X	X	

12.3.4.2 PHY-SAP sublayer-to-sublayer service primitives

Table 25 indicates the primitives for sublayer-to-sublayer interactions.

Table 25—PHY-SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm	Response
PHY-TXSTART	X		X	
PHY-TXEND	X		X	
PHY-CCARESET	X		X	
PHY-CCA		X		
PHY-RXSTART		X		
PHY-RXEND		X		

12.3.4.3 PHY-SAP service primitives parameters

Table 26 shows the parameters used by one or more of the PMD-SAP service primitives.

Table 26—PHY-SAP service primitive parameters

Parameter	Associated primitive	Value
DATA	PHY-DATA.request PHY-DATA.indication	Octet value X'00'–X'FF'
TXVECTOR	PHY-TXSTART.request	A set of parameters
STATUS	PHY-CCA.indication	BUSY, IDLE
RXVECTOR	PHY-RXSTART.indication	A set of parameters
RXERROR	PHY-RXEND.indication	NoError, FormatViolation, Carrier-Lost, UnsupportedRate

12.3.4.4 Vector descriptions

Several service primitives include a parameter vector. This vector is a list of parameters that may vary depending on the PHY type. Table 27 lists the parameter values required by the MAC or PHY in each of the parameter vectors. Parameters in the vectors that are management rather than MAC may be specific to the PHY and are listed in the clause covering that PHY.

Table 27—Vector descriptions

Parameter	Associate vector	Value
DATARATE	TXVECTOR, RXVECTOR	PHY dependent. The name of the field used to specify the Tx data rate and report the Rx data rate may vary for different PHYs.
LENGTH	TXVECTOR, RXVECTOR	PHY dependent

12.3.5 PHY-SAP detailed service specification

The following subclause describes the services provided by each PHY sublayer primitive.

12.3.5.1 PHY-DATA.request

12.3.5.1.1 Function

This primitive defines the transfer of an octet of data from the MAC sublayer to the local PHY entity.

12.3.5.1.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-DATA.request (DATA)

The DATA parameter is an octet of value X'00' through X'FF'.

12.3.5.1.3 When generated

This primitive is generated by the MAC sublayer to transfer an octet of data to the PHY entity. This primitive can only be issued following a transmit initialization response (PHY-TXSTART.confirm) from the PHY layer.

12.3.5.1.4 Effect of receipt

The receipt of this primitive by the PHY entity causes the PLCP transmit state machine to transmit an octet of data. When the PHY entity receives the octet, it will issue a PHY-DATA.confirm to the MAC sublayer.

12.3.5.2 PHY-DATA.indication

12.3.5.2.1 Function

This primitive indicates the transfer of data from the PHY sublayer to the local MAC entity.

12.3.5.2.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-DATA.indication (DATA)

The DATA parameter is an octet of value X'00' through X'FF'.

12.3.5.2.3 When generated

The PHY-DATA.indication is generated by a receiving PHY entity to transfer the received octet of data to the local MAC entity. The time between receipt of the last bit of the provided octet from the wireless medium and the receipt of this primitive by the MAC entity will be the sum of aRXRFDelay + aRxPLCPDelay.

12.3.5.2.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

12.3.5.3 PHY-DATA.confirm

12.3.5.3.1 Function

This primitive issued by the PHY sublayer to the local MAC entity to confirm the transfer of data from the MAC entity to the PHY sublayer.

12.3.5.3.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-DATA.confirm

This primitive has no parameters.

12.3.5.3.3 When generated

This primitive will be issued by the PHY sublayer to the MAC entity whenever the PLCP has completed the transfer of data from the MAC entity to the PHY sublayer. The PHY sublayer will issue this primitive in response to every PHY-DATA.request primitive issued by the MAC sublayer.

12.3.5.3.4 Effect of receipt

The receipt of this primitive by the MAC will cause the MAC to start the next MAC entity request.

12.3.5.4 PHY-TXSTART.request

12.3.5.4.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity to start the transmission of an MPDU.

12.3.5.4.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-TXSTART.request (TXVECTOR)

The TXVECTOR represents a list of parameters that the MAC sublayer provides the local PHY entity in order to transmit an MPDU. This vector contains both PLCP and PHY management parameters. The required PHY parameters are listed in 12.3.4.4.

12.3.5.4.3 When generated

This primitive will be issued by the MAC sublayer to the PHY entity whenever the MAC sublayer needs to begin the transmission of an MPDU.

12.3.5.4.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity will be to start the local transmit state machine.

12.3.5.5 PHY-TXSTART.confirm

12.3.5.5.1 Function

This primitive issued by the PHY sublayer to the local MAC entity to confirm the start of a transmission. The PHY sublayer will issue this primitive in response to every PHY-TXSTART.request primitive issued by the MAC sublayer.

12.3.5.5.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-TXSTART.confirm

There are no parameters associated with this primitive.

12.3.5.5.3 When generated

This primitive will be issued by the PHY sublayer to the MAC entity whenever the PHY has received a PHY-TXSTART.request from the MAC entity and is ready to begin receiving data octets.

12.3.5.5.4 Effect of receipt

The receipt of this primitive by the MAC entity will cause the MAC to start the transfer of data octets.

12.3.5.6 PHY-TXEND.request

12.3.5.6.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity that the current transmission of the MPDU is completed.

12.3.5.6.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-TXEND.request

There are no parameters associated with this primitive.

12.3.5.6.3 When generated

This primitive will be generated whenever the MAC sublayer has received the last PHY-DATA.confirm from the local PHY entity for the MPDU currently being transferred.

12.3.5.6.4 Effect of receipt

The effect of receipt of this primitive by the local PHY entity will be to stop the transmit state machine.

12.3.5.7 PHY-TXEND.confirm

12.3.5.7.1 Function

This primitive issued by the PHY sublayer to the local MAC entity to confirm the completion of a transmission. The PHY sublayer issues this primitive in response to every PHY-TXEND.request primitive issued by the MAC sublayer.

12.3.5.7.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-TXEND.confirm

There are no parameters associated with this primitive.

12.3.5.7.3 When generated

This primitive will be issued by the PHY sublayer to the MAC entity whenever the PHY has received a PHY-TXEND.request immediately after transmitting the end of the last bit of the last data octet indicating the last data octet has been transferred.

12.3.5.7.4 Effect of receipt

The receipt of this primitive by the MAC entity provides the time reference for the contention backoff protocol.

12.3.5.8 PHY-CCARESET.request

12.3.5.8.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity to reset the clear channel assessment (CCA) state machine.

12.3.5.8.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-CCARESET.request

There are no parameters associated with this primitive.

12.3.5.8.3 When generated

This primitive is generated by the MAC sublayer for the local PHY entity at the end of a NAV timer. This request can be used by some PHY implementations that may synchronize antenna diversity with slot timings.

12.3.5.8.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity is to reset the PLCP CS/CCA assessment timers to the state appropriate for the end of a received frame.

12.3.5.9 PHY-CCARESET.confirm

12.3.5.9.1 Function

This primitive issued by the PHY sublayer to the local MAC entity to confirm that the PHY has reset the CCA state machine.

12.3.5.9.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-CCARESET.request

There are no parameters associated with this primitive.

12.3.5.9.3 When generated

This primitive is issued by the PHY sublayer to the MAC entity whenever the PHY has received a PHY-CCARESET.request

12.3.5.9.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

12.3.5.10 PHY-CCA.indication

12.3.5.10.1 Function

This primitive is an indication by the PHY sublayer to the local MAC entity of the current state of the medium.

12.3.5.10.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-CCA.indication (STATE)

The STATE parameter can be one of two values: BUSY or IDLE. The parameter value is BUSY if the channel assessment by the PHY sublayer determines that the channel is not available. Otherwise, the value of the parameter is IDLE.

12.3.5.10.3 When generated

This primitive is generated every time the status of the channel changes from channel idle to channel busy or from channel busy to channel idle. This includes the period of time when the PHY sublayer is receiving data. The PHY sublayer maintains the channel busy indication until the period indicated by the length field in a valid PLCP Header has expired.

12.3.5.10.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

12.3.5.11 PHY-RXSTART.indication

12.3.5.11.1 Function

This primitive is an indication by the PHY sublayer to the local MAC entity that the PLCP has received a valid start frame delimiter (SFD) and PLCP Header.

12.3.5.11.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-RXSTART.indication (RXVECTOR)

The RXVECTOR represents a list of parameters that the PHY sublayer provides the local MAC entity upon receipt of a valid PLCP Header. This vector may contain both MAC and MAC management parameters. The required parameters are listed in 12.3.4.4.

12.3.5.11.3 When generated

This primitive is generated by the local PHY entity to the MAC sublayer whenever the PHY has successfully validated the PLCP Header error check CRC at the start of a new PLCP PDU.

12.3.5.11.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

12.3.5.12 PHY-RXEND.indication

12.3.5.12.1 Function

This primitive is an indication by the PHY sublayer to the local MAC entity that the MPDU currently being received is completed.

12.3.5.12.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-RXEND.indication (RXERROR)

The RXERROR parameter can convey one or more of the following values: NoError, FormatViolation, CarrierLost, or UnsupportedRate. A number of error conditions may occur after the PLCP's receive state machine has detected what appeared to be a valid preamble and SFD. The following describes the parameter returned for each of those error conditions.

— *NoError*. This value is used to indicate that no error occurred during the receive process in the PLCP.
— *FormatViolation*. This value is used to indicate that the format of the received PLCPPDU was in error.
— *CarrierLost*. This value is used to indicate that during the reception of the incoming MPDU, carrier was lost and no further processing of the MPDU could be accomplished.

— *UnsupportedRate*. This value is used to indicate that during the reception of the incoming PLCP-PDU, a nonsupported date rate was detected.

12.3.5.12.3 When generated

This primitive is generated by the PHY sublayer for the local MAC entity to indicate that the receive state machine has completed a reception with or without errors.

12.3.5.12.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

13. PHY management

The attribute definitions and templates of PHY management are described in this clause. Both the PHY-dependent and the PHY-independent portions of the physical layer management information base (MIB) are included. Not all attributes in the following subclause are supported by every PHY. Each PHY contains a Managed Object list, which defines the PHY-specific values required for each PHY implementation. The ASN.1 encoding of the MIB is presented in Annex D. In any discrepancy between the definition in this clause and that in Annex D, the definition in the annex shall take precedence.

13.1 PHY MIB

13.1.1 PHY attributes

13.1.1.1 agPhyOperationGroup

aPHYType,
aRegDomainsSupported,
aCurrentRegDomain,
aSlotTime,
aCCATime,
aRxTxTurnaroundTime,
aTxPLCPDelay,
aRxTxSwitchTime,
aTxRampOnTime,
aTxRFDelay,
aSIFSTime,
aRxRFDelay,
aRxPLCPDelay,
aMACProcessingDelay,
aTxRampOffTime,
aPreambleLength,
aPLCPHeaderLength,
aMPDUDurationFactor,
aAirPropagationTime,
aTempType,
aCWmin,
aCWmax;

13.1.1.2 agPhyRateGroup

aSupportedDataRatesTx,
aSupportedDataRatesRx;
aMPDUMaxLength;

13.1.1.3 agPhyAntennaGroup

aCurrentTxAntenna,
aDiversitySupport;

13.1.1.4 agPhyTxPowerGroup

aNumberSupportedPowerLevels,
aTxPowerLevel1,

aTxPowerLevel2,
aTxPowerLevel3,
aTxPowerLevel4,
aTxPowerLevel5,
aTxPowerLevel6,
aTxPowerLevel7,
aTxPowerLevel8,
aCurrentTxPowerLevel;

13.1.1.5 agPhyFHSSGroup

aHopTime,
aCurrentChannelNumber,
aMaxDwellTime,
aCurrentSet,
aCurrentPattern,
aCurrentIndex;

13.1.1.6 agPhyDSSSGroup

aCurrentChannel,
aCCAModeSupported,
aCurrentCCAMode,
aEDThreshold;

13.1.1.7 agPhyIRGroup

aCCAWatchdogTimerMax,

aCCAWatchdogCountMax,

aCCAWatchdogTimerMin,

aCCAWatchdogCountMin;

13.1.1.8 agPhyStatusGroup

aSynthesizerLocked;

13.1.1.9 agPhyPowerSavingGroup

aCurrentPowerState,
aDozeTurnonTime;

13.1.1.10 agAntennasList

aSupportedTxAntennas,
aSupportedRxAntennas,
aDiversitySelectionRx;

