DRAFT Supplement to STANDARD [for] Information Technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements-

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band.

Sponsor

LAN MAN Standards Committee
of the
IEEE Computer Society

Abstract: Changes and additions to IEEE Std. 802.11 to support the higher rate Physical layer for operation in the 2.45 GHz band are provided.

Keywords: LAN, Local Area Network, Wireless, Radio Frequency

Introduction

(This introduction is not part of P802.11B/D1.0, Draft Standard for Wireless LAN Physical Layer Standards)

This standard is part of a family of standards for Local Area Networks (LANs). This supplement covers an extension to IEEE Std 802.11- 1999 to increase the data rates in the 2.4 GHz band to greater than 10 Mbit/s

Participants

At the time the draft of this standard was sent to sponsor ballot, the IEEE 802.11 working group had the following voting members:

Vic Hayes, Chair
Stuart J. Kerry, Vice Chair
Al Petrick, Co-Vice Chair
George Fishel, Secretary

Bob O'Hara, Chair and editor 802.11-rev
Allen Heberling, State-diagram editor
Michael Fischer, State-diagram editor
Dean M. Kawaguchi, Chair PHY group
David Bagby, Chair MAC group

Naftali Chayat, Chair Task Group a
Hitoshi Takanashi, Editor 802.11a

John Fakatselis, Chair Task Group b
Carl F. Andren, Editor 802.11b

Voting Members:

Jeff Abramowitz
Reza Ahy
Keith B. Amundsen
James R. Baker
Kevin M. Barry
Phil Belanger
John Biddick
Simon Black
Timothy J. Blaney
Jan Boer
Ronald Brockmann
Wesley Brodsky
John H. Cafarella
Wen-Chiang Chen
Ken Clements
Wim Diepstraten
Peter Ecclesine
Richard Eckard
Darwin Engwer
Greg Ennis
Jeff Fischer
John Fisher
Ian Gifford
Motohiro Gochi
Tim Godfrey
Steven D. Gray
Jan Haagh
Karl Hannestad
Kei Hara
Chris Heegard
Robert Heile
Juha Heiskala
Maarten Hoeben
Masayuki Ikeda
Donald C. Johnson
Tal Kaitz
Ad Kamerman
Mika Kasslin
Patrick Kinney
Steven Knudsen
Bruce P. Kraemer
David S. Landeta
James S. Li
Stanley Ling
Michael D. McInnis
Gene Miller
Akira Miura
Henri Moelard
Masaharu Mori
Masahiro Morikura
Richard van Nee
Erwin R. Noble
Tomoki Ohsawa
Kazuhiro Okanoue
Richard H. Paine
Roger Pandanda
Victoria M. Poncini
Gregory S. Rawlins
Stanley A. Reible
Frits Riep
William Roberts
Kent G. Rollins
Clemens C.W. Ruppel
Anil K. Sanwalka
Roy Sebring
Tie-Jun Shan
Stephen J. Shellhammer
Matthew B. Shoemake
Thomas Siep
Donald I. Sloan
Gary Spiess
Satoru Toguchi
Cherry Tom
Mike Trompower
Tom Tsoulogiannis
Bruce Tuch
Sarosh Vesuna
Ikuo Wakayama
Robert M. Ward Jr.
Mark Webster
Leo Wilz
Harry Worstell
Lawrence W. Yonge III
Chris Zegelin
Johnny Zweig
Jim Zyren

The following persons were on the balloting committee:

This section is usually supplied
by IEEE Balloting Center staff.

Table of Contents

DRAFT Supplement to STANDARD [for] Information Technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements-

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band.

[This supplement is based on the current edition of IEEE Std 802.11, 1999 Edition.

NOTE—The editing instructions contained in this supplement define how to merge the material contained herein into the existing base standard to form the new comprehensive standard as created by the addition of IEEE Std 802.11b-1999.

The editing instructions are shown in ***bold italic***. Three editing instructions are used: change, delete, and insert. ***Change*** is used to make small corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed either by using ~~strikethrough~~ (to remove old material) or underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. Editorial notes will not be carried over into future editions.

Change the following paragraphs as indicated:

3.8 Basic Service Set (BSS) basic rate set:

The set of data transfer rates that all the stations in a BSS will be capable of using to receive and transmit frames to/from the wireless medium (WM). The BSS basic rate set data rates are preset for all stations in the BSS.

4.0 Abbreviations and acronyms

Insert the following abbreviations alphabetically in the list in 4.0:

CCK Complementary Code Keying
High Rate High Rate Direct Sequence Spread Spectrum with or without Options enabled
HR/DSSS High Rate Direct Sequence Spread Spectrum using the long preamble and header
HR/DSSS/short High Rate Direct Sequence Spread Spectrum using the optional short preamble and header mode
HR/DSSS/PBCC High Rate Direct Sequence Spread Spectrum using the optional Packet Binary Convolutional Coding mode and the long preamble and header
HR/DSSS/PBCC/short High Rate Direct Sequence Spread Spectrum using the optional Packet Binary Convolutional Coding mode and the optional short preamble and header

7.2.3.1 Beacon frame format

Change notes 1 and 2 of this table as shown.

Table 5 Beacon frame body

Order	**Information**	**Note**
1	Timestamp	
2	Beacon interval	
3	Capability information	
4	SSID	
5	Supported rates	
6	FH Parameter Set	1
7	DS Parameter Set	2
8	CF Parameter Set	3
9	IBSS Parameter Set	4
10	TIM	5

Notes:
1—The FH Parameter Set information element is ~~only~~ present within Beacon frames generated by STAs using frequency-hopping PHYs.
2—The DS Parameter Set information element is ~~only~~ present within Beacon frames generated by STAs using direct sequence PHYs.
3—The CF Parameter Set information element is only present within Beacon frames generated by APs supporting a PCF.
4—The IBSS Parameter Set information element is only present within Beacon frames generated by STAs in an IBSS.
5—The TIM information element is only present within Beacon frames generated by APs.

7.2.3.9 Probe Request frame format

Change notes 1 and 2 of this table as shown.

Table 12—Probe Response frame body

Order	Information	Note
1	Timestamp	
2	Beacon interval	
3	Capability information	
4	SSID	
5	Supported rates	
6	FH Parameter Set	1
7	DS Parameter Set	2
8	CF Parameter Set	3
9	IBSS Parameter Set	4

NOTES
1—The FH Parameter Set information element is ~~only~~ present within Probe Response frames generated by STAs using frequency-hopping PHYs.
2—The DS Parameter Set information element is ~~only~~ present within Probe Response frames generated by STAs using direct sequence PHYs.
3—The CF Parameter Set information element is only present within Probe Response frames generated by APs supporting a PCF.
4—The IBSS Parameter Set information element is only present within Probe Response frames generated by STAs in an IBSS.

7.3.1.4 Capability Information Field

Insert three subfields to the capability information field figure and supporting text as shown:

The Capability Information Field contains a number of subfields that are used to indicate requested or advertised capabilities.

The length of the Capability Information Field is two octets. The Capability Information Field consists of the following subfields: ESS, IBSS, CF-Pollable, CF-Poll Request, ~~and~~ Privacy, Short Preamble, PBCC, and Channel Agility. The format of the Capability Information Field is as illustrated in Figure 27.

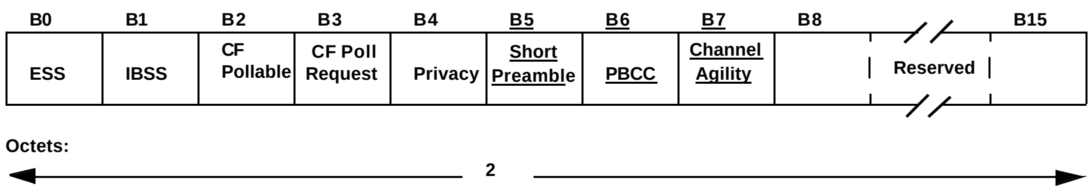


Figure 27 -- Capability Information Fixed Field

.

Insert the following text after the text in 7.3.1.4.

APs (as well as STAs in IBSSs) shall set the Short Preamble subfield to 1 in transmitted Beacon, Probe Response, Association Response and Reassociation Response management MMPDUs to indicate that the use of the short preamble option, as described in subclause 18.2.2.2 is allowed within this BSS. To indicate that the use of the short preamble option is not allowed the Short Preamble subfield shall be set to 0 in Beacon, Probe Response, Association Response and Reassociation Response management MMPDUs transmitted within the BSS.

STAs shall set the Short Preamble subfield to 1 in transmitted Association Request and Reassociation Request MMPDUs when the MIB attribute dot11ShortPreambleOptionImplemented is true. Otherwise STAs shall set the Short Preamble subfield to 0 in transmitted Association Request and Reassociation Request MMPDUs.

APs (as well as STAs in IBSSs) shall set the PBCC subfield to 1 in transmitted Beacon, Probe Response, Association Response and Reassociation Response management MMPDUs to indicate that the use of the PBCC modulation option, as described in subclause 18.4.6.6 is allowed within this BSS. To indicate that the use of the PBCC modulation option is not allowed the PBCC subfield shall be set to 0 in Beacon, Probe Response, Association Response and Reassociation Response management MMPDUs transmitted within the BSS.

STAs shall set the PBCC subfield to 1 in transmitted Association Request and Reassociation Request MMPDUs when the MIB attribute dot11PBCCOptionImplemented is true. Otherwise STAs shall set the PBCC subfield to 0 in transmitted Association Request and Reassociation Request MMPDUs.

Bit 7 of the Capabilities Information Field shall be used to indicate the usage of channel agility by the HR/DSSS PHY. STAs shall set the Channel Agility bit to 1 when channel agility is in use and shall set it to 0 otherwise.

Bits 8 to 15 of the Capability Information Field are reserved.

7.3.1.9 Status Code Field

Add three status codes as shown to table 19:

Table 19 --- Status Codes

Status code	Meaning
19	Association denied due to requesting station not supporting the short preamble option.
20	Association denied due to requesting station not supporting the PBCC modulation option.
21	Association denied due to requesting station not supporting the channel agility option.

7.3.2.2 Supported Rates element

Change the paragraph as follows:

The Supported Rates element specifies ~~all~~ the values ~~rates that this station is capable of receiving~~ in the Operational Rate Set parameter as described in the MLME_Join.request and MLME_Start.request primitives. The information field is encoded as 1 to 8 octets where each octet describes a single supported rate ~~in units of 500 kbit/s~~.

Within Beacon, Probe Response, Association Response, and Reassociation Response management frames, each supported rate belonging to the ~~BSSBasicRateSet~~ BSS basic rate set, is encoded as an octet with the msb (bit 7) set to 1 (e.g., a 1 Mbit/s rate belonging to the ~~BSSBasicRateSet~~ BSS basic rate set is encoded as X'82'). Rates not belonging to the ~~BSSBasicRateSet~~ BSS basic rate set are encoded with the msb set to 0 (e.g., a 2 Mbit/s rate not belonging to the ~~BSSBasicRateSet~~ BSS basic rate set is encoded as X'04'). The msb of each Supported Rate octet in other management frame types is ignored by receiving STAs.

~~BSSBasicRateSet~~ The BSS basic rate set information in Beacon and Probe Response management frames is delivered to the management entity in an STA via the BSSBasicRateSet parameter in the MLME_Scan.confirm primitive. It is used by the management entity in an STA ~~s in order~~ to avoid associating with a BSS if the STA cannot receive and transmit all the data rates in the ~~BSSBasicRateSet~~ BSS basic rate set. See Figure 36.

9.2 DCF

Change the second to the last paragraph as shown:

The medium access protocol allows for stations to support different sets of data rates. All STAs shall be able to receive and transmit at all the data rates in the aBasicRateSet specified parameter of the MLME_Join.request and MLME_Start.request primitives ~~and transmit at one or more of the aBasicRateSet data rates.~~ To support the proper operation of the RTS/CTS and the Virtual Carrier Sense mechanism, all STAs shall be able to detect the RTS and CTS frames. For this reason the RTS and CTS frames shall be transmitted at one of the rates in the BSS basic rate set ~~aBasicRateSet rates.~~ (See subclause 9.6 for a description of multirate operation).

9.6 Multirate support

Change the existing subclause as follows:

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~~ BSS basic rate set ~~(see 10.3.10.1), or at one of the rates in the PHY mandatory rate set~~ so that they will be understood by all STAs in the BSS.

All frames with multicast and broadcast RA shall be transmitted at one of the rates included in the ~~BSSBasicRateSet~~ BSS basic rate set, regardless of their type or subtype.

Data and/or management MPDUs with a unicast ~~immediate address~~ RA 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.

~~In order to~~ To allow the transmitting STA to calculate the contents of the Duration/ID field, the responding STA shall transmit its Control Response and Management Response frames (either CTS or ACK) at the highest rate in the BSS basic rate set that is less than or equal to the rate of ~~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.~~ In addition the Control Response frame shall be sent using the same PHY options as the received frame.

For the HR/DSSS PHY, the time required to transmit a frame, for use in the Duration/ID field, is determined using the PLME-TXTIME.request primitive and the PLME-TXTIME.confirm primitive, both defined in 18.3.4.

10.3.1PLME_start

Change "set" to "sets" in the Name and Description columns for the PHY Parameter Set.

10.3.2.2 PLME_scan.confirm

Change "set" to "sets" in the Name and Description columns for the PHY Parameter Set.

10.3.2.2.2. Semantics of the service primitive

Change wording in the table in the BSSBasicRateSet Description to:

The set of data rates that must be supported by all STAs that desire to join this BSS. The STAs must be able to receive and transmit at each of the data rates listed in the set.

10.3.3.1.2 Semantics of the service primitive

Change Table as follows:

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	greater than or equal to 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 500kbit/s)~~ that the STA desires to 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~~ This set is a superset of the ~~BSSBasicRateSet~~ BSS basic rate set advertised by the BSS.

10.3.10.1.2 Semantics of the service primitive

Change the table as follows:

Name	Type	Valid Range	Description
SSID	octet string	1 - 32 octets	The SSID of the BSS.
BSSType	Enumeration	INFRASTRUCTURE, INDEPENDENT	The type of the BSS.
Beacon Period	integer	greater than or equal to 1	The Beacon period of the BSS (in TU).
DTIM Period	integer	As defined in 7.3.12.6	The DTIM Period of the BSS (in Beacon Periods)
CF parameter set	As defined in Frame Format	As defined in 7.3.2.5	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 7.3.2.3 or 7.3.2.4	The parameter set relevant to the PHY.
IBSS parameter set	As defined in Frame Format	As defined in 7.3.2.7	The parameter set for the IBSS, if BSS is an IBSS.

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 7.3.1.4	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 that desire to join this BSS. The STA that is creating the BSS must be able to receive and transmit 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 desires to 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~~ This set is a superset of the BSS basic rate set ~~BSSBasicRateSet~~ advertised by the BSS.

10.4.4 PLME_DSSSTESTMODE

Add switches for the new options.

PLME-DSSSTESTMODE.request (
TEST_ENABLE,
TEST_MODE,
SCRAMBLE_STATE,
SPREADING_STATE,
DATA_TYPE,
DATA_RATE;
PREAMBLE_TYPE;
MODULATION_CODE_TYPE;
)

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 modes 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. For example: all one, all zeros, and random data patterns
DATA_RATE	Integer	2, 4, 11, 22	Selects between 1 and 2 Mbit/s operation 02 = 1 Mbit/s 04 = 2 Mbit/s 11 = 5.5 Mbit/s 22 = 11 Mbit/s
PREAMBLE_TYPE	Boolean	null, 0,1	Selects the preamble length. Can be null
MODULATION_CODE_TYPE	Boolean	null, 0, 1	Selects the modulation code 0 = CCK 1 = PBCC Can be null

Add a paragraph 18 as follows:

18 High rate direct sequence spread spectrum (HR/DSSS) PHY specification

18.1 Overview

This clause specifies the high rate extension of the physical layer for the Direct Sequence Spread Spectrum (DSSS) system (clause 15 in IEEE Std 802.11- 1999) hereinafter known as the High Rate PHY for the 2.4GHz band designated for ISM applications. The Radio Frequency LAN system is aimed at the 2.4 GHz bands designated for ISM applications as provided in the USA according to Code of Federal Regulations, Title 47, Section 15.247, in Europe by ETS 300-328 and other countries according to subclause 18.4.6.2.

This extension of the DSSS system builds on the data rate capabilities as described in clause 15 in IEEE Std 802.11- 1999 to provide 5.5 and 11 Mbit/s payload data rates in addition to the 1 and 2 Mbps rates. To provide the higher rates, 8 chip Complementary Code Keying (CCK) is employed as the modulation scheme. The chipping rate is 11 MHz, which is the same as the DSSS system as described in IEEE Std 802.11- 1999 clause 15, thus providing the same occupied channel bandwidth. The basic new capability described in this clause is called High Rate Direct Sequence Spread Spectrum (HR/DSSS). The basic High Rate PHY uses the same PLCP preamble and header as the DSSS PHY so both PHYs can co-exist in the same BSS and can use the rate switching mechanism as provided.

In addition to providing higher speed extensions to the DSSS system, a number of optional features are described that will allow the performance of the Radio Frequency LAN system to be improved as technology allows the implementation of these options to become cost effective.

An optional mode replacing the CCK modulation with Packet Binary Convolutional Coding (HR/DSSS/PBCC) is provided.

Another optional mode which allows data throughput at the higher rates (2, 5.5 and 11 Mbit/s) to be significantly, increased by using a shorter PLCP preamble, is also provided. This mode is called HR/DSSS/short or HR/DSSS/PBCC/short. This short preamble mode can co-exist with DSSS, HR/DSSS, or HR/DSSS/PBCC under limited circumstances such as on different channels or with appropriate CCA mechanisms.

An optional capability for channel agility is also provided for. This option allows an implementation to overcome some inherent difficulty with static channel assignements (a tone jammer), without burdening all implementations with the added cost of this capability. This option can also be used to implement 802.11 compliant systems that are interoperable with both FH and DS modulations. See informative Annex F for more details.

18.1.1 Scope

This clause specifies the Physical Layer Entity for the Higher Rate Direct Sequence Spread Spectrum (DSSS) extension and the changes that have to be made to the base standard to accommodate the High Rate PHY.

The High Rate PHY layer consists of 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 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. The PHY exchanges PHY Protocol Data Units (PPDU) that contain PLCP Service Data Units (PSDU). The MAC uses the PHY service, so each MPDU corresponds to a PSDU that is carried in a PPDU.

b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a wireless medium between two or more STAs each using the High Rate system.

18.1.2 High Rate PHY functions

The 2.4 GHz High Rate PHY architecture is depicted in the ISO/IEC basic reference model shown in Figure 11. The High Rate PHY contains three functional entities: the PMD function, the physical layer convergence function, and the layer management function. Each of these functions is described in detail in the following subclauses. For the purposes of MAC and MAC Management when channel agility is both present and enabled (see 18.3.2 and Annex C), the High Rate PHY shall be interpreted to be both a High Rate and a frequency hopping physical layer.

The High Rate PHY service shall be provided to the MAC through the PHY service primitives described in Clause 12 of IEEE Std 802.11- 1999.

18.1.2.1 PLCP sublayer

To allow the MAC to operate with minimum dependence on the PMD sublayer, a physical layer convergence procedure (PLCP) sublayer is defined. This function simplifies the PHY service interface to the MAC services.

18.1.2.2 Physical Medium Dependent Sublayer (PMD) sublayer

The PMD sublayer provides a means and method of transmitting and receiving data through a wireless medium (WM) between two or more STAs each using the High Rate system.

18.1.2.3 Physical layer management entity (PLME)

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

18.1.3 Service specification method and notation

The models represented by figures and state diagrams are intended to be illustrations of functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the High Rate PHY compliant developer.

The service of a layer or sublayer is a 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 is independent of any particular implementation.

18.2 High Rate PLCP sublayer

18.2.1 Overview

This subclause provides a convergence procedure for the 2, 5.5 and 11 Mbit/s specification in which PSDUs are converted to and from PPDUs. During transmission, the PSDU shall be appended to a PLCP preamble and header to create the PPDU. Two different preambles and headers are defined: the mandatory supported long preamble and header which interoperates with the current 1 and 2 Mbit/s DSSS specification as described in IEEE Std 802.11- 1999, and an optional short preamble and header. At the receiver, the PLCP preamble and header are processed to aid in demodulation and delivery of the PSDU.

The optional short preamble and header is intended for applications where maximum throughput is desired and interoperability with legacy and non short preamble capable equipment is not a consideration. That is, it is expected to be used only in networks of like equipment that can all handle the optional mode.

18.2.2 PPDU format

Two different preambles and headers are defined: the mandatory supported long preamble and header which is interoperable with the current 1 and 2 Mbit/s DSSS specification as described in IEEE Std 802.11- 1999, and an optional short preamble and header.

18.2.2.1 Long PLCP PPDU format

Figure 1 shows the format for the interoperable (long) PPDU including the High Rate PLCP Preamble, the High Rate PLCP Header, and the PSDU. The PLCP Preamble contains the following fields: Synchronization (Sync) and Start Frame Delimiter (SFD). The PLCP Header contains the following fields: Signaling (SIGNAL), Service (SERVICE), Length (LENGTH), and CCITT CRC-16 field. Each of these fields is described in detail in 18.2.3. The format for the PPDU including the long High Rate PLCP preamble, the long High Rate PLCP header and the PSDU do not differ from the IEEE Std 802.11- 1999 for 1 and 2 Mbit/s. The only exceptions are the encoding of the rate in the SIGNAL Field and the use of bits in the SERVICE field to resolve an ambiguity in PSDU length in octets when the length is expressed in whole microseconds and to indicate if the optional PBCC mode is being used

.

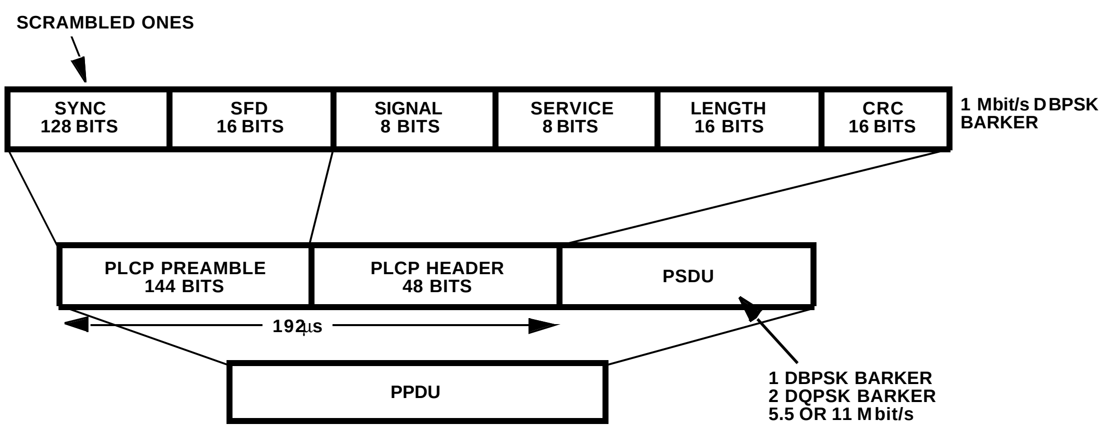


Figure 1—Long PLCP PPDU format

18.2.2.2 Short PLCP PPDU format (Optional)

The short PLCP preamble and header (HR/DSSS/short) is defined as optional. The short preamble and header may be used to minimize overhead and thus maximize the network data throughput. The format of the PPDU with HR/DSSS/short is depicted in Figure 2.

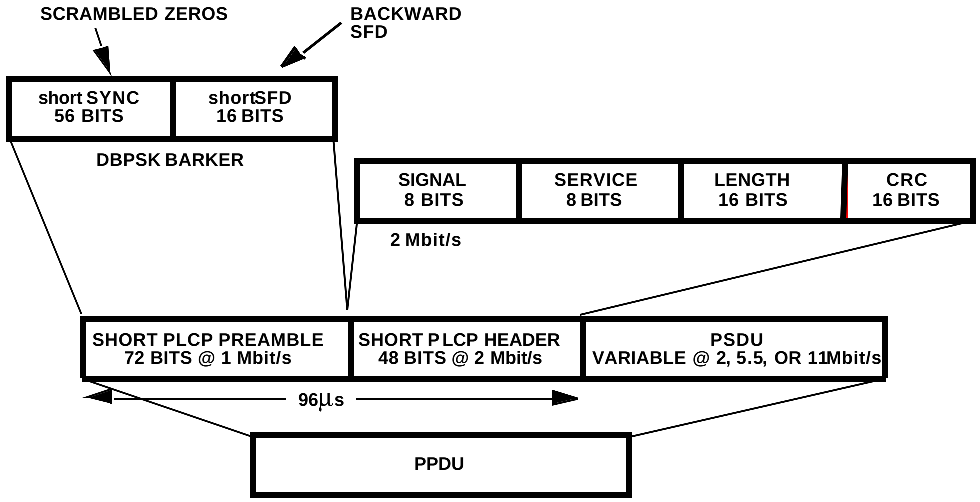


Figure 2—Short PLCP PPDU format

A transmitter using the short PLCP will only be interoperable with another receiver which is also capable of receiving this short PLCP. To interoperate with a receiver that is not capable of receiving a short preamble and header, the transmitter shall use the long PLCP preamble and header. The short PLCP preamble uses the 1 Mbit/s Barker code spreading with DBPSK modulation. The short PLCP header uses the 2 Mbit/s Barker code spreading with DQPSK modulation and the PSDU is transmitted at 2Mbit/s, 5.5 Mbit/s or 11 Mbit/s.

Stations not implementing this option that do active scanning will get a response even when the network is using short preambles since all management traffic is returned with the same type preamble as received.

18.2.3 PLCP PPDU field definitions

In the following PLCP field definition subclauses, the definitions for the Long (i.e. clause 15) PLCP fields are described first. Subsequently, the definitions of the short PLCP are defined. The names for the short PLCP fields are preceded with the term Short.

18.2.3.1 Long PLCP Synchronization Field (SYNC)

The SYNC field shall consist of 128 bits of scrambled "1" bits. This field is provided so the receiver can perform the necessary synchronization operations. The initial state of the scrambler (seed) shall be [1101100], where the left most bit specifies the value to put in the first delay element (Z^1) in Figure 5 and the right most bit specifies the value to put in the last delay element in the scrambler.

To support the reception of DSSS signals generated with implementations based on clause 15, the receiver shall also be capable of synchronization on a SYNC field derived from any non-zero scrambler initial state.

18.2.3.2 Long PLCP Start Frame Delimiter (SFD)

The SFD shall be provided to indicate the start of PHY dependent parameters within the PLCP Preamble. The SFD shall be a 16-bit field, [1111 0011 1010 0000], where the right most bit shall be transmitted first in time.

18.2.3.3 Long PLCP Signal (SIGNAL) field

The 8-bit signal field indicates to the PHY the modulation that shall be used for transmission (and reception) of the PSDU. The data rate shall be equal to the SIGNAL field value multiplied by 100 kbit/s. The High Rate PHY supports four mandatory rates given by the following 8 bit words which represent the rate in units of 100 kbit/s, where the lsb shall be transmitted first in time:

a) X'0A' (msb to lsb) for 1 Mbit/s
b) X'14' (msb to lsb) for 2 Mbit/s
c) X'37' (msb to lsb) for 5.5 Mbit/s
d) X'6E' (msb to lsb) for 11 Mbit/s

The High Rate PHY rate change capability is described in 18.2.3.14. This field shall be protected by the CCITT CRC-16 frame check sequence described in 18.2.3.6.

18.2.3.4 Long PLCP SERVICE (SERVICE) field

Three bits have been defined in the SERVICE field to support the high rate extension. The right most bit (bit 7) shall be used to supplement the LENGTH field described in 18.2.3.5. Bit 3 shall be used to indicate whether the modulation method is CCK <0> or PBCC <1> as shown in Table 1. Bit 2 is used to indicate that the transmit frequency and symbol clocks are derived from the same oscillator. This Locked Clocks bit shall be set by the PHY layer based on its implementation configuration. The SERVICE field shall be transmitted b0 first in time and shall be protected by the CCITT CRC-16 frame check sequence described in 18.2.3.6. An IEEE802.11 compliant device shall set the values of the bits b0, b1, b4, b5 and b6 to 0.

Table 1. SERVICE field definitions

b0	b1	b2	b3	b4	b5	b6	b7
Reserved	Reserved	Locked Clocks Bit 0 = not 1 = locked	Mod. Selection Bit 0 = CCK 1 = PBCC	Reserved	Reserved	Reserved	Length Extension Bit

18.2.3.5 Long PLCP Length (LENGTH) field

The PLCP length field shall be an unsigned 16 bit integer which indicates the number of microseconds required to transmit the PSDU. The transmitted value shall be determined from the LENGTH and DataRate parameters in the TXVECTOR issued with the PHY-TXSTART.request primitive described in subclause 18.4.4.2.

The length field provided in the TXVECTOR is in octets and is converted to microseconds for inclusion in the PLCP LENGTH field. The LENGTH field is calculated as follows: Since there is an ambiguity in the number of octets that is described by a length in integer microseconds for any data rate over 8 Mbit/s, a Length Extension bit shall be placed at bit position b7 in the SERVICE field to indicate when the smaller potential number of octets is correct.

a) 5.5Mbit/s CCK Length = number of octets * 8/5.5, rounded up to the next integer.
b) 11Mbit/s CCK Length = number of octets * 8/11, rounded up to the next integer and the service field b7 bit shall indicate a '0' if the rounding took less than 8/11 or a '1' if the rounding took more than or equal to 8/11.
c) 5.5 Mbit/s PBCC Length = (number of octets + 1)* 8/5.5, rounded up to the next integer.

d) 11 Mbit/s PBCC Length = (number of octets + 1)* 8/11, rounded up to the next integer and the service field b7 bit shall indicate a ‘0’ if the rounding took less than 8/11 or a ‘1’ if the rounding took more than or equal to 8/11.

At the receiver, the number of octets in the MPDU is calculated as follows:

a) 5.5 Mbit/s CCK number of octets = Length * 5.5/8, rounded down to the next integer
b) 11 Mbit/s CCK number of octets = Length * 11/8, rounded down to the next integer, minus 1 if the service field b7 bit is a ‘1’.
c) 5.5 Mbit/s PBCC number of octets = (Length * 5.5/8) -1, rounded down to the next integer
d) 11 Mbit/s PBCC number of octets = (Length * 11/8) -1, rounded down to the next integer, minus 1 if the service field b7 bit is a ‘1’.

An example for an 11 Mbit/s calculation described in psuedocode form is shown below.
At the transmitter, the values of the Length field and Length Extension bit are calculated as follows:

```
LENGTH’ = ((number of octets + P) *8) / R
LENGTH = Ceiling(LENGTH’)

IF (R = 11) AND (LENGTH - LENGTH’) >= 8/11)
Then LengthExtension = 1
Else LengthExtension = 0
```

Where:
R = data rate in Mbit/s
P = 0 for CCK, =1 for PBCC
Ceiling(X) returns the smallest integer value greater than or equal to X.

At the receiver, the number of octets in the MPDU is calculated as follows:

number of octets = Floor(((Length*R) / 8) - P) - LengthExtension

Where:
R = data rate in Mbit/s
P = 0 for CCK, =1 for PBCC
Floor(X) returns the largest integer value less than or equal to X.

Table 2 shows an example calculation for several packet lengths of CCK at 11 Mbit/s:

Table 2-Example of LENGTH calculations for CCK

TX Octets	Octets *8/11	LENGTH	Length Extension bit	LENGTH *11/8	floor(X)	RX Octets
1023	744	744	0	1023	1023	1023
1024	744.7273	745	0	1024.375	1024	1024
1025	745.4545	746	0	1025.75	1025	1025
1026	746.1818	747	1	1027.125	1027	1026

Table 3 shows an example calculation for several packet lengths of PBCC at 11 Mbit/s:

Table 3-Example of LENGTH calculations for PBCC

TX Octets	(Octets *8/11) + 1	LENGTH	Length Extension bit	(LENGTH *11/8) - 1	floor(X)	RX Octets
1023	744.7273	745	0	1023.375	1023	1023
1024	745.4545	746	0	1024.750	1024	1024
1025	746.1818	747	1	1026.125	1026	1025
1026	746.9091	747	0	1026.125	1026	1026

This example illustrates why normal rounding or truncation of the number will not produce the right result. The length field is defined in units of microseconds, and must correspond to the actual length and the number of octets must be exact.

The lsb (least significant bit) shall be transmitted first in time. This field shall be protected by the CCITT CRC-16 frame check sequence described in subclause 18.2.3.6.

