
**IEEE P802.11
Wireless LANs**

**Proposal for
Packet Binary Convolutional Codes (PBCC)
As an Optional Forward Error Control (FEC)
For the 2.4 GHz High-Speed PHY**

Date: September 9, 1998

Author: Chris Heegard, Matthew B. Shoemake, and Stanley Ling



heegard@alantro.com, shoemake@alantro.com, sling@alantro.com

Abstract

The 64-state binary convolutional code, presented in the Alantro proposal for the 2.4 GHz high-speed PHY, provides better performance in terms of SNR and multipath rejection than the CCK codes specified in the current draft of the 802.11b standard. We have demonstrated the practicality and usefulness of these codes in our previous presentations. The performance improvements possible with these codes is achieved at the cost of increased receiver complexity.

We propose that the high-speed PHY standard allow the use of a modulation scheme based on these convolutional codes (PBCC) as an option. The proposed mechanism for using these codes provides full interoperability between STAs that support the optional codes and ones that do not. Inclusion of these codes as an option would allow the high-speed PHY to be used in a greater range of applications than would be possible with only the CCK modulation.

Two methods for providing support for the use of optional codes is presented in this paper. The first method uses the encoding of the rate information in the MAC to specify not only the data rate but also the code used for transmitting the data. The second method requires the addition of an information element called "Supported Codes" to frames that contain the "Supported Rates" element.

Each of the methods presented has advantages and disadvantages that are discussed below. Our recommendation is that method 1 be adopted as the mechanism for supporting optional codes for the high-speed PHY.

1 Introduction

The 64-state binary convolutional codes, presented in the Alantro proposal for the 2.4 GHz high-speed PHY [98/82], provide better performance in terms of SNR and multipath rejection than the CCK codes specified in the current draft of the 802.11b standard. We have demonstrated the usefulness of these codes in our previous presentations [98/82, 98/84, 98/85, 98/186]. The performance improvements possible with this coding scheme are again presented here in section 3. System diagrams are provided to enable full verification of our results by interested parties.

The system proposed here, which includes binary convolutional codes with scrambling in a packet-based system, shall be referred to as packet binary convolutional coding or PBCC. The increase in performance that can be achieved by PBCC makes it an ideal forward error control (FEC) solution for the high-speed PHY. We propose that the high-speed PHY standard allow the use of a modulation scheme based on PBCC as an option.

Two methods for providing support for the use of optional codes are presented here. The first method uses the rate information encoding in the MAC to specify not only the data rate but also the code used for transmitting the data. The second method requires the addition of an information element called "Supported Codes" to frames that contain the "Supported Rates" element. The proposed mechanism for switching between codes is similar to the rate switching mechanism currently in the standard. Both methods provide full interoperability between STAs that support the optional codes and ones that do not. A detailed description of the first method and the changes required to the current draft are presented in section XX. The second method and the changes required to support it are presented in section YY.

In addition to support for optional codes the current DS PLCP header has to be modified to support rates higher than 8 Mb/s. We propose that four service bits be defined in the PLCP header for the high-speed PHY. The "Byte Boundary" bits (bit 0-2 of the service field) be defined to allow the exact length (in bytes) of the MPDU to be determined from the length field which is specified in microseconds. The length of the frame will be the maximum number of bytes that would fit into the time specified in the length field minus the number encoded in these bits. Using three bits allows data rates up to 64 Mb/s to be supported with the current DS PLCP header. Bit 3 called the "Code" bit, specifies the code used to transmit the frame. A value of 0 indicates that the mandatory code is in use, and a value of 1 indicates that the optional code is in use. This approach allows transparent backward compatibility with the current DS PHY.

2 Binary Convolutional Coding Scheme Description

This coding scheme uses a binary convolutional coding scheme with a 64-state binary convolutional code and a cover sequence. The output of the BCC is encoded jointly onto the I and Q channels, as further documented in Section 2.1. This provides enhanced multi-path performance and reduced complexity over using two generators and encoding the I and Q channel independently. The cover sequence also provides added multi-path immunity.

The encoder for this scheme is shown in Figure 2.1. Incoming data is first encoded with a binary convolutional code that is well suited for difficult channels such as wireless communications channels. The encoded data is then scrambled before transmission through the channel.