13.1.2 PHY object class

```
PHY MANAGED OBJECT CLASS
DERIVED FROM "ISO/IEC 10165-2":top;
CHARACTERIZED BY
        pPHYbase        PACKAGE
                BEHAVIOUR
                        bPHYbase BEHAVIOUR
                    DEFINED AS "The PHY object class provides the necessary support for all the
                    required PHY operational information, which may vary from PHY to PHY and from
                    STA to STA, to be communicated to upper layers."
                        ATTRIBUTES

                aPHYType                                GET,
                aRegDomainsSupported                    GET,
                aCurrentRegDomain                       GET-REPLACE,
                aSlotTime                               GET,
                aCCATime                                GET,
                aRxTxTurnaroundTime                     GET,
                aTxPLCPDelay                            GET,
                aRxTxSwitchTime                         GET,
                aTxRampOnTime                           GET,
                aTxRFDelay                              GET,
                aSIFSTime                               GET,
                aRxRFDelay                              GET,
                aRxPLCPDelay                            GET,
                aMACProcessingDelay                     GET,
                aTxRampOffTime                          GET,
                aPreambleLength                         GET,
                aPLCPHeaderLength                       GET,
                aMPDUDurationFactor                     GET,
                aAirPropagationTime                     GET,
                aTempType                               GET,
                aCWmin                                  GET,
                aCWmax                                  GET,
                aSupportedDataRatesTx                   GET,
                aSupportedDataRatesRx                   GET,
                aMPDUMaxLength                          GET,
                 aSupportedTxAntennas                   GET,
                aCurrentTxAntenna                       GET-REPLACE,
                aSupportedRxAntennas                    GET,
                aDiversitySupport                       GET,
                aDiversitySelectionRx                   GET-REPLACE,
                aNumberSupportedPowerLevels             GET,
                aTxPowerLevel1                          GET,
                aTxPowerLevel2                          GET,
                aTxPowerLevel3                          GET,
                aTxPowerLevel4                          GET,
                aTxPowerLevel5                          GET,
                aTxPowerLevel6                          GET,
                aTxPowerLevel7                          GET,
                aTxPowerLevel8                          GET,
                aCurrentTxPowerLevel                    GET-REPLACE,
                aHopTime                                GET,
                aCurrentChannelNumber                   GET-REPLACE,
                aMaxDwellTime                           GET,
                 aCurrentSet                            GET-REPLACE,
                aCurrentPattern                         GET-REPLACE,
```

```
            aCurrentIndex                          GET-REPLACE,
            aCurrentChannel                        GET-REPLACE,
            aCCAModeSupported                      GET,
            aCurrentCCAMode                        GET-REPLACE,
            aEDThreshold                           GET-REPLACE,
            aSynthesizerLocked                     GET,
            aCurrentPowerState                     GET-REPLACE,
            aDozeTurnonTime                        GET;

        ATTRIBUTE GROUPS
            agPhyOperationGroup,
            agPhyRateGroup,
            agPhyAntennaGroup,
            agPhyTxPowerGroup,
            agPhyFHSSGroup,
            agPhyDSSSGroup,
            agPhyIRGroup,
            agPhyStatusGroup,
            agPhyPowerSavingGroup,
            agAntennaListGroup;
        ACTIONS
            acPHYreset;
        NOTIFICATIONS
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) };
```

13.1.3 PHY attribute group templates

13.1.3.1 agPhyOperationGroup

```
PhyOperationGroup ATTRIBUTE GROUP
    GROUP ELEMENTS
        aPHYType,
        aRegDomainsSupported,
        aCurrentRegDomain,
        aSlotTime,
        aCCATime,
        aRxTxTurnaroundTime,
        aTxPLCPDelay,
        aRxTxSwitchTime,
        aTxRampOntime,
        aTxRFDelay,
        aSIFSTime,
        aRxRFDelay,
        aRxPLCPDelay,
        aMACProcessingDelay,
        aTxRampOffTime,
        aPreambleLength,
        aPLCPHeaderLength,
        aMPDUDurationFactor,
        aAirPropagationTime,
        aTempType,
        aCWmin,
        aCWmax;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
    PhyOperationGroup(0) };
```


13.1.3.2 agPhyRateGroup

```
PhyRateGroup ATTRIBUTE GROUP
    GROUP ELEMENTS
        aSupportedDataRatesTx,
        aSupportedDataRatesRx,
        aMPDUMaxLength,
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
    PhyRateGroup(1) };
```

13.1.3.3 agPhyAntennaGroup

```
PhyAntennaGroup ATTRIBUTE GROUP
    GROUP ELEMENTS
        aCurrentTxAntenna,
        aDiversitySupport;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
    PhyAntennaGroup(2) };
```

13.1.3.4 agPhyTxPowerGroup

```
PhyTxPowerGroup ATTRIBUTE GROUP
    GROUP ELEMENTS
        aNumberSupportedPowerLevels,
        aTxPowerLevel1,
        aTxPowerLevel2,
        aTxPowerLevel3,
        aTxPowerLevel4,
        aTxPowerLevel5,
        aTxPowerLevel6,
        aTxPowerLevel7,
        aTxPowerLevel8,
        aCurrentTxPowerLevel;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
    PhyTxPowerGroup(3) };
```

13.1.3.5 agPhyFHSSGroup

```
PhyFHSSGroup ATTRIBUTE GROUP
    GROUP ELEMENTS
        aHopTime,
        aCurrentChannelNumber,
        aMaxDwellTime,
        aCurrentSet,
        aCurrentPattern,
        aCurrentIndex;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
    PhyFHSSGroup(4) };
```

13.1.3.6 agPhyDSSSGroup

```
PhyStatusGroup ATTRIBUTE GROUP
    GROUP ELEMENTS
        aCurrentChannel,
        aCCAModeSupported,
        aCurrentCCAMode,
```

aEDThreshold;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
PhyDSSSGroup(5) };

13.1.3.7 agPhyIRGroup

PhyStatusGroup ATTRIBUTE GROUP
GROUP ELEMENTS
aCCAWatchdogTimerMax,
aCCAWatchdogCountMax,
aCCAWatchdogTimerMin,
aCCAWatchdogCountMin;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
PhyIRGroup(6) };

13.1.3.8 agPhyStatusGroup

PhyStatusGroup ATTRIBUTE GROUP
GROUP ELEMENTS
aSynthesizerLocked;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
PhyStatusGroup(7) };

13.1.3.9 agPhyPowerSavingGroup

PhyStatusGroup ATTRIBUTE GROUP
GROUP ELEMENTS
aCurrentPowerState,
aDozeTurnonTime;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
PhyPowerSavingGroup(8) };

13.1.3.10 agAntennaListGroup

agAntennasList ATTRIBUTE GROUP
GROUP ELEMENTS
aSupportedTxAntennas,
aSupportedRxAntennas,
aDiversitySelectionRx;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
AntennaListGroup(9) };

13.1.4 PHY attribute templates

13.1.4.1 aPHYType

PHYType ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
“This is an 8-bit integer value that identifies the PHY type supported by the attached PLCP and PMD. Currently defined values and their corresponding PHY types are:
FHSS 2.4 GHz = 01 , DSSS 2.4 GHz = 02, IR Baseband = 03”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) PHYType(1) };

13.1.4.2 aRegDomainsSupported

RegDomainsSupported ATTRIBUTE
WITH APPROPRIATE SYNTAX
Null Terminated list of byte integers;
BEHAVIOUR DEFINED AS
"There are different operational requirements dependent on the regulatory domain. This attribute list describes the regulatory domains the PLCP and PMD support in this implementation. Currently defined values and their corresponding Regulatory Domains are:
FCC (USA) = X'10', IC (Canada) = X'20', ETSI (most of Europe) = X'30', Spain = X'31', France = X'32', MKK (Japan) = X'40', list terminator = X'00'";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) RegDomainsSupported(2) };

13.1.4.3 aCurrentRegDomain

CurrentRegDomain ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current regulatory domain that this instance of the PMD is supporting. This octet corresponds to one of the RegDomains listed in aRegDomainsSupported";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CurrentRegDomain(3) };

13.1.4.4 aSlotTime

SlotTime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The time in microseconds that the MAC will use for defining the PIFS and DIFS periods. The SlotTime is defined as a function of the following the equation:
CCATime + RxTxTurnaroundTime + AirPropagationTime + aMACProcessingDelay.
AirPropagationTime is defined as 1 μs.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) SlotTime(4) };

13.1.4.5 aCCATime

CCATime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The minimum time in microseconds that the CCA mechanism has available to assess the medium within every time slot to determine whether the medium is busy or idle";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CCATime(5) };

13.1.4.6 aRxTxTurnaroundTime

RxTxTurnaroundTime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The maximum time in microseconds that the PHY requires to change from receiving to transmitting the start of the first symbol. The following equation is used to derive the RxTxTurnaroundTime:

aTxPLCPDelay + aRxTxSwitchTime + aTxRampOnTime + aTxRFDelay.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) RxTxTurnaroundTime(6) };

13.1.4.7 aTxPLCPDelay

TxPLCPDelay ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The nominal time in microseconds that the PLCP uses to deliver a symbol from the MAC interface to the transmit data path of the PMD";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxPLCPDelay(7) };

13.1.4.8 aRxTxSwitchTime

RxTxSwitchTime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The nominal time in microseconds that the PMD takes to switch from Receive to Transmit";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) RxTxSwitchTime(8) };

13.1.4.9 aTxRampOnTime

TxRampOnTime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The maximum time in microseconds that the PMD takes to turn the Transmitter on";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxRampOnTime(9) };

13.1.4.10 aTxRFDelay

TxRFDelayWITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The nominal time in microseconds between the issuance of a PMD-DATA.request to the PMD and the start of the corresponding symbol at the air interface. The start of a symbol is defined to be 1/2 symbol period prior to the center of the symbol for FH, or 1/2 chip period prior to the center of the first chip of the symbol for DS, or 1/2 slot time prior to the center of the corresponding slot for IR.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxRFDelay(10) };

13.1.4.11 aSIFSTime

SIFSTime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The nominal time in microseconds that the MAC and PHY will require to receive the last symbol of a frame at the air interface, process the frame, and respond with the first symbol on the air interface of the earliest possible response frame. The following equation is used to determine the SIFSTime:
aRxRFDelay + aRxPLCPDelay + aMACProcessingDelay +

aRxTxTurnaroundTime";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) SIFSTime(11) };

13.1.4.12 aRxRFDelay

RxRFDelay ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The nominal time in microseconds between the end of a symbol at the air interface to the issuance of a PMD-DATA.indicate to the PLCP. The end of a symbol is defined to be 1/2 symbol period after the center of the symbol for FH, or 1/2 chip period after the center of the last chip of the symbol for DS, or 1/2 slot time after the center of the corresponding slot for IR.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) RxRFDelay(12) };

13.1.4.13 aRxPLCPDelay

RxPLCPDelay ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The nominal time in microseconds that the PLCP uses to deliver a bit from the PMD receive path to the MAC";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) RxPLCPDelay(13) };

13.1.4.14 aMACProcessingDelay

MACProcessingDelay ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The nominal time in microseconds that the MAC uses to process a frame and prepare a response to the frame";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) MACProcessingDelay(14) };

13.1.4.15 aTxRampOffTime

TxRampOffTime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The nominal time in microseconds that the PMD takes to turn the Transmit Power Amplifier off";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxRampOffTime(15) };

13.1.4.16 aPreambleLength

PreambleLength ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current PHY's Preamble Length in microseconds. If the actual value of the length of the modulated pre-

amble is not an integral number of microseconds, the value shall be rounded up to the next higher value.”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) PreambleLength (16) };

13.1.4.17 aPLCPHeaderLength

PLCPHeaderLength ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
“The current PHY’s PLCP Header Length in microseconds. If the actual value of the length of the modulated header is not an integral number of microseconds, the value shall be rounded up to the next higher value.”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) PLCPHdrLength (17) };

13.1.4.18 aMPDUDurationFactor

MPDUDurationFactor ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
“The overhead added by the PHY to the MPDU as it is transmitted through the wireless medium.”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) MPDUDurationFactor(18) };

13.1.4.19 aAirPropagationTime

AirPropagationTime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
“The anticipated time it takes a transmitted signal to go from the transmitting station to the receiving station.”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) AirPropagationTime(19) };

13.1.4.20 aTempType

TempType ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
“There are different operating temperature requirements dependent on the anticipated environmental conditions. This attribute describes the current PHY’s operating temperature range capability. Currently defined values and their corresponding temperature ranges are:
Type 1 = X'01'—Commercial range of 0 to 40 °C,
Type 2 = X'02'—Industrial range of –20 to 55 °C,
Type 3 = X'03'—Industrial range of –30 to 70 °C.”;
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TempType (20) };

13.1.4.21 aCWmin

CWmin ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The minimum size of the contention window, in units of aSlotTime.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CWmin (21) };

13.1.4.22 aCWmax

CWmax ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The maximum size of the contention window, in units of aSlotTime.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CWmax(22) };

13.1.4.23 aSupportedDataRatesTX

SupportedDataRates ATTRIBUTE
WITH APPROPRIATE SYNTAX
Null Terminated list of byte integers;
BEHAVIOUR DEFINED AS
"The transmit bit rates supported by the PLCP and PMD, represented by a count from X'00'–X'7F', corresponding to data rates in increments of 500 kbit/s from 0 to 63.5 Mbit/s subject to limitations of each individual PHY.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) SupportedDataRatesTX (23) };

13.1.4.24 aSupportedDataRatesRX

SupportedDataRXRates ATTRIBUTE
WITH APPROPRIATE SYNTAX
Null Terminated list of byte integers;
BEHAVIOUR DEFINED AS
"The receive bit rates supported by the PLCP and PMD, represented by a count from X'00'–X'7F', corresponding to data rates in increments of 500 kbit/s from 0 to 63.5 Mbit/s.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) SupportedDataRatesRX(24) };

13.1.4.25 aMPDUMaxLength

MPDUMaxLength ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The maximum number of octets of an MPDU that can be conveyed by a PLCPPDU";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) MPDUMaxLength (25) };

13.1.4.26 aSupportedTxAntennas

SupportedTxAntennas ATTRIBUTE
WITH APPROPRIATE SYNTAX
Null terminated list of integers;
BEHAVIOUR DEFINED AS
"A list of one or more antennas that can be used as the transmit antenna. Each antenna is represented by an integer, starting with antenna 1, and through antenna N, where N ≤ 255.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) SupportedTxAntennas (26) };

13.1.4.27 aCurrentTxAntenna

CurrentTxAntenna ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current antenna being used to transmit. This value is one of the values appearing in aSupportedTx-Antennas.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) MPDUMaxLength1M (27) };