18.2.3.6 PLCP CRC (CCITT CRC-16) field

The SIGNAL, SERVICE, and LENGTH fields shall be protected with a CCITT CRC-16 FCS (frame check sequence). The CCITT CRC-16 FCS shall be the one's complement of the remainder generated by the modulo 2 division of the protected PLCP fields by the polynomial:

$$x^{16} + x^{12} + x^5 + 1$$

The protected bits shall be processed in transmit order. All FCS calculations shall be made prior to data scrambling. A schematic of the processing is shown in Figure 3

As an example, the SIGNAL, SERVICE, and LENGTH fields for a DBPSK signal with a PPDU length of 192 µs (24 octets) would be given by the following:

0101 0000 0000 0000 0000 0011 0000 0000 (leftmost bit (b0) transmitted first in time)
b0..b48

The one's complement FCS for these protected PLCP Preamble bits would be the following:

0101 1011 0101 0111 (leftmost bit (b0) transmitted first in time)
b0.........................b16

Figure 3 depicts this example.

An illustrative example of the CCITT CRC-16 FCS using the information from Figure 3 follows in Figure 4.

18.2.3.7 Long PLCP Data Modulation and Modulation Rate Change

The long PLCP preamble and header shall be transmitted using the 1 Mbit/s DBPSK modulation. The SIGNAL and SERVICE fields combined shall indicate the modulation which shall be used to transmit the PSDU. The SIGNAL field indicates the rate and the SERVICE field indicates the modulation. The transmitter and receiver shall initiate the modulation and rate indicated by the SIGNAL and SERVICE fields starting

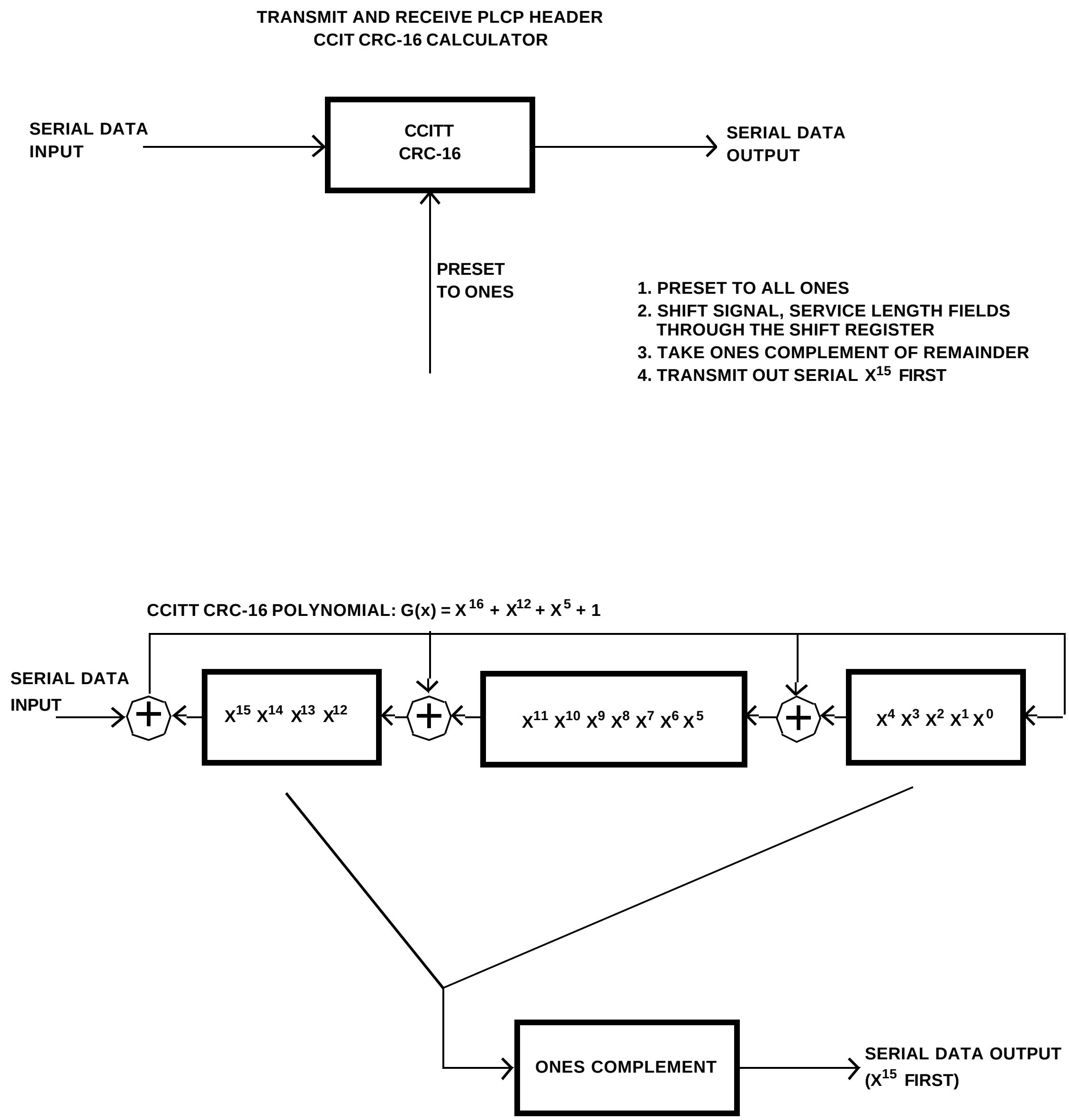


Figure 3—CCITT CRC-16 Implementation

with the first octet of the PSDU. The PSDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in subclause 18.4.4.1.

18.2.3.8 Short PLCP Synchronization (shortSYNC)

The shortSYNC field shall consist of 56 bits of scrambled "0" bits. This field is provided so the receiver can perform the necessary synchronization operations. The initial state of the scrambler (seed) shall be [001 1011], where the left end bit specifies the value to place in the first delay element (Z^1) in Figure 5 and the right end bit specifies the value to place in the last delay element (Z^7).

Data	CRC registers msb lsb	
	1111111111111111	; initialize preset to 1's
0	1110111111011111	
1	1101111110111110	
0	1010111101011101	
1	0101111010111010	
0	1011110101110100	
0	0110101011001001	
0	1101010110010010	
0	1011101100000101	
0	0110011000101011	
0	1100110001010110	
0	1000100010001101	
0	0000000100111011	
0	0000001001110110	
0	0000010011101100	
0	0000100111011000	
0	0001001110110000	
0	0010011101100000	
0	0100111011000000	
0	1001110110000000	
0	0010101100100001	
0	0101011001000010	
0	1010110010000100	
1	0101100100001000	
1	1010001000110001	
0	0101010001000011	
0	1010100010000110	
0	0100000100101101	
0	1000001001011010	
0	0001010010010101	
0	0010100100101010	
0	0101001001010100	
0	1010010010101000	
	0101101101010111	; one's complement, result = CRC FCS parity

Figure 4—Example CRC calculation

18.2.3.9 Short PLCP Start Frame Delimiter Field (shortSFD)

The shortSFD shall be a 16 bit field and be the time reverse of the field of the SFD in the long PLCP preamble (subclause18.2.3.2). The field is the bit pattern 0000 0101 1100 1111. The right end bit shall be transmitted first in time. A receiver not configured to use the short header option will not detect this SFD.

18.2.3.10 Short PLCP SIGNAL Field (shortSignal)

The 8 bit SIGNAL Field of the short header indicates to the PHY the data rate which shall be used for transmission (and reception) of the PSDU. A PHY operating with a HR/DSSS/short option supports three mandatory rates given by the following 8 bit words, where the lsb shall be transmitted first in time and the number represents the rate in units of 100 kBit/s:

a) X'14'(msb to lsb) for 2 Mbit/s

b) X'37'(msb to lsb) for 5.5 Mbit/s

c) X'6E'(msb to lsb) for 11 Mbit/s

18.2.3.11 Short PLCP SERVICE Field (shortService)

The SERVICE field in the short header shall be the same as the SERVICE field described in subclause 18.2.3.4.

18.2.3.12 Short PLCP Length Field (shortLENGTH)

The LENGTH field in the short header shall be the same as the LENGTH field described in subclause 18.2.3.5

18.2.3.13 Short CCITT CRC-16 Field (shortCRC)

The CRC in the short header shall be the same as the CRC field as defined in subclause 18.2.3.6. The CRC-16 is calculated over the shortSIGNAL, shortSERVICE, and shortLENGTH fields.

18.2.3.14 Short PLCP Data Modulation and Modulation rate Change

The short PLCP preamble shall be transmitted using the 1 Mbit/s DBPSK modulation. The short PLCP header shall be transmitted using the 2 Mbit/s modulation. The SIGNAL and SERVICE fields combined shall indicate the modulation which shall be used to transmit the PSDU. The SIGNAL field indicates the rate and the SERVICE field indicates the modulation. The transmitter and receiver shall initiate the modulation and rate indicated by the SIGNAL and SERVICE fields starting with the first octet of the PSDU. The PSDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in subclause 18.4.4.1.

18.2.4 PLCP/High Rate PHY data scrambler and descrambler

The polynomial $G(z) = z^{-7} + z^{-4} + 1$ shall be used to scramble *all* bits transmitted. The feedthrough configuration of the scrambler and descrambler is self-synchronizing, which requires no prior knowledge of the transmitter initialization of the scrambler for receive processing. Figure 5 and Figure 6 show typical implementations of the data scrambler and descrambler, but other implementations are possible.

The scrambler shall be initialized as specified in subclause 18.2.3.8 for the short PLCP and subclause 18.2.3.1 for the long PLCP. For a long preamble, this shall result in the scrambler registers Z^1 through Z^7 in Figure 5 having the data pattern: [1101100] (i.e. Z^1=1… Z^7=0) when the scrambler is first started. The scrambler shall be initialized with the reverse pattern, [0011011] when transmitting the optional short preamble.

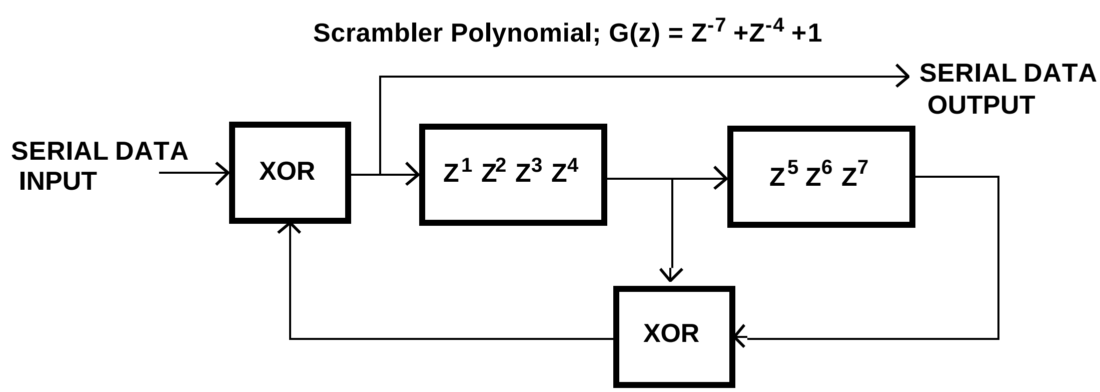


Figure 5—Data scrambler

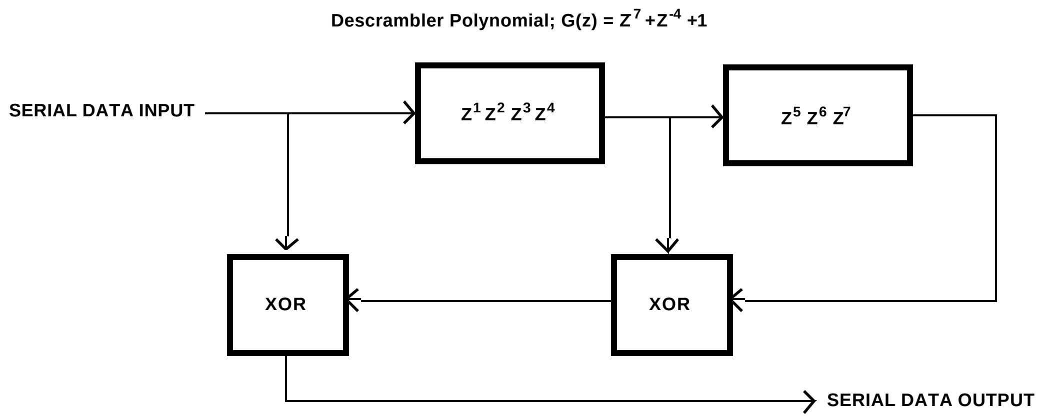


Figure 6—Data descrambler

18.2.5 PLCP transmit procedure

The transmit procedures for a High Rate PHY using the long PLCP preamble and header are the same as those described in IEEE 802.11 Std- 1999, subclauses 15.2.7 and 15.2.8 and do not change apart from the ability to transmit 5.5 and 11 Mbit/s.

The procedures for a transmitter employing HR/DSSS/short and HR/DSSS/PBCC/short are the same except for length and rate changes. The decision to use a long or short PLCP is beyond the scope of this standard.

The PLCP transmit procedure is shown in Figure 7.

A PHY-TXSTART.request(TXVECTOR) primitive will be issued by the MAC to start the transmission of a PPDU. In addition to DATARATE and LENGTH other transmit parameters such as PREAMBLE_TYPE and MODULATION are set via the PHY-SAP with the PHY-TXSTART.request(TXVECTOR) as described in 18.3.5. The SIGNAL, SERVICE and LENGTH fields of the PLCP header are calculated as described in subclause 18.2.3.

The PLCP shall issue PMD_ANTSEL, PMD_RATE, and PMD_TXPWRLVL primitives to configure the PHY. The PLCP shall then issue a PMD_TXSTART.request and the PHY entity shall immediately initiate data scrambling and transmission of the PLCP Preamble based on the parameters passed in the PHY-TXSTART.request primitive. The time required for TX power on ramp described in 18.4.7.6 shall be included in the PLCP synchronization field. Once the PLCP Preamble transmission is complete, data shall be exchanged between the MAC and the PHY by a series of PHY-DATA.request(DATA) primitives issued by the MAC and PHY-DATA.confirm primitives issued by the PHY. The modulation and rate change, if any, shall be initiated with the first data symbol of the PSDU as described in 18.2.3.7 and 18.2.3.14. The PHY proceeds with PSDU transmission through a series of data octet transfers from the MAC. At the PMD layer, the data octets are sent in lsb to msb order and presented to the PHY layer through PMD_DATA.request primitives. Transmission can be prematurely terminated by the MAC through the primitive PHY-TXEND.request. PHY-TXSTART shall be disabled by the issuance of the PHY-TXEND.request. Normal termination occurs after the transmission of the final bit of the last PSDU octet calculated from the number supplied in the PHY preamble LENGTH and SERVICE fields using the equations specified in 18.2.3.5. The PPDU transmission shall be completed and the PHY entity shall enter the receive state (i.e., PHY-TXSTART shall be disabled). It is recommended that modulation continue during power-down to prevent radiating a CW carrier. Each PHY-TXEND.request is acknowledged with a PHY-TXEND.confirm primitive from the PHY.

A typical state machine implementation of the PLCP transmit procedure is provided in Figure 8.

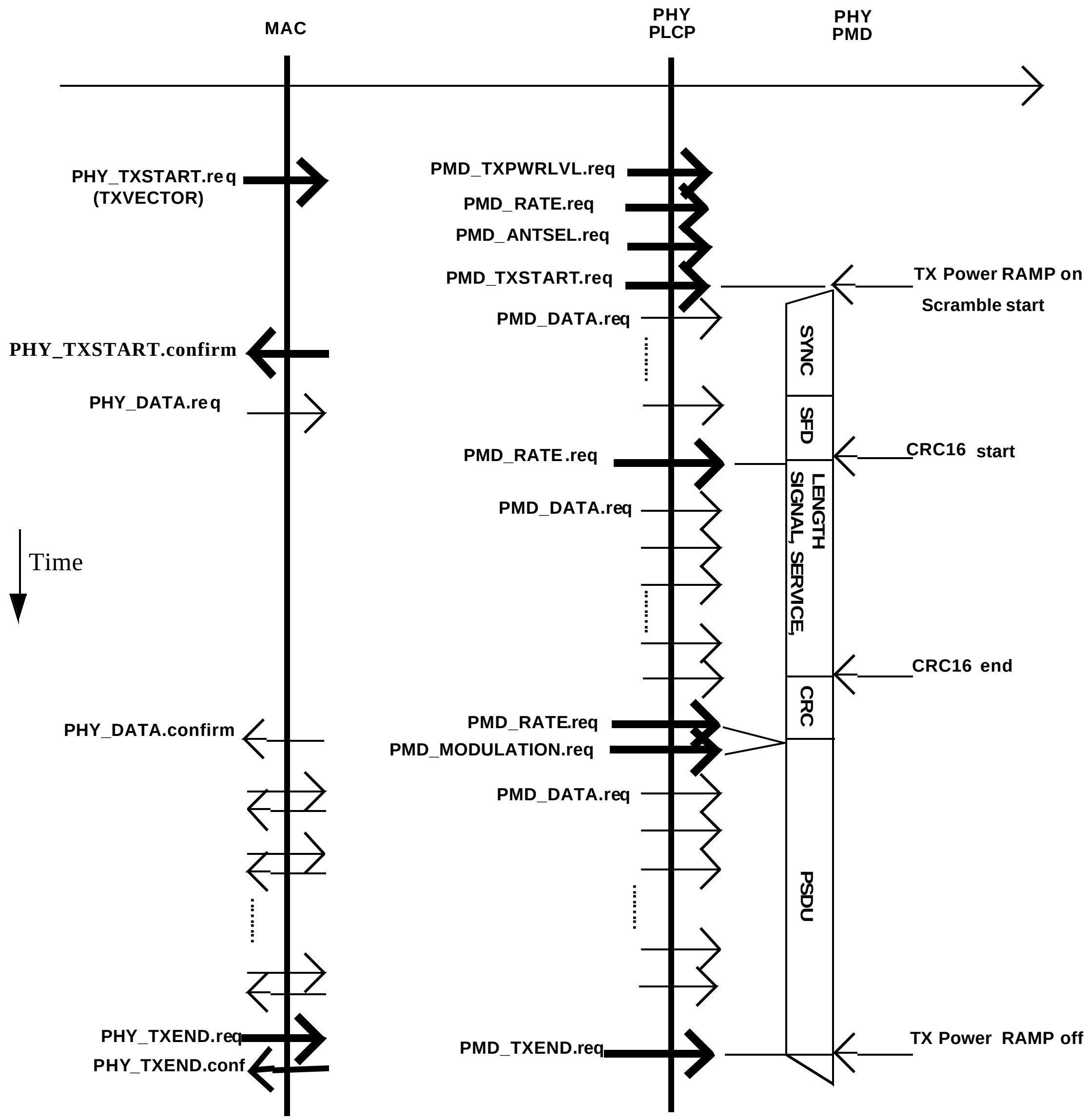


Figure 7—PLCP Transmit Procedure

18.2.6 PLCP receive procedure

The receive procedures for receivers configured to receive the mandatory and optional PLCPs, Rates and Modulations are described in this section. A receiver that supports this high rate extension of the standard is capable of receiving 5.5 Mbit/s and 11 Mbit/s in addition to 1 and 2 Mbit/s. If the PHY implements the short preamble option, it shall detect both short and long preamble formats and indicate which type of preamble was received in the RXVECTOR. If the PHY implements the PBCC modulation option it shall detect either CCK or PBCC modulations as indicated in the SIGNAL field and shall report the type of modulation used in the RXVECTOR.

The receiver shall implement the CCA procedure as define in subclause 18.4.8.4.

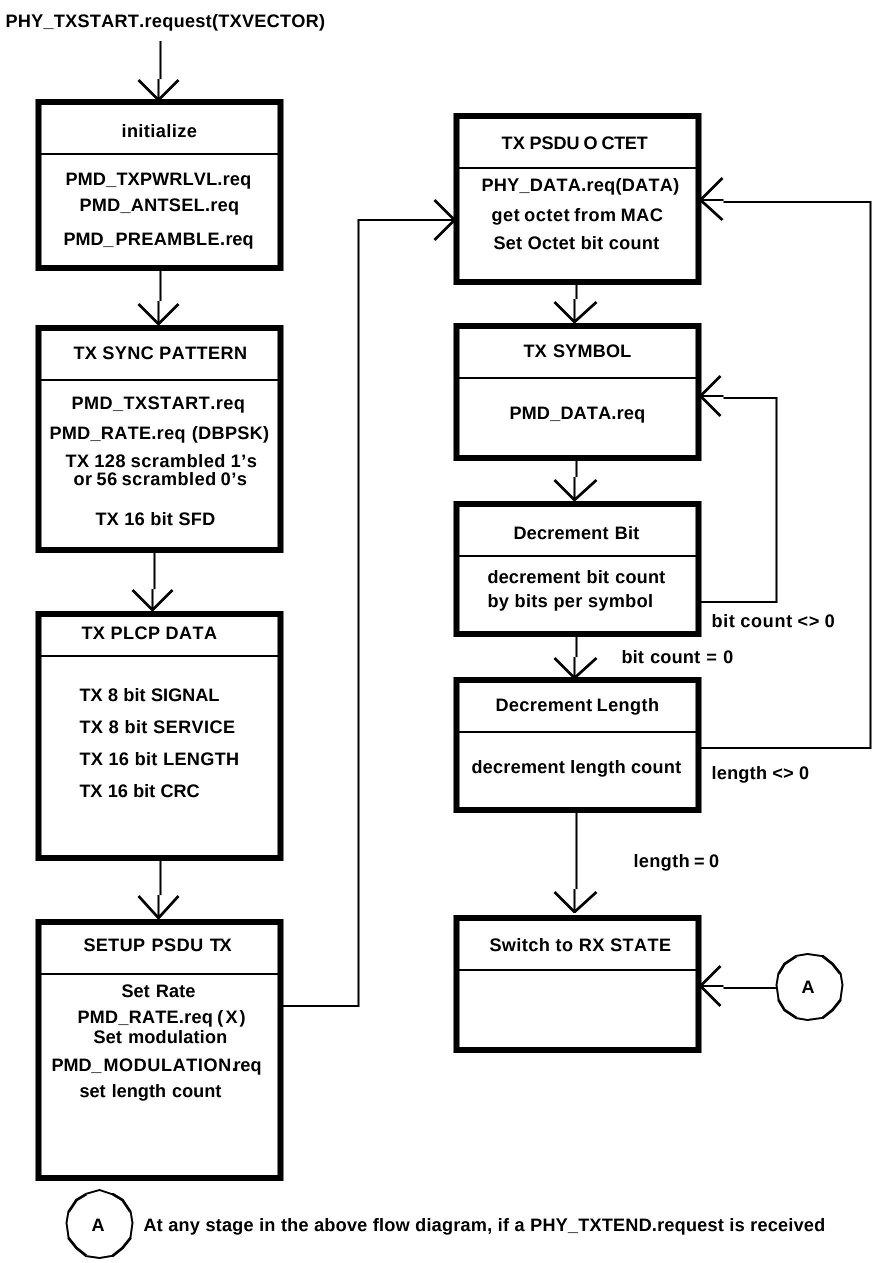


Figure 8—PLCP transmit state machine

Upon receiving a PPDU the receiver shall distinguish between a long and short header format by the value of the SFD as specified in 18.2.2 The receiver shall demodulate a long PLCP header using BPSK at 1 Mbit/s. The receiver shall demodulate a short PLCP header using QPSK at 2 Mbit/s. The receiver shall use the SIGNAL and SERVICE fields of the PLCP header to determine the data rate and the modulation of the PSDU.

The PLCP receive procedure is shown in Figure 9.

In order to receive data, PHY-TXSTART.request shall be disabled so that the PHY entity is in the receive state. Further, through station management via the PLME, the PHY is set to the appropriate channel and the

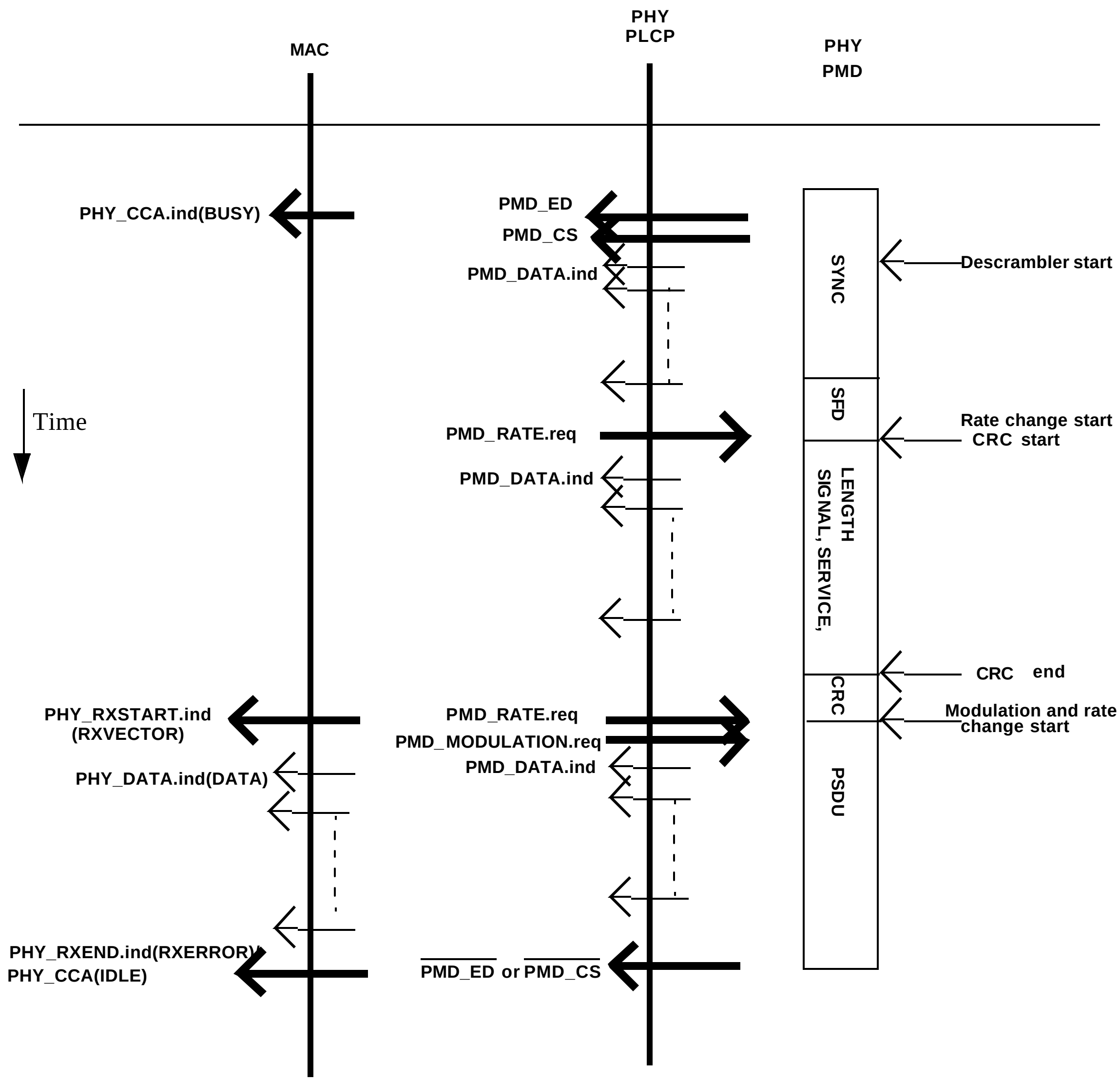


Figure 9—PLCP receive procedure

CCA method is chosen. Other receive parameters such as receive signal strength indication (RSSI), signal quality (SQ), and indicated DATARATE may be accessed via the PHY-SAP.

Upon receiving the transmitted energy, according to the selected CCA mode, the PMD_ED shall be enabled (according to 18.4.8.4) as the RSSI strength reaches the ED_THRESHOLD and/or PMD_CS shall be enabled after code lock is established. These conditions are used to indicate activity to the MAC via PHY-CCA.indicate according to 18.4.8.4. PHY-CCA.indicate(BUSY) shall be issued for energy detection and/or code lock prior to correct reception of the PLCP header. The PMD primitives PMD_SQ and PMD_RSSI are issued to update the RSSI and SQ parameters reported to the MAC.

After PHY-CCA.indicate is issued, the PHY entity shall begin searching for the SFD field. Once the SFD field is detected, CCITT CRC-16 processing shall be initiated and the PLCP SIGNAL, SERVICE and LENGTH fields are received. The CCITT CRC-16 FCS shall be processed. If the CCITT CRC-16 FCS check fails, the PHY receiver shall return to the RX Idle state as depicted in Figure 10. Should the status of CCA return to the IDLE state during reception prior to completion of the full PLCP processing, the PHY receiver shall return to the RX Idle state.

If the PLCP Header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indicate(RXVECTOR) shall be issued. The RXVECTOR associated with this primitive includes the SIGNAL field, the SERVICE field, the PSDU length in octets (calculated from the LENGTH field in microseconds and the DATARATE in Mbit/s in accordance with the formula in subclause 18.2.3.5), RXPREAMBLE_TYPE (which is an enumerated type taking on values SHORTPREAMBLE or LONGPREAMBLE), the antenna used for receive (RX_ANTENNA), RSSI, and SQ.

The received PSDU bits are assembled into octets and presented to the MAC using a series of PHY-DATA.indicate(DATA) primitive exchanges. The rate and modulation change indicated in the SIGNAL field shall be initiated with the first symbol of the PSDU as described in 18.2.5. The PHY proceeds with PSDU reception. After the reception of the final bit of the last PSDU octet indicated by the PLCP Preamble LENGTH field, the receiver shall be returned to the RX Idle state as shown in Figure 10. A PHY-RXEND.indicate(NoError) primitive shall be issued. A PHY-CCA.indicate(IDLE) primitive shall be issued following a change in PHYCS (PHY carrier sense) and/or PHYED (PHY energy detection) according to the selected CCA method.

In the event that a change in PHYCS or PHYED would cause the status of CCA to return to the IDLE state before the complete reception of the PSDU as indicated by the PLCP LENGTH field, the error condition PHY-RXEND.indicate(CarrierLost) shall be reported to the MAC. The High Rate PHY shall ensure that the CCA will indicate a busy medium for the intended duration of the transmitted PPDU.

If the PLCP Header is successful, but the indicated rate or modulation in the SIGNAL and SERVICE fields are not within the capabilities of the receiver, a PHY-RXSTART.indicate shall not be issued. The PHY shall issue the error condition PHY-RXEND.indicate(UnsupportedRate). If the PLCP Header is invalid, a PHY-RXSTART.indicate shall not be issued and the PHY shall issue the error condition PHY-RXEND.indicate(FormatViolation). Also, in both cases, the High Rate PHY shall ensure that the CCA shall indicate a busy medium for the intended duration of the transmitted PSDU as indicated by the LENGTH field. The intended duration is indicated by the LENGTH field (LENGTH $\times$ 1 μs).

A typical state machine implementation of the PLCP receive procedure is provided in Figure 10.