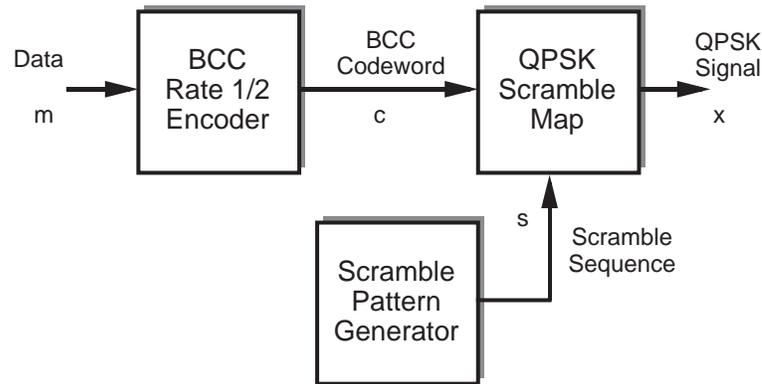


Figure 2.1

2.1 - Binary Convolutional Code

The binary convolutional code that is used is a 64-state, rate $\frac{1}{2}$ 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 $G = [133, 175]$. This code provides a good trade-off between additive white Gaussian noise (AWGN) performance and performance in multi-path environments.

Since the system is packet based, the encoder is defined as being in state zero, i.e. all memory elements contain zero, at the beginning of each packet. The encoder must also be placed in a known state at the end of each packet to prevent the data bits near the end of the packet from being substantially less reliable than those early on in the packet. To place the encoder in a known state at the end of a packet, six deterministic bits are input immediately following the last data bit input to the convolutional encoder. These bits are all zero, which places the encoder in the zero state.

An encoder block diagram is shown in Figure 2.1.1. It consists of six memory elements. For every data bit input, two output bits are generated.

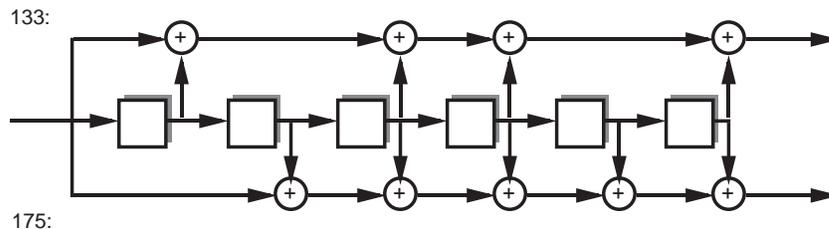


Figure 2.1.1

2.2 – Pseudo-random Cover Sequence

The output of the binary convolutional code described in Section 2.1 is mapped a constellation using one of two possible modes. One mode uses BPSK and the other 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 and used to produce two PSK symbols. This yields a throughput of one bit per symbol in QPSK mode and one-half a bit per second in BPSK mode.

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 2.2.1.

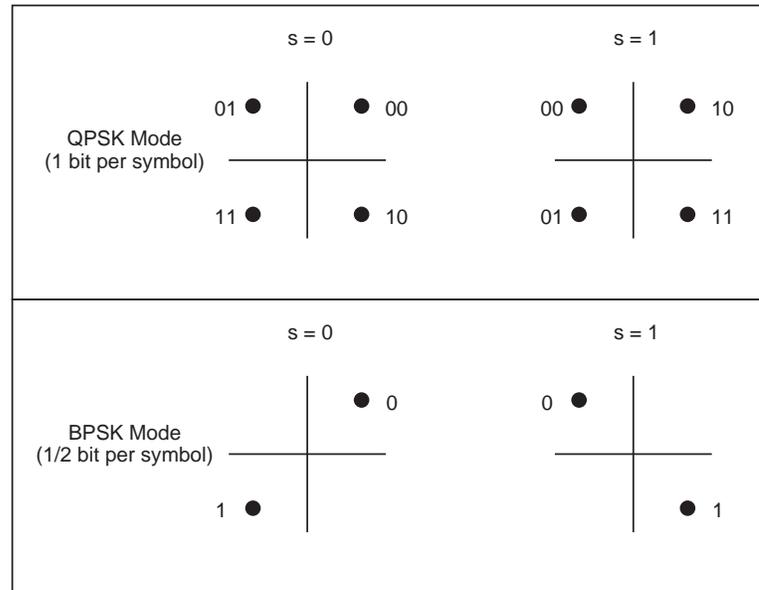


Figure 2.2.1

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 2.2.1.