13.1.4.28 aSupportedRxAntennas

SupportedRxAntennas ATTRIBUTE
WITH APPROPRIATE SYNTAX
Null terminated list of integers;
BEHAVIOUR DEFINED AS
"A list of one or more antennas that can be used as the receive antenna. Each antenna is represented by an integer, starting with antenna 1, and through antenna N, where N ≤ 255.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) SupportedRxAntennas (28) };

13.1.4.29 aDiversitySupport

DiversitySupport ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"This implementation's support for diversity, encoded as:
X'01'—diversity is available and is performed over the fixed list of antennas defined in aDiversitySelectionRx.
X'02'—diversity is not supported.
X'03'—diversity is supported and control of diversity is also available, in which case the attribute aDiversitySelectionRx can be dynamically modified by the LME.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) MPDUMaxLength1M (29) };

13.1.4.30 aDiversitySelectionRx

DiversitySelectionRx ATTRIBUTE
WITH APPROPRIATE SYNTAX
Null terminated list of integers;
BEHAVIOUR DEFINED AS
"A list of one or more antennas that can be used as receive antennas. Each antenna is represented by an integer, starting with antenna 1, and through antenna N, where N ≤ 255.";
REGISTERED AS

{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) DiversitySelectionRx (30) };

13.1.4.31 aNumberSupportedPowerLevels

NumberSupportedPowerLevels ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The number of power levels supported by the PMD. This attribute can have a value of 1 to 8.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7)
NumberSupportedPowerLevels (31) };

13.1.4.32 aTxPowerLevel1

TxPowerLevel1 ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The transmit output power for LEVEL1 in mW. This is also the default power level.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxPowerLevel1 (32) };

13.1.4.33 aTxPowerLevel2

TxPowerLevel2 ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The transmit output power for LEVEL2 in mW.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxPowerLevel2 (33) };

13.1.4.34 aTxPowerLevel3

TxPowerLevel3 ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The transmit output power for LEVEL3 in mW.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxPowerLevel3 (34) };

13.1.4.35 aTxPowerLevel4

TxPowerLevel4 ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The transmit output power for LEVEL4 in mW.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxPowerLevel4 (35) };

13.1.4.36 aTxPowerLevel5

TxPowerLevel5 ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The transmit output power for LEVEL5 in mW.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxPowerLevel5 (36) };

13.1.4.37 aTxPowerLevel6

TxPowerLevel6 ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The transmit output power for LEVEL6 in mW.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxPowerLevel6 (37) };

13.1.4.38 aTxPowerLevel7

TxPowerLevel7 ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The transmit output power for LEVEL7 in mW.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxPowerLevel7 (38) };

13.1.4.39 aTxPowerLevel8

TxPowerLevel8 ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The transmit output power for LEVEL8 in mW.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) TxPowerLevel8 (39) };

13.1.4.40 aCurrentTxPowerLevel

CurrentTxPowerLevel ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The TxPowerLevelN currently being used to transmit data. Some PHYs also use this value to determine the receiver sensitivity requirements for CCA.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CurrentTxPowerLevel (40) };

13.1.4.41 aHopTime

HopTime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The time in microseconds for the PMD to change from channel 2 to channel 80.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) HopTime (41) };

13.1.4.42 aCurrentChannelNumber

CurrentChannelNumber ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current channel number of the frequency output by the RF synthesizer.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CurrentChannelNumber (42) };

13.1.4.43 aMaxDwellTime

MaxDwellTime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The maximum time in TU that the transmitter is permitted to operate on a single channel.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) MaxDwellTime (43) };

13.1.4.44 aCurrentSet

CurrentSet ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current set of patterns that the PHY LME is using to determine the hop sequence.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CurrentSet (44) };

13.1.4.45 aCurrentPattern

CurrentPattern ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current pattern that the PHY LME is using to determine the hop sequence.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CurrentPattern (45) };

13.1.4.46 aCurrentIndex

CurrentIndex ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current index value the PHY LME is using to determine the CurrentChannelNumber.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CurrentIndex (46) };

13.1.4.47 aCurrentChannel

CurrentChannel ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current operating frequency channel of the DSSS PHY. Valid channel numbers are as defined in 15.4.6.2.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CurrentChannel (47) };

13.1.4.48 aCCAModeSupported

CCAModeSupported ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"A list of the DSSS PHY CCA modes that are supported. Valid values are:
energy detect only (ED_ONLY) = 01,
carrier sense only (CS_ONLY) = 02,
carrier sense and energy detect (ED_and_CS)= 03.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CCAModeSupported (48) };

13.1.4.49 aCurrentCCAMode

CurrentCCAMode ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current CCA method in operation. Valid values are:
energy detect only (ED_ONLY) = 01,
carrier sense only (CS_ONLY) = 02,
carrier sense and energy detect (ED_and_CS)= 03.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CurrentCCAMode (49) };

13.1.4.50 aEDThreshold

EDThreshold ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current Energy Detect Threshold being used by the DSSS PHY.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) EDThreshold (50) };

13.1.4.51 aSynthesizerLocked

SynthesizerLocked ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"This is an indication that the PMD's synthesizer is locked to the current channel specified by aCurrentChannelNumber. X'00' represents unlocked while X'FF' represents locked.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) SynthesizerLocked (51) };

13.1.4.52 aCurrentPowerState

CurrentPowerState ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The current power state of the PHY. Valid values are ACTIVE=01 and DOZE=02.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CurrentPowerState (52) };

13.1.4.53 aDozeTurnonTime

DozeTurnonTime ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"The time in microseconds required by the PHY to progress from the Doze power down state to the ACTIVE operating state.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) DozeTurnonTime (53) };

13.1.4.54 aCCAWatchdogTimerMax

CCAWatchdogTimerMax ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"This parameter, together with CCAWatchdogCountMax, determines when energy detected in the channel can be ignored.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CCAWatchdogTimerMax (54) };

13.1.4.55 aCCAWatchdogCountMax

CCAWatchdogCountMax ATTRIBUTE
WITH APPROPRIATE SYNTAX
integer;
BEHAVIOUR DEFINED AS
"This parameter, together with CCAWatchdogTimerMax, determines when energy detected in the channel can be ignored.";
REGISTERED AS
{ iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CCAWatchdogCountMax (55) };

13.1.4.56 aCCAWatchdogTimerMin

CCAWatchdogTimerMin ATTRIBUTE
WITH APPROPRIATE SYNTAX
    integer;
BEHAVIOUR DEFINED AS
    "The minimum value to which CCAWatchdogTimerMax can be set.";
REGISTERED AS
    { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CCAWatchdogTimerMin (56) };

13.1.4.57 aCCAWatchdogCountMin

CCAWatchdogCountMin ATTRIBUTE
WITH APPROPRIATE SYNTAX
    integer;
BEHAVIOUR DEFINED AS
    "The minimum value to which CCAWatchdogCountMin can be set.";
REGISTERED AS
    { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attribute(7) CCAWatchdogCountMin (57) };

14. Frequency-Hopping spread spectrum (FHSS) PHY specification for the 2.4 GHz Industrial, Scientific, and Medical (ISM) band

14.1 Overview

14.1.1 Overview of FHSS PHY

The PHY services provided to the IEEE 802.11 wireless LAN MAC for the 2.4 GHz frequency-hopping spread spectrum (FHSS) system are described in this clause. The FHSS PHY consists of the following two protocol functions:

a) A physical layer convergence function, which adapts the capabilities of the physical medium dependent (PMD) system to the PHY service. This function is supported by the physical layer convergence procedure (PLCP), which defines a method of mapping the IEEE 802.11 MAC sublayer protocol data units (MPDUs) into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.
b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a wireless medium (WM) between two or more STAs.

14.1.2 FHSS PHY functions

The 2.4 GHz FHSS PHY architecture is shown in Figure 11. The FHSS PHY contains three functional entities: the PMD function, the physical layer convergence function, and the physical layer management function. Each of these functions is described in detail in the following subclauses.

The FHSS PHY service is provided to the MAC entity at the STA through a PHY service access point (SAP) as shown in called the PHY-SAP, as shown in Figure 11. A set of primitives might also be defined that describe the interface between the physical layer convergence protocol sublayer and the PMD sublayer, called the PMD-SAP.

14.1.2.1 PLCP sublayer

To allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a PHY convergence sublayer is defined. This function simplifies provision of a PHY service interface to the IEEE 802.11 MAC services.

14.1.2.2 Physical layer management entity (PLME)

The PLME performs management of the local PHY functions in conjunction with the MAC management entity.

14.1.2.3 PMD sublayer

The PMD sublayer provides a transmission interface used to send and receive data between two or more STAs.

14.1.3 Service specification method and notation

The models represented by state diagrams in the following subclauses are intended as the primary specifications of the functions provided. It is important to distinguish, however, between a model and a real implementation. The models are optimized for simplicity and clarity of presentation, while any realistic implementation may place heavier emphasis on efficiency and suitability to a particular implementation technology.

The service of a layer or sublayer is the set of capabilities that it offers to a user in the next higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition of service is independent of any particular implementation.

14.2 FHSS PHY specific service parameter lists

14.2.1 Overview

The architecture of the IEEE 802.11 MAC is intended to be PHY independent. Some PHY implementations require medium management state machines running in the MAC sublayer in order to meet certain PMD requirements. These PHY dependent MAC state machines reside in a sublayer defined as the MAC sublayer management entity (MLME). The MLME in certain PMD implementations may need to interact with the physical layer management entity (PLME) as part of the normal PHY-SAP primitives. These interactions are defined by the PLME parameter list currently defined in the PHY Service Primitives as TXVECTOR and RXVECTOR. The list of these parameters and the values they may represent are defined in the specific PHY specifications for each PMD. This subclause addresses the TXVECTOR and RXVECTOR for the FHSS PHY.

All of the values included in the TXVECTOR or RXVECTOR described in this subclause are considered mandatory unless otherwise specified. The 1 Mbit/s and 2 Mbit/s are the only rates currently supported. Other indicated data rates are for possible future use.

14.2.2 TXVECTOR parameters

The parameters in Table 28 are defined as part of the TXVECTOR parameter list in the PHY-TXSTART.request service primitive.

Table 28—TXVECTOR parameters

Parameter	Associate primitive	Value
LENGTH	PHY-TXSTART.request (TXVECTOR)	1–4095
DATARATE	PHY-TXSTART.request (TXVECTOR)	1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5

14.2.2.1 TXVECTOR LENGTH

The LENGTH parameter has the value of 1 to 4095. This parameter is used to indicate the number of octets in the MPDU that the MAC is currently requesting the PHY to transmit. This value is used by the PHY to determine the number of octet transfers that will occur between the MAC and the PHY after receiving a request to start a transmission.

14.2.2.2 TXVECTOR DATARATE

The DATARATE parameter describes the bit rate at which the PLCP should transmit the PSDU. Its value can be any of the rates as defined in Table 28, and supported by the conformant FH PHY.

14.2.3 RXVECTOR parameters

The parameters in Table 29 are defined as part of the RXVECTOR parameter list in the PHY-RXSTART.indicate service primitive.

Table 29—RXVECTOR parameters

Parameter	Associate primitive	Value
LENGTH	PHY-RXSTART.indicate (RXVECTOR)	1–4095
RSSI	PHY-RXSTART.indicate (RXVECTOR)	0–RSSI Max
DATARATE	PHY-RXSTART.request (RXVECTOR)	1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5

14.2.3.1 TRXVECTOR LENGTH

The LENGTH parameter has the value of 1 to 4095. This parameter is used to indicate the value contained in the LENGTH field that the PLCP has received in the PLCP Header. The MAC and PLCP will use this value to determine the number of octet transfers that will occur between the two sublayers during the transfer of the received PSDU.

14.2.3.2 RXVECTOR RSSI

The receive signal strength indicator (RSSI) is an optional parameter that has a value of 0 through RSSI Max. This parameter is a measure by the PHY sublayer of the energy observed at the antenna used to receive the current PPDU. RSSI shall be measured between the beginning of the start frame delimiter (SFD) and the end of the PLCP header error check (HEC). RSSI is intended to be used in a relative manner. Absolute accuracy of the RSSI reading is not specified.

14.3 FHSS PLCP sublayer

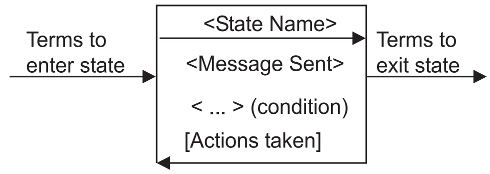
14.3.1 Overview

This subclause provides a convergence procedure to map MPDUs into a frame format designed for FHSS radio transceivers. The procedures for transmission, carrier sense, and reception are defined for single and multiple antenna diversity radios.

14.3.1.1 State diagram notation

The operation of the procedures can be described by state diagrams. Each diagram represents the domain and consists of a group of connected, mutually exclusive states. Only one state is active at any given time. Each state is represented by a rectangle as shown in Figure 68. These are divided into two parts by a horizontal line. In the upper part the state is identified by a name. The lower part contains the name of any signal that is generated. Actions are described by short phrases and enclosed in brackets.

Each permissible transition between the states is represented graphically by an arrow from the initial to the terminal state. A transition that is global in nature (for example, an exit condition from all states to the IDLE or RESET state) is indicated by an open arrow. Labels on transitions are qualifiers that must be fulfilled before the transition will be taken. The label UCT designates an unconditional transition. Qualifiers described by short phrases are enclosed in parentheses.