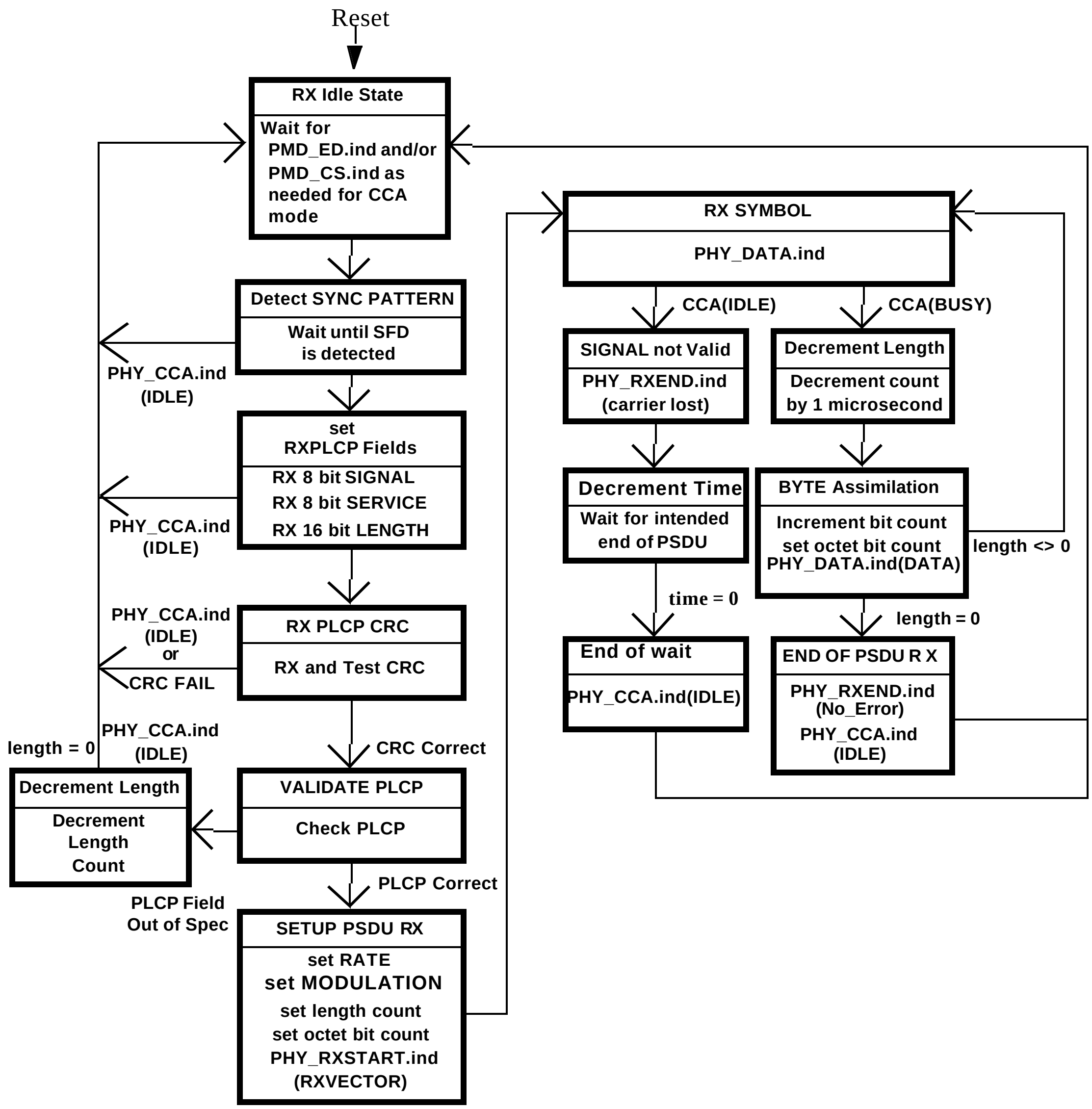


Figure 10—PLCP receive state machine

18.3 High Rate physical layer management entity (PLME)

18.3.1 PLME_SAP sublayer management primitives

Table 4 lists the MIB attributes that may be accessed by the PHY sublayer entities and intralayer or higher layer management entities (LME). These attributes are accessed via the PLME-GET, PLME-SET, and PLME-RESET primitives defined in IEEE Std 802.11- 1999 Clause 10.

18.3.2 High Rate PHY MIB

All High Rate PHY MIB attributes are defined in Annex D of IEEE Std 802.11- 1999, with specific values defined in Table 4.

Table 4—MIB Attribute Default Values/Ranges

Managed object	Default value/range	Operational semantics
dot11PhyOperationTable		
dot11PHYType	High Rate -2.4(X’05’)	Static
dot11TempType	Implementation dependent	Static
dot11CurrentRegDomain	Implementation dependent	Static
dot11ShortPreambleOptionImplemented	Implementation dependent	Static
dot11PBCCOptionImplemented	Implementation dependent	Static
dot11ChannelAgility Present	Implementation dependent	Static
dot11ChannelAgilityEnabled	False/Boolean	Dynamic
dot11PhyAntennaTable		
dot11CurrentTxAntenna	Implementation dependent	Dynamic
dot11DiversitySupport	Implementation dependent	Static
dot11CurrentRxAntenna	Implementation dependent	Dynamic
dot11PhyTxPowerTable		
dot11NumberSupportedPowerLevels	Implementation dependent	Static
dot11TxPowerLevel1	Implementation dependent	Static
dot11TxPowerLevel2	Implementation dependent	Static
dot11TxPowerLevel3	Implementation dependent	Static
dot11TxPowerLevel4	Implementation dependent	Static
dot11TxPowerLevel5	Implementation dependent	Static
dot11TxPowerLevel6	Implementation dependent	Static
dot11TxPowerLevel7	implementation dependent	Static
dot11TxPowerLevel8	Implementation dependent	Static
dot11CurrentTxPowerLevel	Implementation dependent	Dynamic
dot11PhyDSSSTable		
dot11CurrentChannel	Implementation dependent	Dynamic
dot11CCAModeSupported	Implementation dependent	Static
dot11CurrentCCAMode	Implementation dependent	Dynamic
dot11EDThreshold	Implementation dependent	Dynamic
dot11AntennasListTable		
dot11SupportTxAntenna	Implementation dependent	Static
dot11SupportRxAntenna	Implementation dependent	Static
dot11DiversitySelectionRx	Implementation dependent	Dynamic

Table 4—MIB Attribute Default Values/Ranges *(continued)*

Managed object	Default value/range	Operational semantics
dot11RegDomainsSupportedTable		
dot11RegDomainsSupported	Implementation dependent	Static
dot11SupportedDataRatesTx	Table Tx X'02', X'04,' X’0B’, X’16’	Static
dot11SupportedDataRatesRx	Table Rx X'02', X'04,' X’0B’, X’16’	Static
NOTE—The column titled “Operational semantics” contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. MIB attributes defined as dynamic can be modified by some management entities.		

18.3.3 DS PHY characteristics

The static DS PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 5. The definitions of these characteristics are in IEEE Std 802.11- 1999 subclause 10.4.3.

Table 5—High Rate PHY Characteristics

Characteristic	Value
aSlotTime	20 µs
aSIFSTime	10 µs
aCCATime	≤15 µs
aRxTxTurnaroundTime	≤5 µs
aTxPLCPDelay	Implementors may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxPLCPDelay	Implementors may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met.
aRxTxSwitchTime	≤5 µs
aTxRampOnTime	Implementors may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.

Table 5—High Rate PHY Characteristics

Characteristic	Value
aTxRampOffTime	Implementors may choose any value for this delay as long as the requirements of aSIFSTime are met.
aTxRFDelay	Implementors may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxRFDelay	Implementors may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met.
aAirPropagationTime	1 µs
aMACProcessingDelay	0 (not applicable)
aPreambleLength	144 µs
aPLCPHeaderLength	48 bits
aMPUMaxLength	$14 \leq x \leq (2^{12} - 1)$
aCWmin	31
aCWmax	1023

18.3.4 High Rate TXTIME calculation

The value of the TXTIME parameter returned by the PLME-TXTIME.confirm primitive shall be calculated according to the following equation:

TXTIME = PreambleLength + PLCPHeaderTime + Ceiling(((LENGTH+PBCC) x 8) / DATARATE)

Where LENGTH and DATARATE are values from the TXVECTOR parameter of the corresponding PLME-TXTIME.request primitive. PBCC has a value of 1 if the SIGNAL value from the TXVECTOR parameter specifies PBCC and has a value of 0 otherwise. The value of PreambleLength is 144 microseconds if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates "LONGPREAMBLE" or 72 microseconds if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates "SHORTPREAMBLE". The value of PLCPHeaderTime is 48 microseconds if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates "LONGPREAMBLE" or 24 microseconds if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates "SHORTPREAMBLE". LENGTH is in units of octets. DATARATE is in units of Mbit/s. Ceiling is a function which returns the smallest integer value greater than or equal to its argument value.

18.3.5 Vector Descriptions

Several service primitives include a parameter vector. These vectors are a list of parameters as described in Table 6. DATARATE and LENGTH are described in subclause 12.3.4.4 in IEEE Std 802.11- 1999. The remaining parameters are considered to be management parameters and are specific to this PHY.

Table 6, Parameter Vectors

Parameter	Associated Vector	Value
DATARATE	RXVECTOR, TXVECTOR	The rate used to transmit the PSDU in Mbit/s
LENGTH	RXVECTOR, TXVECTOR	The length of the PSDU in octets.

PREAMBLE_TYPE	RXVECTOR, TXVECTOR	The preamble used for the transmission of this PPDU. This is an enumerated type that can take the value SHORTPREAMBLE or LONGPREAMBLE.
MODULATION	RXVECTOR, TXVECTOR	The modulation used for the transmission of this PSDU. This is an integer where 0 means CCK and 1 means PBCC.

18.4 High Rate PMD sublayer

18.4.1 Scope and field of application

This subclause describes the Physical Medium Dependent (PMD) services provided to the PLCP for the High Rate PHY. 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 High Rate physical layer is shown in Figure 11.

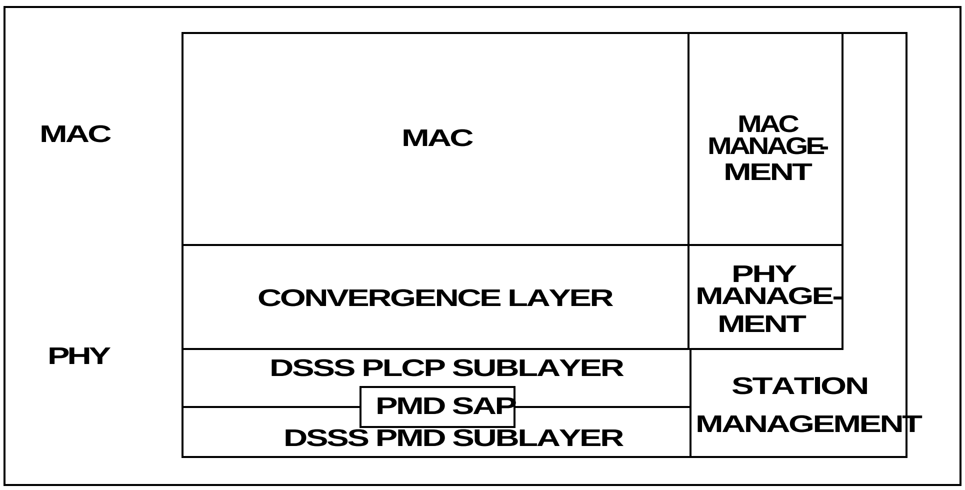


Figure 11. Layer Reference Model

18.4.2 Overview of service

The High Rate PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which data is transmitted or received from the medium. The combined function of High Rate PMD sublayer primitives and parameters for the receive function results in a data stream, timing information, and associated received signal parameters being delivered to the PLCP sublayer. A similar functionality is provided for data transmission.

18.4.3 Overview of interactions

The primitives associated with the PLCP sublayer to the High Rate PMD fall into two basic categories:

a) Service primitives that support PLCP peer-to-peer interactions

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

18.4.4 Basic service and options

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

18.4.4.1 PMD_SAP peer-to-peer service primitives

Table 7 indicates the primitives for peer-to-peer interactions.

Table 7—PMD_SAP Peer-to-Peer Service Primitives

Primitive	Request	Indicate	Confirm	Response
PMD_DATA	X	X		

18.4.4.2 PMD_SAP sublayer-to-sublayer service primitives

Table 8 indicates the primitives for sublayer-to-sublayer interactions.

Table 8—PMD_SAP Sublayer-to-Sublayer Service Primitives

Primitive	Request	Indicate	Confirm	Response
PMD_TXSTART	X			
PMD_TXEND	X			
PMD_ANTSEL	X	X		
PMD_TXPWRLVL	X			
PMD_MODULATIO N	X	X		
PMD_PREAMBLE	X	X		
PMD_RATE	X	X		
PMD_RSSI		X		
PMD_SQ		X		
PMD_CS		X		
PMD_ED	X	X		

18.4.4.3 PMD_SAP service primitive parameters

18.4.5 PMD_SAP detailed service specification

The following subclauses describe the services provided by each PMD primitive.

18.4.5.1 PMD_DATA.request

18.4.5.1.1 Function

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

18.4.5.1.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
TXD_UNIT	PMD_DATA.request	0,1: 1 Mbit/s 00,01,11,10:2 Mbit/s X'0' - X'F': 5.5 Mbit/s X'00' - X'FF': 11 Mbit/s	This parameter represents a single block of data, which, in turn, is used by the PMD to be differentially encoded into a transmitted symbol. The symbol itself is spread by the PN code prior to transmission.

18.4.5.1.3 When generated

This primitive is generated by the PLCP sublayer to request transmission of a symbol. The data clock for this primitive is supplied by PMD layer based on the PN code repetition.

18.4.5.1.4 Effect of receipt

The PMD performs the differential encoding, PN code modulation and transmission of the data.

18.4.5.2 PMD_DATA.indicate

18.4.5.2.1 Function

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

18.4.5.2.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
RXD_UNIT	PMD_DATA.indicate	0,1: 1 Mbit/s 00,01,11,10:2 Mbit/s X'0' - X'F': 5.5 Mbit/s X'00' - X'FF': 11 Mbit/s	This parameter represents a single symbol that has been demodulated by the PMD entity.

18.4.5.2.3 When generated

This primitive, which is generated by the PMD entity, forwards received data to the PLCP sublayer. The data clock for this primitive is supplied by PMD layer based on the PN code repetition.

18.4.5.2.4 Effect of receipt

The PLCP sublayer either interprets the bit or bits that are recovered as part of the PLCP convergence procedure or passes the data to the MAC sublayer as part of the PSDU.

18.4.5.3 PMD_MODULATION.request

18.4.5.3.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the modulation code that is used by the High Rate PHY for transmission.

18.4.5.3.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
MODULATION	PMD_MODULATION.request PMD_MODULATION.indicate	1MbBarker, 2MbBarker, 5.5CCK, 11CCK , 5.5PBCC or 11PBCC	In Receive mode, the MODULATION parameter informs the PLCP layer which of the PHY data modulations was used to process the PSDU portion of the PPDU. Subclause 18.4.6.3 provides further information on the High Rate PHY modulation codes.

18.4.5.3.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY modulation code used for the PSDU portion of a PPDU. The PMD_MODULATION.request primitive is normally issued prior to issuing the PMD_TXSTART command.

18.4.5.3.4 Effect of receipt

The receipt of PMD_MODULATION selects the modulation that is used for all subsequent PSDU transmissions. This code is used for transmission only. The High Rate PHY shall still be capable of receiving all the required High Rate PHY modulations. This primitive, which is generated by the PMD entity, sets the state of the PHY for demodulation of the appropriate modulation.

18.4.5.4 PMD_PREAMBLE.request

18.4.5.4.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the preamble mode that is used by the High Rate PHY for transmission.

18.4.5.4.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
PREAMBLE	PMD_PREAMBLE.request	‘0’ for long ‘1’ for short	PREAMBLE selects which of the High Rate PHY preamble types is used for PLCP transmission. Subclause 18.2.2 provides further information on the High Rate PHY preamble modes.

18.4.5.4.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY preamble mode used for the PLCP portion of a PPDU. The PMD_PREAMBLE.request primitive is normally issued prior to issuing the PMD_TXSTART command.

18.4.5.4.4 Effect of receipt

The receipt of PMD_PREAMBLE selects the preamble mode that is used for all subsequent PSDU transmissions. This mode is used for transmission only. The High Rate PHY shall still be capable of receiving all the required High Rate PHY preambles. This primitive sets the state of the PHY for modulation of the appropriate mode.

18.4.5.5 PMD_PREAMBLE.indicate

18.4.5.5.1 Function

This primitive, which is generated by the PMD sublayer, indicates which preamble mode was used to receive the PLCP portion of the PPDU.

18.4.5.5.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
PREAMBLE	PMD_PREAMBLE.indicate	‘0’ for long ‘1’ for short	In receive mode, the PREAMBLE parameter informs the PLCP layer which of the High Rate PHY preamble modes was used to send the PLCP portion of the PPDU.

18.4.5.5.3 When generated

This primitive is generated by the PMD sublayer when the PLCP Preamble has been properly detected.

18.4.5.5.4 Effect of receipt

This parameter is provided to the PLCP layer for information only.

18.4.5.6 PMD_TXSTART.request

18.4.5.6.1 Function

As a result of receiving a PHY_DATA.request from the MAC, the PLCP issues this primitive, which initiates PPDU transmission by the PMD layer.

18.4.5.6.2 Semantics of the service primitive

This primitive has no parameters.

18.4.5.6.3 When generated

This primitive is generated by the PLCP sublayer to initiate the PMD layer transmission of the PPDU. The PHY-DATA.request primitive is provided to the PLCP sublayer prior to issuing the PMD_TXSTART command.

18.4.5.6.4 Effect of receipt

PMD_TXSTART initiates transmission of a PPDU by the PMD sublayer.

18.4.5.7 PMD_TXEND.request

18.4.5.7.1 Function

This primitive, which is generated by the PHY PLCP sublayer, ends PPDU transmission by the PMD layer.

18.4.5.7.2 Semantics of the service primitive

This primitive has no parameters.

18.4.5.7.3 When generated

This primitive is generated by the PLCP sublayer to terminate the PMD layer transmission of the PPDU.

18.4.5.7.4 Effect of receipt

PMD_TXEND terminates transmission of a PPDU by the PMD sublayer.

18.4.5.8 PMD_ANTSEL.request

18.4.5.8.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the antenna used by the PHY for transmission or reception (when diversity is disabled).

18.4.5.8.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
ANT_STATE	PMD_ANTSEL.request PMD_ANTSEL.indicate	1 to 256	ANT_STATE selects which of the available antennas should be used for transmit. The number of available antennas is determined from the MIB table parameters aSuprtRxAntennas and aSuprtTxAntennas.

18.4.5.8.3 When generated

This primitive is generated by the PLCP sublayer to select a specific antenna for transmission (or reception when diversity is disabled).

18.4.5.8.4 Effect of receipt

PMD_ANTSEL immediately selects the antenna specified by ANT_STATE.

18.4.5.9 PMD_TXPWRLVL.request

18.4.5.9.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the power level used by the PHY for transmission.

18.4.5.9.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
TXPWR_LEVEL	PHY-TXPWR_LEVEL.request	0, 1, 2, 3 (max of 4 levels)	TXPWR_LEVEL selects which of the optional transmit power levels should be used for the current PPDU transmission. The number of available power levels is determined by the MIB parameter dot11NumberSupportedPowerLevels. Subclause 18.4.7.2 provides further information on the optional High Rate PHY power level control capabilities.

18.4.5.9.3 When generated

This primitive is generated by the PLCP sublayer to select a specific transmit power. This primitive is applied prior to setting PMD_TXSTART into the transmit state.

18.4.5.9.4 Effect of receipt

PMD_TXPWRLVL immediately sets the transmit power level given by TXPWR_LEVEL.

18.4.5.10 PMD_RATE.request

18.4.5.10.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the data rate that shall be used by the High Rate PHY for transmission.

18.4.5.10.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
RATE	PMD_RATE.indicate PMD_RATE.request	X'0A' for 1 Mbit/s X'14' for 2 Mbit/s X'37' for 5.5 Mbit/s X'6E' for 11 Mbit/s	RATE selects which of the High Rate PHY data rates is used for PSDU transmission. Subclause 18.4.6.3 provides further information on the High Rate PHY data rates. The High Rate PHY rate change capability is fully described in 18.2.

18.4.5.10.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY data rate used for the PSDU portion of a PPDU.

18.4.5.10.4 Effect of receipt

The receipt of PMD_RATE selects the rate that is used for all subsequent PSDU transmissions. This rate is used for transmission only. The High Rate PHY shall still be capable of receiving all the required High Rate PHY data rates.

18.4.5.11 PMD_RSSI.indicate

18.4.5.11.1 Function

This optional primitive may be generated by the PMD to provide the received signal strength to the PLCP..

18.4.5.11.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
RSSI	PMD_RSSI.indicate	0–8 bits of RSSI	The RSSI is a measure of the RF energy received by the High Rate PHY.

18.4.5.11.3 When generated

This primitive is generated by the PMD when the High Rate PHY is in the receive state. It is continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

18.4.5.11.4 Effect of receipt

This parameter is provided to the PLCP layer for information only. The RSSI may be used in conjunction with SQ as part of a CCA scheme.

18.4.5.12 PMD_SQ.indicate

18.4.5.12.1 Function

This optional primitive, may be generated by the PMD to provide an indication of the signal quality (SQ) of the High Rate PHY PN code correlation to the PLCP. SQ is a measure of the quality of BARKER code lock, providing an effective measure during the full reception of a PLCP preamble and header.

18.4.5.12.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
SQ	PMD_SQ.indicate	0–8 bits of SQ	This primitive is a measure of the signal quality received by the HR/DSSS PHY.

18.4.5.12.3 When generated

This primitive is generated by the PMD when the High Rate PHY is in the receive state and Barker code lock is achieved. It is continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

18.4.5.12.4 Effect of receipt

This parameter is provided to the PLCP layer for information only. The SQ may be used in conjunction with RSSI as part of a CCA scheme.

18.4.5.13 PMD_CS.indicate

This primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has acquired (locked) the Barker code and data is being demodulated.

18.4.5.13.1 Function

This primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has acquired (locked) the Barker code and data is being demodulated.

18.4.5.13.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
PMD_CS	PMD_CS.indicate	'0' for DISABLED '1' for ENABLED	The PMD_CS (carrier sense) primitive in conjunction with PMD_ED provide CCA status through the PLCP layer PHYCCA primitive. PMD_CS indicates a binary status of ENABLED or DISABLED. PMD_CS is ENABLED when the correlator SQ indicated in PMD_SQ is greater than the correlation threshold. PMD_CS is DISABLED when the PMD_SQ falls below the correlation threshold.

18.4.5.13.3 When generated

This primitive is generated by the PMD sublayer when the High Rate PHY is receiving a PPDU and the PN code has been acquired.

18.4.5.13.4 Effect of receipt

This indicator is provided to the PLCP for forwarding to the MAC entity for information purposes through the PHYCCA indicator. This parameter shall indicate that the RF medium is busy and occupied by a High Rate PHY signal. The High Rate PHY should not be placed into the transmit state when PMD_CS is ENABLED.

18.4.5.14 PMD_ED.indicate

18.4.5.14.1 Function

This optional primitive may be generated by the PMD to provide an indication that the receiver has detected RF energy indicated by the PMD_RSSI primitive that is above a predefined threshold.

18.4.5.14.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
PMD_ED	PMD_ED.indicate	‘0’ for DISABLED ‘1’ for ENABLED	The PMD_ED (energy detect) primitive, along with the PMD_SQ, provides CCA status at the PLCP layer through the PHYCCA primitive. PMD_ED indicates a binary status of ENABLED or DISABLED. PMD_ED is ENABLED when the RSSI indicated in PMD_RSSI is greater than the ED_THRESHOLD parameter. PMD_ED

18.4.5.14.3 When generated

This primitive is generated by the PHY sublayer when the PHY is receiving RF energy from any source that exceeds the ED_THRESHOLD parameter.

18.4.5.14.4 Effect of receipt

This indicator is provided to the PLCP for forwarding to the MAC entity for information purposes through the PMD_ED indicator. This parameter shall indicate that the RF medium may be busy with an RF energy source that is not High Rate PHY compliant. If a High Rate PHY source is being received, the PMD_CS function is enabled shortly after the PMD_ED function is enabled.

18.4.5.15 PMD_ED.request

18.4.5.15.1 Function

This optional primitive may be generated by the PLCP to set a set a value for the energy detect ED THRESHOLD.

18.4.5.15.2 Semantics of the service primitive

The primitive provides the following parameters:

Parameter	Associated primitive	Value	Description
PMD_ED	PMD_ED.request	ED_THRESHOLD	ED_THRESHOLD is the threshold that the RSSI indicated is greater than in order for PMD_ED to be enabled. PMD_ED is DISABLED when the PMD_RSSI falls below the energy detect threshold.

18.4.5.15.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY energy detect threshold.

18.4.5.15.4 Effect of receipt

The receipt of PMD_ED immediately changes the energy detection threshold as set by the ED_THRESHOLD parameter.

18.4.6 PMD operating specifications, general

The following subclauses provide general specifications for the High Rate PMD sublayer. These specifications apply to both the Receive and the Transmit functions and general operation of a High Rate PHY.

Wireless LANS implemented in accordance with this standard are subject to equipment certification and operation requirements established by regional and national regulatory administrations. The PMD specification establishes minimum technical requirements for interoperability, based upon established regulations at the time this standard was issued. These regulations are subject to revision, or may be superceeded. Requirements that are subject to local geographic regulations are annotated within the PMD specification. Regulatory requirements that do not effect interoperability are not addressed in this standard. Implementers are refered to the following regulatory sources for further information. Operation in countries within defined regulatory domains may be subject to additional regulations.

The documents listed in 14.6.2 and below specify the current regulatory requirements for various geographic areas at the time the standard was developed. They are provided for geographic information only, and are subject to change or revision at any time.

Table —Additional Regulatory requirement list

Geographic area	Approval standards	Documents	Approval authority
Japan	**Ministry of Public management, Home affairs, Post and Tele-communication (MPHPT)**	**MPHPT Ordinance for regulating Radio Equipment, Article 49-20**	**MPHPT**

18.4.6.1 Operating frequency range

The High Rate PHY shall operate in the frequency range of 2.4 GHz to 2.4835 GHz as allocated by regulatory bodies in the USA, Europe, and Japan or in the 2.471 GHz to 2.497 GHz frequency band as allocated by regulatory authority in Japan.

18.4.6.2 Number of operating channels

The channel center frequencies and CHNL_ID numbers shall be as shown in Table 9. The FCC (US), IC (Canada), MPHPT (Japan), and ETSI (Europe) specify operation from 2.4 GHz to 2.4835 GHz. For Japan, operation is aditionally specified as 2.471 GHz to 2.497 GHz. France allows operation from 2.4465 GHz to 2.4835 GHz, and Spain allows operation from 2.445 GHz to 2.475 GHz. For each supported regulatory domain, all channels in Table 9 marked with “X” shall be supported.

Table 9—High Rate PHY Frequency Channel Plan

		Regulatory domains						
CHNL_ID	**Frequency**	**X'10' FCC**	**X'20' IC**	**X'30' ETSI**	**X'31' Spain**	**X'32' France**	**X'40' Japan**	**X'41' Japan**
1	2412 MHz	X	X	X	—	—	—	X
2	2417 MHz	X	X	X	—	—	—	X
3	2422 MHz	X	X	X	—	—	—	X
4	2427 MHz	X	X	X	—	—	—	X
5	2432 MHz	X	X	X	—	—	—	X
6	2437 MHz	X	X	X	—	—	—	X
7	2442 MHz	X	X	X	—	—	—	X
8	2447 MHz	X	X	X	—	—	—	X
9	2452 MHz	X	X	X	—	—	—	X

Table 9—High Rate PHY Frequency Channel Plan

10	2457 MHz	X	X	X	X	X	—	X
11	2462 MHz	X	X	X	X	X	—	X
12	2467 MHz	—	—	X	—	X	—	X
13	2472 MHz	—	—	X	—	X	—	X
14	2484 MHz	—	—	—	—	—	X	

In a multiple cell network topology, overlapping and/or adjacent cells using different channels can operate simultaneously without interference if the distance between the center frequencies is at least 25 MHz. Channel 14 shall be designated specifically for operation in Japan.

18.4.6.3 Modulation and channel data rates

Four modulation formats and data rates are specified for the High Rate PHY. The basic access rate shall be based on 1 Mbit/s DBPSK modulation. The enhanced access rate shall be based on 2 Mbit/s DQPSK. The extended Direct Sequence specification defines two additional data rates. The high rate access rates shall be based on the Complementary Code Keying (CCK) modulation scheme for 5.5 Mbit/s and 11 Mbit/s. An optional Packet Binary Convolutional Coding (PBCC) mode is also provided for potentially enhanced performance.

18.4.6.4 Spreading sequence and modulation for 1 and 2 Mbit/s

The following 11-chip Barker sequence shall be used as the PN code sequence for the 1 and 2 Mbit/s modulation:

+1, –1, +1, +1, –1, +1, +1, +1, –1, –1, –1

The leftmost chip shall be output first in time. The first chip shall be aligned at the start of a transmitted symbol. The symbol duration shall be exactly 11 chips long.

The DBPSK encoder for the basic access rate is specified in Table 10. The DQPSK encoder is specified in Table 11. (In the tables, $+j\omega$ shall be defined as counterclockwise rotation.)

Table 10—1 Mbit/s DBPSK Encoding Table

Bit input	Phase change ($+j\omega$)
0	0
1	π

Table 11—2 Mbit/s DQPSK Encoding Table

Dibit pattern (d0,d1) d0 is first in time	Phase change (+jω)
00	0
01	π/2
11	π
10	3π/2 (–π/2)

18.4.6.5 Spreading Sequences and modulation for CCK modulation at 5.5 and 11 Mbit/s

For the CCK modulation modes, the spreading code length is 8 and is based on complementary codes. The chipping rate is 11 Mchip/s. The symbol duration shall be exactly 8 complex chips long.

The following formula shall be used to derive the CCK code words that shall be used for spreading both 5.5 and 11 Mbit/s:

$$c=\{e^{j(\varphi_1+\varphi_2+\varphi_3+\varphi_4)}, e^{j(\varphi_1+\varphi_3+\varphi_4)}, e^{j(\varphi_1+\varphi_2+\varphi_4)}, -e^{j(\varphi_1+\varphi_4)}, e^{j(\varphi_1+\varphi_2+\varphi_3)}, e^{j(\varphi_1+\varphi_3)}, -e^{j(\varphi_1+\varphi_2)}, e^{j\varphi_1}\}$$

, where C is the code word

C ={ c0 .. c7}

The terms: φ1, φ2, φ3, and φ4 are defined in subclause 18.4.6.5.2 for 5.5 Mbit/s and subclause 18.4.6.5.3 for 11Mbit/s.

This formula creates 8 complex chips (c0 to c7) where c0 is transmitted first in time.

This is a form of the generalized Hadamard transform encoding where φ1 is added to all code chips, φ2 is added to all odd code chips, φ3 is added to all odd pairs of code chips and φ4 is added to all odd quads of code chips.

The phases φ1 modify the phase of all code chips of the sequence and shall be DQPSK encoded for 5.5 and 11 Mbit/s. This shall take the form of rotating the whole symbol by the appropriate amount relative to the phase of the preceding symbol. Note that the chip c7 of the symbol defined above is the chip that indicates the symbol’s phase and it is transmitted last.

18.4.6.5.1 Codes for CCK

The 4th and 7th chips are rotated 180 degrees (as shown) which helps to optimize the sequence correlation properties and minimize DC offsets in the codes. This can be seen by the minus sign on the 4th and 7th terms in the equation in subclause 18.4.6.5.

18.4.6.5.2 CCK 5.5 Mbit/s Modulation

At 5.5 Mbit/s 4 bits (d0 to d3; d0 first in time) are transmitted per symbol.

The data bits d0 and d1 encode $\varphi 1$ based on DQPSK. The DQPSK encoder is specified in Table 12. (In the tables, +jω shall be defined as counterclockwise rotation.). The phase change for $\varphi 1$ is relative to the phase $\varphi 1$ of the preceding symbol. For the header to PSDU transition, the phase change for $\varphi 1$ is relative to the phase of the preceding DQPSK (2 Mbit/s) symbol. That is, the phase of the last symbol of the CRC-16 is the reference phase for the first symbol generated from the PSDU octets. See the definition in subclause 18.4.6.4 for the reference phase of this Barker coded symbol. A “+1” chip in the Barker code shall represent the same carrier phase as a “+1” chip in the CCK code.

All odd numbered symbols generated from the PSDU octets shall be given an extra 180 degree (π) rotation in addition to the standard DQPSK modulation as shown in Table 12. The symbols of the PSDU shall be numbered starting with “0” for the first symbol for the purposes of determining odd and even symbols. That is, the PSDU transmission starts on an even numbered symbol.

Table 12. DQPSK Encoding Table

Dibit pattern (d(0),d(1)) **d(0) is first in time**	**Even Symbols** **Phase Change (+jω)**	**Odd Symbols** **Phase Change (+jω)**
00	0	π
01	$\pi/2$	$3\pi/2$ ($-\pi/2$)
11	π	0
10	$3\pi/2$ ($-\pi/2$)	$\pi/2$

The data dibits d2, and d3 CCK encode the basic symbol as specified in Table 13. This table is derived from the formula above by setting $\varphi 2 = (d2*\pi) + \pi/2$, $\varphi 3 = 0$, and $\varphi 4 = d3*\pi$. In the table d2 and d3 are in the order shown and the complex chips are shown c0 to c7 (left to right) with c0 transmitted first in time.

Table 13. 5.5 Mbit/s CCK Encoding Table

d2, d3	c1	c2	c3	c4	c5	c6	c7	c8
00	1*j*	1	1*j*	-1	1*j*	1	-1*j*	1
01	-1*j*	-1	-1*j*	1	1*j*	1	-1*j*	1
10	-1*j*	1	-1*j*	-1	-1*j*	1	1*j*	1
11	1*j*	-1	1*j*	1	-1*j*	1	1*j*	1

18.4.6.5.3 CCK 11 Mbit/s modulation.

At 11 Mbit/s, 8 bits (d0 to d7; d0 first in time) are transmitted per symbol.