Key:

( )	=	condition, for example, (if no_collision)
[ ]	=	action, for example, [resetPLSfunctions]
*	=	logical AND
+	=	logical OR
UCT	=	unconditional transition

Figure 68—State diagram notation example

State transitions and sending and receiving of messages occur instantaneously. When a state is entered and the condition to leave that state is not immediately fulfilled, the state executes continuously, sending the messages and executing the actions contained in the state in a continuous manner.

Some devices described in this standard are allowed to have two or more ports. State diagrams capable of describing the operation of devices with an unspecified number of ports require qualifier notation that allows testing for conditions at multiple ports. The notation used is a term that includes a description in parentheses of which ports must meet the term for the qualifier to be satisfied (e.g., ANY and ALL). It is also necessary to provide for term-assignment statements that assign a name to a port that satisfies a qualifier. The following convention is used to describe a term-assignment statement that is associated with a transition:

a) The character “:” (colon) is a delimiter used to denote that a term assignment statement follows.
b) The character “<” (left arrow) denotes assignment of the value following the arrow to the term preceding the arrow.

The state diagrams contain the authoritative statement of the procedures they depict; when apparent conflicts between descriptive text and state diagrams arise, the state diagrams are to take precedence. This does not, however, override any explicit description in the text that has no parallel in the state diagrams.

The models presented by state diagrams are intended as the primary specifications to be provided. It is important to distinguish, however, between a model and a real implementation. The models are optimized for simplicity and clarity of presentation, while any realistic implementation may place heavier emphasis on efficiency and suitability to a particular implementation technology. It is the functional behavior of any unit that must match the standard, not its internal structure. The internal details of the model are useful only to the extent that they specify the external behavior clearly and precisely.

14.3.2 PLCP frame format

The PLCP protocol data unit (PPDU) frame format provides for the asynchronous transfer of MAC sublayer MPDUs from any transmitting STA to all receiving STAs within the wireless LAN’s BSS. The PPDU illustrated in Figure 69 consists of three parts: a PLCP Preamble, a PLCP Header, and a PSDU. The PLCP Preamble provides a period of time for several receiver functions. These functions include antenna diversity, clock and data recovery, and field delineation of the PLCP Header and the PSDU. The PLCP Header is used to specify the length of the whitened PSDU field and support any PLCP management information. The PPDU contains the PLCP Preamble, the PLCP Header, and the PSDU modified by the PPDU data whitener.

PLCP Preamble		PLCP Header			Whitened PSDU
Sync	Start Frame Delimiter	PLW	PSF	Header Error Check	
80 bits	16 bits	12 bits	4 bits	16 bits	Variable number of octets

Figure 69—PLCP frame format

14.3.2.1 PLCP Preamble

The PLCP Preamble contains two separate subfields; the Preamble Synchronization (SYNC) field and the Start Frame Delimiter (SFD), to allow the PHY circuitry to reach steady-state demodulation and synchronization of bit clock and frame start.

14.3.2.1.1 Preamble SYNC field

The Preamble SYNC field is an 80-bit field containing an alternating zero-one pattern, transmitted starting with zero and ending with one, to be used by the PHY sublayer to detect a potentially receivable signal, select an antenna if diversity is utilized, and reach steady-state frequency offset correction and synchronization with the received packet timing.

14.3.2.1.2 Start Frame Delimiter (SFD)

The SFD consists of the 16-bit binary pattern 0000 1100 1011 1101 (transmitted leftmost bit first). The first bit of the SFD follows the last bit of the sync pattern. The SFD defines the frame timing.

14.3.2.2 PLCP Header field

The PLCP Header field contains three separate subfields: a 12-bit PSDU Length Word (PLW), a 4-bit PLCP Signaling field (PSF), and a 16-bit PLCP HEC field.

14.3.2.2.1 PSDU length word

The PSDU length word (PLW) is passed from the MAC as a parameter within the PHY-TXSTART.request primitive. The PLW specifies the number of octets contained in the PSDU. Its valid values are X'001'–X'FFF', representing counts of one to 4095 octets. The PLW is transmitted lsb first and msb last. The PLW is used by the receiving STA, in combination with the 32/33 coding algorithm specified in this clause, to determine the last bit in the packet.

14.3.2.2.2 PLCP Signaling field (PSF)

The 4-bit PSF is defined in Table 30. The PSF is transmitted bit 0 first and bit 3 last.

Table 30—PSF bit descriptions

Bit	Parameter name	Parameter values	Description
0	Reserved	Default = 0	Reserved
1:3	PLCP_BITRATE	b1 b2 b3 = Data Rate 0 0 0 = 1.0 Mbit/s, 0 0 1 = 1.5 Mbit/s, 0 1 0 = 2.0 Mbit/s, 0 1 1 = 2.5 Mbit/s, 1 0 0 = 3.0 Mbit/s, 1 0 1 = 3.5 Mbit/s, 1 1 0 = 4.0 Mbit/s, 1 1 1 = 4.5 Mbit/s	This field indicates the data rate of the whitened PSDU from 1 Mbit/s to 4.5 Mbit/s in 0.5 Mbit/s increments.

14.3.2.2.3 Header Error Check (HEC) field

The HEC field is a 16-bit CCITT CRC-16 error detection field. The HEC uses the CCITT CRC-16 generator polynomial G(x) as follows:

$$G(x) = x^{16}+x^{12}+x^{5}+1$$

The HEC shall be the one's complement of the sum (modulo 2) of the following:

a) The remainder of $x^k \times (x^{15} + x^{14} + \ldots + x^2 + x^1 + 1)$ divided (modulo 2) by $G(x)$, where k is the number of bits in the PSF and PLW fields of the PLCP Header;
b) The remainder after multiplication by x^{16} and then division (modulo 2) by $G(x)$ of the content (treated as a polynomial) of the PSF and PLW fields.

The HEC shall be transmitted with the coefficient of the highest term first.

As a typical implementation, at the transmitter, the initial remainder of the division is preset to all ones and is then modified by division of the PSF and PLW fields by the generator polynomial, $G(x)$. The one's complement of this remainder is inserted in the HEC field with the most significant bit transmitted first.

At the receiver, the initial remainder of the division is again preset to all ones. The division of the received PSF, PLW, and HEC fields by the generator polynomial, $G(x)$, results, in the absence of transmission errors, in a unique nonzero value, which is the following polynomial $R(x)$:

$$R(x) = x^{12} + x^{11} + x^{10} + x^8 + x^3 + x^2 + x^1 + 1$$

14.3.2.3 PLCP data whitener

The PLCP data whitener uses a length-127 frame-synchronous scrambler followed by a 32/33 bias-suppression encoding to randomize the data and to minimize the data dc bias and maximum run lengths. Data octets are placed in the transmit serial bit stream lsb first and msb last. The frame synchronous scrambler uses the generator polynomial $S(x)$ as follows:

$S(x) = x^7 + x^4 + 1$

and is illustrated in Figure 70. The 127-bit sequence generated repeatedly by the scrambler is (leftmost bit used first) 00001110 11110010 11001001 00000010 00100110 00101110 10110110 00001100 11010100 11100111 10110100 00101010 11111010 01010001 10111000 1111111. The same scrambler is used to scramble transmit data and to descramble receive data. The data whitening starts with the first bit of the PSDU, which follows the last bit of the PLCP Header. The specific bias suppression encoding and decoding method used is defined in Figure 74 and Figure 79. The format of the packet after data whitening is as shown in Figure 71.

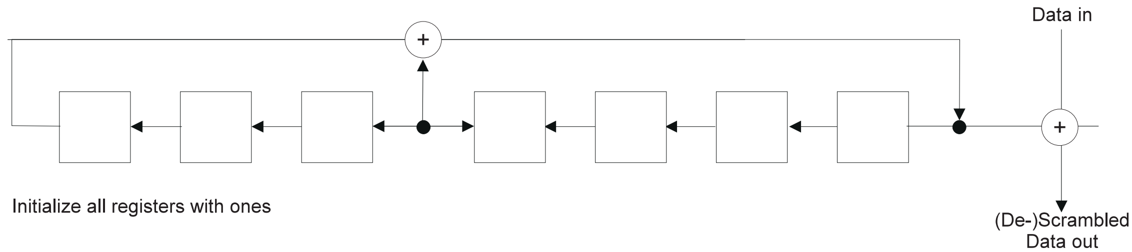


Figure 70—Frame synchronous scrambler/descrambler

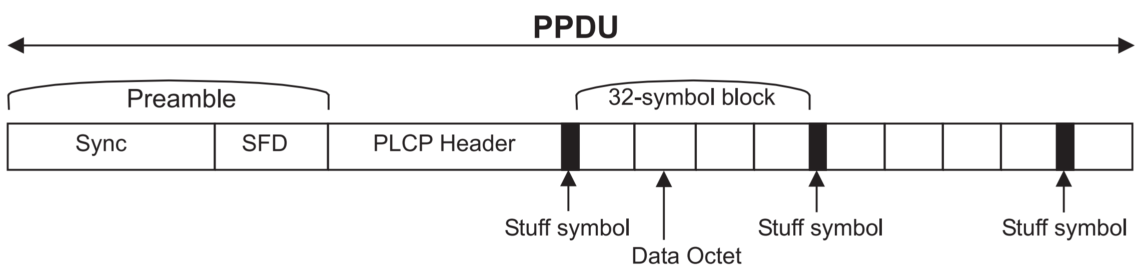


Figure 71—PLCP data whitener format

14.3.3 PLCP state machines

The PLCP consists of three state machines, as illustrated in the overview diagram of Figure 72: the transmit (TX), carrier sense/clear channel assessment (CS/CCA), and receive (RX) state machines. The three PLCP state machines are defined in the subclauses below; Figure 72 is not a state diagram itself. Execution of the PLCP state machines normally is initiated by the FH PLME state machine and begins at the CS/CCA state machine. The PLCP returns to the FH PLME state machine upon interrupt to service a PLME service request, such as PLME-SET, PLME-RESET, etc.

14.3.3.1 PLCP transmit procedure

The PLCP transmit procedure is invoked by the CS/CCA procedure immediately upon receiving a *PHY-TXSTART.request(TXVECTOR)* from the MAC sublayer. The CSMA/CA protocol is performed by the MAC with the PHY PLCP in the CS/CCA procedure prior to executing the transmit procedure.

14.3.3.1.1 Transmit state machine

The PLCP transmit state machine illustrated in Figure 73 includes functions that must be performed prior to, during, and after PPDU data transmission. Upon entering the transmit procedure in response to a *PHY-TXSTART.request (TXVECTOR)* from the MAC, the PLCP shall switch the PHY PMD circuitry from

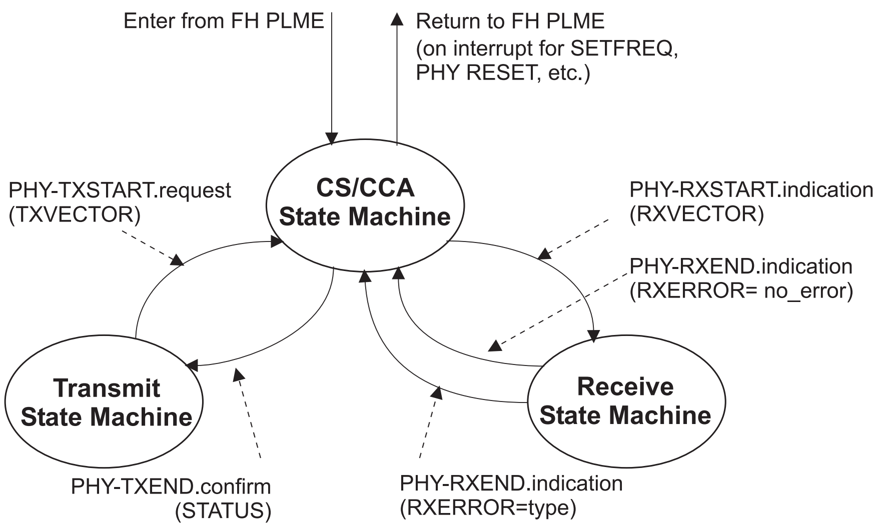


Figure 72—PLCP top-level state diagram

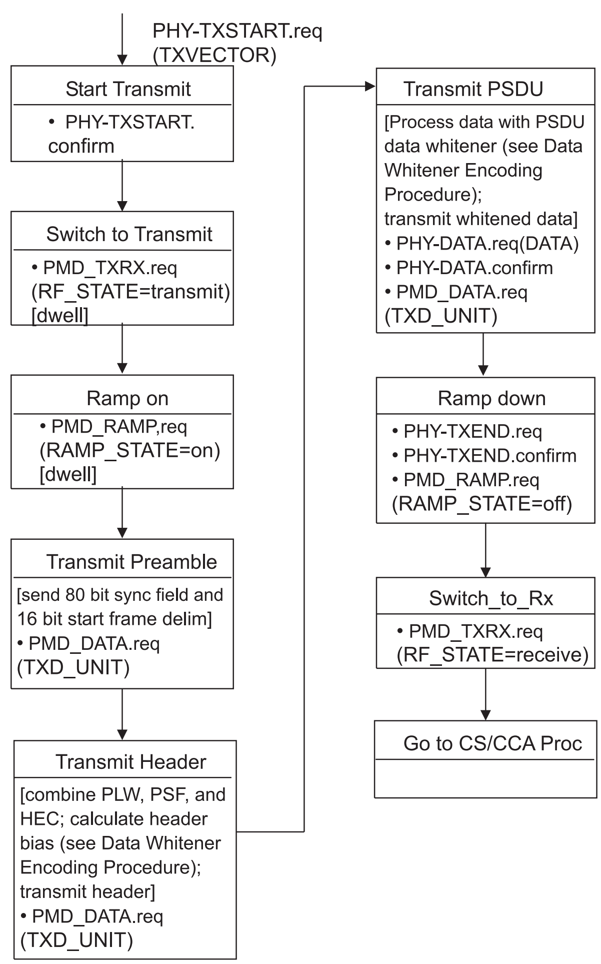


Figure 73—Transmit state machine

receive to transmit state; ramp on the transmit power amplifier in the manner prescribed in 14.6; and transmit the preamble sync pattern and SFD. The PLCP shall generate the PLCP Header as defined in 14.3.2.2 in sufficient time to send the bits at their designated bit slot time. The PLCP shall add the PLCP Header to the start of the PSDU data.

Prior to transmitting the first PSDU data bit, the PLCP shall send a *PHY-TXSTART.confirm* message to the MAC indicating that the PLCP is ready to receive an MPDU data octet. The MAC will pass an MPDU data octet to the PHY with a *PHY-DATA.request(DATA)*, which the PHY will respond to with a *PHY-DATA.confirm*. This sequence of *PHY-DATA.request(DATA)* and *PHY-DATA.confirm* shall be executed until the last data octet is passed to the PLCP. During transmission of the PSDU data, each bit of the PSDU shall be processed by the data whitener algorithm defined in Figure 74 and described in 14.3.2.3. Each PSDU data octet is processed and transmitted lsb first and msb last.

```
Data whitener encoding algorithm:

/*        If msb of stuff symbol = 1 then the next block is inverted; 0 = not inverted */
/*        Accumulate PLCP Header; begin stuffing on first bit of the PSDU */

/********* Calculate number of 32-symbol BSE blocks required to send PSDU;
                no padding is necessary when the number of symbols is not a multiple of 32 *********/
Input parameter: number_of_PSDU_octets, rate;                 /* rate is 1 or 2*/
number_of_symbols= (number_of_PSDU_octets *8) /rate;
number_of_blocks_in_packet = truncate{(number_of_symbols + 31) / 32)};

/********* Accumulate the bias in the header to use in calculating the inversion state of the first
                block of PSDU data *********/
Read in header {b(1),...,b(32)};                 /* b(1) is first bit in */
header_bias = Sum{weight(b(1)),...,weight(b(32))};
                        /* calculate bias in header; weights are defined in Table 31*/
Transmit {b(1),...,b(32)};                       /* no stuffing on header */
accum=header_bias;                               /* initialize accum */
Initialize scrambler to all ones;

/********* Whiten the PSDU data with scrambler and BSE encoder *********/
For n = 1 to number_of_blocks_in_packet
{
        b(0) = 0 for 1 Mbit/s; b(0)=00 for 2 Mbit/s;            /* b(0) is the stuff symbol */
        N = min(32, number_of_symbols);                         /* N= block size in symbols */
        Read in next symbol block {b(1),...,b(N)};              /* b(n) = {0,1} or {0,1,2,3};
        1 - 8 octets, use PHY-DATA.req(DATA), PHY-DATA.confirm for each octet*/
        Scramble {b(1),...,b(N)};                                   /* see 14.3.2.3*/
        bias_next_block = Sum{weight(b(0)),...,weight(b(N))};       /* calculate bias with b(0)=0 */

        /***** if accum and bias of next block has the same sign, then invert block;
                if accum=0 or bias_next_block=0, don't invert *****/
        If {[accum * bias_next_block > 0] then
         {
                Invert {b(0),...,b(N)};         /* Invert deviation, or, negate msb of symbol */
                bias_next_block = - bias_next_block;
         }

        accum = accum + bias_next_block;
        transmit {b(0),...,b(N)};                       /* b(0) is first symbol out */
        number_of_symbols = number_of_symbols - N
}
```

Figure 74—Data whitener encoding procedure

After the last MPDU octet is passed to the PLCP, the MAC will indicate the end of the frame with a *PHY-TXEND.request*. After the last bit of the PSDU data has completed propagation through the radio and been transmitted into the air, the PLCP shall complete the transmit procedure by sending a *PHY-TXEND.confirm* to the MAC sublayer, ramp off the power amplifier in the manner prescribed in 14.6, and switch the PHY PMD circuitry from transmit to receive state. The execution shall then return to the CS/CCA procedure.

The weights assigned to each value of the symbols are defined in Table 31 for the 1 Mbit/s (2GFSK) and 2 Mbit/s (4GFSK) symbols.

Table 31—PLCP field bit descriptions

2GFSK	4GFSK	Weight
	10	3
1		2
	11	1
Center	Center	0
	01	–1
0		–2
	00	–3

14.3.3.1.2 Transmit state timing

The transmit timing illustrated in Figure 75 is defined from the instant that the *PHY-TXSTART.request(TXVECTOR)* is received from the MAC sublayer. The PLCP shall switch the PMD circuitry from receive to transmit, turn on and settle the transmitter, and begin transmitting the first bit of the preamble at the antenna within a maximum of 20 µs of receiving the *PHY-TXSTART.request(TXVECTOR)*. The PLCP Preamble shall be transmitted at 1 Mbit/s and be completed in 96 µs. The PLCP Header shall be transmitted at 1 Mbit/s and be completed in 32 µs. The variable length PSDU shall be transmitted at the selected data rate. After the last bit of the PSDU data has completed propagation through the radio and been transmitted onto the air, the PLCP shall send the *PHY-TXEND.confirm* to the MAC sublayer. The PLCP shall turn off the transmitter, reducing the output energy to less than the specified off-mode transmit power within the time specified in 14.6. At the end of the power amplifier ramp down period, the PLCP shall switch the PMD circuitry from transmit to receive.

14.3.3.2 Carrier sense/clear channel assessment (CS/CCA) procedure

The PLCP CS/CCA procedure is executed while the receiver is turned on and the STA is not currently receiving or transmitting a packet. The CS/CCA procedure is used for two purposes: to detect the start of a network signal that can be received (CS) and to determine whether the channel is clear prior to transmitting a packet (CCA).

14.3.3.2.1 CS/CCA state machine

Timing for priority (PIFS, DIFS), contention backoff (slot times), and CS/CCA assessment windows are defined relative to the end of the last bit of the last packet on the air. The CS/CCA state machine is shown in Figure 76. The PLCP shall perform a CS/CCA assessment on a minimum of one antenna within a MAC contention backoff slot time of 50 µs. The PLCP shall be capable of detecting within the slot time an FH PHY conformant signal that is received at the selected antenna up to 22 µs after the start of the slot time with the synchronous detection performance specified in 14.6.15.3. Subclause 14.6.15.3 specifies detection performance with zero-one sync patterns and with random data patterns. If a start of a transmission is asynchronous with the BSS and arrives after the start of the slot but at least 16 µs prior to the end of the slot, the PLCP shall indicate a busy channel prior to the end of the slot time with the asynchronous detection performance specified

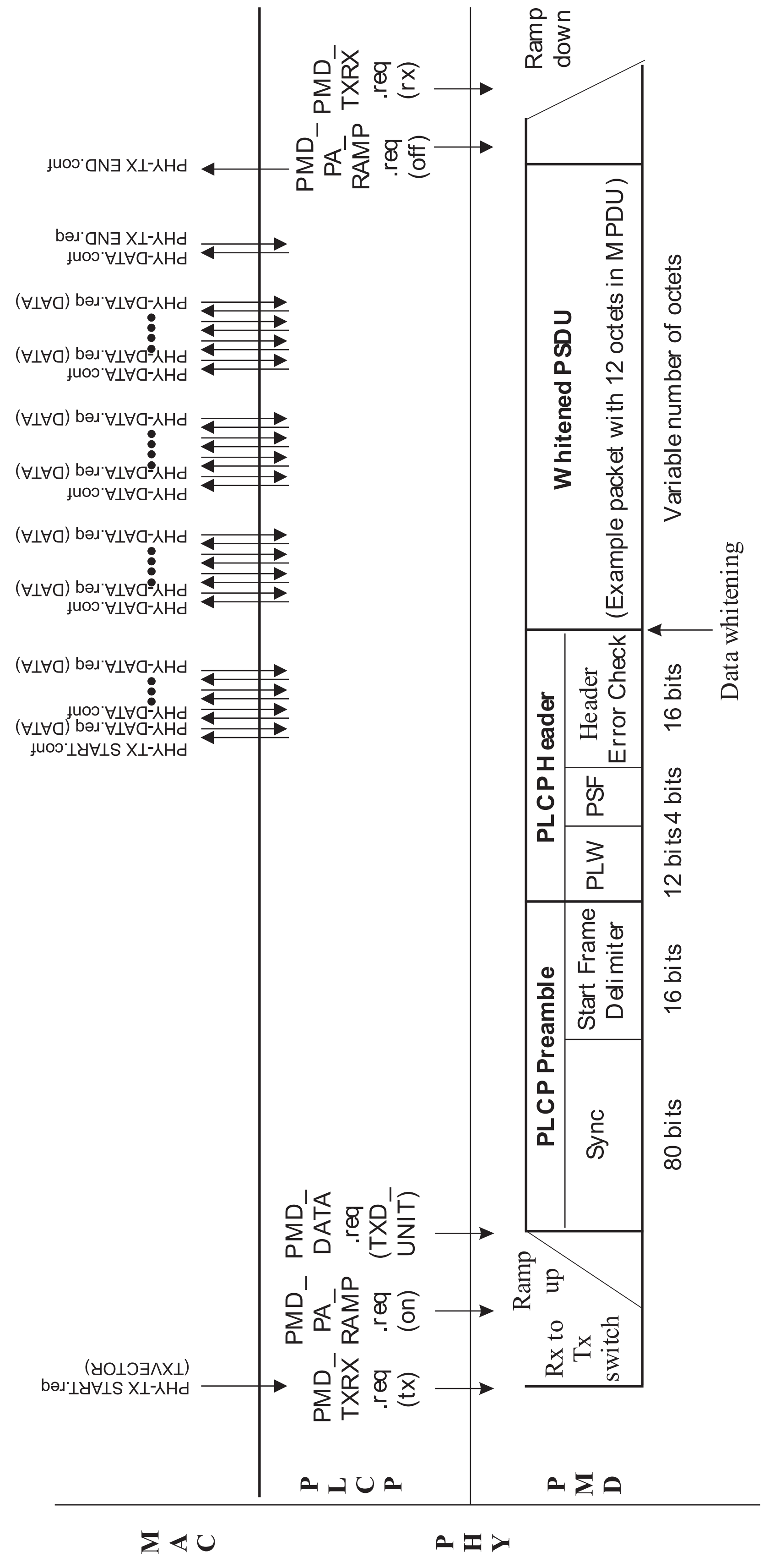


Figure 75—Transmit state timing

in 14.6.15.3. The CCA indication immediately prior to transmission shall be performed on an antenna with essentially the same free space gain and gain pattern as the antenna to be used for transmission. The method of determining CS/CCA is unspecified except for the detection performance of a conformant method as specified in 14.6.15.3.

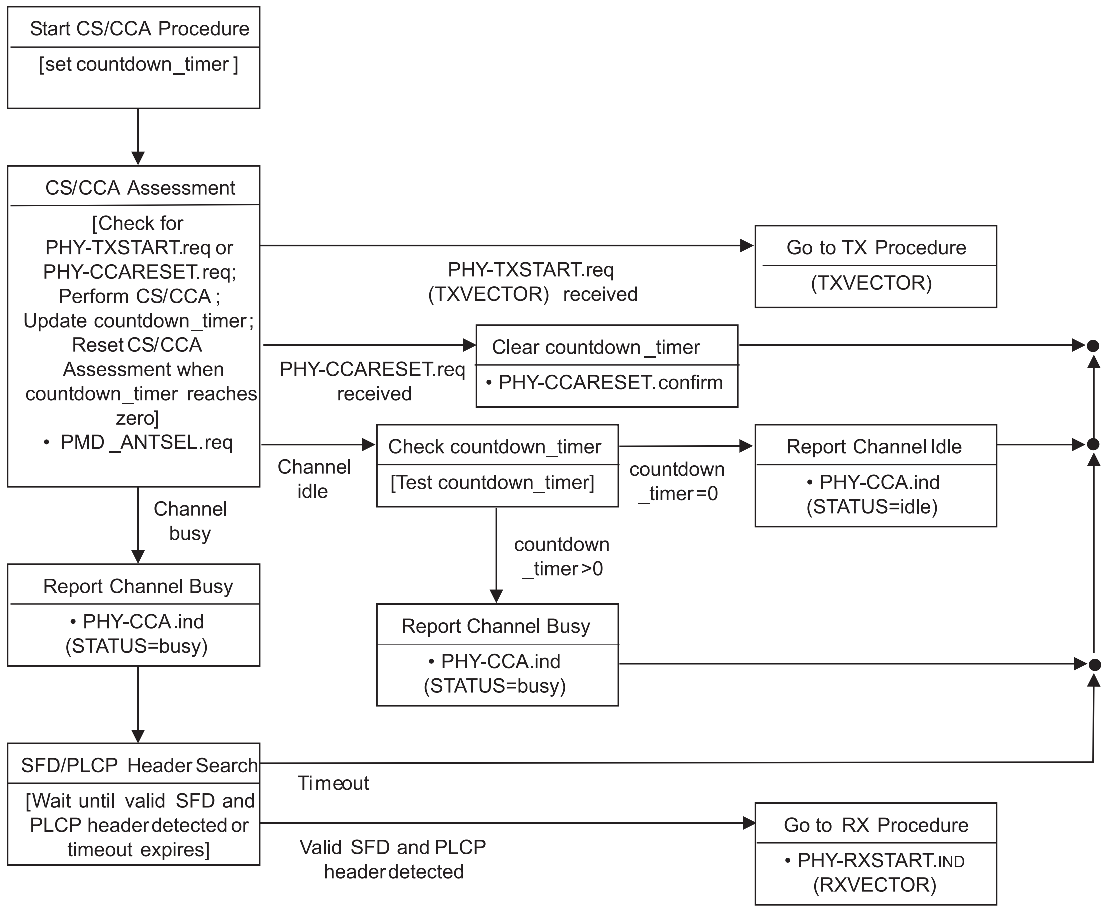


Figure 76—CS/CCA state machine

If a *PHY-TXSTART.request (TXVECTOR)* is received, the CS/CCA procedure shall exit to the transmit procedure within 1 µs. If a *PHY-CCARESET.request* is received, the PLCP shall reset the CS/CCA state machine to the state appropriate for the end of a complete received frame. This service primitive is generated by the MAC at the end of a NAV period. The PHY shall indicate completion of the request by sending a *PHY-CCARESET.confirm* to the MAC.