The first dibit (d0,d1) encodes φ1 based on DQPSK. The DQPSK encoder is specified in Table 12 above. The phase change for φ1 is relative to the phase φ1 of the preceding symbol. In the case of header to PSDU transition, the phase change for φ1 is relative to the phase of the preceding DQPSK symbol. All odd numbered symbols of the PSDU are given an extra 180 degree (π) rotation in accordance with the DQPSK modulation as shown in Table 12. Symbol numbering starts with "0" for the first symbol of the PSDU.

The data dibits: (d2,d3), (d4,d5), (d6,d7) encode φ2, φ3, and φ4 respectively based on QPSK as specified in Table 14. Note that this table is binary, not Grey, coded.

Table 14. QPSK Encoding Table

Dibit pattern (d(i),d(i+1)) d(i) is first in time	Phase
00	0
01	π/2
10	π
11	3π/2 (-π/2)

18.4.6.6 DSSS/PBCC Data Modulation and Modulation Rate (Optional)

This optional coding scheme uses a binary convolutional coding with a 64-state binary convolutional code (BCC) and a cover sequence. The output of the BCC is encoded jointly onto the I and Q channels, as further documented below.

The encoder for this scheme is shown in Figure 12. Incoming data is first encoded with a binary convolutional code. A cover code is applied to the encoded data prior to transmission through the channel.

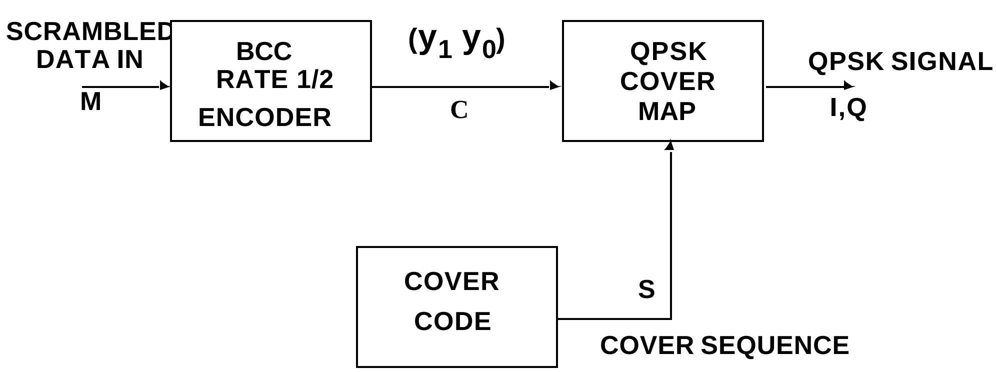


Figure 12. PBCC Modulator Scheme

The binary convolutional code that is used is a 64-state, rate ½ code. The generator matrix for the code is given as:

$$G = [D^6 + D^4 + D^3 + D + 1, \quad D^6 + D^5 + D^4 + D^3 + D^2 + 1]$$

or in octal notation, it is given by

$$G = [133, \quad 175]$$

Since the system is frame (PPDU) based, the encoder shall be in state zero, i.e. all memory elements contain zero, at the beginning of each PPDU. The encoder must also be placed in a known state at the end of each PPDU to prevent the data bits near the end of the PPDU from being substantially less reliable than those early on in the PPDU. To place the encoder in a known state at the end of a PPDU, at least six deterministic bits must be input immediately following the last data bit input the convolutional encoder. This is achieved by appending one octet containing all zeros to the end of the PPDU prior to transmission and discarding the final octet of each received PPDU. In this manner, the decoding process can be completed on the last data bits reliably.

An encoder block diagram is shown in Figure 13. It consists of six memory elements. For every data bit input, two output bits are generated.

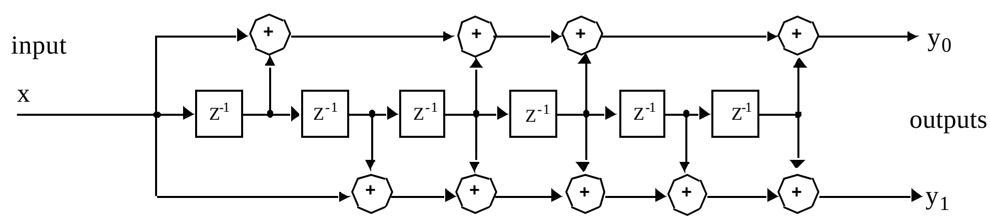


Figure 13. PBCC Convolutional Encoder

The output of the binary convolutional code described in above is mapped to a constellation using one of two possible rates. The 5.5 Mbps rate uses BPSK and the 11 Mbps rate uses QPSK. In QPSK mode each pair of output bits from the binary convolutional code is used to produce one symbol, while in BPSK mode each pair of bits from the BCC is taken serially (y0 first) and used to produce two BPSK symbols. This yields a throughput of one bit per symbol in QPSK mode and one-half a bit per symbol in BPSK mode.

The phase of the first complex chip of the PSDU shall be defined with respect to the phase of the last chip of the PCLP header, i.e. the last chip of the CRC check. The bits (y_1 y_0) = (0,0) shall indicate the same phase as the last chip of the CRC check. The other three combinations of (y_1 y_0) shall be defined with respect to this reference phase as shown in Figure 14.

The mapping from BCC outputs to PSK constellation points in BPSK and QPSK modes is determined by a pseudo-random cover sequence. This is shown for both modes in Figure 14. Note that this is an absolute phase table, not differential as in CCK.

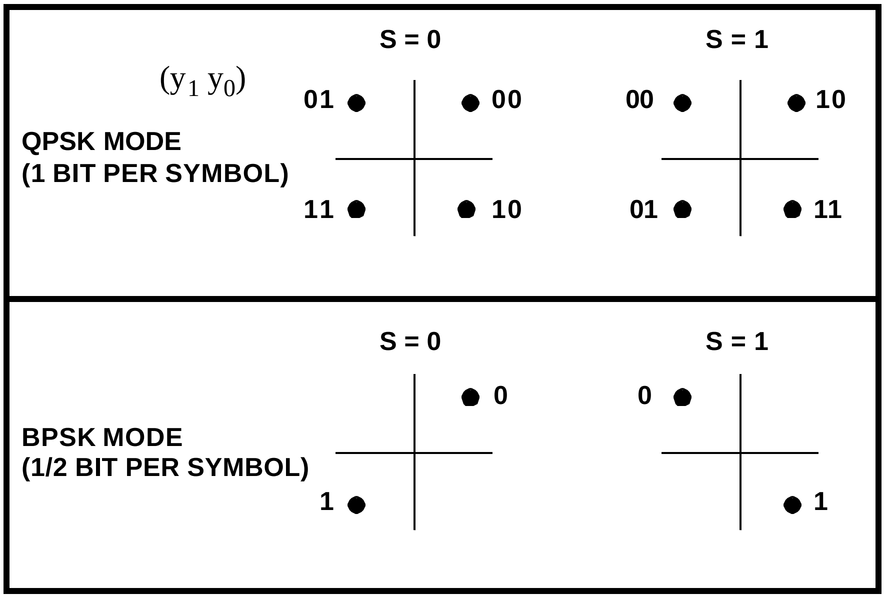


Figure 14. Cover Code Mapping

The pseudo-random cover sequence is generated from a seed sequence. The 16-bit seed sequence is 0011001110001011, where the first bit of the sequence in time is the left most bit. This sequence in octal notation is given as 150714, where the least significant bit is the first in time. This seed sequence is used to generate the pseudo-random cover sequence of length 256 bits that is used in the mapping of the current PSK symbol. It is the current binary value of this sequence at every given point in time that is taken as s in Figure 14.

This sequence of 256 bits is produced by taking the first sixteen bits of the sequence as the seed sequence, the second sixteen bits as the seed sequence cyclically left rotated by three, the third sixteen bits as the seed sequence cyclically left rotated by six, etc. If ci is the ith bit of the seed sequence, where 0 <= I <= 15, then the sequence that is used to cover the data is given row-wise as follows:

c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15
c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2
c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5
c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8
c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11
c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14
c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1
c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4
c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7
c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10
c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0
c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3
c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6
c10 c11 c12c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9
c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12

For PPDUs with more than 256 data bits this sequence of 256 bits is simply repeated.

18.4.6.7 Channel Agility (Optional)

This channel agility option allows an implementation to overcome some inherent difficulty with static channel assignements (a tone jammer), without burdening all implementations with the added cost of this capability. When the channel agility option is enabled the PHY shall meet the requirements on channel switching and settling time as described in subclause 18.4.6.12. and the hop sequences described below. This option can also be used to implement 802.11 compliant systems that are interoperable between both FH and DS modulations. Annex F contains a description of the expected behavior when such networks are employed.

18.4.6.7.1 Hop Sequences

The hop sequences for each of the specified geographical areas are defined with two sets. High Rate frequency channels referred to in this subclause are defined in Table 9.

The first set (Figure 15 and Figure 17) uses non-overlapping frequency channels to allow the High Rate systems to minimize interference degradation. The synchronization of frequency hopping is performed by the MAC sub-layer management entity as defined in the IEEE 802.11 Standard, 1999, subclause 11.1.5 for the FH PHY. The PLME SAP service primitives to command a new frequency channel is as defined in the IEEE 802.11 Standard, 1999, subclause 10.4.

The second set (Figure 16 and Figure 18) uses half overlapping frequency channels with 10 MHz center frequency spacing to enable interoperability with 1 and 2 Mbit/s FH systems hopping with the approved 802.11 hop sequences. The High Rate hop frequency is calculated from the specific 1 MHz channel chosen for a given hop by picking the closest High Rate channel within the set. Where there is a choice of two DSSS channels, the lower one shall be the one chosen. Therefore, the chosen channel shall be no more than +/-5 MHz of the channel center of the FH channel. When operating on the FH channels beyond +/-5 MHz of the closest High Rate channel specified in the set, the High Rate mode shall not be used and all FH transmissions shall occur at the 1 or 2 Mbit/s rates.

18.4.6.7.2 Operating channels

The operating channels for specified geographical areas are defined in Table 15 and Table 16.

Table 15. North American Operating Channels

Set	Number of Channels	HR/DSSS Channel Numbers
1	3	1, 6, 11
2	6	1, 3, 5, 7, 9, 11

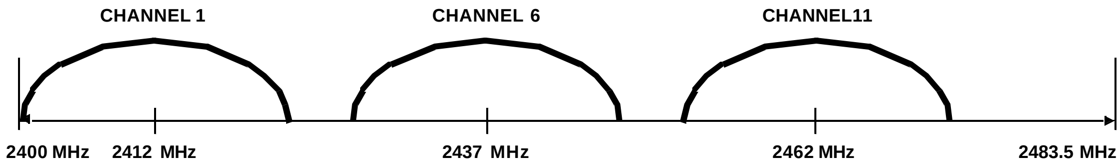


Figure 15. North American Channel Selection - Non overlapping

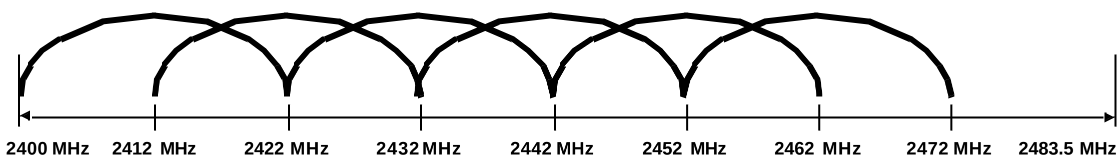


Figure 16. North American Channel Selection - Overlapping

Table 16. Europe Operating Channels (except France and Spain)

Set	Number of Channels	HR/DSS Channel Numbers
1	3	1, 7, 13
2	7	1, 3, 5, 7, 9, 11, 13

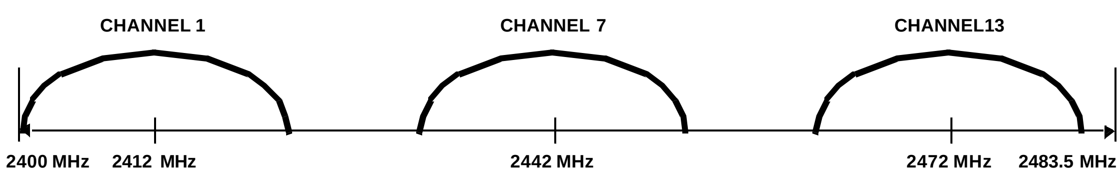


Figure 17. European Channel Selection - Non overlapping

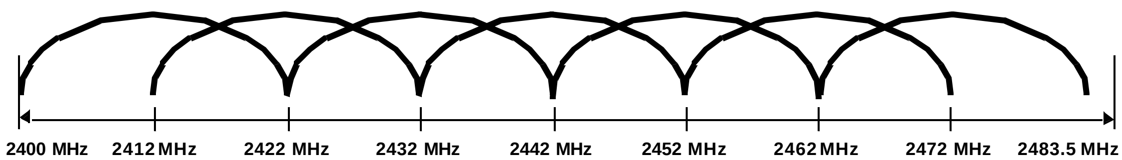


Figure 18. European Channel Selection - Overlapping

18.4.6.7.3 Hop Patterns

A frequency hopping pattern, Fx, consists of a permutation of all frequency channels defined in Table 15, and Table 16. For a given pattern number, x, the hopping sequence can be written as:

Fx = {fx (1), fx (2),...fx (p)}

where,

fx (i) = channel number (as defined in subclause14.6.4) for ith frequency in xth hopping pattern

p = number of hops in pseudo-random hopping pattern before repeating sequence (79 for North America and most of Europe)

The frequency hopping patterns for Set 1 of each geographic area is based on the Hop Patterns in Table 17 and Table 18.

The frequency hopping patterns for Set 2 of each geographic area is defined by the 1/2 Mbit/s FH PHY hop sequences as described in the FH PHY subclause 14.6.8. Given the hopping pattern number, x, and the index for the next frequency, i (in the range 1 to p), the DS channel number as defined in subclause 18.4.6.2 shall be selected with the following algorithm:

North America:

f'x (i) = f''x (i) for 1 <= f''x (i) <= 11;

f'x (i) = null for f''x (i) < 1 and f''x (i) > 11;

f''x (i) = 2 * Int [({[b(i) + x] mod (79) +2} – 6) / 10] – 1

with b(i) defined in Table 42 of subclause 14.6.8,

Most of Europe:

f'x (i) = f''x (i) for 1 <= f''x (i) <= 13;

f'x (i) = null for f''x (i) < 1 and f''x (i) > 13;

f''x (i) = 2 * Int [({[b(i) + x] mod (79) +2} – 6) / 10] – 1

with b(i) defined in Table 42 of subclause 14.6.8,

Table 17. North America Set 1 Hop Patterns

Index	Pattern 1	Pattern 2
1	1	1
2	6	11
3	11	6

Table 18. Europe Set 1 Hop Patterns (except France and Spain)

Index	Pattern 1	Pattern 2
1	1	1
2	7	13
3	13	7

18.4.6.8 Transmit and receive in-band and out-of-band spurious emissions

The High Rate PHY conforms with in-band and out-of-band spurious emissions as set by regulatory bodies. For the USA, refer to FCC 15.247, 15.205, and 15.209. For Europe, refer to ETS 300–328. For Japan refer to MPT ordinance for Regulating Radio Equipment, Article 7.

18.4.6.9 Transmit-to-receive turnaround time

The TX-to-RX turnaround time shall be less than 10 µs, including the power-down ramp specified in 18.4.7.6.

The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol to valid CCA detection of the incoming signal. The CCA should occur within 25 µs (10 µs for turnaround time plus 15 µs for energy detect) or by the next slot boundary occurring after the 25 µs has elapsed (refer to 18.4.8.4). A receiver input signal 3 dB above the ED threshold described in 18.4.8.4 shall be present at the receiver.

18.4.6.10 Receive-to-transmit turnaround time

The RX-to-TX turnaround time shall be measured at the MAC/PHY interface, using PHYTX-START.request and shall be 5 µs. This includes the transmit power up ramp described in 18.4.7.6.

18.4.6.11 Slot time

The slot time for the High Rate PHY shall be the sum of the RX-to-TX turnaround time (5 µs) and the energy detect time (15 µs specified in 18.4.8.4). The propagation delay shall be regarded as being included in the energy detect time.

18.4.6.12 Channel switching/settling time

When the channel agility option is enabled, the time to change from one operating channel frequency to another as specified in 18.4.6.2 is 224 µs. A conformant PMD meets this switching time specification when the operating channel center frequency has settled to within +/- 60 kHz of the nominal channel center. Stations shall not transmit until after the channel change settling time.

18.4.6.13 Transmit and receive antenna port impedance

The impedance of the transmit and receive antenna port(s) shall be 50 Ω if the port is exposed.

18.4.6.14 Transmit and receive operating temperature range

Three temperature ranges for full operation compliance to the High Rate PHY are specified in Clause 13. Type 1 shall be defined as 0°C to 40°C, and is designated for office environments. Type 2 shall be defined as –30°C to +70°C, and is designated for industrial environments.

18.4.7 PMD transmit specifications

The following subclauses describe the transmit functions and parameters associated with the PMD sublayer.

18.4.7.1 Transmit power levels

The maximum allowable output power as measured in accordance with practices specified by the regulatory bodies is shown in Table 19. In the USA, the radiated emissions should also conform with the ANSI uncontrolled radiation emission standards (IEEE Std C95.1-1991).

Table 19—Transmit Power Levels

Maximum output power	Geographic location	Compliance document
1000 mW	USA	FCC 15.247
100 mW (EIRP)	Europe	ETS 300–328
See Table 115a	Japan	MPT ordinance for Regulating Radio Equipment, Article 49-20

Table 115a, Transmit power levels in Japan

Maximum output power	Modulation/Frequency range	Compliance document
10 mW/MHz	**for FH-SS or DS-SS modulation and operation in 2.471 GHz - 2.497 GHz**	**MPHPT ordinance for Regulating Radio Equipment, article 49-20**
10 mW/MHz	**for DS-SS modulation and operation in 2.400GHz - 2.4835 GHz**	**MPHPT ordinance for Regulating Radio Equipment, article 49-20**
3 mW/MHz	**for FH-SS modulation and operation in 2.400 GHz - 2.4835 GHz**	**MPHPT ordinance for Regulating Radio Equipment, article 49-20**

18.4.7.2 Transmit power level control

Power control shall be provided for transmitted power greater than 100 mW. A maximum of four power levels may be provided. At a minimum, a radio capable of transmission greater than 100 mW shall be capable of switching power back to 100 mW or less.

18.4.7.3 Transmit spectrum mask

The transmitted spectral products shall be less than –30 dBr (dB relative to the SINx/x peak) for f_c – 22 MHz < f < f_c –11 MHz, f_c +11 MHz < f < f_c + 22 MHz, –50 dBr for f < f_c –22 MHz, and f > f_c + 22 MHz, where f_c is the channel center frequency. The transmit spectral mask is shown in Figure 19. The measurements shall be made using 100 kHz resolution bandwidth and a 100 kHz video bandwidth.

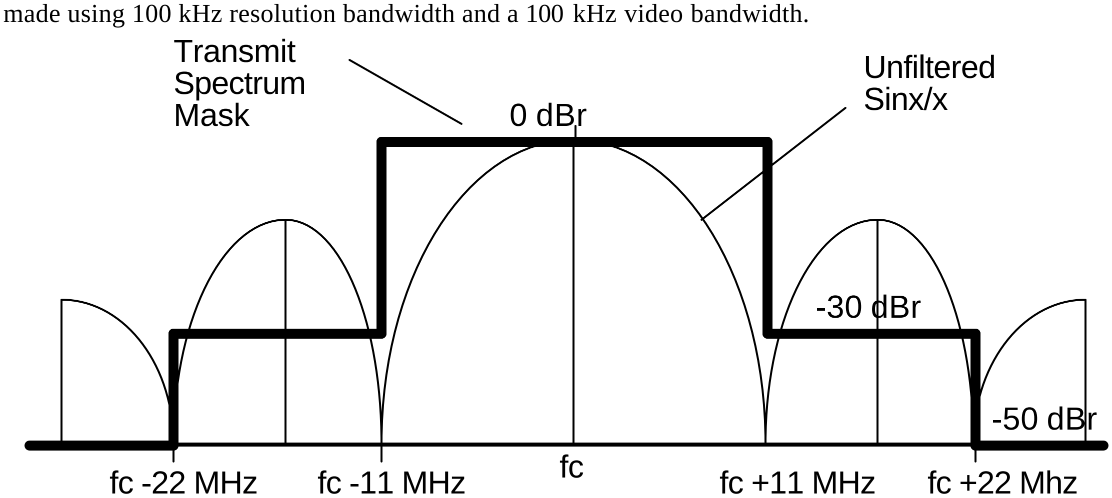


Figure 19—Transmit spectrum mask

18.4.7.4 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ±25 ppm maximum.

18.4.7.5 Chip clock frequency tolerance

The PN code chip clock frequency tolerance shall be better than ±25 ppm maximum. It is highly recommended that the chip clock and the transmit frequency be locked (coupled) for optimum demodulation performance. If these clocks are locked, it is recommended that bit 2 of the SERVICE field be set to a 1 as indicated in paragraph 18.2.3.4.

18.4.7.6 Transmit power-on and power-down ramp

The transmit power-on ramp for 10% to 90% of maximum power shall be no greater than 2 µs. The transmit power-on ramp is shown in Figure 20.

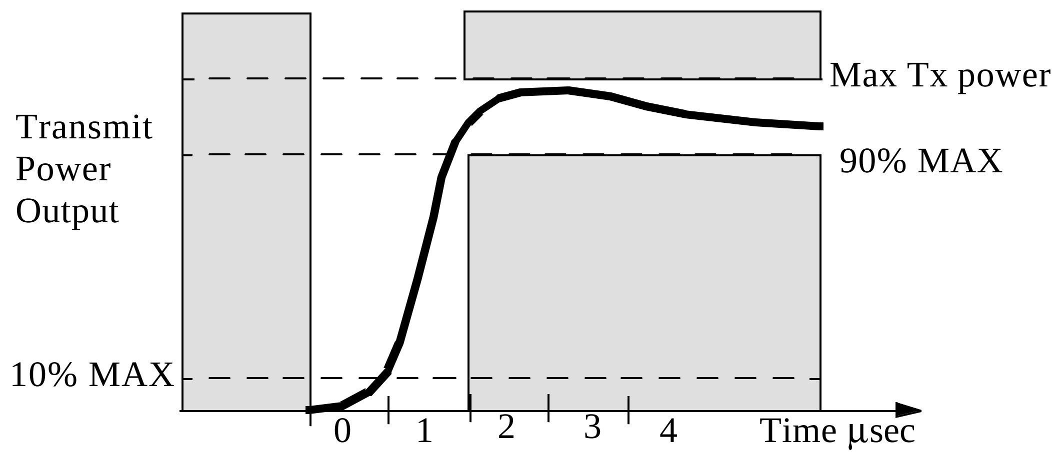


Figure 20—Transmit Power-On Ramp

The transmit power-down ramp for 90% to 10% maximum power shall be no greater than 2 µs. The transmit power down ramp is shown in Figure 21.

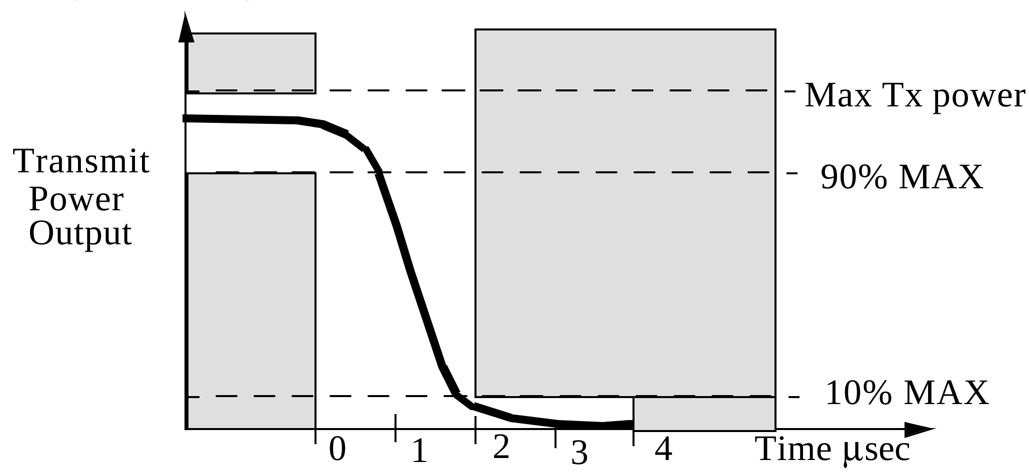


Figure 21—Transmit Power-Down Ramp

The transmit power ramps shall be constructed such that the High Rate PHY emissions conform with spurious frequency product specification defined in 18.4.6.8.

18.4.7.7 RF carrier suppression

The RF carrier suppression, measured at the channel center frequency, shall be at least 15 dB below the peak SIN(x)/x power spectrum. The RF carrier suppression shall be measured while transmitting a repetitive 01 data sequence with the scrambler disabled using DQPSK modulation. A 100 kHz resolution bandwidth shall be used to perform this measurement.

18.4.7.8 Transmit modulation accuracy

The transmit modulation accuracy requirement for the High Rate PHY shall be based on the difference between the actual transmitted waveform and the ideal signal waveform. Modulation accuracy shall be determined by measuring the peak vector error magnitude measured during each chip period. Worst-Case vector error magnitude shall not exceeded 0.35 for the normalized sampled chip data. The ideal complex I and Q constellation points associated with DQPSK modulation (0.707,0.707), (0.707, –0.707), (–0.707, 0.707), (–0.707, –0.707) shall be used as the reference. These measurements shall be from baseband I and Q sampled data after recovery through a reference receiver system.

Figure 22 illustrates the ideal DQPSK constellation points and range of worst-case error specified for modulation accuracy.

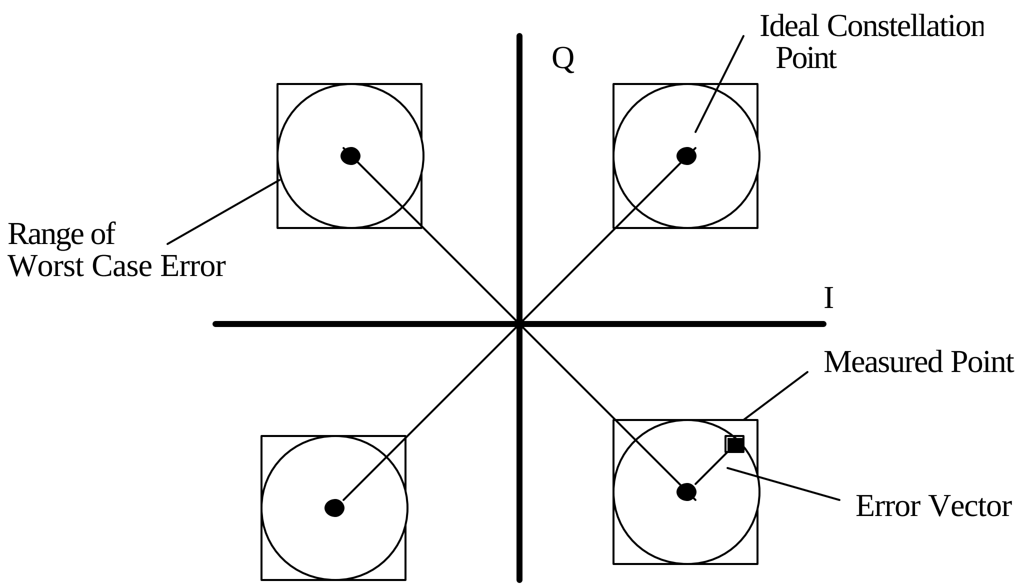


Figure 22—Modulation Accuracy Measurement Example

Error vector measurement requires a reference receiver capable of carrier lock. All measurements shall be made under carrier lock conditions. The distortion induced in the constellation by the reference receiver shall be calibrated and measured. The test data error vectors described below shall be corrected to compensate for the reference receiver distortion.

The IEEE 802.11 compatible radio shall provide an exposed TX chip clock, which shall be used to sample the I and Q outputs of the reference receiver.

The measurement shall be made under the conditions of continuous DQPSK transmission using scrambled all 1's.

The eye pattern of the I channel shall be used to determine the I and Q sampling point. The chip clock provided by the vendor radio shall be time delayed such that the samples fall at a 1/2 chip period offset from the mean of the zero crossing positions of the eye (see Figure 23). This is the ideal center of the eye and may not be the point of maximum eye opening.

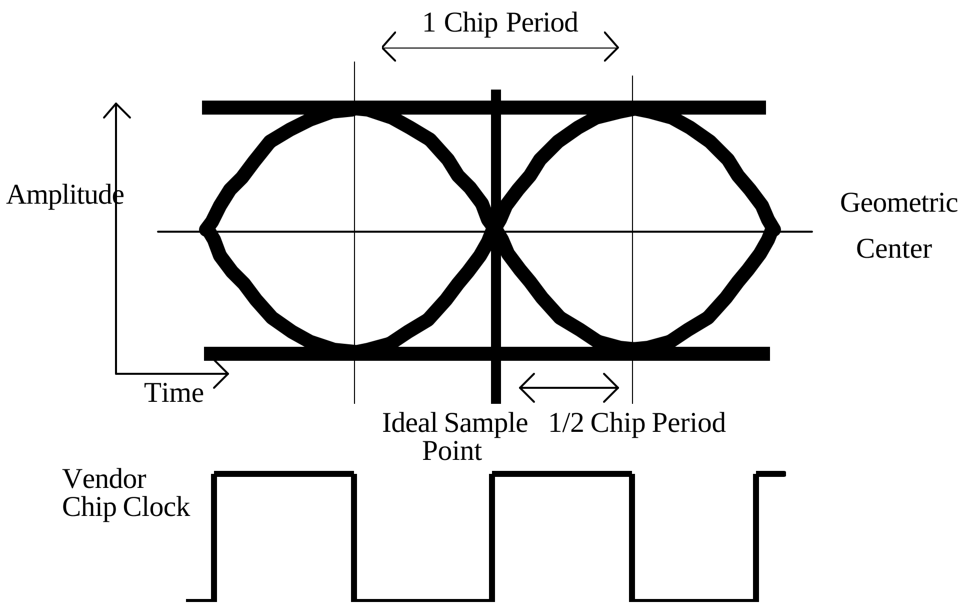


Figure 23—Chip Clock Alignment with Baseband Eye Pattern

Using the aligned chip clock, 1000 samples of the *I* and *Q* baseband outputs from the reference receiver are captured. The vector error magnitudes shall be calculated as follows:

Calculate the dc offsets for *I* and *Q* samples.

$$I_{mean} = \sum_{n=0}^{999} I(n)/1000$$

$$Q_{mean} = \sum_{n=0}^{999} Q(n)/1000$$

Calculate the dc corrected *I* and *Q* samples for all *n* =1000 sample pairs.

$$I_{dc}(n) = I(n) - I_{mean}$$

$$Q_{dc}(n) = Q(n) - Q_{mean}$$

Calculate the average magnitude of *I* and *Q* samples.

$$I_{\mathrm{mag}} = \sum_{n=0}^{999} |I_{\mathrm{dc}}(n)| / 1000$$

$$Q_{\mathrm{mag}} = \sum_{n=0}^{999} |Q_{\mathrm{dc}}(n)| / 1000$$

Calculate the normalized error vector magnitude for the $I_{\mathrm{dc}}(n)/Q_{\mathrm{dc}}(n)$ pairs.

$$V_{\mathrm{err}}(n) = \left[\frac{1}{2} \times (\{|I_{\mathrm{dc}}(n)| - I_{\mathrm{mag}}\}^2 + \{|Q_{\mathrm{dc}}(n)| - Q_{\mathrm{mag}}\}^2)\right]^{\frac{1}{2}} - V_{\mathrm{correction}}$$

with $V_{\mathrm{correction}}$ = error induced by the reference receiver system.

A vendor High Rate PHY implementation shall be compliant if for all n =1000 samples the following condition is met:

$V_{\mathrm{err}}(n) < 0.35$

18.4.8 PMD receiver specifications

The following subclauses describe the receive functions and parameters associated with the PMD sublayer.

18.4.8.1 Receiver minimum input level sensitivity

The frame error ratio (FER) shall be less than 8×10^{-2} at an PSDU length of 1024 octets for an input level of –76 dBm measured at the antenna connector. This FER shall be specified for 11 Mbit/s CCK modulation. The test for the minimum input level sensitivity shall be conducted with the energy detection threshold set less than or equal to –76 dBm.

18.4.8.2 Receiver maximum input level

The receiver shall provide a maximum FER of 8×10^{-2} at an PSDU length of 1024 octets for a maximum input level of –10 dBm measured at the antenna. This FER shall be specified for 11 Mbit/s CCK modulation.

18.4.8.3 Receiver adjacent channel rejection

Adjacent channel rejection is defined between any two channels with ≥25 MHz separation in each channel group defined in 18.4.6.2.

The adjacent channel rejection shall be equal to or better than 35 dB with an FER of 8×10^{-2} using 11 Mbit/s CCK modulation described in 18.4.6.3 and an PSDU length of 1024 octets.

The adjacent channel rejection shall be measured using the following method:

Input a 11 Mbit/s CCK modulated signal at a level 6 dB greater than specified in 18.4.8.1. In an adjacent channel (≥25 MHz separation as defined by the channel numbering), input a signal modulated in a similar fashion that adheres to the transmit mask specified in 18.4.7.3 to a level 41 dB above the level specified in

18.4.8.1. The adjacent channel signal shall be derived from a separate signal source. It cannot be a frequency shifted version of the reference channel. Under these conditions, the FER shall be no worse than 8×10^{-2}.

18.4.8.4 CCA

The High Rate PHY shall provide the capability to perform CCA according to at least one of the following three methods:

a) CCA Mode 1*:* Energy above threshold. CCA shall report a busy medium upon detecting any energy above the ED threshold.
b) CCA Mode 4: Carrier Sense with timer. CCA shall start a timer whose duration is 3.65 ms and report a busy medium only upon the detection of a High Rate PHY signal. CCA shall report an idle medium after the timer expires and no High Rate PHY signal is detected. The 3.65 ms timeout is the duration of the longest possible 5.5 Mbit/s PSDU.
c) CCA Mode 5: A combination of Carrier Sense and energy above threshold. CCA shall report busy at least while a High Rate PPDU with energy above the ED threshold is being received at the antenna.

The energy detection status shall be given by the PMD primitive, PMD_ED. The carrier sense status shall be given by PMD_CS. The status of PMD_ED and PMD_CS is used in the PLCP convergence procedure to indicate activity to the MAC through the PHY interface primitive PHY-CCA.indicate.

A busy channel shall be indicated by PHY-CCA.indicate of class BUSY.

Clear channel shall be indicated by PHY-CCA.indicate of class IDLE.

The PHY MIB attribute dot11CCAModeSupported shall indicate the appropriate operation modes. The PHY shall be configured through the PHY MIB attribute dot11CurrentCCAMode.

The CCA shall be TRUE if there is no energy detect or carrier sense. The CCA parameters are subject to the following criteria:

a) If a valid High Rate signal is detected during its preamble within the CCA assessment window, the energy detection threshold shall be less than or equal to –76 dBm for TX power > 100 mW, –73 dBm for 50 mW < TX power $\leq$ 100 mW, and –70 dBm for TX power $\leq$ 50 mW.
b) With a valid signal (according to the CCA mode of operation) present at the receiver antenna within 5 μs of the start of a MAC slot boundary, the CCA indicator shall report channel busy before the end of the slot time. This implies that the CCA signal is available as an exposed test point. Refer to IEEE Std 802.11- 1999 Figure 47 for a slot time boundary definition.
c) In the event that a correct PLCP Header is received, the High Rate PHY shall hold the CCA signal inactive (channel busy) for the full duration as indicated by the PLCP LENGTH field. Should a loss of carrier sense occur in the middle of reception, the CCA shall indicate a busy medium for the intended duration of the transmitted PPDU. Upon reception of a correct PLCP Header, the timer of CCA Mode 2 shall be overridden by this requirement.

Conformance to High Rate PHY CCA shall be demonstrated by applying an equivalent High Rate compliant signal, above the appropriate ED threshold (a), such that all conditions described in b) and c) above are demonstrated.

Annex A Protocol Implementation Conformance Statement (PICS) Proforma (normative)

Add the following table to Annex A:

A.4.3 - IUT Configuration

Insert an entry CF7 to the table

Item	IUT Configuration	References	Status	Support
	What is the configuration of the IUT?			
* CF1	Access Point (AP)	5.2	O.1	Yes o No o
* CF2	Independent Station (NOT an AP)	5.2	O.1	Yes o No o
* CF3	Frequency Hopping Spread Spectrum PHY Layer for the 2.4GHz Band		O.2	Yes o No o
* CF4	Direct Sequence Spread Spectrum PHY Layer for the 2.4GHz Band		O.2	Yes o No o
* CF5	Infrared PHY Layer		O.2	Yes o No o
* CF7	High Speed PHY layer		O.2	Yes o No o

Insert a new section 4.9 for the optional HR/DSSS parameters.

A 4.9 - High Rate Direct Sequence Physical Layer Functions

Are the following PHY features supported?

Item	PHY Feature	References	Status	Support
HRDS1	Long Preamble and Header Procedures	18.2	M	Yes o No o
HRDS1.1	Long DS Preamble prepended on TX	18.2.1	M	Yes o No o
HRDS1.2	Long PLCP integrity check generation	18.2.3, 18.2.3.6	M	Yes o No o
HRDS1.3	TX Rate change capability	18.2.3.3	M	Yes o No o
HRDS1.4	Supported Data Rates	18.1, 18.2.3.3	M	Yes o No o
HRDS1.5	Data scrambler	18.2.4	M	Yes o No o
HRDS1.6	Scrambler initialization	18.2.4	M	Yes o No o
*HRDS2	Channel Agility Option	18.3.2	O	Yes o No o
*HRDS3	Short Preamble and Header Procedures	18.2	O	Yes o No o
HRDS3.1	Short Preamble prepended on TX	18.2.2	HRDS3: M	Yes o No o N/A o
HRDS3.2	Short Header Transmission	18.2.3.8, 18.2.3.9, 18.2.3.10, 18.2.3.11, 18.2.3.12 ,18.2.3.13, 18.2.3.14	HRDS3: M	Yes o No o N/A o
HRDS4	Long Preamble process on RX	18.2.6	M	Yes o No o
HRDS4.1	PLCP format	18.2.6	HRDS4: M	Yes o No o
HRDS4.2	PLCP integrity check verify	18.2.6	HRDS4: M	Yes o No o
HRDS4.3	RX Rate change capability	18.2.6	HRDS4: M	Yes o No o
HRDS4.4	Data whitener descrambler	18.2.6	HRDS4: M	Yes o No o
*HRDS5	Short Preamble process on RX	18.2.6	HRDS5: M	Yes o No o N/A o
HRDS5.1	PLCP format	18.2.6	HRDS5: M	Yes o No o N/A o

HRDS5.2	PLCP integrity check verify	18.2.6	HRDS5: M	Yes o No o N/A o
HRDS5.3	RX Rate change capability	18.2.6	HRDS5: M	Yes o No o N/A o
HRDS5.4	Data whitener descrambler	18.2.6	HRDS5: M	Yes o No o N/A o
*HRDS6	Operating channel capability			
* HRDS6.1	North America (FCC)	18.4.6.2	HRDS6: O.3	Yes o No o N/A o
HRDS6.1.1	channel 1	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
HRDS6.1.2	channel 2	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
HRDS6.1.3	channel 3	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
HRDS6.1.4	channel 4	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
HRDS6.1.5	channel 5	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
HRDS6.1.6	channel 6	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
HRDS6.1.7	channel 7	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
HRDS6.1.8	channel 8	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
HRDS6.1.9	channel 9	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
HRDS6.1.10	channel 10	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
HRDS6.1.11	channel 11	18.4.6.2	HRDS6.1 :M	Yes o No o N/A o
* HRDS6.2	Canada (IC)	18.4.6.2	HRDS6: O.3	Yes o No o N/A o
HRDS6.2.1	channel 1	18.4.6.2	HRDS6.2 :M	Yes o No o N/A o
HRDS6.2.2	channel 2	18.4.6.2	HRDS6.2 :M	Yes o No o N/A o
HRDS6.2.3	channel 3	18.4.6.2	HRDS6.2 :M	Yes o No o N/A o
HRDS6.2.4	channel 4	18.4.6.2	HRDS6.2 :M	Yes o No o N/A o
HRDS6.2.5	channel 5	18.4.6.2	HRDS6.2 :M	Yes o No o N/A o
HRDS6.2.6	channel 6	18.4.6.2	HRDS6.2 :M	Yes o No o N/A o
HRDS6.2.7	channel 7	18.4.6.2	HRDS6.2 :M	Yes o No o N/A o
HRDS6.2.8	channel 8	18.4.6.2	HRDS6.2 :M	Yes o No o N/A o
HRDS6.2.9	channel 9	18.4.6.2	HRDS6.2 :M	Yes o No o N/A o
HRDS6.2.10	channel 10	18.4.6.2	HRDS6.2 :M	Yes o No o N/A o

HRDS6.2.11	channel 11	18.4.6.2	HRDS6.2:M	Yes o No o N/A o
* HRDS6.3	Europe (ETSI)	18.4.6.2	HRDS6:O.3	Yes o No o N/A o
HRDS6.3.1	channel 1	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.2	channel 2	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.3	channel 3	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.4	channel 4	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.5	channel 5	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.6	channel 6	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.7	channel 7	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.8	channel 8	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.9	channel 9	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.10	channel 10	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.11	channel 11	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.12	channel 12	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
HRDS6.3.13	channel 13	18.4.6.2	HRDS6.3:M	Yes o No o N/A o
* HRDS6.4	France	18.4.6.2	HRDS6:O.3	Yes o No o N/A o
HRDS6.4.1	channel 10	18.4.6.2	HRDS6.4:M	Yes o No o N/A o
HRDS6.4.2	channel 11	18.4.6.2	HRDS6.4:M	Yes o No o N/A o
HRDS6.4.3	channel 12	18.4.6.2	HRDS6.4:M	Yes o No o N/A o
HRDS6.4.4	channel 13	18.4.6.2	HRDS6.4:M	Yes o No o N/A o
* HRDS6.5	Spain	18.4.6.2	HRDS6:O.3	Yes o No o N/A o
HRDS6.5.1	channel 10	18.4.6.2	HRDS6.5:M	Yes o No o N/A o
HRDS6.5.2	channel 11	18.4.6.2	HRDS6.5:M	Yes o No o N/A o
* HRDS6.6	Japan	18.4.6.2	HRDS6:O.3	Yes o No o N/A o
HRDS6.6.1	Channel 1	18.4.6.2	HRDS6.6:M	Yes o No o N/A o
HRDS6.6.2	Channel 2	18.4.6.2	HRDS6.6:M	Yes o No o N/A o
HRDS6.6.3	Channel 3	18.4.6.2	HRDS6.6:M	Yes o No o N/A o

HRDS6.6.4	Channel 4	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS6.6.5	Channel 5	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS6.6.6	Channel 6	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS6.6.7	Channel 7	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS6.6.8	Channel 8	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS6.6.9	Channel 9	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS6.6.10	Channel 10	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS6.6.11	Channel 11	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS6.6.12	Channel 12	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS6.6.13	Channel 13	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS6.6.14	Channel 14	18.4.6.2	HRDS6.6: M	Yes o No o N/A o
HRDS8	Hop Sequences		HRDS2: M	Yes o No o N/A o
HRDS8	CCK Bits to Symbol Mapping			
HRDS8.1	5.5 Mbit/s	18.4.6.5	HRDS8: M	Yes o No o
HRDS8.2	11 Mbit/s	18.4.6.5	HRDS8: M	Yes o No o
*HRDS9	PBCC Bits to Symbol Mappings	18.4.6.6	O	
HRDS9.1	5.5 Mbit/s	18.4.6.6	HRDS9: M	Yes o No o
HRDS9.2	11 Mbit/s	18.4.6.6	HRDS9: M	Yes o No o
*HRDS10	CCA functionality	18.4.8.4		
HRDS10.1	CCA mode 1, Energy Only (RSSI above threshold)	18.4.8.4	HRDS10: O.4	Yes o No o
HRDS10.2	CCA mode 4, Carrier sense with timer	18.4.8.4	HRDS10: O.4	Yes o No o
HRDS10.3	CCA mode 5, Energy detect with high rate CS	18.4.8.4	HRDS10: O.4	Yes o No o
HRDS10.4	Hold CCA busy for packet duration of a correctly received PLCP but carrier lost during reception of MPDU	18.2.6	M	Yes o No o
HRDS10.5	Hold CCA busy for packet duration of a correctly received but out of spec PLCP	18.2.6	M	Yes o No o
HRDS11	Transmit antenna selection	18.4.5.8,	O	Yes o No o
HRDS12	Receive antenna diversity	18.4.5.8, 18.4.5.9	O	Yes o No o
*HRDS13	antenna port(s) availability	18.4.6.8	O	Yes o No o
HRDS13.1	if available (50 ohm impedance)	18.4.6.8	HRDS13: M	Yes o No o N/A o
*HRDS14	transmit power level support	18.4.5.9, 18.4.7.2	O	Yes o No o
HRDS14.1	if greater than 100mW capability	18.4.7.2	HRDS14: M	Yes o No o N/A o

*HRDS15	radio type (temperature range)	18.4.6.14		
HRDS15.1 HRDS15.2	Type 1 Type 2	18.4.6.14 18.4.6.14	HRDS15: O.5 HRDS15: O.5	Yes o No o N/A o Yes o No o N/A o
HRDS16	Spurious Emissions conformance	18.4.6.8	M	Yes o No o
HRDS17	TX - RX turnaround time	18.4.6.9	M	Yes o No o
HRDS18	RX - TX turnaround time	18.4.6.10	M	Yes o No o
HRDS19	Slot Time	18.4.6.11	M	Yes o No o
HRDS20	ED reporting time	18.4.6.10, 18.4.8.4	M	Yes o No o
HRDS21	minimum transmit power level	18.4.7.2	M	Yes o No o
HRDS22	transmit spectral mask conformance	18.4.7.3	M	Yes o No o
HRDS23	transmitted center frequency tolerance	18.4.7.4	M	Yes o No o
HRDS24	chip clock frequency tolerance	18.4.7.5	M	Yes o No o
HRDS25	transmit power on ramp	18.4.7.6	M	Yes o No o
HRDS26	transmit power down ramp	18.4.7.6	M	Yes o No o
HRDS27	RF carrier suppression	18.4.7.7	M	Yes o No o
HRDS28	transmit modulation accuracy	18.4.7.8	M	Yes o No o
HRDS29	receiver minimum input level sensitivity	18.4.8.1	M	Yes o No o
HRDS30	receiver maximum input level	18.4.8.2	M	Yes o No o
HRDS31	receiver adjacent channel rejection	18.4.8.3	M	Yes o No o
HRDS32 HRDS32.1	Management Information Base PHY Object Class	13.1, 18.3.2, Annex C 13.1, 18.3.3,	M M	Yes o No o Yes o No o

Annex C: Formal Description of MAC Operation (Normative)

For the HR/DSSS PHY, replace the use of aMPDUDurationFactor, aPreambleLength, and aPLCPHeaderLength with the use of PLME-TXTIME.request and PLME-TXTIME.confirm primitives in the formal description, updating the following diagrams:

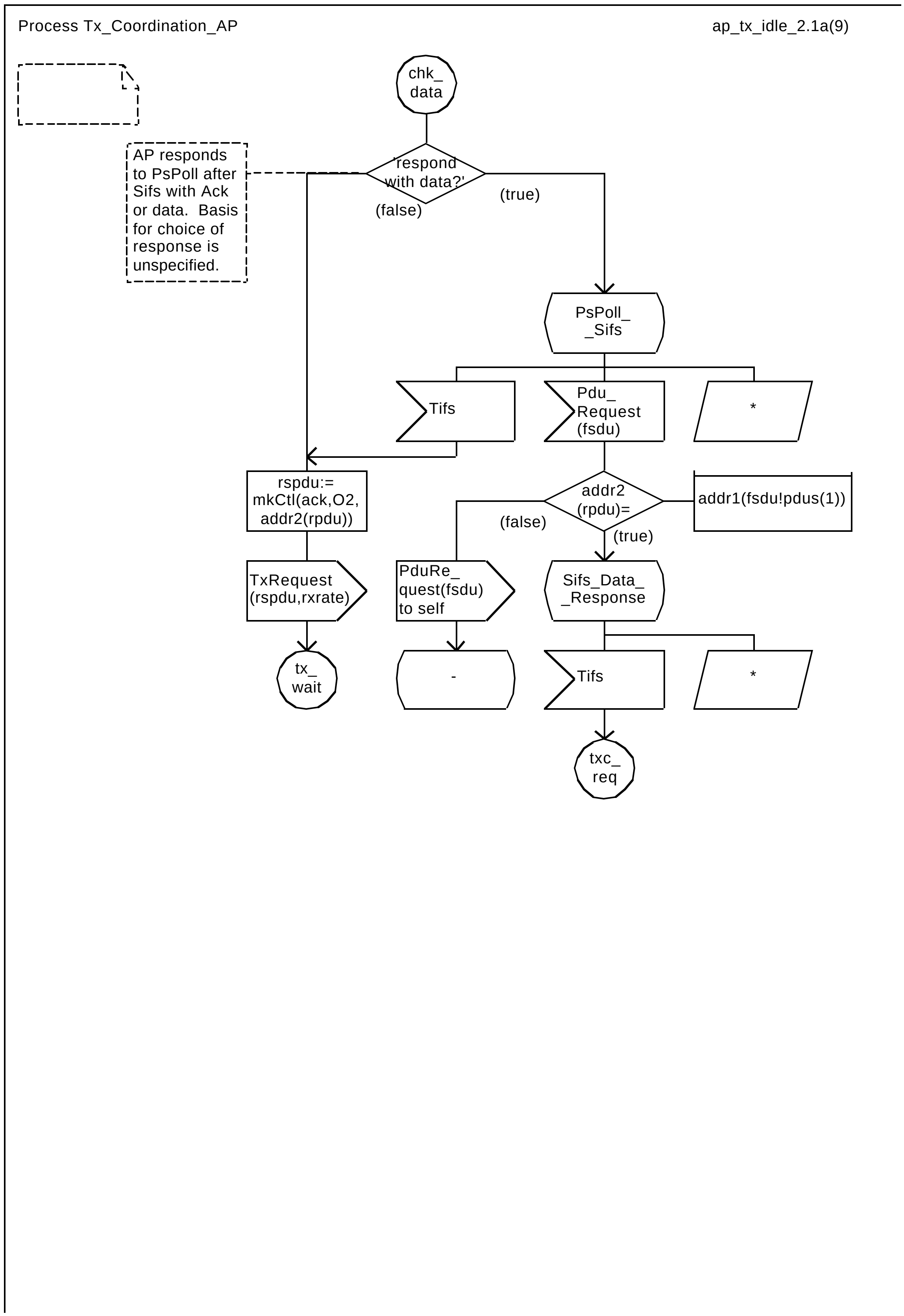
Package macsorts

3127_e\PHY_Params(31)

```
/*******************************************************************
 *       Generic PHY parameter set sort
 *******************************************************************/
/* Generic PHY parameter element for signals related to Beacons */
/* and Probe Responses that are PHY-type independent. */
syntype PhyParms = Octetstring   endsyntype PhyParms;
```

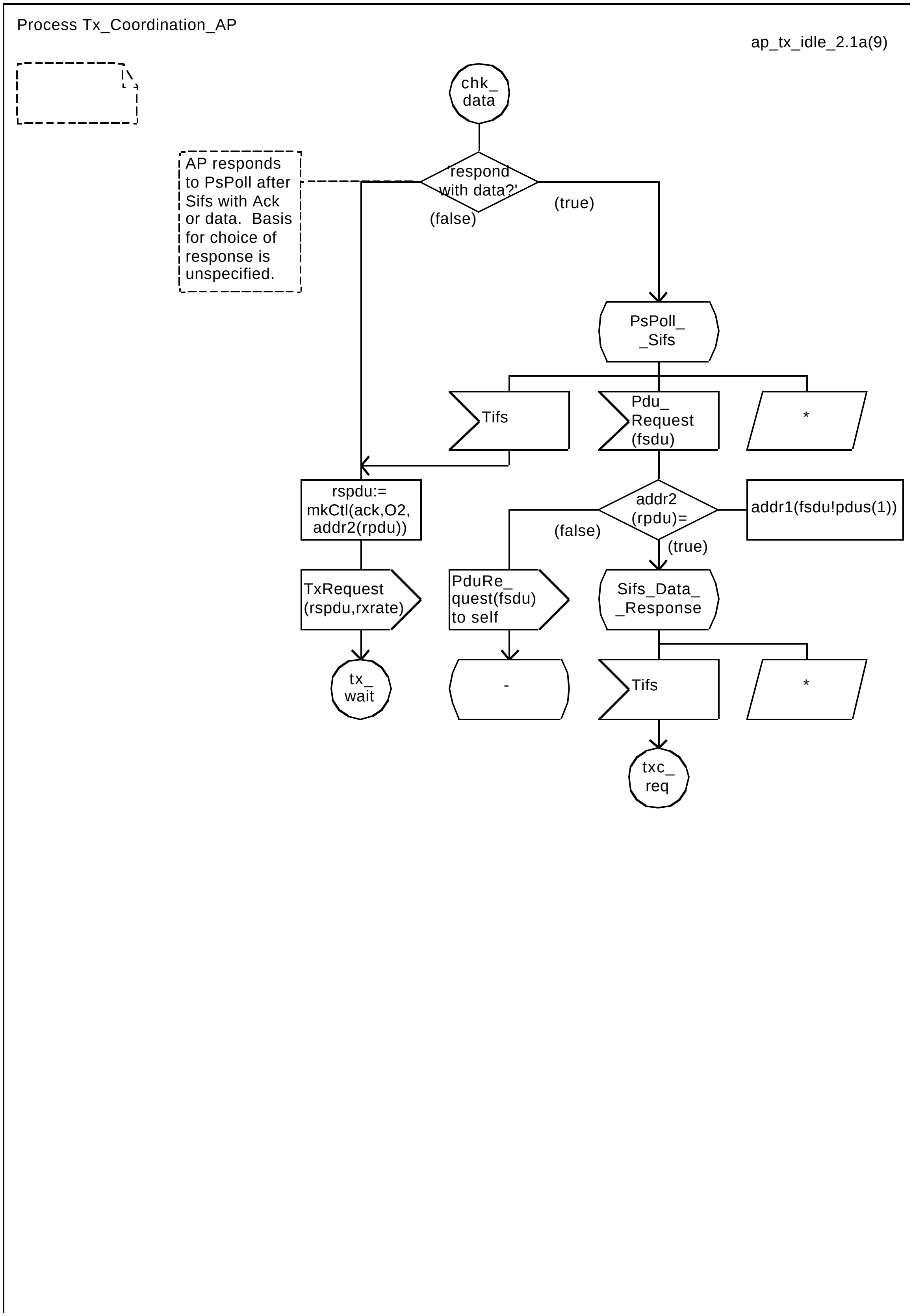
```
NEWTYPE PhyChrstcs struct
  aSlotTime Usec;
  aSifsTime Usec;
  aCCATime Usec;
  aRxTxTurnaroundTime Usec;
  aTxPLCPDelay Usec;
  aRxPLCPDelay Usec;
  aRxTxSwitchTime Usec;
  aTxRampOnTime Usec;
  aTxRampOffTime Usec;
  aTxRFDelay Usec;
  aRxRFDelay Usec;
  aAirPropagationTime Usec;
  aMACProcessingDelay Usec;
  aPreambleLength Usec;
  aPLCPHeaderLength Usec;
  aMPDUMaxLength Integer;
  aCWmin Integer;
  aCWmax Integer;
EndNewType PhyChrstcs;
```

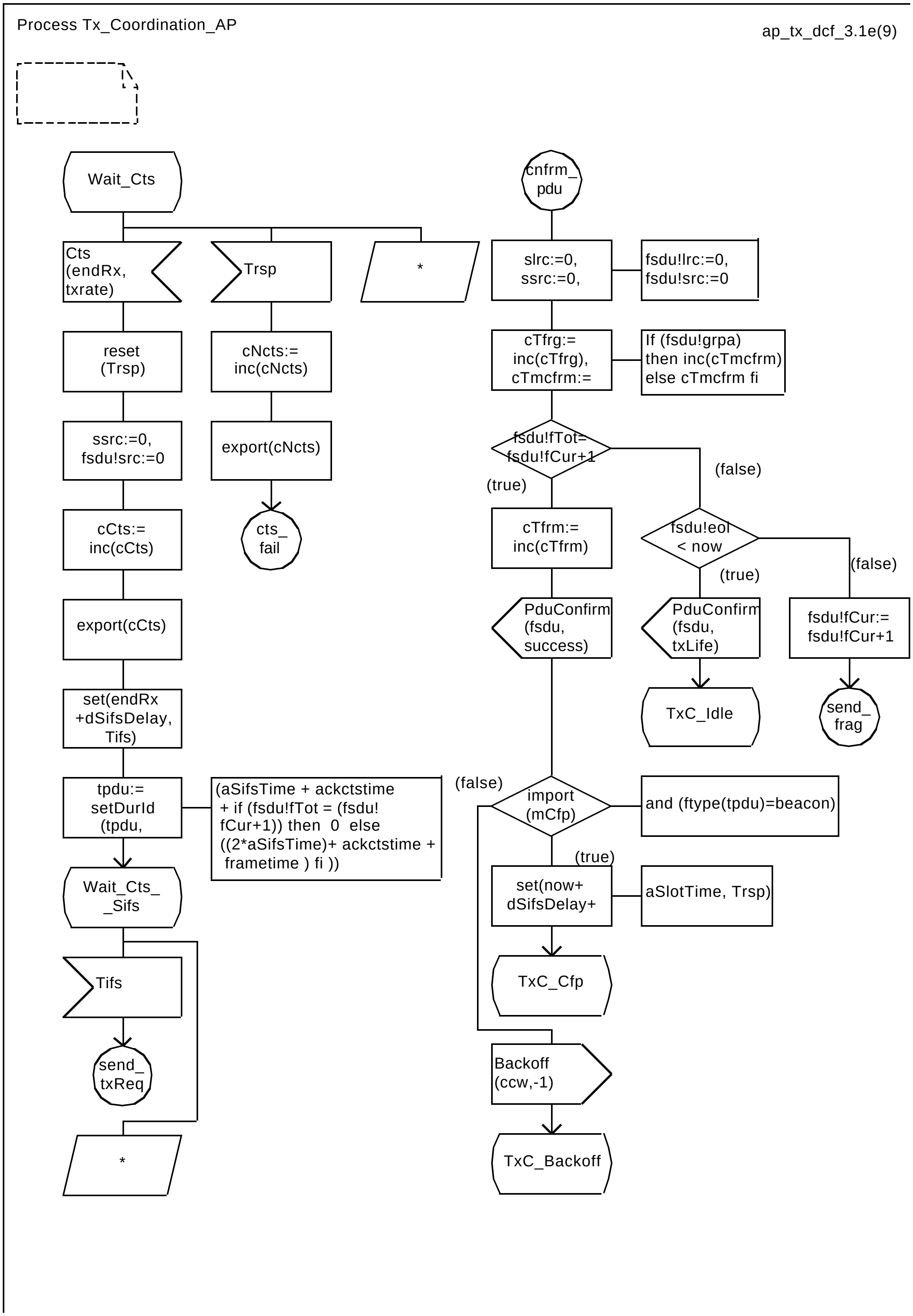
```
use macsorts ;

Package macmib                                              3204_e\PhyOperation(5)

/*******************************************************************
 *       PhyOperation Table
 *       (values shown are mostly for FH PHY)
 *******************************************************************/
synonym FHphy Integer = 01; /* enumerated dot11PHYType value */
synonym DSphy Integer = 02; /* enumerated dot11PHYType value */
synonym IRPhy Integer = 03; /* enumerated dot11PHYType value */
synonym dot11PHYType  Integer = FHphy;
remote dot11CurrentRegDomain  Integer nodelay;
synonym dot11TempType Integer = 01;

/*********************************************************************************************
*  PhyCharacteristic Parameters (values shown are mostly for FH PHY )
*********************************************************************************************/
/* NOTE:  The PhyCharacteristics are defined as synonyms because
   their values are static during MAC operation.  It is assumed
   that , during each initialization of MAC operation, current
   values for each of these parameters are obtained from the
   PHY using the PlmeCharacteristics primitive. */
remote procedure TxTime;  returns Integer;
synonym aSlotTime  Usec = (aCcaTime + aRxTxTurnaroundTime +
    aAirPropagationTime + aMacProcessingTime);
synonym aCcaTime  Usec = 27;
synonym aRxTxTurnaroundTime Usec = (aTxPlcpDelay + aRxTxSwitchTime +
    aTxRampOnTime + aTxRfDelay);
synonym aTxPlcpDelay  Usec = 1;
synonym aRxTxSwitchTime  Usec = 10;
synonym aTxRampOnTime  Usec = 8;
synonym aTxRfDelay  Usec = 1;
synonym aSifsTime Usec = (aRxRfDelay + aRxPlcpDelay +
    aMacProcessingTime + aRxTxTurnaroundTime);
synonym aRxRfDelay  Usec = 4;
synonym aRxPlcpDelay  Usec = 2;
synonym aMacProcessingTime  Usec = 2;
synonym aTxRampOffTime  Usec = 8;
synonym aPreambleLength  Usec = 96;
synonym aPlcpHeaderLength  Usec = 32;
synonym aMpduMaxLength Integer = 4095;
synonym aAirPropagationTime  Usec = 1;
synonym aCWmax  Integer = 1023;
synonym aCWmin  Integer = 15;
```

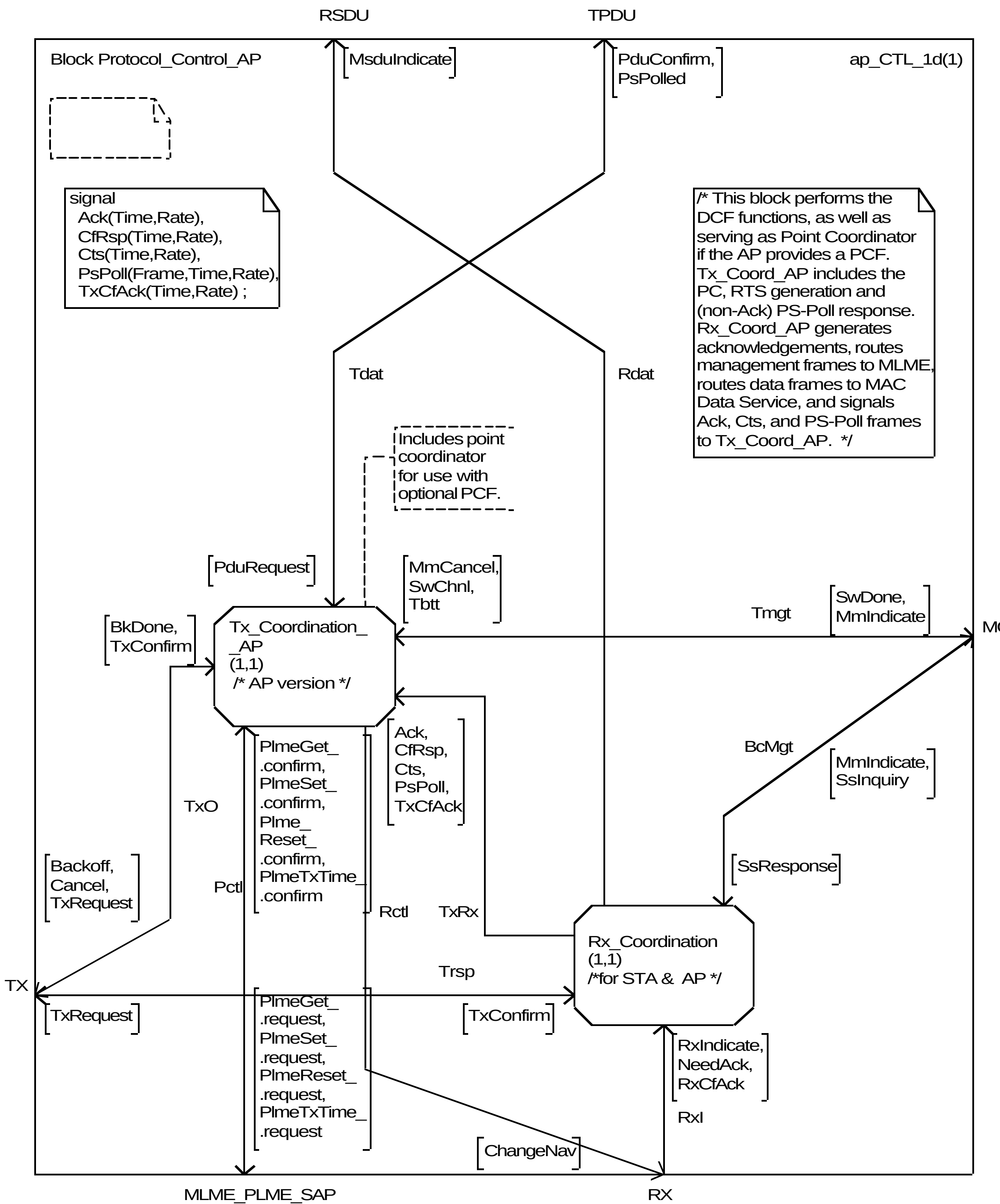

Process Tx_Coordination_AP
ap_tx_idle_2.1a(9)
chk_
data
AP responds
to PsPoll after
Sifs with Ack
or data. Basis
for choice of
response is
unspecified.
'respond
with data?'
(false)
(true)
PsPoll_
_Sifs
Tifs
Pdu_
Request
(fsdu)
*
rspdu:=
mkCtl(ack,O2,
addr2(rpdu))
addr2
(rpdu)=
addr1(fsdu!pdus(1))
(false)
(true)
TxRequest
(rspdu,rxrate)
PduRe_
quest(fsdu)
to self
Sifs_Data_
_Response
tx_
wait
-
Tifs
*
txc_
req