If a CS/CCA assessment returns a channel idle result, the PHY shall send a *PHY-CCA.indicate(STATUS=idle)* to the MAC.

If a CS/CCA assessment returns a channel busy result, the PHY shall send a *PHY-CCA.indicate(STATUS=busy)* to the MAC. Upon a channel busy assessment, the PLCP shall stop any antenna switching prior to the earliest possible arrival time of the SFD and detect a valid SFD and PLCP Header if received. A valid PLCP Header is defined as containing valid PLCP Length Word and PHY Signaling field values and a valid HEC field. If a valid SFD/PLCP Header is detected, the CS/CCA procedure shall send a PHY-RXSTART.indicate(RXVECTOR) message to the MAC sublayer and exit to the receive procedure. The PLCP shall dwell and search for the SFD/PLCP Header for a minimum period longer than the latest possible arrival time of the SFD/PLCP Header. Indication of a busy channel does not necessarily lead to the successful reception of a frame.

The octet/bit count remaining may be a nonzero value when returning from the receive procedure if a signal in the process of being received was lost prior to the end as determined from the Length field of a valid PLCP Header. The countdown timer shall be set to the octet/bit count and used to force the CS/CCA indication to remain in the BUSY state until the predicted end of the frame regardless of actual CS/CCA indications.

However, if the CS/CCA procedure indicates the start of a new frame within the countdown timer period, it is possible to transition to the receive procedure prior to the end of the countdown timer period. If the PHY transitions to receive under these conditions, the countdown timer shall be reset to the longer of (1) the remaining time of the current frame and (2) the length of the new frame.

When a nonzero countdown timer reaches zero, the PLCP shall reset the CS/CCA state machine to the state appropriate for the end of a complete received frame and the CS/CCA indication shall reflect the state of the channel.

If the receive procedure encountered an unsupported rate error, the PLCP shall keep the CS/CCA state at Busy for the duration of the frame by setting the countdown timer to the value corresponding to the calculated time based on the information in the PLCP Header and the 33/32 expansion factor.

14.3.3.2.2 CS/CCA state timing

Timing for priority (PIFS, DIFS), contention backoff (slot times), and CS/CCA assessment windows is defined relative to the end of the last bit of the last packet on the air. The PLCP shall perform a CS/CCA assessment on a minimum of one antenna within a slot time. The appropriate CS/CCA indication shall be available prior to the end of each 50 μs slot time with the performance specified in 14.6. See Figure 77.

If a STA has not successfully received the previous packet, the perceived packet end time and slot boundary times will have a higher uncertainty for that STA.

14.3.3.3 PLCP receive procedure

The PLCP receive procedure is invoked by the PLCP CS/CCA procedure upon detecting a portion of the preamble sync pattern followed by a valid SFD and PLCP Header.

14.3.3.3.1 Receive state machine

The PLCP receive procedure shown in Figure 78 includes functions that must be performed while the PPDU is being received. The PLCP receive procedure begins upon detection of a valid SFD and PLCP Header in the CS/CCA procedure. The PLCP shall set a PPDU octet/bit counter to indicate the last bit of the packet, receive the PPDU bits, and perform the data whitening decoding procedure shown in Figure 79 on each PPDU bit. The PLCP shall pass correctly received data octets to the MAC with a series of *PHY-DATA.indicate(DATA)*. After the last PPDU bit is received and the last octet is passed to the MAC, the PLCP shall send a *PHY-RXEND.indicate(RXERROR=no_error)* to the MAC sublayer. Upon error-free completion of a packet reception, the PLCP shall exit the receive procedure and return to the PLCP CS/CCA procedure with the octet/bit count set to 0.

If the PLCP Header was decoded without a CRC error but encountered an unsupported rate, then the PLCP shall immediately complete the receive procedure with a *PHY-RXEND.indicate (RXERROR = unsupported_rate)* to the MAC, and return to the CS/CCA procedure with the octet/bit count remaining and the data rate value contained in the PLCP Header.

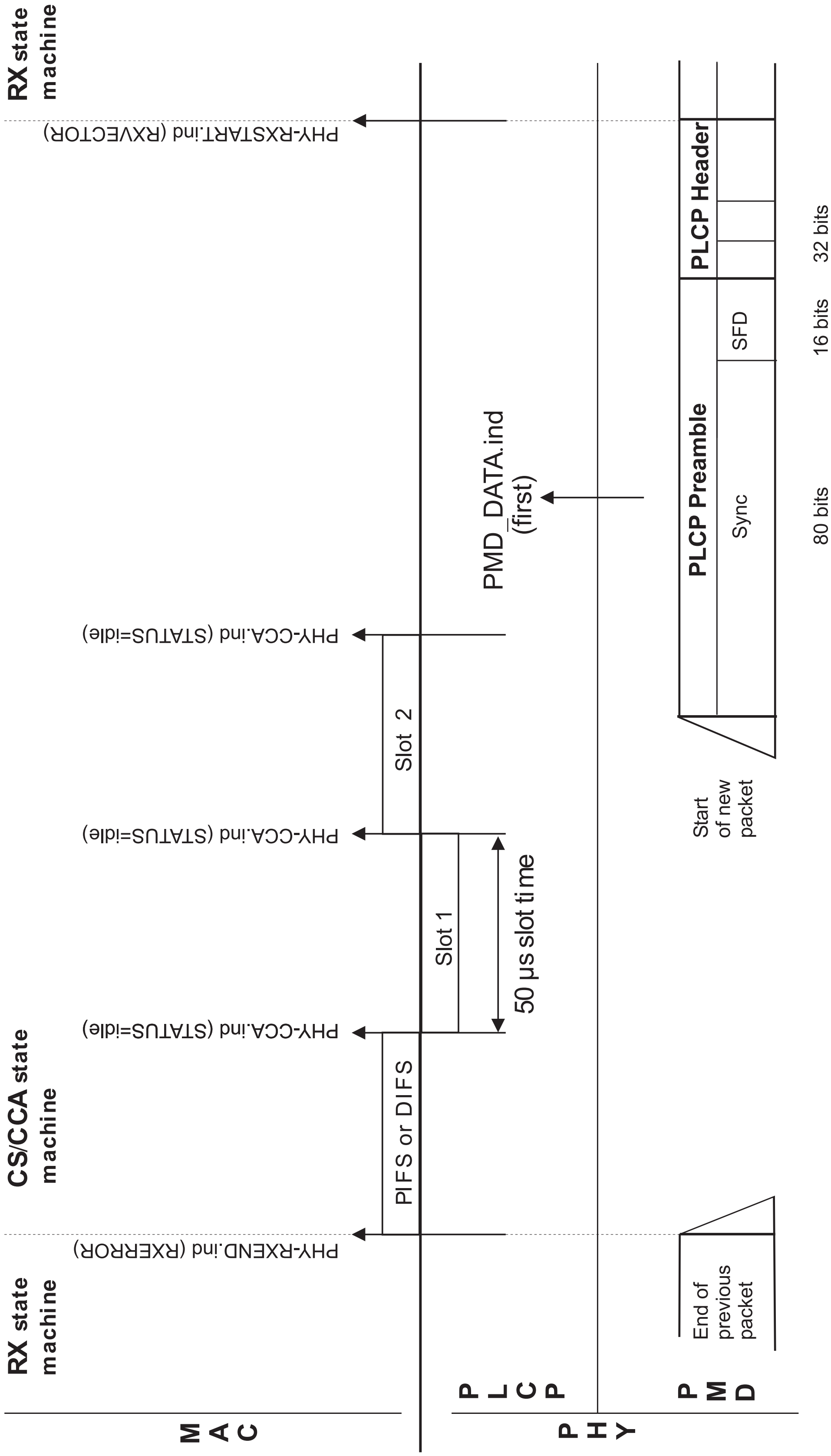


Figure 77—CS/CCA state timing

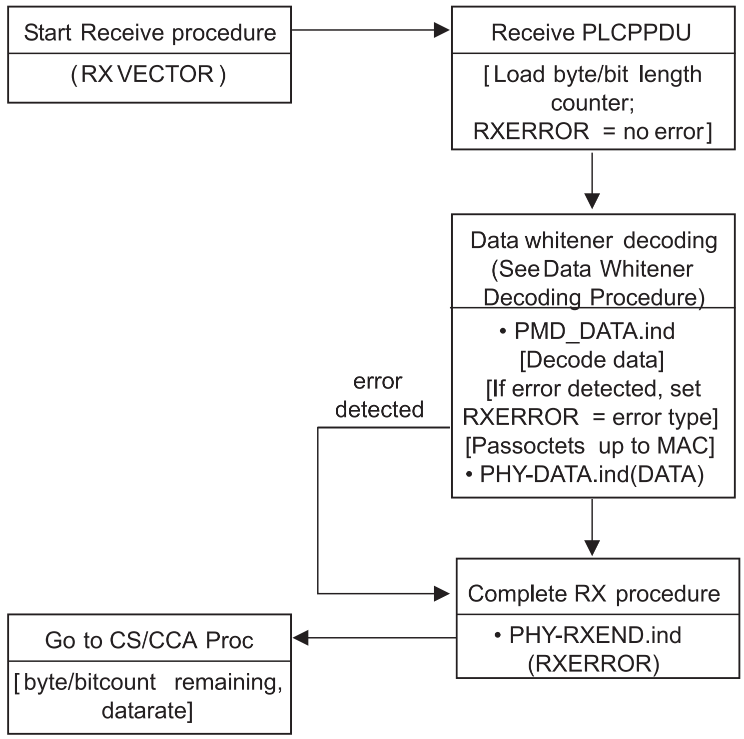


Figure 78—Receive state machine

If an error was detected during the reception of the PPDU, the PLCP shall immediately complete the receive procedure with a *PHY-RXEND.indicate(RXERROR=carrier_lost)* to the MAC, and return to the CS/CCA procedure with the octet/bit count remaining and the data rate value contained in the PLCP Header.

Data whitener decoding algorithm:

```
/*          If msb of stuff symbol = 1 then the next block is inverted; 0 = not inverted */
/*          Stuffing begins on first symbol of PLCP Header following the SFD */
/*          Algorithm begins after verifying validity of header with HEC */

/********* Read header *********/
Read in header {b(1),...,b(32)};                          /* b(1) is first bit in */

Get number_of_PSDU_octets, rate from header;              /* rate is 1 or 2 */
number_of_symbols = (number_of_PSDU_octets*8)/rate
number_of_blocks_in_packet = truncate{(number_of_symbols + 31) / 32};
Initialize scrambler to all ones;

/********* De-whiten the PPDU data with BSE decoder and de-scrambler *********/
For n = 1 to number_of_blocks_in_packet
{
        N = min(32, # of symbols remaining);        /* N= block size in symbols */
        Read in next block {b(0),...,b(N)};/                  * b(n) = {0,1} or {0,1,2,3} */

        If {[msb of b(0)=1] then Invert {b(1),...,b(N)};      /* if invert bit=true */
        Descramble {b(1),...,b(N)};                 /* see 14.3.2.3*/
        Send {b(1),...,b(N)} to MAC
                              /* 1 - 8 octets; use PHY-DATA.ind(DATA) for each octet. */
}
```

Figure 79—Data whitener decoding procedure

14.3.3.3.2 Receive state timing

The receive state timing shown in Figure 80 is defined to begin upon detection of a valid SFD and PLCP Header in the CS/CCA procedure. The PLCP shall begin receiving the variable length whitened PSDU immediately after the end of the last bit of the PLCP Header. The PLCP shall send a *PHY-RXEND.indicate(RXERROR)* after receiving the last PPDU data bit.

If any error was detected during the reception of the PPDU, the PLCP may send a PHY-RXEND.indicate(RXERROR) and terminate the receive procedure before the last bit arrives.

14.4 PLME SAP layer management

14.4.1 Overview

This subclause describes the services provided by the FHSS PLME to the upper layer management entities. The PLME/PMD services are defined in terms of service primitives. These primitives are abstract representations of the services and are not intended to restrict implementations.

14.4.2 FH PHY specific MAC sublayer management entity (MLME) procedures

14.4.2.1 Overview

The specific MAC sublayer management entity (MLME) procedures required for operating the FHSS PHY are specified in this portion of the subclause. The relationship between the MLME and FH PLME procedures is also described.