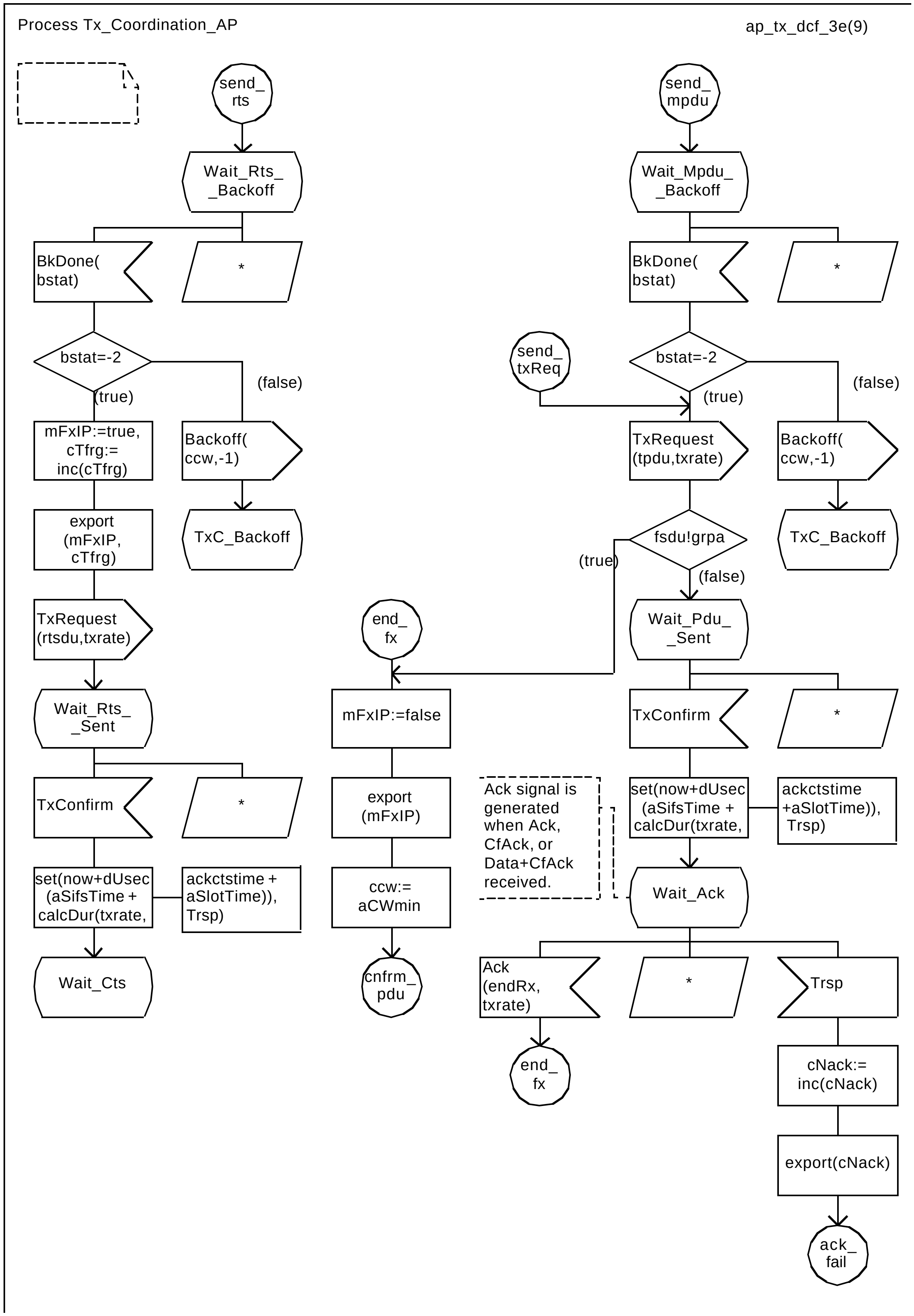
Package macsorts 3125_e\RateAndDurationSorts(31)

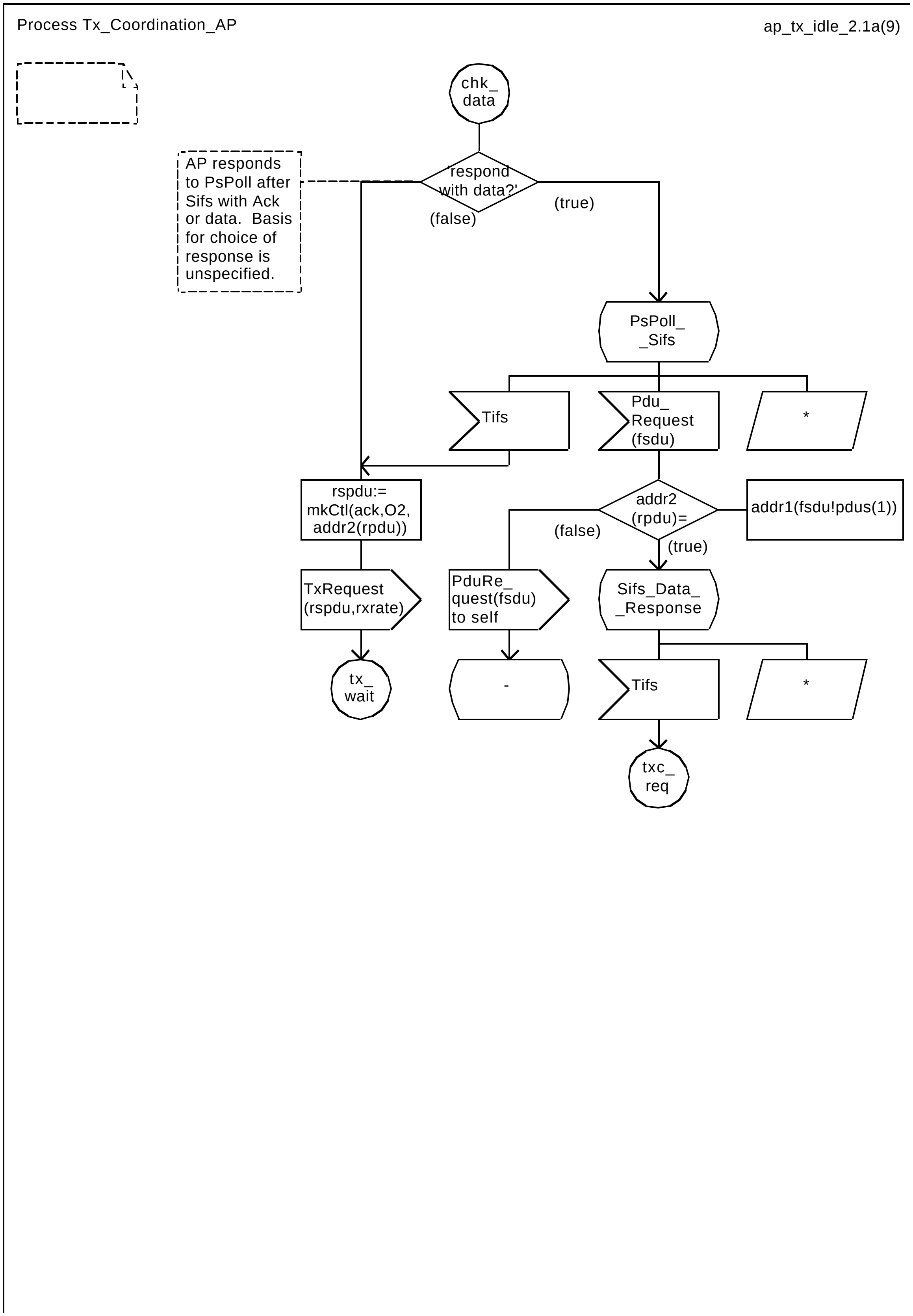
```
/*******************************************************************
 *        Multi-rate support sorts
 *******************************************************************/
newtype Rate   inherits Octet   operators all;
  adding  operators
    calcDur : Rate, Integer -> Integer; /* converts (rate,bitCount) to integer usec */
    rateVal : Rate -> Rate; /* clears high-order bit */
    basicRate : Rate -> Rate; /* sets high-order bit */
    isBasic : Rate -> Boolean; /* true if high-order bit set */
  axioms
    for all r in Rate(    for all i in Integer(    for all b in Boolean(
          calcDur(r, i) == ((((10000000 + (octetVal(r and 0x7F) - 1)) /
              (500 * octetVal(r and 0x7F))) * i) + 9999) / 10000;
          rateVal(r) == r and 0x7F;          basicRate(r) == r or 0x80;
          isBasic(r) == (r and 0x80) = 0x80; )));
endnewtype Rate;
syntype RateString = Octetstring   endsyntype RateString;
```

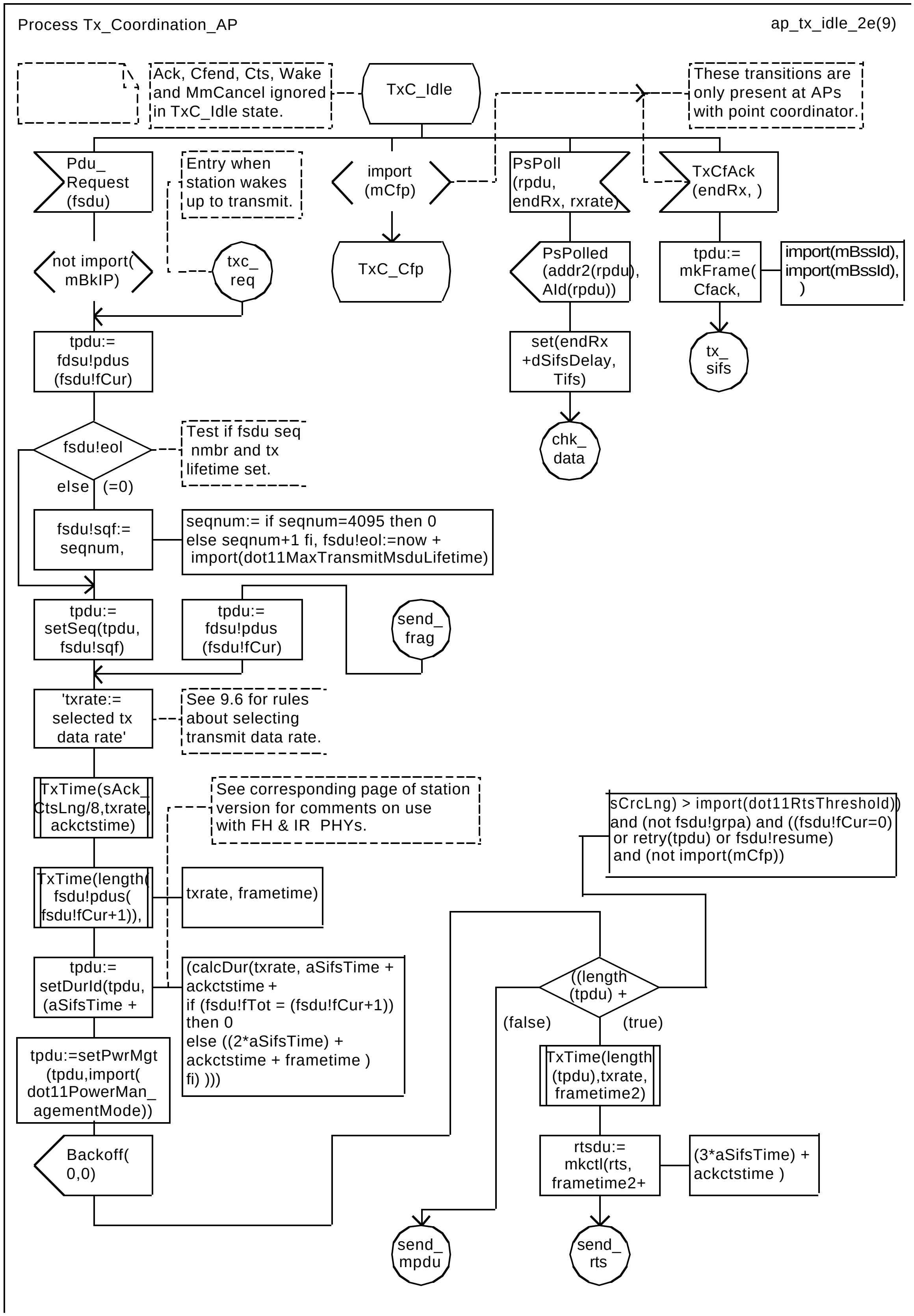

Process Tx_Coordination_AP
ap_tx_idle_2.1a(9)
chk_ data
AP responds to PsPoll after Sifs with Ack or data. Basis for choice of response is unspecified.
'respond with data?'
(true)
(false)
PsPoll_ _Sifs
Tifs
Pdu_ Request (fsdu)
*
rspdu:= mkCtl(ack,O2, addr2(rpdu))
addr2 (rpdu)=
addr1(fsdu!pdus(1))
(false)
(true)
TxRequest (rspdu,rxrate)
PduRe_ quest(fsdu) to self
Sifs_Data_ _Response
tx_ wait
-
Tifs
*
txc_ req

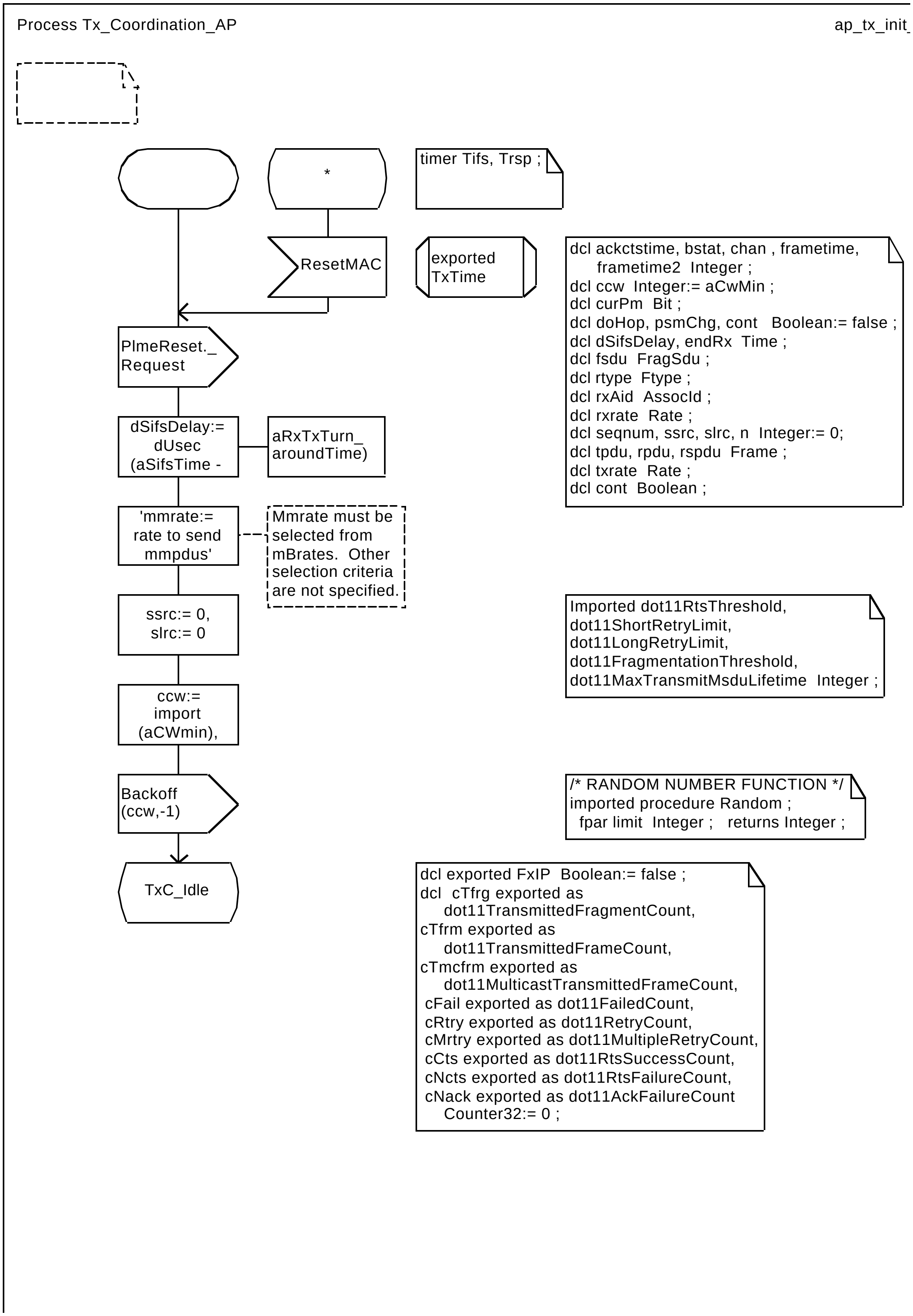

Process Tx_Coordination_AP
ap_tx_dcf_3.1e(9)
Wait_Cts
Cts (endRx, txrate)
Trsp
*
reset (Trsp)
cNcts:= inc(cNcts)
ssrc:=0, fsdu!src:=0
export(cNcts)
cts_ fail
cCts:= inc(cCts)
export(cCts)
set(endRx +dSifsDelay, Tifs)
tpdu:= setDurId (tpdu,
(aSifsTime + ackctstime + if (fsdu!fTot = (fsdu! fCur+1)) then 0 else ((2*aSifsTime)+ ackctstime + frametime) fi))
Wait_Cts_ _Sifs
Tifs
send_ txReq
*
cnfrm_ pdu
slrc:=0, ssrc:=0,
fsdu!lrc:=0, fsdu!src:=0
cTfrg:= inc(cTfrg), cTmcfrm:=
If (fsdu!grpa) then inc(cTmcfrm) else cTmcfrm fi
fsdu!fTot= fsdu!fCur+1
(true)
(false)
cTfrm:= inc(cTfrm)
fsdu!eol < now
(true)
(false)
PduConfirm (fsdu, success)
PduConfirm (fsdu, txLife)
fsdu!fCur:= fsdu!fCur+1
TxC_Idle
send_ frag
(false)
import (mCfp)
and (ftype(tpdu)=beacon)
(true)
set(now+ dSifsDelay+
aSlotTime, Trsp)
TxC_Cfp
Backoff (ccw,-1)
TxC_Backoff


RSDU
TPDU
Block Protocol_Control_AP
MsduIndicate
PduConfirm,
PsPolled
ap_CTL_1d(1)
signal
Ack(Time,Rate),
CfRsp(Time,Rate),
Cts(Time,Rate),
PsPoll(Frame,Time,Rate),
TxCfAck(Time,Rate) ;
/* This block performs the DCF functions, as well as serving as Point Coordinator if the AP provides a PCF. Tx_Coord_AP includes the PC, RTS generation and (non-Ack) PS-Poll response. Rx_Coord_AP generates acknowledgements, routes management frames to MLME, routes data frames to MAC Data Service, and signals Ack, Cts, and PS-Poll frames to Tx_Coord_AP. */
Tdat
Rdat
Includes point coordinator for use with optional PCF.
PduRequest
MmCancel,
SwChnl,
Tbtt
Tmgt
SwDone,
MmIndicate
BkDone,
TxConfirm
Tx_Coordination_
_AP
(1,1)
/* AP version */
Ack,
CfRsp,
Cts,
PsPoll,
TxCfAck
BcMgt
MmIndicate,
SsInquiry
TxO
PlmeGet_
.confirm,
PlmeSet_
.confirm,
Plme_
Reset_
.confirm,
PlmeTxTime_
.confirm
SsResponse
Backoff,
Cancel,
TxRequest
Pctl
Rctl
TxRx
Rx_Coordination
(1,1)
/*for STA & AP */
Trsp
TX
TxRequest
PlmeGet_
.request,
PlmeSet_
.request,
PlmeReset_
.request,
PlmeTxTime_
.request
TxConfirm
RxIndicate,
NeedAck,
RxCfAck
RxI
ChangeNav
MLME_PLME_SAP
RX


Process Tx_Coordination_AP
ap_tx_dcf_3e(9)
send_ rts
Wait_Rts_ _Backoff
BkDone(bstat)
*
bstat=-2
(true)
(false)
mFxIP:=true, cTfrg:= inc(cTfrg)
Backoff(ccw,-1)
export (mFxIP, cTfrg)
TxC_Backoff
TxRequest (rtsdu,txrate)
Wait_Rts_ _Sent
TxConfirm
*
set(now+dUsec (aSifsTime + calcDur(txrate,
ackctstime + aSlotTime)), Trsp)
Wait_Cts
end_ fx
mFxIP:=false
export (mFxIP)
ccw:= aCWmin
cnfrm_ pdu
send_ mpdu
Wait_Mpdu_ _Backoff
BkDone(bstat)
*
send_ txReq
bstat=-2
(true)
(false)
TxRequest (tpdu,txrate)
Backoff(ccw,-1)
fsdu!grpa
(true)
(false)
TxC_Backoff
Wait_Pdu_ _Sent
TxConfirm
*
Ack signal is generated when Ack, CfAck, or Data+CfAck received.
set(now+dUsec (aSifsTime + calcDur(txrate,
ackctstime +aSlotTime)), Trsp)
Wait_Ack
Ack (endRx, txrate)
*
Trsp
end_ fx
cNack:= inc(cNack)
export(cNack)
ack_ fail


Process Tx_Coordination_AP
ap_tx_idle_2.1a(9)
chk_ data
AP responds to PsPoll after Sifs with Ack or data. Basis for choice of response is unspecified.
'respond with data?'
(false)
(true)
PsPoll_ _Sifs
Tifs
Pdu_ Request (fsdu)
*
rspdu:= mkCtl(ack,O2, addr2(rpdu))
addr2 (rpdu)=
addr1(fsdu!pdus(1))
(false)
(true)
TxRequest (rspdu,rxrate)
PduRe_ quest(fsdu) to self
Sifs_Data_ _Response
tx_ wait
-
Tifs
*
txc_ req


Process Tx_Coordination_AP
ap_tx_idle_2e(9)
Ack, Cfend, Cts, Wake and MmCancel ignored in TxC_Idle state.
TxC_Idle
These transitions are only present at APs with point coordinator.
Pdu_ Request (fsdu)
Entry when station wakes up to transmit.
import (mCfp)
PsPoll (rpdu, endRx, rxrate)
TxCfAck (endRx,)
not import(mBkIP)
txc_ req
TxC_Cfp
PsPolled (addr2(rpdu), AId(rpdu))
tpdu:= mkFrame(Cfack,
import(mBssId), import(mBssId),)
tpdu:= fdsu!pdus (fsdu!fCur)
set(endRx +dSifsDelay, Tifs)
tx_ sifs
chk_ data
fsdu!eol
Test if fsdu seq nmbr and tx lifetime set.
else
(=0)
fsdu!sqf:= seqnum,
seqnum:= if seqnum=4095 then 0 else seqnum+1 fi, fsdu!eol:=now + import(dot11MaxTransmitMsduLifetime)
tpdu:= setSeq(tpdu, fsdu!sqf)
tpdu:= fdsu!pdus (fsdu!fCur)
send_ frag
'txrate:= selected tx data rate'
See 9.6 for rules about selecting transmit data rate.
TxTime(sAck_ CtsLng/8,txrate, ackctstime)
See corresponding page of station version for comments on use with FH & IR PHYs.
sCrcLng) > import(dot11RtsThreshold)) and (not fsdu!grpa) and ((fsdu!fCur=0) or retry(tpdu) or fsdu!resume) and (not import(mCfp))
TxTime(length(fsdu!pdus(fsdu!fCur+1)),
txrate, frametime)
tpdu:= setDurId(tpdu, (aSifsTime +
(calcDur(txrate, aSifsTime + ackctstime + if (fsdu!fTot = (fsdu!fCur+1)) then 0 else ((2*aSifsTime) + ackctstime + frametime) fi))))
((length (tpdu) +
(false)
(true)
tpdu:=setPwrMgt (tpdu,import(dot11PowerMan_ agementMode))
TxTime(length (tpdu),txrate, frametime2)
Backoff(0,0)
rtsdu:= mkctl(rts, frametime2+
(3*aSifsTime) + ackctstime)
send_ mpdu
send_ rts


Process Tx_Coordination_AP
ap_tx_init_
*
timer Tifs, Trsp ;
ResetMAC
exported TxTime
dcl ackctstime, bstat, chan , frametime, frametime2 Integer ;
dcl ccw Integer:= aCwMin ;
dcl curPm Bit ;
dcl doHop, psmChg, cont Boolean:= false ;
dcl dSifsDelay, endRx Time ;
dcl fsdu FragSdu ;
dcl rtype Ftype ;
dcl rxAid AssocId ;
dcl rxrate Rate ;
dcl seqnum, ssrc, slrc, n Integer:= 0;
dcl tpdu, rpdu, rspdu Frame ;
dcl txrate Rate ;
dcl cont Boolean ;
PlmeReset._ Request
dSifsDelay:= dUsec (aSifsTime -
aRxTxTurn_ aroundTime)
'mmrate:= rate to send mmpdus'
Mmrate must be selected from mBrates. Other selection criteria are not specified.
ssrc:= 0, slrc:= 0
Imported dot11RtsThreshold, dot11ShortRetryLimit, dot11LongRetryLimit, dot11FragmentationThreshold, dot11MaxTransmitMsduLifetime Integer ;
ccw:= import (aCWmin),
Backoff (ccw,-1)
/* RANDOM NUMBER FUNCTION */
imported procedure Random ;
fpar limit Integer ; returns Integer ;
TxC_Idle
dcl exported FxIP Boolean:= false ;
dcl cTfrg exported as dot11TransmittedFragmentCount,
cTfrm exported as dot11TransmittedFrameCount,
cTmcfrm exported as dot11MulticastTransmittedFrameCount,
cFail exported as dot11FailedCount,
cRtry exported as dot11RetryCount,
cMrtry exported as dot11MultipleRetryCount,
cCts exported as dot11RtsSuccessCount,
cNcts exported as dot11RtsFailureCount,
cNack exported as dot11AckFailureCount
Counter32:= 0 ;

```
use macsorts ;
use macmib ;
```

System Access_Point

AP_signallists_

```
signallist
MlmeRequestSignals=
  MlmeAssociate.request,
  MlmeAuthenticate.request,
  MlmeDeauthenticate.request,
  MlmeDisassociate.request,
  MlmeGet.request,
  MlmeJoin.request,
  MlmePowermgt.request,
  MlmeReassociate.request,
  MlmeReset.request,
  MlmeScan.request,
  MlmeSet.request,
  MlmeStart.request ;
```

```
signallist
MlmeConfirmSignals=
  MlmeAssociate.confirm,
  MlmeAuthenticate.confirm,
  MlmeDeauthenticate.confirm,
  MlmeDisassociate.confirm,
  MlmeGet.confirm,
  MlmeJoin.confirm,
  MlmePowermgt.confirm,
  MlmeReassociate.confirm,
  MlmeReset.confirm,
  MlmeScan.confirm,
  MlmeSet.confirm,
  MlmeStart.confirm ;
```

```
signallist
MlmeIndicationSignals=
  MlmeAuthenticate.indication,
  MlmeDeauthenticate.indication,
  MlmeDisassociate.indication,
  MlmeAssociate.indication,
  MlmeReassociate.indication ;
```

```
signallist
SmtRequestSignals=
  MlmeAssociate.request,
  MlmeAuthenticate.request,
  MlmeDeauthenticate.request,
  MlmeDisassociate.request,
  MlmeJoin.request,
  MlmeReassociate.request,
  MlmeScan.request,
  MlmeStart.request ;
```

```
signallist
SmtConfirmSignals=
  MlmeAssociate.confirm,
  MlmeAuthenticate.confirm,
  MlmeDeauthenticate.confirm,
  MlmeDisassociate.confirm,
  MlmeJoin.confirm,
  MlmeReassociate.confirm,
  MlmeScan.confirm,
  MlmeStart.confirm ;
```

```
signallist
SmtIndicationSignals=
  MlmeAuthenticate.indication,
  MlmeDeauthenticate.indication,
  MlmeDisassociate.indication,
  MlmeAssociate.indication,
  MlmeReassociate.indication ;
```

```
signallist
PhyTxRequestSignals=
  PhyTxStart.request,
  PhyTxEnd.request,
  PhyData.request ;
```

```
signallist
PhyTxConfirmSignals=
  PhyTxStart.confirm,
  PhyTxEnd.confirm,
  PhyData.confirm ;
```

```
signallist
PhyRxSignals=
  PhyRxStart.indication,
  PhyRxEnd.indication,
  PhyData.indication,
  PhyCca.indication,
  PhyCcarst.confirm ;
```

```
signallist
PlmeRequestSignals=
  PlmeCharacteristics.request,
  PlmeGet.request,
  PlmeSet.request,
  PlmeReset.request,
  PlmeTxTime.request ;
```

```
signallist
PlmeConfirmSignals=
  PlmeCharacteristics.confirm,
  PlmeGet.confirm,
  PlmeSet.confirm,
  PlmeReset.confirm,
  PlmeTxTime.confirm ;
```

```
use macsorts ;
use macmib ;
```