14.4.2.2 FH synchronization

The MLME of a compliant FH PHY STA shall perform the FH time synchronization procedure as defined in 11.1.5. This procedure provides for synchronized frequency hopping for all compliant FH PHY STAs within a single BSS or ad hoc network. The FH PLME accepts PLME-SET.request commands from the MLME to change the tune frequency at the time determined by the MLME. The tune frequency is changed by updating any combination of the Set, Pattern, and Index PHY MIB parameters.

14.4.3 FH PHY layer management entity state machines

14.4.3.1 Overview

This portion of this subclause describes the FH PHY layer management state machines to turn the PMD on/off, reset the PLCP state machine, and change the frequency hop channel.

14.4.3.2 PLME state machine

The PLME state machine in Figure 81 begins with a *PLME-SET.request (aCurrentPowerState= ON)* request, which turns on the PHY circuitry, resets the PLME and PLCP state machines, and sends a *PLME-SET.confirm*. The MAC then sends a series of three *PLME-SET.request* primitives to update the aCurrentSet, aCurrentPattern, and aCurrentIndex PHY MIB parameters, which together tune the PMD to the selected channel. The PLME then transfers execution to the PLCP state machine as defined in 14.3.3.

Upon receiving a PLME request from a higher-level LME, the PLCP shall return execution to the PLME state machine and process the request. A *PLME-RESET.request* shall cause a reset to the PLME and PLCP state machines. A *PLME-SET.request* updating the aCurrentIndex or a combination of the aCurrentSet, aCurrentPattern, and aCurrentIndex shall cause the PLCP to terminate a receive or CS/CCA process and

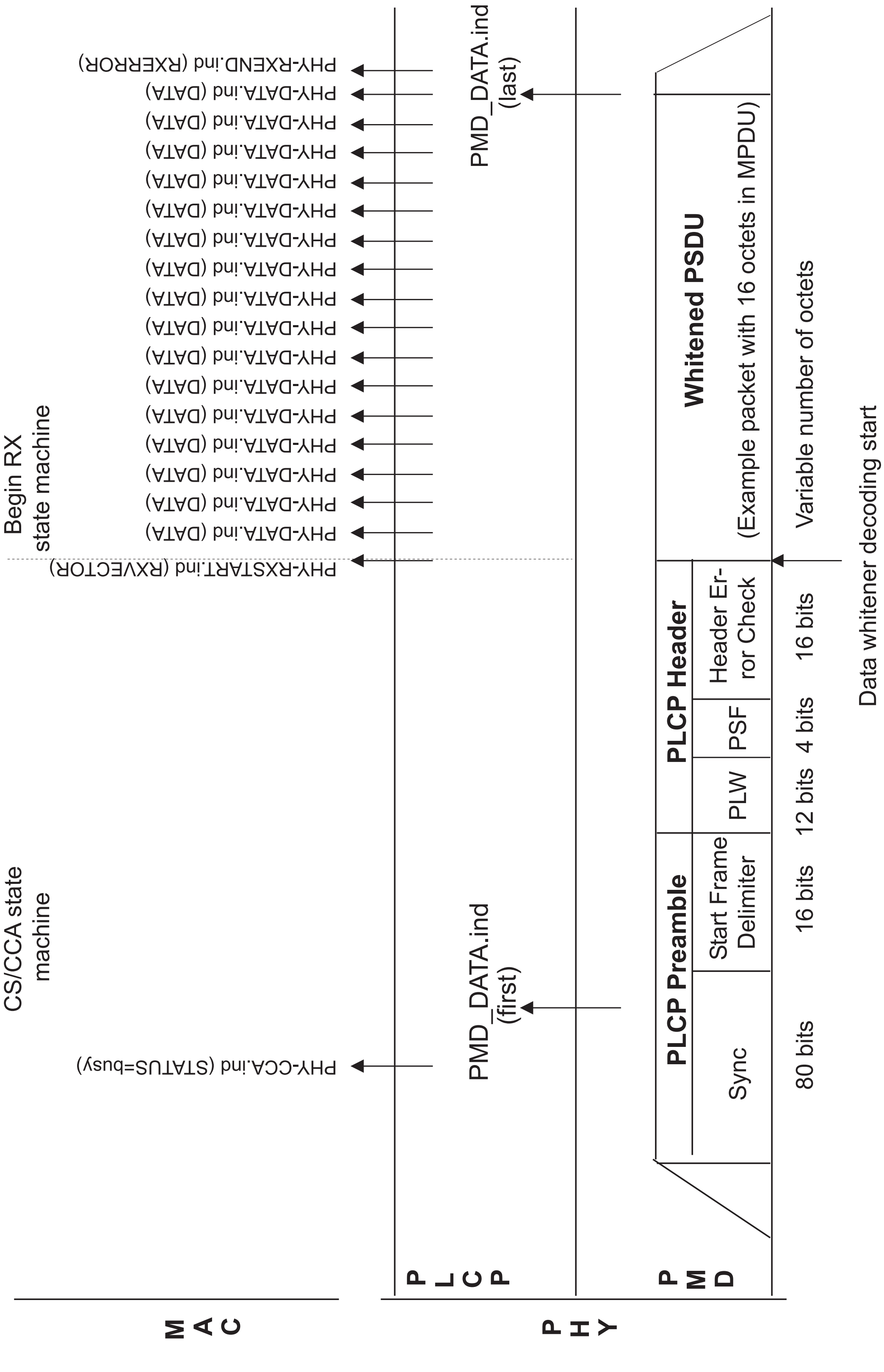


Figure 80—Receive timing

change frequency before returning to the PLCP state machine. A *PLME-SET.request(aCurrentPowerState=OFF)* request shall cause the PLCP to terminate a receive or CS/CCA process, power down the PMD circuitry, and return the PLME state machine to the idle state. *PLME-SET.request*s to any parameter other than the ones identified within this paragraph shall be executed and control returned to the PLCP state machine. The MAC should not send a PLME request while the PLCP is in the transmit state.

All *PLME-GET.request*s shall be processed in parallel and with no interruption to the execution of any state machine in process.

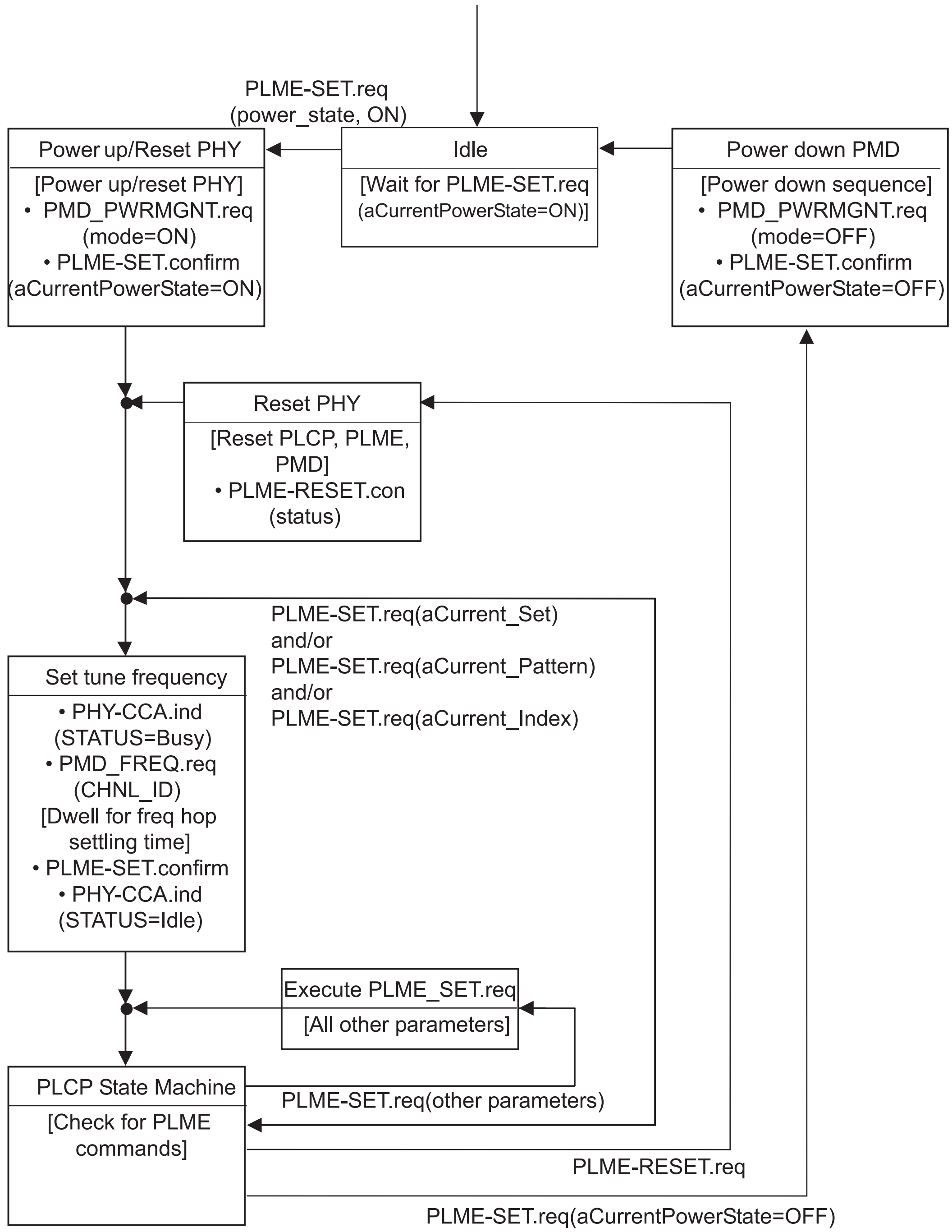


Figure 81—PLME state machine

14.4.3.3 PLME management primitives

The FH PLME uses the generic management primitives defined in 10.2 to manage all FH PHY parameters.

14.5 FHSS PMD sublayer services

14.5.1 Scope and field of application

The PMD services provided to the PLCP for the FHSS PHY are described in this subclause. Also defined in this subclause are the functional, electrical, and RF characteristics required for interoperability of implementations conforming to this specification. The relationship of this specification to the entire FHSS PHY is shown in Figure 82.

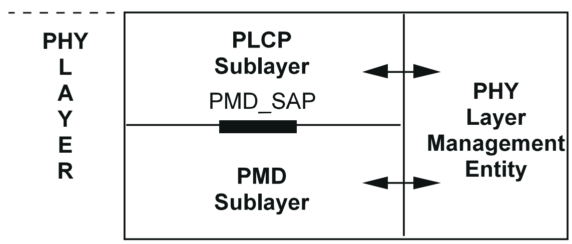


Figure 82—PMD layer reference model

14.5.2 Overview of services

In general, the FHSS PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which the signals required by these primitives are imposed onto the medium. In the FHSS PMD sublayer at the receiver the process is reversed. The combined function of the transmitting and receiving FHSS PMD sublayers results in a data stream, timing information, and receive parameter information being delivered to the receiving PLCP sublayer.

14.5.3 Overview of interactions

The primitives associated with the IEEE 802.11 PLCP sublayer to the FHSS PMD sublayer falls into two basic categories:

a) Service primitives that support PLCP peer-to-peer interactions
b) Service primitives that have local significance and support sublayer-to-sublayer interactions

14.5.4 Basic service and options

All of the service primitives described in this subclause are considered mandatory unless otherwise specified.

14.5.4.1 PMD_SAP peer-to-peer service primitives

Table 32 indicates the primitives for peer-to-peer interactions.

Table 32—PMD-SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_DATA	X	X		

14.5.4.2 PMD_SAP sublayer-to-sublayer service primitives

Table 33 indicates the primitives for sublayer-to-sublayer interactions.

Table 33—PMD_SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_TXRX	X			
PMD_PA_RAMP	X			
PMD_ANTSEL	X			
PMD_TXPWRLVL	X			
PMD_FREQ	X			
PMD_RSSI		X		
PMD_PWRMGMT	X			

14.5.4.3 PMD_SAP service primitives parameters

Table 34 shows the parameters used by one or more of the PMD_SAP service primitives.

Table 34—List of parameters for PMD primitives

Parameter	Associate primitive	Value
TXD_UNIT	PMD_DATA.request	1 Mbit/s: 0, 1 2 Mbit/s: 0, 1, 2, 3
RXD_UNIT	PMD_DATA.indicate	1 Mbit/s: 0, 1 2 Mbit/s: 0, 1, 2, 3
RF_STATE	PMD_TXRX.request	TRANSMIT, RECEIVE
RAMP_STATE	PMD_PA_RAMP.request	ON, OFF
ANTENNA_STATE	PMD_ANTSEL.request	1 to 255
TXPWR_LEVEL	PMD_TXPWRLVL.request	LEVEL1, LEVEL2, LEVEL3, LEVEL 4
CHNL_ID	PMD_FREQ.request	2 through 80 inclusive
STRENGTH	PMD_RSSI.indicate	0 to RSSI Max
MODE	PMD_PWRMGMT.request	ON, OFF

14.5.5 PMD_SAP detailed service specification

This subclause describes the services provided by each PMD primitive.

14.5.5.1 PMD_DATA.request

14.5.5.1.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

14.5.5.1.2 Semantics of the service primitive

The primitive shall provide the following parameters:

PMD_DATA.request (TXD_UNIT)

The TXD_UNIT parameter can take on one of two values: ONE or ZERO. This parameter represents a single data bit. The effect of this parameter is that the PMD will properly modulate the medium to represent ONEs or ZEROs as defined in the FHSS PMD modulation specifications for a given data rate.