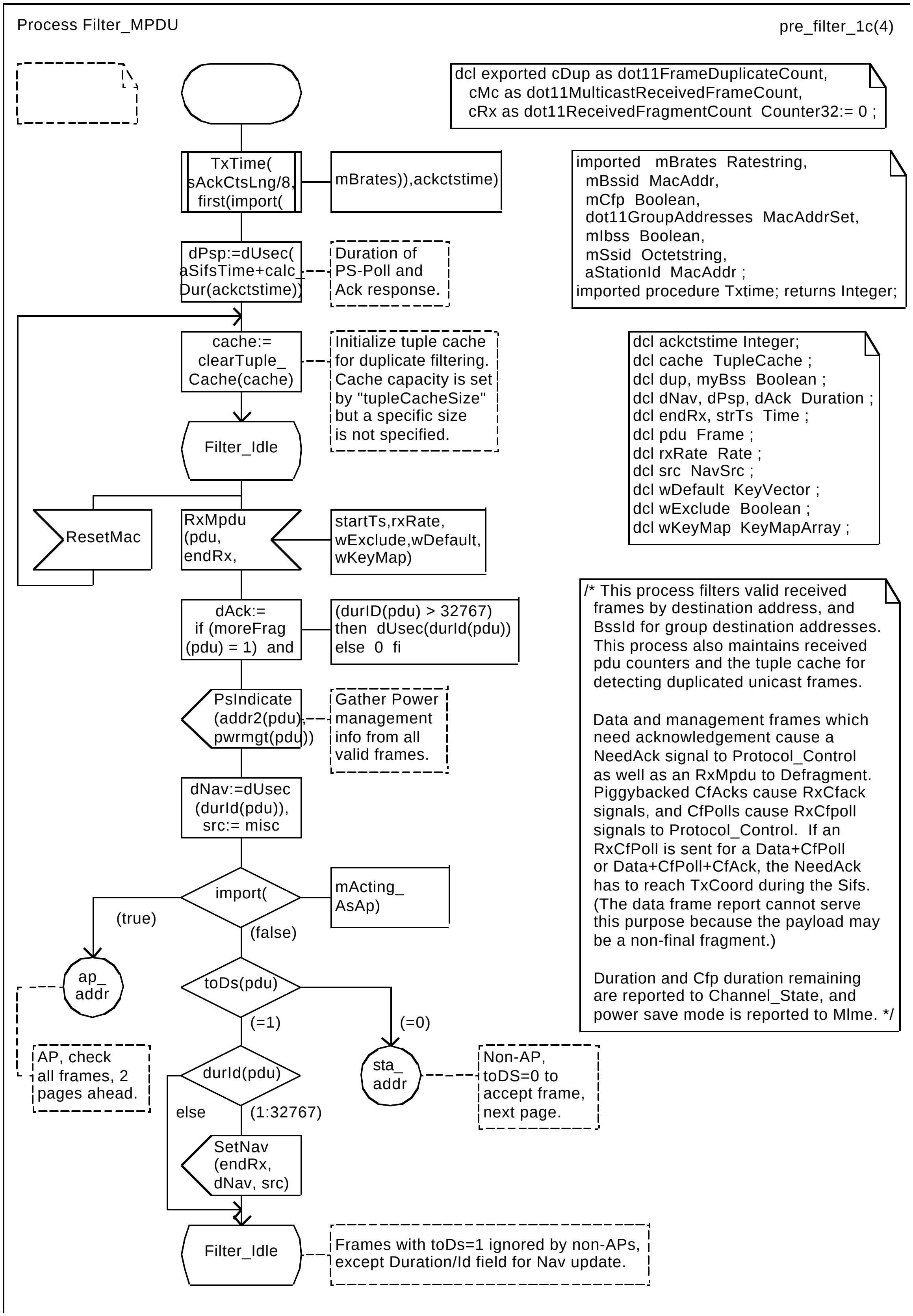
System Access_Point

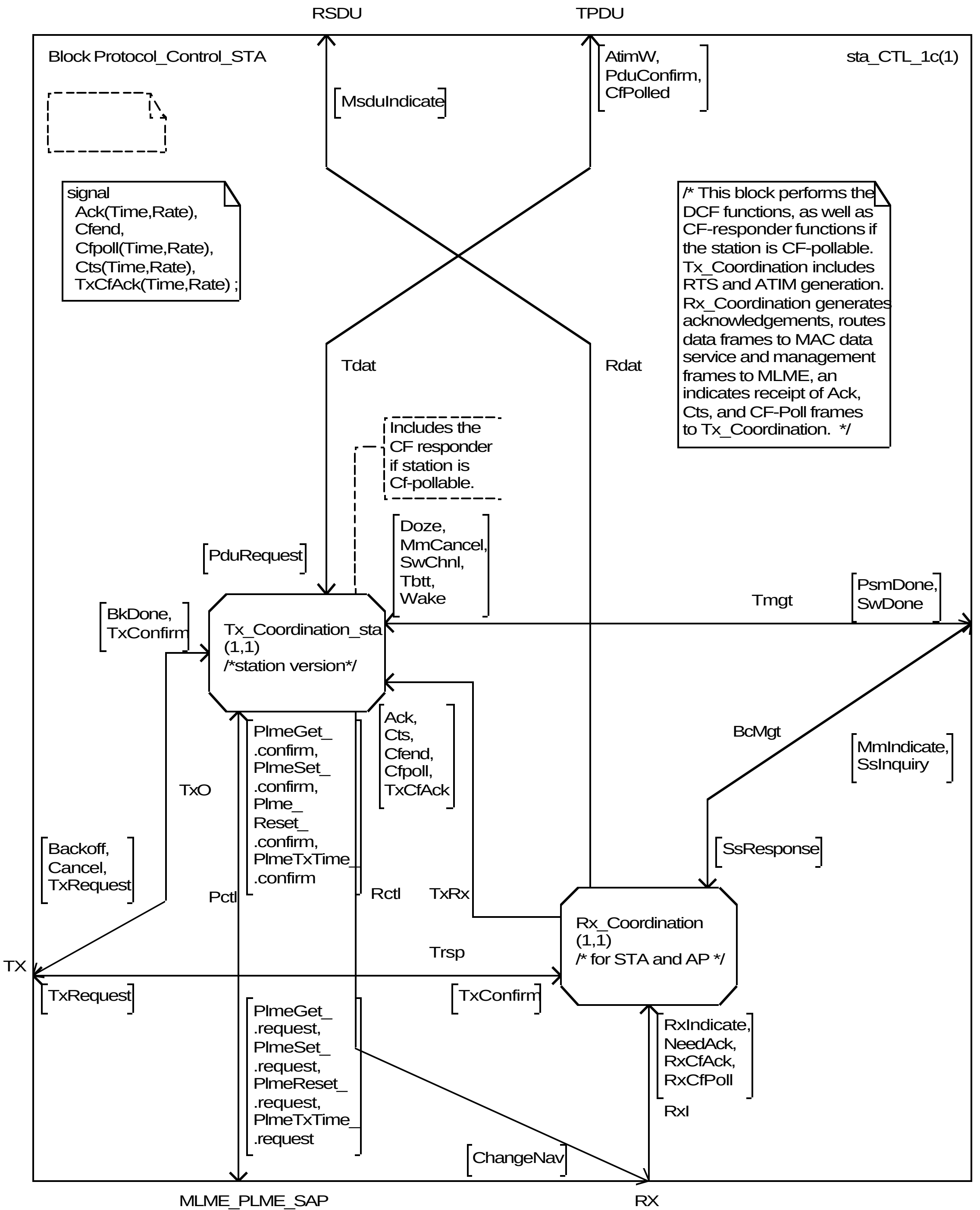
AP_signa

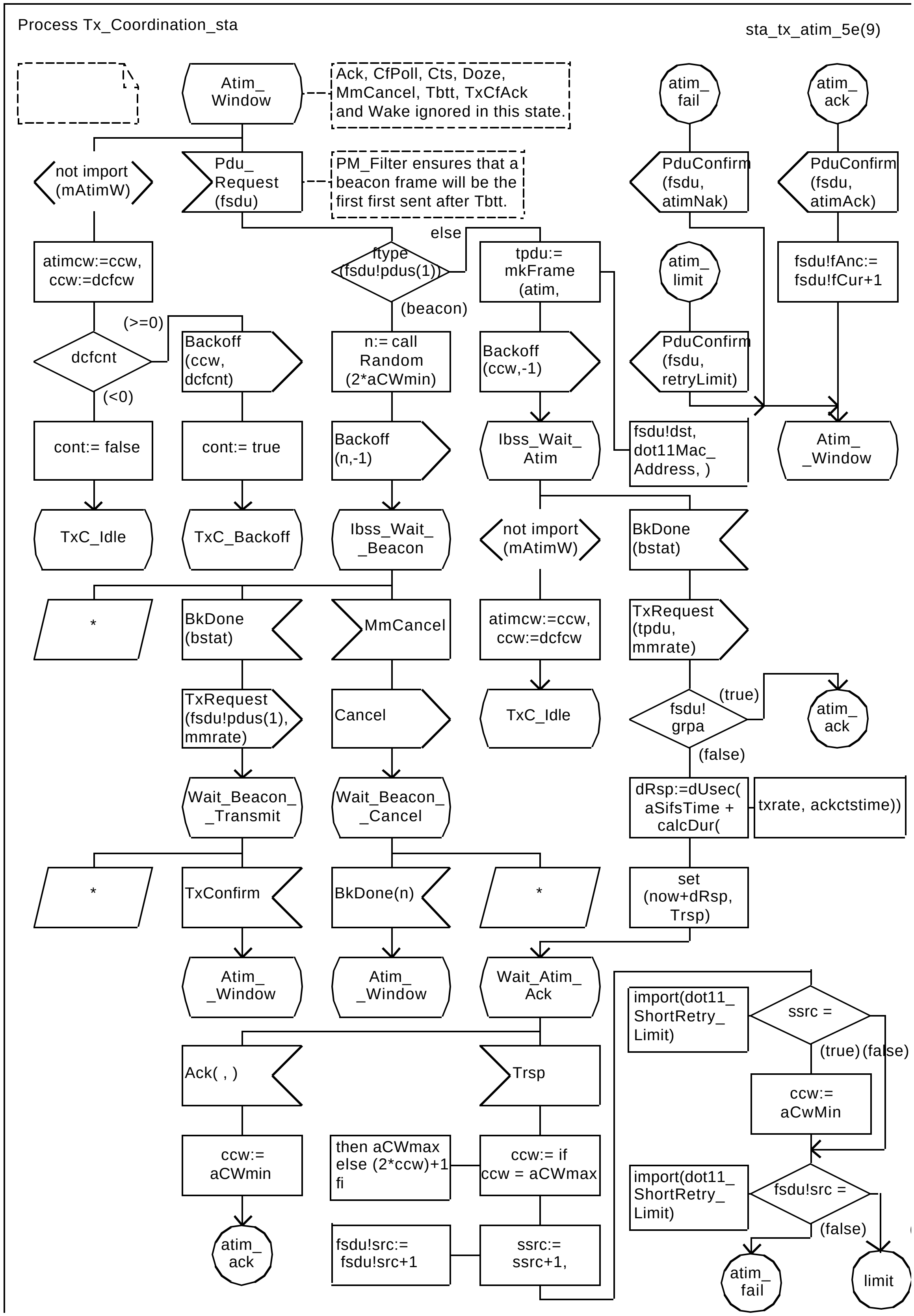
```
newtype DsStatus  literals
  assoc, disassoc, reassoc, unknown
endnewtype DsStatus ;
```

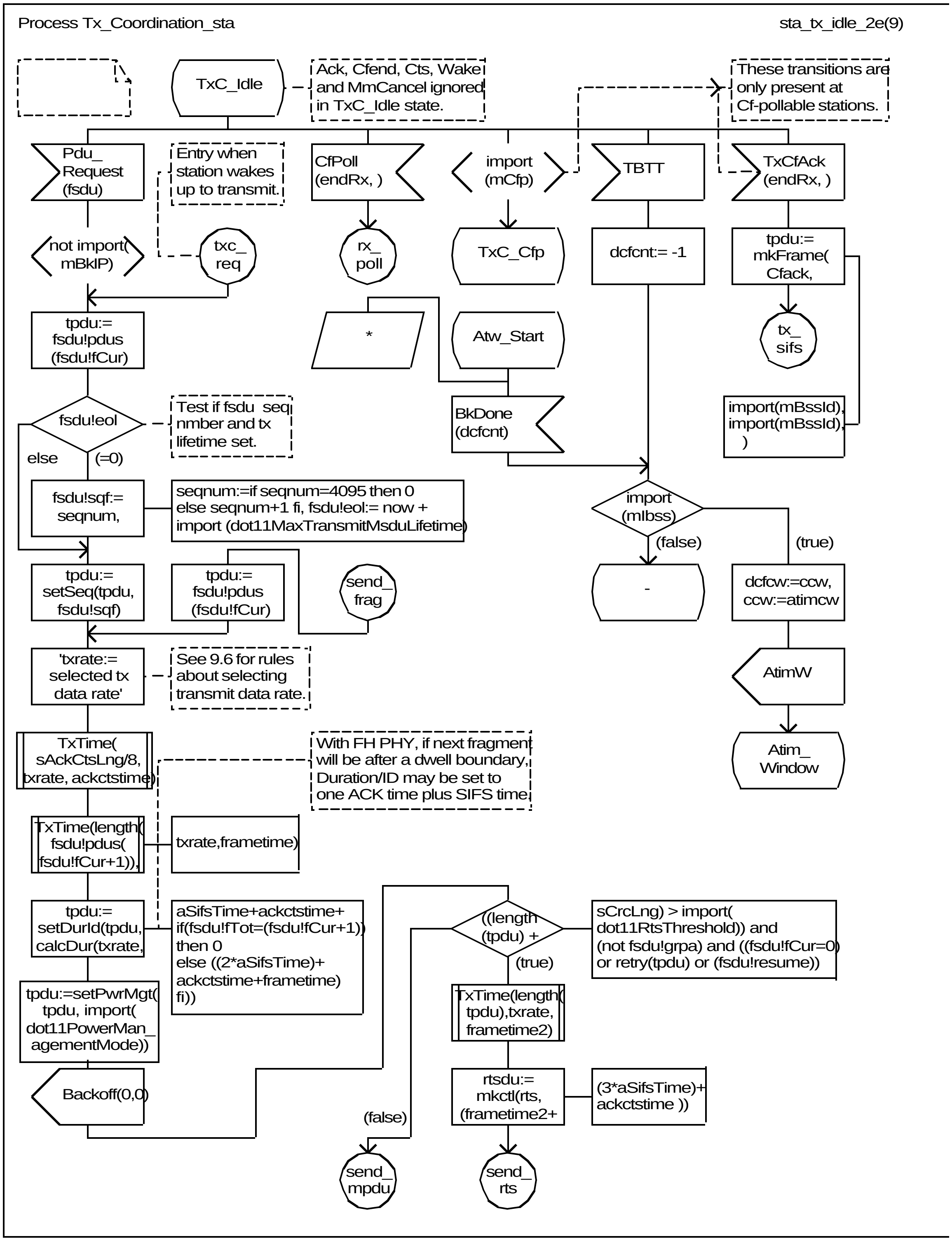
```
signal
  AsChange(Frame,DsStatus),
  Backoff(Integer,Integer),
  BkDone(Integer),
  Busy,
  Cancel,
  ChangeNav(Time,Duration,NavSrc),
  DsInquiry(MacAddr,MacAddr),
  DsNotify(MacAddr,DsStatus),
  DsResponse(MacAddr,MacAddr,DsStatus),
  FromDsm(MacAddr,MacAddr,Octetstring),
  Idle,
  MaUnitdata.indication(MacAddr,MacAddr,
    Routing,Octetstring,RxStatus,
    CfPriority,ServiceClass),
  MaUnitdata.request(MacAddr,MacAddr,
    Routing,Octetstring,CfPriority,ServiceClass),
  MaUnitdataStatus.indication(MacAddr,
    MacAddr,TxStatus,CfPriority,ServiceClass),
  MlmeAssociate.confirm(MlmeStatus),
  MlmeAssociate.indication(MacAddr),
  MlmeAssociate.request(MacAddr,Kusec,Capability,Integer),
  MlmeAuthenticate.confirm
    (MacAddr,AuthType,MlmeStatus),
  MlmeAuthenticate.indication(MacAddr,AuthType),
  MlmeAuthenticate.request(MacAddr,AuthType,Kusec),
  MlmeDeauthenticate.confirm(MacAddr,MlmeStatus),
  MlmeDeauthenticate.indication(MacAddr,ReasonCode),
  MlmeDeauthenticate.request(MacAddr,ReasonCode),
  MlmeDisassociate.confirm(MlmeStatus),
  MlmeDisassociate.indication(MacAddr,ReasonCode),
  MlmeDisassociate.request(MacAddr,ReasonCode),
  MlmeGet.confirm(MibStatus,MibAtrib,MibValue),
  MlmeGet.request(MibAtrib),
  MlmeJoin.confirm(MlmeStatus),
  MlmeJoin.request(BssDscr,Integer,Usec,Ratestring),
  MlmePowermgt.confirm(MlmeStatus),
  MlmePowermgt.request(PwrSave,Boolean,Boolean),
  MlmeReassociate.confirm(MlmeStatus),
  MlmeReassociate.indication(MacAddr),
  MlmeReassociate.request(MacAddr,Kusec,Capability,Integer),
  MlmeReset.confirm(MlmeStatus),
  MlmeReset.request,
  MlmeScan.confirm(BssDscrSet,MlmeStatus),
  MlmeScan.request(BssTypeSet,MacAddr,Octetstring,
    ScanType,Usec,Intstring,Kusec,Kusec),
  MlmeSet.confirm(MibStatus,MibAtrib),
  MlmeSet.request(MibAtrib,MibValue),
  MlmeStart.confirm(MlmeStatus),
  MlmeStart.request(Octetstring,BssType,Kusec,
    Integer,CfParms,PhyParms,IbssParms,Usec,
    Capability,Ratestring,Ratestring) ;
```

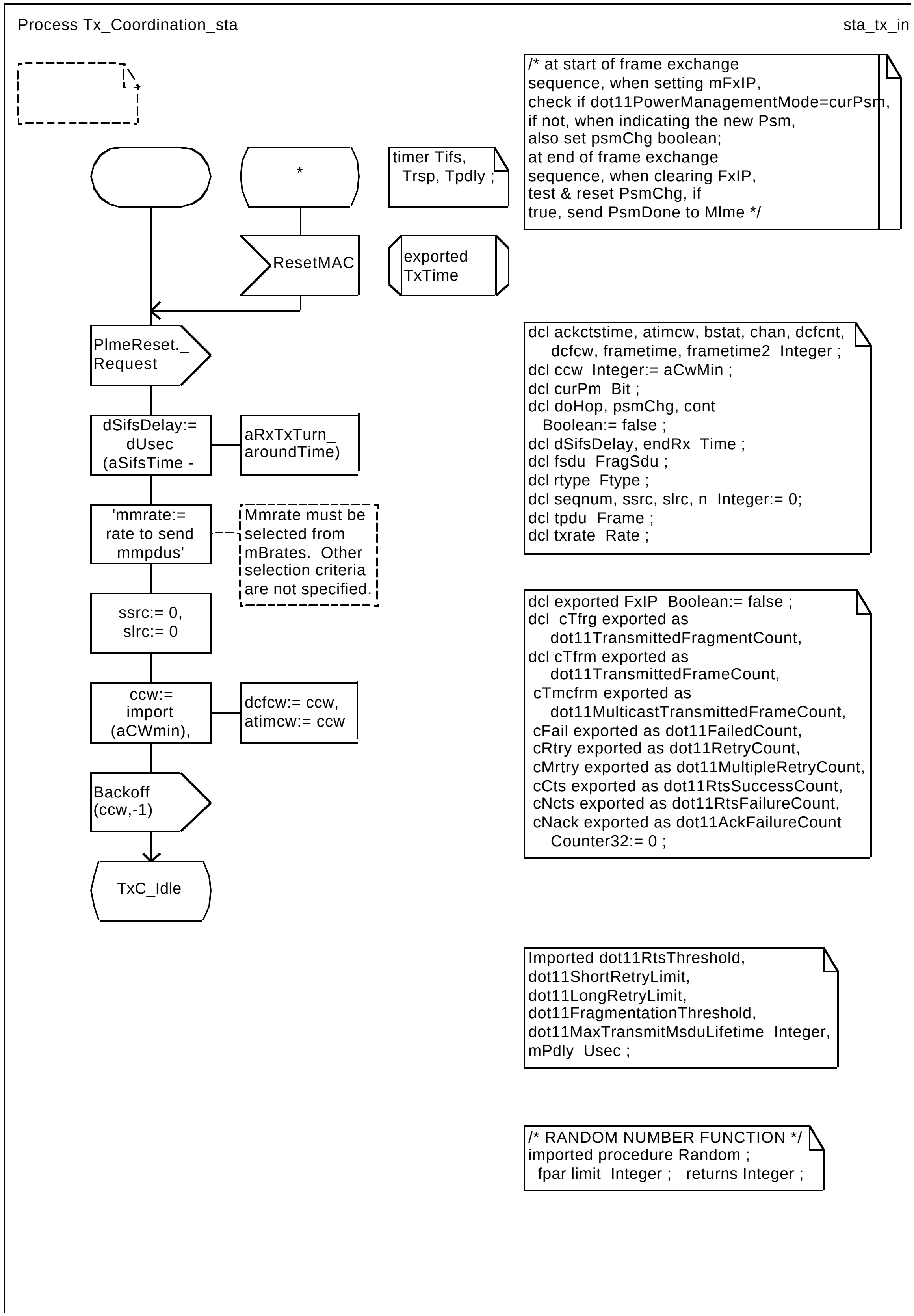
```
signal
  MmCancel,
  MmConfirm(Frame,TxStatus),
  MmIndicate(Frame,Time,Time,StateErr),
  MmRequest(Frame,Imed,Rate),
  MsduConfirm(Frame,CfPriority,TxStatus),
  MsduIndicate(Frame,CfPriority),
  MsduRequest(Frame,CfPriority),
  NeedAck(MacAddr,Time,Duration,Rate),
  PduConfirm(FragSdu,TxResult),
  PduRequest(FragSdu),
  PhyCca.indication(Ccastatus),
  PhyCcarst.confirm,
  PhyCcarst.request,
  PhyData.confirm,
  PhyData.indication(Octet),
  PhyData.request(Octet),
  PhyRxEnd.indication(PhyRxStat),
  PhyRxStart.indication(Integer,Rate),
  PhyTxEnd.confirm,
  PhyTxEnd.request,
  PhyTxStart.confirm,
  PhyTxStart.request(Integer,Rate),
  PlmeCharacteristics.confirm(PhyChrstcs),
  PlmeCharacteristics.request,
  PlmeGet.confirm(MibStatus,
    MibAtrib,MibValue),
  PlmeGet.request(MibAtrib),
  PlmeReset.confirm(Boolean),
  PlmeReset.request,
  PlmeSet.confirm(MibStatus,MibAtrib),
  PlmeSet.request(MibAtrib,MibValue),
  PlmeTxTime.confirm(Integer),
  PlmeTxTime.request(Integer, Rate),
  PsmDone,
  PsPolled(MacAddr,AsocId),
  PsChange(MacAddr,PsMode),
  PsIndicate(MacAddr,PsMode),
  PsInquiry(MacAddr),
  PsResponse(MacAddr,PsMode),
  ResetMAC,
  RxCfAck(MacAddr),
  RxIndicate(Frame,Time,Time,Rate),
  Slot,
  SsInquiry(MacAddr),
  SsResponse(MacAddr,
    StationState,StationState),
  SwChnl(Integer,Boolean),
  SwDone,
  ToDsm(MacAddr,MacAddr,Octetstring),
  TxConfirm,
  TxRequest(Frame,Rate) ;
```

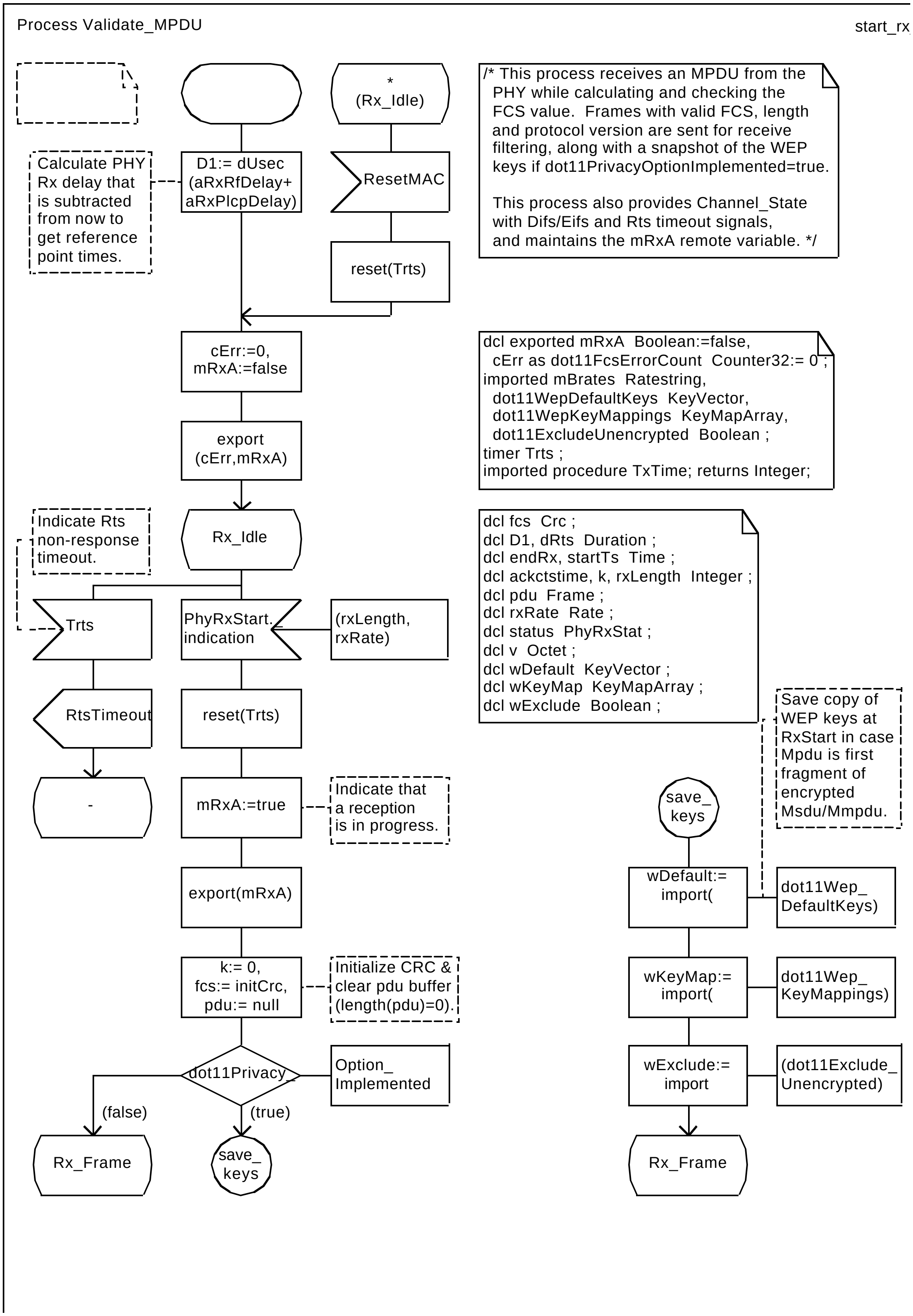

Process Filter_MPDU
pre_filter_1c(4)
dcl exported cDup as dot11FrameDuplicateCount,
cMc as dot11MulticastReceivedFrameCount,
cRx as dot11ReceivedFragmentCount Counter32:= 0 ;
TxTime(
sAckCtsLng/8,
first(import(
mBrates)),ackctstime)
imported mBrates Ratestring,
mBssid MacAddr,
mCfp Boolean,
dot11GroupAddresses MacAddrSet,
mIbss Boolean,
mSsid Octetstring,
aStationId MacAddr ;
imported procedure Txtime; returns Integer;
dPsp:=dUsec(
aSifsTime+calc_
Dur(ackctstime))
Duration of
PS-Poll and
Ack response.
cache:=
clearTuple_
Cache(cache)
Initialize tuple cache
for duplicate filtering.
Cache capacity is set
by "tupleCacheSize"
but a specific size
is not specified.
dcl ackctstime Integer;
dcl cache TupleCache ;
dcl dup, myBss Boolean ;
dcl dNav, dPsp, dAck Duration ;
dcl endRx, strTs Time ;
dcl pdu Frame ;
dcl rxRate Rate ;
dcl src NavSrc ;
dcl wDefault KeyVector ;
dcl wExclude Boolean ;
dcl wKeyMap KeyMapArray ;
Filter_Idle
ResetMac
RxMpdu
(pdu,
endRx,
startTs,rxRate,
wExclude,wDefault,
wKeyMap)
/* This process filters valid received
frames by destination address, and
BssId for group destination addresses.
This process also maintains received
pdu counters and the tuple cache for
detecting duplicated unicast frames.
Data and management frames which
need acknowledgement cause a
NeedAck signal to Protocol_Control
as well as an RxMpdu to Defragment.
Piggybacked CfAcks cause RxCfack
signals, and CfPolls cause RxCfpoll
signals to Protocol_Control. If an
RxCfPoll is sent for a Data+CfPoll
or Data+CfPoll+CfAck, the NeedAck
has to reach TxCoord during the Sifs.
(The data frame report cannot serve
this purpose because the payload may
be a non-final fragment.)
Duration and Cfp duration remaining
are reported to Channel_State, and
power save mode is reported to Mlme. */
dAck:=
if (moreFrag
(pdu) = 1) and
(durID(pdu) > 32767)
then dUsec(durId(pdu))
else 0 fi
PsIndicate
(addr2(pdu),
pwrmgt(pdu))
Gather Power
management
info from all
valid frames.
dNav:=dUsec
(durId(pdu)),
src:= misc
import(
mActing_
AsAp)
(true)
(false)
ap_
addr
toDs(pdu)
(=1)
(=0)
AP, check
all frames, 2
pages ahead.
durId(pdu)
sta_
addr
Non-AP,
toDS=0 to
accept frame,
next page.
else
(1:32767)
SetNav
(endRx,
dNav, src)
Filter_Idle
Frames with toDs=1 ignored by non-APs,
except Duration/Id field for Nav update.


RSDU
TPDU
Block Protocol_Control_STA
sta_CTL_1c(1)
MsduIndicate
AtimW, PduConfirm, CfPolled
signal Ack(Time,Rate), Cfend, Cfpoll(Time,Rate), Cts(Time,Rate), TxCfAck(Time,Rate) ;
/* This block performs the DCF functions, as well as CF-responder functions if the station is CF-pollable. Tx_Coordination includes RTS and ATIM generation. Rx_Coordination generates acknowledgements, routes data frames to MAC data service and management frames to MLME, an indicates receipt of Ack, Cts, and CF-Poll frames to Tx_Coordination. */
Tdat
Rdat
Includes the CF responder if station is Cf-pollable.
PduRequest
Doze, MmCancel, SwChnl, Tbtt, Wake
Tmgt
PsmDone, SwDone
BkDone, TxConfirm
Tx_Coordination_sta (1,1) /*station version*/
PlmeGet_ .confirm, PlmeSet_ .confirm, Plme_ Reset_ .confirm, PlmeTxTime_ .confirm
Ack, Cts, Cfend, Cfpoll, TxCfAck
BcMgt
MmIndicate, SsInquiry
TxO
SsResponse
Backoff, Cancel, TxRequest
Pctl
Rctl
TxRx
Rx_Coordination (1,1) /* for STA and AP */
TX
Trsp
TxRequest
TxConfirm
PlmeGet_ .request, PlmeSet_ .request, PlmeReset_ .request, PlmeTxTime_ .request
RxIndicate, NeedAck, RxCfAck, RxCfPoll
RxI
ChangeNav
MLME_PLME_SAP
RX


Process Tx_Coordination_sta
sta_tx_atim_5e(9)
Atim_ Window
Ack, CfPoll, Cts, Doze, MmCancel, Tbtt, TxCfAck and Wake ignored in this state.
atim_ fail
atim_ ack
not import (mAtimW)
Pdu_ Request (fsdu)
PM_Filter ensures that a beacon frame will be the first first sent after Tbtt.
PduConfirm (fsdu, atimNak)
PduConfirm (fsdu, atimAck)
else
atimcw:=ccw, ccw:=dcfcw
ftype (fsdu!pdus(1))
tpdu:= mkFrame (atim,
atim_ limit
fsdu!fAnc:= fsdu!fCur+1
(>=0)
(beacon)
dcfcnt
Backoff (ccw, dcfcnt)
n:= call Random (2*aCWmin)
Backoff (ccw,-1)
PduConfirm (fsdu, retryLimit)
(<0)
cont:= false
cont:= true
Backoff (n,-1)
Ibss_Wait_ Atim
fsdu!dst, dot11Mac_ Address,)
Atim_ _Window
TxC_Idle
TxC_Backoff
Ibss_Wait_ _Beacon
not import (mAtimW)
BkDone (bstat)
*
BkDone (bstat)
MmCancel
atimcw:=ccw, ccw:=dcfcw
TxRequest (tpdu, mmrate)
TxRequest (fsdu!pdus(1), mmrate)
Cancel
TxC_Idle
(true)
fsdu! grpa
atim_ ack
(false)
Wait_Beacon_ _Transmit
Wait_Beacon_ _Cancel
dRsp:=dUsec(aSifsTime + calcDur(
txrate, ackctstime))
*
TxConfirm
BkDone(n)
*
set (now+dRsp, Trsp)
Atim_ _Window
Atim_ _Window
Wait_Atim_ Ack
import(dot11_ ShortRetry_ Limit)
ssrc =
(true)
(false)
Ack(,)
Trsp
ccw:= aCwMin
ccw:= aCWmin
then aCWmax else (2*ccw)+1 fi
ccw:= if ccw = aCWmax
import(dot11_ ShortRetry_ Limit)
fsdu!src =
(false)
atim_ ack
fsdu!src:= fsdu!src+1
ssrc:= ssrc+1,
atim_ fail
limit


Process Tx_Coordination_sta
sta_tx_idle_2e(9)
TxC_Idle
Ack, Cfend, Cts, Wake and MmCancel ignored in TxC_Idle state.
These transitions are only present at Cf-pollable stations.
Pdu_ Request (fsdu)
Entry when station wakes up to transmit.
CfPoll (endRx,)
import (mCfp)
TBTT
TxCfAck (endRx,)
not import(mBkIP)
txc_ req
rx_ poll
TxC_Cfp
dcfcnt:= -1
tpdu:= mkFrame(Cfack,
tpdu:= fsdu!pdus (fsdu!fCur)
*
Atw_Start
tx_ sifs
fsdu!eol
Test if fsdu seq nmber and tx lifetime set.
BkDone (dcfcnt)
import(mBssId), import(mBssId),)
else
(=0)
fsdu!sqf:= seqnum,
seqnum:=if seqnum=4095 then 0 else seqnum+1 fi, fsdu!eol:= now + import (dot11MaxTransmitMsduLifetime)
import (mIbss)
(false)
(true)
tpdu:= setSeq(tpdu, fsdu!sqf)
tpdu:= fsdu!pdus (fsdu!fCur)
send_ frag
-
dcfcw:=ccw, ccw:=atimcw
'txrate:= selected tx data rate'
See 9.6 for rules about selecting transmit data rate.
AtimW
TxTime(sAckCtsLng/8, txrate, ackctstime)
With FH PHY, if next fragment will be after a dwell boundary, Duration/ID may be set to one ACK time plus SIFS time.
Atim_ Window
TxTime(length(fsdu!pdus(fsdu!fCur+1)),
txrate,frametime)
tpdu:= setDurId(tpdu, calcDur(txrate,
aSifsTime+ackctstime+ if(fsdu!fTot=(fsdu!fCur+1)) then 0 else ((2*aSifsTime)+ ackctstime+frametime) fi))
((length (tpdu) +
sCrcLng) > import(dot11RtsThreshold)) and (not fsdu!grpa) and ((fsdu!fCur=0) or retry(tpdu) or (fsdu!resume))
(true)
tpdu:=setPwrMgt(tpdu, import(dot11PowerMan_ agementMode))
TxTime(length(tpdu),txrate, frametime2)
Backoff(0,0)
rtsdu:= mkctl(rts, (frametime2+
(3*aSifsTime)+ ackctstime))
(false)
send_ mpdu
send_ rts


Process Tx_Coordination_sta
sta_tx_ini
/* at start of frame exchange sequence, when setting mFxIP, check if dot11PowerManagementMode=curPsm, if not, when indicating the new Psm, also set psmChg boolean; at end of frame exchange sequence, when clearing FxIP, test & reset PsmChg, if true, send PsmDone to Mlme */
*
timer Tifs, Trsp, Tpdly ;
ResetMAC
exported TxTime
PlmeReset._ Request
dcl ackctstime, atimcw, bstat, chan, dcfcnt, dcfcw, frametime, frametime2 Integer ;
dcl ccw Integer:= aCwMin ;
dcl curPm Bit ;
dcl doHop, psmChg, cont Boolean:= false ;
dcl dSifsDelay, endRx Time ;
dcl fsdu FragSdu ;
dcl rtype Ftype ;
dcl seqnum, ssrc, slrc, n Integer:= 0;
dcl tpdu Frame ;
dcl txrate Rate ;
dSifsDelay:= dUsec (aSifsTime -
aRxTxTurn_ aroundTime)
'mmrate:= rate to send mmpdus'
Mmrate must be selected from mBrates. Other selection criteria are not specified.
ssrc:= 0, slrc:= 0
dcl exported FxIP Boolean:= false ;
dcl cTfrg exported as dot11TransmittedFragmentCount,
dcl cTfrm exported as dot11TransmittedFrameCount,
cTmcfrm exported as dot11MulticastTransmittedFrameCount,
cFail exported as dot11FailedCount,
cRtry exported as dot11RetryCount,
cMrtry exported as dot11MultipleRetryCount,
cCts exported as dot11RtsSuccessCount,
cNcts exported as dot11RtsFailureCount,
cNack exported as dot11AckFailureCount Counter32:= 0 ;
ccw:= import (aCWmin),
dcfcw:= ccw, atimcw:= ccw
Backoff (ccw,-1)
TxC_Idle
Imported dot11RtsThreshold, dot11ShortRetryLimit, dot11LongRetryLimit, dot11FragmentationThreshold, dot11MaxTransmitMsduLifetime Integer, mPdly Usec ;
/* RANDOM NUMBER FUNCTION */
imported procedure Random ;
fpar limit Integer ; returns Integer ;


Process Validate_MPDU
start_rx
*
(Rx_Idle)
/* This process receives an MPDU from the PHY while calculating and checking the FCS value. Frames with valid FCS, length and protocol version are sent for receive filtering, along with a snapshot of the WEP keys if dot11PrivacyOptionImplemented=true.
This process also provides Channel_State with Difs/Eifs and Rts timeout signals, and maintains the mRxA remote variable. */
Calculate PHY Rx delay that is subtracted from now to get reference point times.
D1:= dUsec (aRxRfDelay+ aRxPlcpDelay)
ResetMAC
reset(Trts)
cErr:=0, mRxA:=false
export (cErr,mRxA)
dcl exported mRxA Boolean:=false,
cErr as dot11FcsErrorCount Counter32:= 0 ;
imported mBrates Ratestring,
dot11WepDefaultKeys KeyVector,
dot11WepKeyMappings KeyMapArray,
dot11ExcludeUnencrypted Boolean ;
timer Trts ;
imported procedure TxTime; returns Integer;
Indicate Rts non-response timeout.
Rx_Idle
dcl fcs Crc ;
dcl D1, dRts Duration ;
dcl endRx, startTs Time ;
dcl ackctstime, k, rxLength Integer ;
dcl pdu Frame ;
dcl rxRate Rate ;
dcl status PhyRxStat ;
dcl v Octet ;
dcl wDefault KeyVector ;
dcl wKeyMap KeyMapArray ;
dcl wExclude Boolean ;
Trts
PhyRxStart. indication
(rxLength, rxRate)
RtsTimeout
reset(Trts)
Save copy of WEP keys at RxStart in case Mpdu is first fragment of encrypted Msdu/Mmpdu.
-
mRxA:=true
Indicate that a reception is in progress.
save_ keys
export(mRxA)
wDefault:= import(
dot11Wep_ DefaultKeys)
k:= 0, fcs:= initCrc, pdu:= null
Initialize CRC & clear pdu buffer (length(pdu)=0).
wKeyMap:= import(
dot11Wep_ KeyMappings)
dot11Privacy_
Option_ Implemented
wExclude:= import
(dot11Exclude_ Unencrypted)
(false)
(true)
Rx_Frame
save_ keys
Rx_Frame

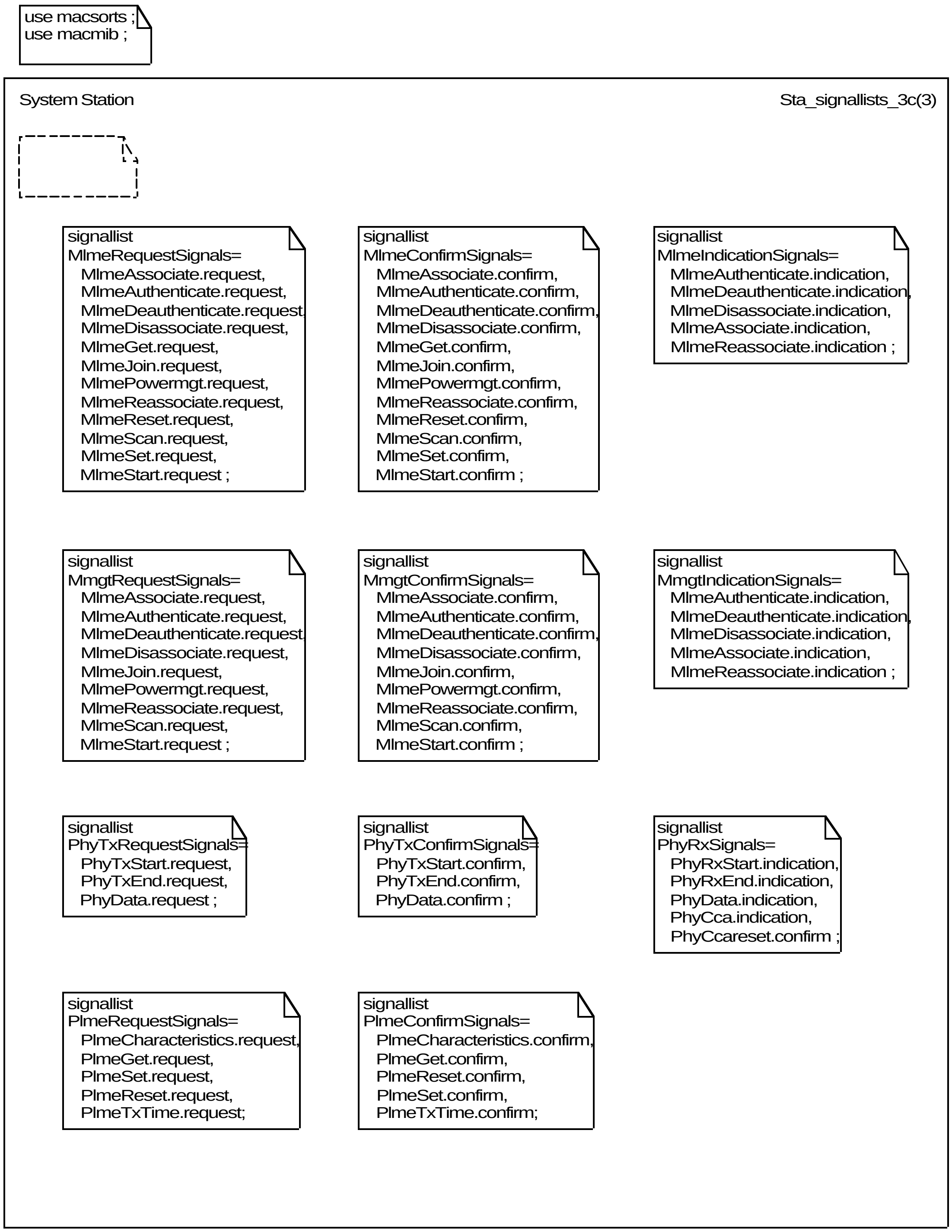
```
use macsorts ;
use macmib ;
```

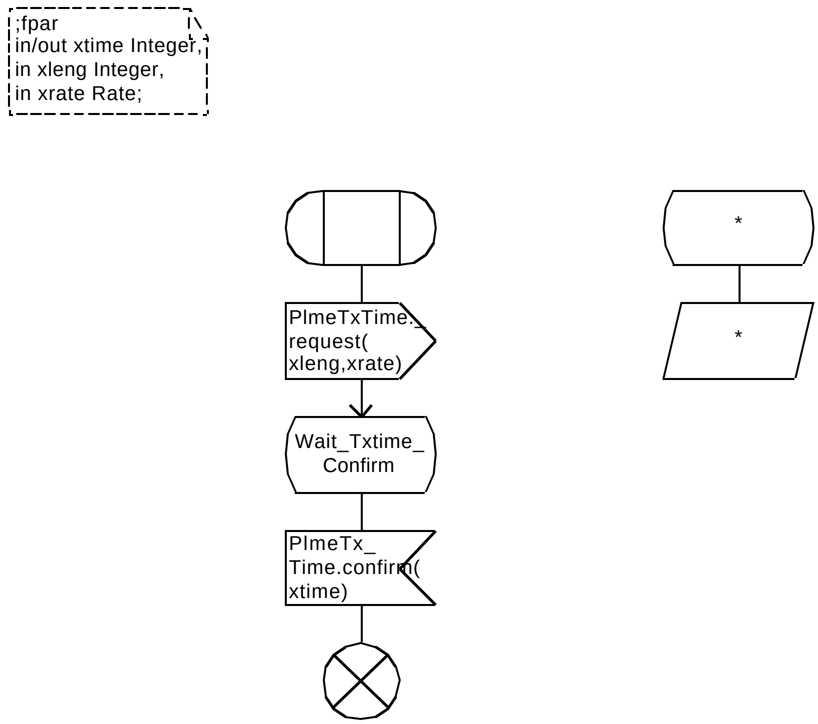
System Station

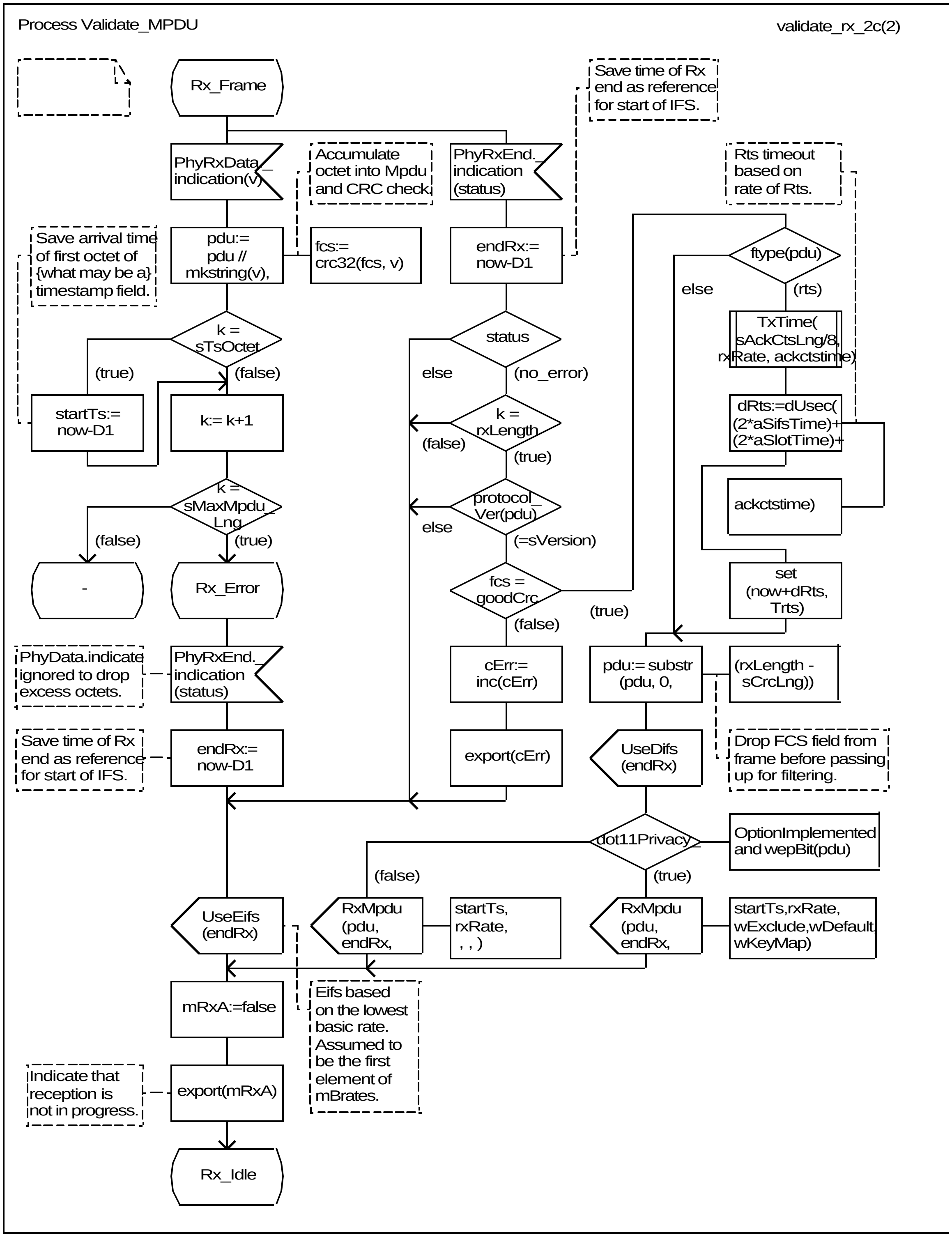
Sta_signals_2d(3)

```
signal
  AtimW,
  Backoff(Integer,Integer),
  BkDone(Integer),
  Busy,
  Cancel,
  CfPolled,
  ChangeNav(Time,Duration,NavSrc),
  Doze,
  Idle,
  MaUnitdata.indication(MacAddr,MacAddr,
    Routing,Octetstring,RxStatus,
    CfPriority,ServiceClass),
  MaUnitdata.request(MacAddr,MacAddr,
    Routing,Octetstring,CfPriority,ServiceClass),
  MaUnitdataStatus.indication(MacAddr,
    MacAddr,TxStatus,CfPriority,ServiceClass),
  MlmeAssociate.confirm(MlmeStatus),
  MlmeAssociate.indication(MacAddr),
  MlmeAssociate.request(MacAddr,Kusec,Capability,Integer),
  MlmeAuthenticate.confirm
    (MacAddr,AuthType,MlmeStatus),
  MlmeAuthenticate.indication(MacAddr,AuthType),
  MlmeAuthenticate.request(MacAddr,AuthType,Kusec),
  MlmeDeauthenticate.confirm(MacAddr,MlmeStatus),
  MlmeDeauthenticate.indication(MacAddr,ReasonCode),
  MlmeDeauthenticate.request(MacAddr,ReasonCode),
  MlmeDisassociate.confirm(MlmeStatus),
  MlmeDisassociate.indication(MacAddr,ReasonCode),
  MlmeDisassociate.request(MacAddr,ReasonCode),
  MlmeGet.confirm(MibStatus,MibAtrib,MibValue),
  MlmeGet.request(MibAtrib),
  MlmeJoin.confirm(MlmeStatus),
  MlmeJoin.request(BssDscr,Integer,Usec,Ratestring),
  MlmePowermgt.confirm(MlmeStatus),
  MlmePowermgt.request(PwrSave,Boolean,Boolean),
  MlmeReassociate.confirm(MlmeStatus),
  MlmeReassociate.indication(MacAddr),
  MlmeReassociate.request(MacAddr,Kusec,Capability,Integer),
  MlmeReset.confirm(MlmeStatus),
  MlmeReset.request(MacAddr,Boolean),
  MlmeScan.confirm(BssDscrSet,MlmeStatus),
  MlmeScan.request(BssTypeSet,MacAddr,Octetstring,
    ScanType,Usec,Intstring,Kusec,Kusec),
  MlmeSet.confirm(MibStatus,MibAtrib),
  MlmeSet.request(MibAtrib,MibValue),
  MlmeStart.confirm(MlmeStatus),
  MlmeStart.request(Octetstring,BssType,Kusec,
    Integer,CfParms,PhyParms,IbssParms,Usec,
    Capability,Ratestring,Ratestring) ;
```

```
signal
  MmCancel,
  MmConfirm(Frame,TxStatus),
  MmIndicate(Frame,Time,Time,StateErr),
  MmRequest(Frame,Imed,Rate),
  MsduConfirm(Frame,CfPriority,TxStatus),
  MsduIndicate(Frame,CfPriority),
  MsduRequest(Frame,CfPriority),
  NeedAck(MacAddr,Time,Duration,Rate),
  PduConfirm(FragSdu,TxResult),
  PduRequest(FragSdu),
  PhyCca.indication(Ccastatus),
  PhyCcarst.confirm,
  PhyCcarst.request,
  PhyData.confirm,
  PhyData.indication(Octet),
  PhyData.request(Octet),
  PhyRxEnd.indication(PhyRxStat),
  PhyRxStart.indication(Integer,Rate),
  PhyTxEnd.confirm,
  PhyTxEnd.request,
  PhyTxStart.confirm,
  PhyTxStart.request(Integer,Rate),
  PlmeCharacteristics.confirm(PhyChrstcs),
  PlmeCharacteristics.request,
  PlmeGet.confirm(MibStatus,
    MibAtrib,MibValue),
  PlmeGet.request(MibAtrib),
  PlmeReset.confirm(Boolean),
  PlmeReset.request,
  PlmeSet.confirm(MibStatus,MibAtrib),
  PlmeSet.request(MibAtrib,MibValue),
  PlmeTxTime.confirm(Integer),
  PlmeTxTime.request(Integer, Rate),
  PsmDone,
  PsChange(MacAddr,PsMode),
  PsIndicate(MacAddr,PsMode),
  PsInquiry(MacAddr),
  PsResponse(MacAddr,PsMode),
  ResetMAC,
  RxCfAck(MacAddr),
  RxIndicate(Frame,Time,Time,Rate),
  Slot,
  SsInquiry(MacAddr),
  SsResponse(MacAddr,
    StationState,StationState),
  SwChnl(Integer,Boolean),
  SwDone,
  TBTT,
  TxConfirm,
  TxRequest(Frame,Rate),
  Wake ;
```


use macsorts ;
use macmib ;
System Station
Sta_signallists_3c(3)
signallist
MlmeRequestSignals=
MlmeAssociate.request,
MlmeAuthenticate.request,
MlmeDeauthenticate.request,
MlmeDisassociate.request,
MlmeGet.request,
MlmeJoin.request,
MlmePowermgt.request,
MlmeReassociate.request,
MlmeReset.request,
MlmeScan.request,
MlmeSet.request,
MlmeStart.request ;
signallist
MlmeConfirmSignals=
MlmeAssociate.confirm,
MlmeAuthenticate.confirm,
MlmeDeauthenticate.confirm,
MlmeDisassociate.confirm,
MlmeGet.confirm,
MlmeJoin.confirm,
MlmePowermgt.confirm,
MlmeReassociate.confirm,
MlmeReset.confirm,
MlmeScan.confirm,
MlmeSet.confirm,
MlmeStart.confirm ;
signallist
MlmeIndicationSignals=
MlmeAuthenticate.indication,
MlmeDeauthenticate.indication,
MlmeDisassociate.indication,
MlmeAssociate.indication,
MlmeReassociate.indication ;
signallist
MmgtRequestSignals=
MlmeAssociate.request,
MlmeAuthenticate.request,
MlmeDeauthenticate.request,
MlmeDisassociate.request,
MlmeJoin.request,
MlmePowermgt.request,
MlmeReassociate.request,
MlmeScan.request,
MlmeStart.request ;
signallist
MmgtConfirmSignals=
MlmeAssociate.confirm,
MlmeAuthenticate.confirm,
MlmeDeauthenticate.confirm,
MlmeDisassociate.confirm,
MlmeJoin.confirm,
MlmePowermgt.confirm,
MlmeReassociate.confirm,
MlmeScan.confirm,
MlmeStart.confirm ;
signallist
MmgtIndicationSignals=
MlmeAuthenticate.indication,
MlmeDeauthenticate.indication,
MlmeDisassociate.indication,
MlmeAssociate.indication,
MlmeReassociate.indication ;
signallist
PhyTxRequestSignals=
PhyTxStart.request,
PhyTxEnd.request,
PhyData.request ;
signallist
PhyTxConfirmSignals=
PhyTxStart.confirm,
PhyTxEnd.confirm,
PhyData.confirm ;
signallist
PhyRxSignals=
PhyRxStart.indication,
PhyRxEnd.indication,
PhyData.indication,
PhyCca.indication,
PhyCcareset.confirm ;
signallist
PlmeRequestSignals=
PlmeCharacteristics.request,
PlmeGet.request,
PlmeSet.request,
PlmeReset.request,
PlmeTxTime.request;
signallist
PlmeConfirmSignals=
PlmeCharacteristics.confirm,
PlmeGet.confirm,
PlmeReset.confirm,
PlmeSet.confirm,
PlmeTxTime.confirm;


;fpar
in/out xtime Integer,
in xleng Integer,
in xrate Rate;
PlmeTxTime._
request(
xleng,xrate)
Wait_Txtime_
Confirm
PlmeTx_
Time.confirm(
xtime)
*
*


Process Validate_MPDU
validate_rx_2c(2)
Rx_Frame
Save time of Rx end as reference for start of IFS.
PhyRxData_ indication(v)
Accumulate octet into Mpdu and CRC check
PhyRxEnd._ indication (status)
Rts timeout based on rate of Rts.
Save arrival time of first octet of {what may be a} timestamp field.
pdu:= pdu // mkstring(v),
fcs:= crc32(fcs, v)
endRx:= now-D1
ftype(pdu)
else
(rts)
k = sTsOctet
(true)
(false)
status
else
(no_error)
TxTime(sAckCtsLng/8, rxRate, ackctstime)
startTs:= now-D1
k:= k+1
k = rxLength
(false)
(true)
dRts:=dUsec((2*aSifsTime)+ (2*aSlotTime)+
k = sMaxMpdu_ Lng
protocol_ Ver(pdu)
else
ackctstime)
(false)
(true)
(=sVersion)
-
Rx_Error
fcs = goodCrc
(true)
set (now+dRts, Trts)
(false)
PhyData.indicate ignored to drop excess octets.
PhyRxEnd._ indication (status)
cErr:= inc(cErr)
pdu:= substr (pdu, 0,
(rxLength - sCrcLng))
Save time of Rx end as reference for start of IFS.
endRx:= now-D1
export(cErr)
UseDifs (endRx)
Drop FCS field from frame before passing up for filtering.
dot11Privacy_
OptionImplemented and wepBit(pdu)
(false)
(true)
UseEifs (endRx)
RxMpdu (pdu, endRx,
startTs, rxRate, , ,)
RxMpdu (pdu, endRx,
startTs,rxRate, wExclude,wDefault, wKeyMap)
mRxA:=false
Eifs based on the lowest basic rate. Assumed to be the first element of mBrates.
Indicate that reception is not in progress.
export(mRxA)
Rx_Idle

Annex D: ASN.1 Encoding of the MAC and PHY Management Information Base (normative)

Add the following variables to the PHY MIB

```
dot11ShortPreambleOptionImplemented OBJECT-TYPE
SYNTAX INTEGER TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This attribute, when true, shall indicate that the short preamble option as defined in subclause 18.2.2.2 is implemented.
The default value of this attribute shall be false."
:= {dot11PhyHRDSSSEntry 1}


dot11PBCCOptionImplemented OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This attribute, when true, shall indicate that the PBCC modulation option as defined in subclause 18.4.6.6 is imple-
mented. The default value of this attribute shall be false."
:= {dot11PhyHRDSSSEntry 2}


dot11PhyOperationEntry:= SEQUENCE {
        dot11PhyOperationGroupTableIndex        Integer32,
        dot11PHYType                            INTEGER,
        dot11CurrentRegDomain                   Integer32,
        dot11CCATime                            Integer32,
        dot11MACProcessingDelay                 Integer32,
        dot11TempType                           INTEGER
        dot11PhyOperationGroupRowStatus         RowStatus,


dot11ChannelAgilityPresent OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
"This attribute indicates that the PHY is capable of channel agility."
::= { dot11PhyHRDSSSEntry 3}


dot11ChannelAgilityEnabled OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
"This attribute indicates that the PHY channel agility functionality is enabled."
:= {dot11PhyHRDSSSEntry 4 }


dot11HRCCAModeSupported OBJECT-TYPE
        SYNTAX INTEGER (1..31)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "dot11HRCCAModeSupported is a bit-significant value, representing
        all of
```

```
        the CCA modes supported by the PHY. Valid values are:
            energy detect only (ED_ONLY) = 01,
            carrier sense only (CS_ONLY) = 02,
            carrier sense and energy detect (ED_and_CS)= 04
            carrier sense with timer (CS_and_Timer)= 08
            high rate carrier sense and energy detect (HRCS_and_ED)= 16
        or the logical sum of any of these values.  In the high rate extension
        PHY, this attribute shall be used in preference to the
        dot11CCAModeSupported attribute."
::= { dot11PhyHRDSSSEntry 5 }
```

Add a new compliance group to the compliance statements just before the section:"OPTIONAL-GROUPS":

```
GROUP dot11PhyHRDSSSComplianceGroup
    DESCRIPTION
        "Implementation of this group is required when object
        dot11PHYType has the value of hrdsss. This group is
        mutually exclusive with the groups dot11PhyDSSSComplianceGroup,
        dot11PhyIRComplianceGroup and dot11PhyFHSSComplianceGroup."
```

Change the following attribute definition (as it was previously modified by 802.11a):

```
dot11PHYType OBJECT-TYPE
SYNTAX INTEGER {fhss(1), dsss(2), irbaseband(3), ofdm(4), hrdsss(5)}
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "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,
        OFDM 5GHz = 04, HRDSSS = 05 "
::= { dot11PhyOperationEntry 1 }
```

Modify the following compliance group in the compliance statements:

```
dot11CCAModeSupported OBJECT-TYPE
    SYNTAX INTEGER (1..7)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "dot11CCAModeSupported is a bit-significant value, representing all
    of
        the CCA modes supported by the PHY. Valid values are:
            energy detect only (ED_ONLY) = 01,
            carrier sense only (CS_ONLY) = 02,
            carrier sense and energy detect (ED_and_CS)= 04
```

~~carrier sense with timer (CS_and_Timer)= 08~~
~~high rate carrier sense and energy detect (HRCS_and_ED)= 16~~

```
        or the arithmetic sum of any of these values.  This attribute shall
        not be used to indicate the CCA modes supported by a higher rate
        extension PHY.  Rather, the dot11HRCCAModeSupported attribute shall
        be used to indicate the CCA modes of the higher rate extension PHY."
::= { dot11PhyDSSSEntry 2 }

dot11CurrentCCAMode OBJECT-TYPE
    SYNTAX INTEGER {edonly(1), csonly(2), edandcs(4), cswithtimer(8),
                    hrcsanded(16) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The current CCA method in operation. Valid values are:
            energy detect only (edonly) = 01,
            carrier sense only (csonly) = 02,
            carrier sense and energy detect (edandcs)= 04.
            carrier sense with timer (cswithtimer)= 08
            high rate carrier sense and energy detect (hrcsanded)= 16"
::= { dot11PhyDSSSEntry 3 }
```

Insert the following into the 802.11 MIB in Annex D, between the section entitled: "conformance information" and the section entitled: "End of dot11SupportedDataRatesRx TABLE":

```
-- **********************************************************************

-- * dot11PhyHRDSSSEntry TABLE

-- **********************************************************************

dot11PhyHRDSSSTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyHRDSSSEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Entry of attributes for dot11PhyHRDSSSEntry. Implemented as a table indexed on ifIndex to allow
        for multiple instances on an Agent."
    ::= { dot11phy 12 }

dot11PhyHRDSSSEntry OBJECT-TYPE
    SYNTAX Dot11PhyHRDSSSEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PhyHRDSSSEntry Table. ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed by ifIndex."
    INDEX {ifIndex}
::= { dot11PhyHRDSSSTable 1 }

Dot11PhyHRDSSSEntry ::= SEQUENCE {
    dot11ShortPreambleOptionImplemented  TruthValue,
    dot11PBCCOptionImplemented TruthValue,
    dot11ChannelAgilityPresent TruthValue,
    dot11ChannelAgilityEnabled TruthValue,
```

```
    dot11HRCCAModeSupported INTEGER }


-- **********************************************************************
-- * End of dot11PhyHRDSSSEntry TABLE
-- **********************************************************************
```

Insert the following text into the 802.11 MIB in Annex D, after the definition of the SMTBase2 Object Group:

```
dot11PhyHRDSSSComplianceGroup OBJECT-GROUP
    OBJECTS {dot11CurrentChannel,
        dot11CCAModeSupported,
        dot11CurrentCCAMode,
        dot11EDThreshold,
        dot11ShortPreambleOptionImplemented,
        dot11PBCCOptionImplemented,
        dot11ChannelAgilityPresent,
        dot11ChannelAgilityEnabled,
        dot11HRCCAModeSupported }
    STATUS current
    DESCRIPTION
        "Attributes that configure the HRDSSS PHY for IEEE 802.11. Implementation of this group is
        required when object dot11PHYType has the value of hrdsss. This group is mutually exclusive with
        the groups dot11PhyDSSSComplianceGroup, dot11PhyIRComplianceGroup and
        dot11PhyFHSSComplianceGroup."
::= { dot11Groups 19 }
```

Insert new annex.

Annex F - High Rate PHY / frequency hopping interoperability (Informative)

The channel agility option described in 18.4.6.7 provides for 802.11 FH PHY interoperability with the High Rate PHY. The frequency hopping patterns as defined within this annex enable synchronization with an FH PHY compliant BSS in North America and most of Europe. In addition, CCA requirements on a High Rate station using this mode provides for CCA detection of 1 MHz wide FH signals within the wideband DS channel selected. FH PHY stations operating in mixed mode FH/DS environments are advised to use similar cross PHY CCA mechanisms. The frequency hopping (channel agility) and cross CCA mechanisms provide the basic mechanisms to enable coexistence and interoperability.

The MAC elements include both DS and FH elements in beacons and probe responses when the channel agility option is turned on. Added capability fields indicate the ability to support the channel agility option and to indicate whether the option is turned on. These fields allow synchronization to the hopping sequence and timing, identification of what modes are being used within a BSS when joining on either High Rate or FHSS sides, and rejection of an association request in some cases.

Interoperability within an infrastructure BSS can be achieved, as an example, using a virtual dual Access Point (AP). A virtual dual AP is defined, for purposes of discussion, as two logically separate APs that exist within a single physical AP with a single radio (one transmit and one receive path). Both FHSS and High Rate logical APs send out their own beacons and DTIMs and other non-directed packets. The two sides interact in the sharing of the medium and the AP’s processor and radio. Addressing and association issues may be handled in one of several ways and are left as an implementation choice.

Minimal interoperability with a non-hopping High Rate or legacy DSSS is provided by the use of a channel at least 1/7 or more of the time. While throughput would be significantly reduced by having a channel only 1/7 of the time, connection and minimal throughput can be provided.

F.1 Additional CCA Recomendations

When the frequency hopping option is utilized, the HR/DSSS PHY should provide the CCA capability to detect 1 MHz wide FH PHY signals operating within the wideband DS channel at levels 10 dB higher than that specified in subclause 18.4.8.4 for wideband HR/DSSS signals. This is in addition to the primary CCA requirements in subclause 18.4.8.4. A timeout mechanism to avoid excessive deferral to constant CW or other non-802.11 type signals is allowed.

FH PHY stations operating in mixed environments should provide similar CCA mechanisms to detect wideband DSSS signals at levels specified in clause 1.4.8.4 but measured within a 1 MHz bandwidth. Signal levels measured in a full DSSS channel will be generally 10 dB or more higher.