14.5.5.1.3 When generated

This primitive is generated by the PLCP sublayer to request the transmission of a single data bit on the PMD sublayer. The bit clock is assumed to be resident or part of the PLCP and this primitive is issued at every clock edge once the PLCP has begun transmitting data.

14.5.5.1.4 Effect of receipt

The receipt of this primitive will cause the PMD entity to encode and transmit a single data bit.

14.5.5.2 PMD_DATA.indicate

14.5.5.2.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

14.5.5.2.2 Semantics of the service primitive

The primitive shall provide the following parameters:

PMD_DATA.indicate (RXD_UNIT)

The RXD_UNIT parameter can take on one of two values: ONE or ZERO. This parameter represents the current state of the medium as determined by the FHSS PMD modulation specifications for a given data rate.

14.5.5.2.3 When generated

The PMD_DATA.indicate is generated to all receiving PLCP entities in the network after a PMD_DATA.request is issued.

14.5.5.2.4 Effect of receipt

The effect of receipt of this primitive by the PLCP is unspecified.

14.5.5.3 PMD_TXRX.request

14.5.5.3.1 Function

This primitive is used to place the PMD entity into the transmit or receive function.

14.5.5.3.2 Semantics of the service primitive

The primitive shall provide the following parameters:

PMD_TXRX.request (RF_STATE)

The RF_STATE parameter can take on one of two values: TRANSMIT or RECEIVE. When the value of the primitive is TRANSMIT, the RF state of the radio is transmit. If the value of the primitive is RECEIVE, the RF state of the radio is receive.

14.5.5.3.3 When generated

This primitive is generated whenever the mode of the radio needs to be set or when changing from transmit to receive or receive to transmit.

14.5.5.3.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the mode of the radio to be in either transmit or receive.

14.5.5.4 PMD_PA_RAMP.request

14.5.5.4.1 Function

This primitive defines the start of the ramp-up or ramp-down of the radio transmitter's power amplifier.

14.5.5.4.2 Semantics of the service primitive

The primitive shall provide the following parameters:

PMD_PA_RAMP.request (RAMP_STATE)

The RAMP_STATE parameter can take on one of two values: ON or OFF. When the value of the primitive is ON, the state of the transmit power amplifier is "on." If the value of the primitive is OFF, the state of the transmit power amplifier is "off."

14.5.5.4.3 When generated

This primitive is issued only during transmit and to establish the initial state. It is generated by the PLCP at the start of the transmit function to turn the transmitter's power amplifier "on." A power amplifier ramp-up period follows the change of state from "off" to "on." After the PLCP has transferred all required data to the PMD entity, this primitive again will be issued by the PLCP to place the transmit power amplifier back into the "off" state. A power amplifier ramp-down period follows the change of state from "on" to "off."

14.5.5.4.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the transmit power amplifier to become on or off.

14.5.5.5 PMD_ANTSEL.request

14.5.5.5.1 Function

This primitive is used to select which antenna the PMD entity will use to transmit or receive data.

14.5.5.5.2 Semantics of the service primitive

The primitive shall provide the following parameters:

PMD_ANTSEL.request (ANTENNA_STATE)

The ANTENNA_STATE parameter can take on values from one to N (where N is the number of antennas supported). When the value of the primitive is a ONE, the PMD will switch to antenna 1 for receive or transmit; if the value of the primitive is TWO, the PMD entity will switch to antenna 2 for receive or transmit, etc.

14.5.5.5.3 When generated

This primitive is generated at various times by the PLCP entity to select an antenna. During receive, this primitive can be used to manage antenna diversity. During transmit, this primitive can be use to select a transmit antenna. This primitive will also be used during CCA.

14.5.5.5.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the radio to select the antenna specified.

14.5.5.6 PMD_TXPWRLVL.request

14.5.5.6.1 Function

This primitive defines the power level the PMD entity will use to transmit data.

14.5.5.6.2 Semantics of the service primitive

The primitive shall provide the following parameters:

PMD_TXPWRLVL.request (TXPOWER_LEVEL)

The TXPOWER_LEVEL parameter can be one of the values listed in Table 35.

Table 35—Transmit power levels

TXPWR_LEVEL	Level description
LEVEL1	Defined as TxPowerLevel1 in MIB
LEVEL2	Defined as TxPowerLevel2 in MIB
LEVEL3	Defined as TxPowerLevel3 in MIB
LEVEL4	Defined as TxPowerLevel4 in MIB
LEVEL5	Defined as TxPowerLevel5 in MIB
LEVEL6	Defined as TxPowerLevel6 in MIB
LEVEL7	Defined as TxPowerLevel7 in MIB
LEVEL8	Defined as TxPowerLevel8 in MIB

14.5.5.6.3 When generated

This primitive is generated as part of the transmit sequence.

14.5.5.6.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the transmit power level to be modify.

14.5.5.7 PMD_FREQ.request

14.5.5.7.1 Function

This primitive defines the frequency the PMD entity will use to receive or transmit data. Since changing the radio frequency is not an immediate function, this primitive serves also as an indication of the start of this process. The completion of this process is dictated by other PMD specifications.

14.5.5.7.2 Semantics of the service primitive

The primitive shall provide the following parameters:

PMD_FREQ.request (CHANNEL_ID)

The CHANNEL_ID parameter can be one of the values listed in Table 38, Table 39, Table 40, or Table 41.

14.5.5.7.3 When generated

This primitive is generated by the PLCP whenever a change to a new frequency is required.

14.5.5.7.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the radio to change to a new frequency defined by the value of the CHNL_ID.

14.5.5.8 PMD_RSSI.indicate

14.5.5.8.1 Function

This primitive transfers a receiver signal strength indication of the physical medium from the PMD sublayer to the PLCP sublayer. This value will be used by the PLCP to perform any diversity or clear channel assessment functions required by the PLCP or other sublayers.

14.5.5.8.2 Semantics of the service primitive

The primitive shall provide the following parameters:

PMD_RSSI.indicate (STRENGTH)

The STRENGTH parameter can be a value from 0 to 15. This parameter is an indication by the PMD sublayer of the magnitude of the energy observed at the selected antenna. This reported value is used to generate the RSSI term in the PHY-RXSTART.ind(RXVECTOR) primitive and might also be used by any diversity function. Since RSSI is only used in a relative manner by the MAC sublayer, this parameter is defined to have no more than 16 values, ranging from 0 through RSSI_Max. The value zero is the weakest signal strength, while RSSI_Max is the strongest signal strength.

14.5.5.8.3 When generated

This primitive is generated continually by the PMD entity to transfer a receive signal strength indication to the PLCP.

14.5.5.8.4 Effect of receipt

The effect of receipt of this primitive by the PLCP is unspecified.

14.5.5.9 PMD_PWRMGMT.request

14.5.5.9.1 Function

This primitive is used by the higher-layer entities to manage or control the power consumption of the PMD when not in use. This allows higher-layer entities to put the radio into a sleep or standby mode when receipt or sending of any data is not expected.

14.5.5.9.2 Semantics of the service primitive

The primitive shall provide the following parameters:

PMD_PWRMGMT.request (MODE)

The MODE parameter can have one of two values: ON or OFF. When the value of the parameter is ON, the PMD entity will enter into a fully functional mode that allows it to send or receive data. When the value of the parameter is OFF, the PMD entity will place itself in a standby or power-saving mode. In the low-power mode, the PMD entity is not expected to be able to perform any request by the PLCP, nor is it expected to indicate any change in PMD state or status.

14.5.5.9.3 When generated

This primitive is delivered by the PLCP but actually is generated by a higher layer management entity.

14.5.5.9.4 Effect of receipt

Upon receipt of this primitive, the PMD entity will enter a fully functional or low power consumption state depending on the value of the primitive's parameter.

14.6 FHSS PMD sublayer, 1.0 Mbit/s

14.6.1 1 Mbit/s PMD operating specifications, general

In general, the PMD accepts convergence layer service primitives and provides the actual means by which the signals required by these primitives are imposed onto the medium. In the PMD sublayer at the receiver, the process is reversed. The combined function of the transmitting and receiving PMD sublayers results in a data stream, timing information, and receive parameter information being delivered to the receiving convergence sublayer.

14.6.2 Regulatory requirements

Wireless LANs implemented in accordance with this standard are subject to equipment certification and operating requirements established by regional and national regulatory administrations. The PMD specification establishes minimum technical requirements for interoperability, based upon established regulations for

Europe, Japan, and North America at the time this standard was issued. These regulations are subject to revision, or may be superseded. Requirements that are subject to local geographic regulations are annotated within the PMD specification. Regulatory requirements that do not affect interoperability are not addressed within this standard. Implementors are referred to the following regulatory sources for further information. Operation in countries within Europe, or other regions outside Japan or North America, may be subject to additional or alternative national regulations.

The documents listed below specify the current regulatory requirements for various geographic areas at the time the standard was developed. They are provided for information only, and are subject to change or revision at any time.

Geographic area	Approval standards	Documents	Approval authority
Europe	European Telecommunications Standards Institute (ETSI)	ETS 300-328, ETS 300-339	National type approval authorities
France	Règle technique applicable aux équipements radioélectriques de transmission de données à large bande fonctionnant dans la bande de fréquences à 2,4 GHz et utilisant la technique de l'étalement de spectre (Édition février 1995)	SP/DGPT/ATAS/23, ETS 300-328, ETS 300-339	Direction Générale des Postes et Télécommunications (DGPT)
Japan	Association of Radio Industries and Businesses (ARIB)	RCR STD-33A	Ministry of Telecommunications (MKK)
North America			
Canada	Industry Canada (IC)	GL36	IC
USA	Federal Communications Commission (FCC)	CFR47, Part 15, Sections 15.205, 15.209, 15.247	FCC
Spain	Supplemento Del Numero 164 Del Boletin Oficial Del Estado (Published 10 July 1991, Revised 25 June 1993)	ETS 300-328, ETS 300-339	Cuadro Nacional De Atribucion De Frecuesias

14.6.3 Operating frequency range

A conformant PMD implementation shall be able to select the carrier frequency (F_c) from the full geographic-specific set of available carrier frequencies. Table 36 summarizes these frequencies for a number of geographic locations.

Table 36—Operating frequency range

Lower Limit	Upper limit	Regulatory range	Geography
2.402 GHz	2.480 GHz	2.400–2.4835 GHz	North America
2.402 GHz	2.480 GHz	2.400–2.4835 GHz	Europe[a]
2.473 GHz	2.495 GHz	2.471–2.497 GHz	Japan
2.447 GHz	2.473 GHz	2.445–2.475 GHz	Spain
2.448 GHz	2.482 GHz	2.4465–2.4835 GHz	France
NOTE—The frequency ranges in this table are subject to the geographic specific regulatory authorities.			

[a]Excluding Spain and France.

14.6.4 Number of operating channels

The number of transmit and receive frequency channels used for operating the PMD entity is 79 for the US and Europe, and 23 for Japan. Table 37 summarizes these frequencies for a number of geographic locations. This is more fully defined in Table 38 through Table 41 of 14.6.5.

Table 37—Number of operating channels

Minimum	Hopping set	Geography
75	79	North America
20	79	Europe[a]
Not applicable	23	Japan
20	27	Spain
20	35	France
NOTE—The number of required hopping channels is subject to the geographic specific regulatory authorities.		

[a]Excluding Spain and France.

14.6.5 Operating channel center frequency

The channel center frequency is defined in sequential 1.0 MHz steps beginning with the first channel, channel 2.402 GHz for the USA and Europe excluding Spain and France, as listed in Table 38. The channel centers for Japan, starting at 2.473 GHz with 1 MHz increments, are listed in Table 39. The channel centers for Spain and France are listed in Table 40 and Table 41, respectively.

Table 38—Requirements in North America and Europe (excluding Spain and France; values specified in GHz)

Channel #	Value	Channel #	Value	Channel #	Value
2	2.402	28	2.428	54	2.454
3	2.403	29	2.429	55	2.455
4	2.404	30	2.430	56	2.456
5	2.405	31	2.431	57	2.457
6	2.406	32	2.432	58	2.458
7	2.407	33	2.433	59	2.459
8	2.408	34	2.434	60	2.460
9	2.409	35	2.435	61	2.461
10	2.410	36	2.436	62	2.462
11	2.411	37	2.437	63	2.463
12	2.412	38	2.438	64	2.464

Table 38—Requirements in North America and Europe (excluding Spain and France; values specified in GHz) (continued)

Channel #	Value	Channel #	Value	Channel #	Value
13	2.413	39	2.439	65	2.465
14	2.414	40	2.440	66	2.466
15	2.415	41	2.441	67	2.467
16	2.416	42	2.442	68	2.468
17	2.417	43	2.443	69	2.469
18	2.418	44	2.444	70	2.470
19	2.419	45	2.445	71	2.471
20	2.420	46	2.446	72	2.472
21	2.421	47	2.447	73	2.473
22	2.422	48	2.448	74	2.474
23	2.423	49	2.449	75	2.475
24	2.424	50	2.450	76	2.476
25	2.425	51	2.451	77	2.477
26	2.426	52	2.452	78	2.478
27	2.427	53	2.453	79	2.479
—	—	—	—	80	2.480

Table 39—Requirements in Japan (values specified in GHz)

Channel #	Value	Channel #	Value	Channel #	Value
73	2.473	81	2.481	89	2.489
74	2.474	82	2.482	90	2.490
75	2.475	83	2.483	91	2.491
76	2.476	84	2.484	92	2.492
77	2.477	85	2.485	93	2.493
78	2.478	86	2.486	94	2.494
79	2.479	87	2.487	95	2.495
80	2.480	88	2.488	—	—