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 c_i is the i th bit of the seed sequence, where $0 \leq i \leq 15$, then the sequence that is used to scramble 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
...
c10 c11 c12 c13 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 packet based systems with more than 256 bits and continuous systems this sequence of 256 bits is simply repeated.

3 Performance

The performance of PBCC is shown in Figures 2.1 and 2.2. The performance with no multipath interference is shown as the curve with T_{rms} of zero. The multipath performance is shown for T_{rms} up to 300 ns. It is seen that this scheme tolerates significant multipath distortion without significantly lose in decibels. The performance is not monotonic with T_{rms} due to the fact that the signal to noise ratio for the multipath channel specified for use by TGB is a random variable with lower variance for larger T_{rms} .

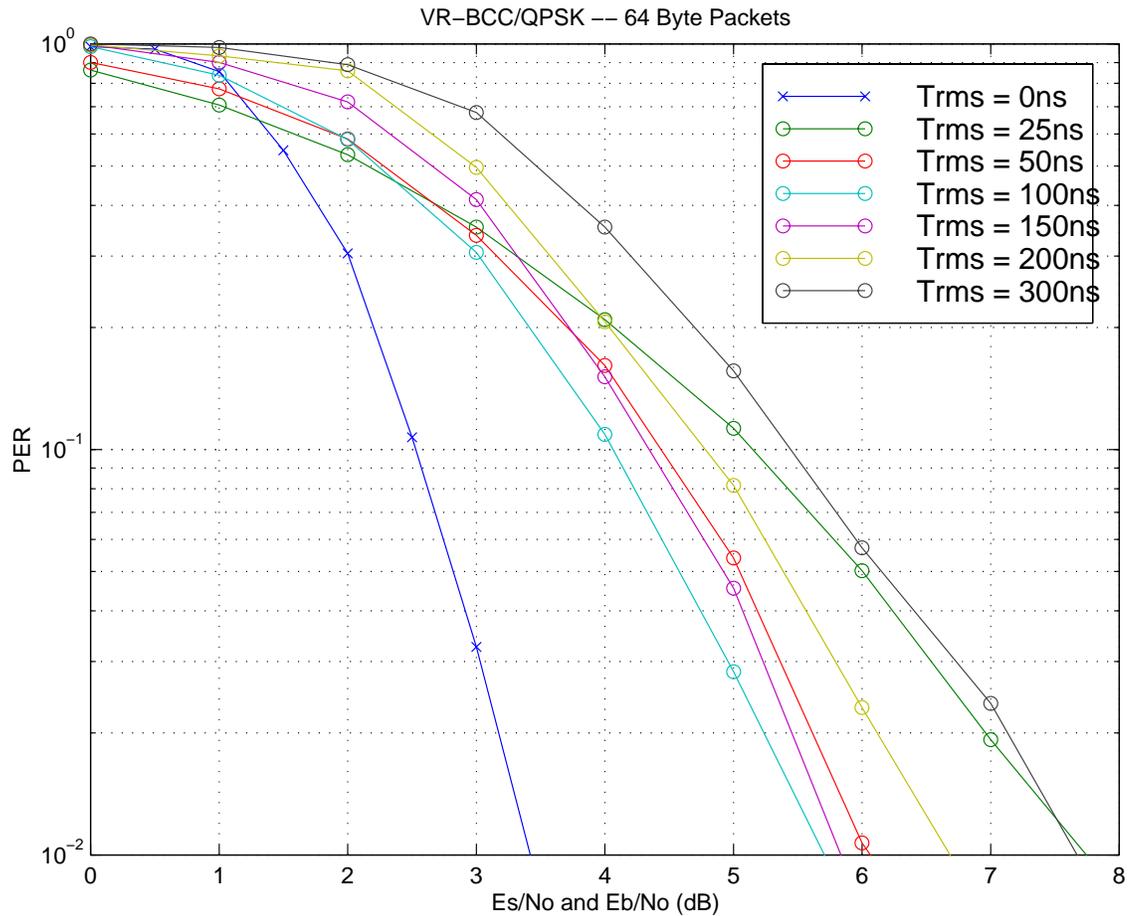


Figure 2.1

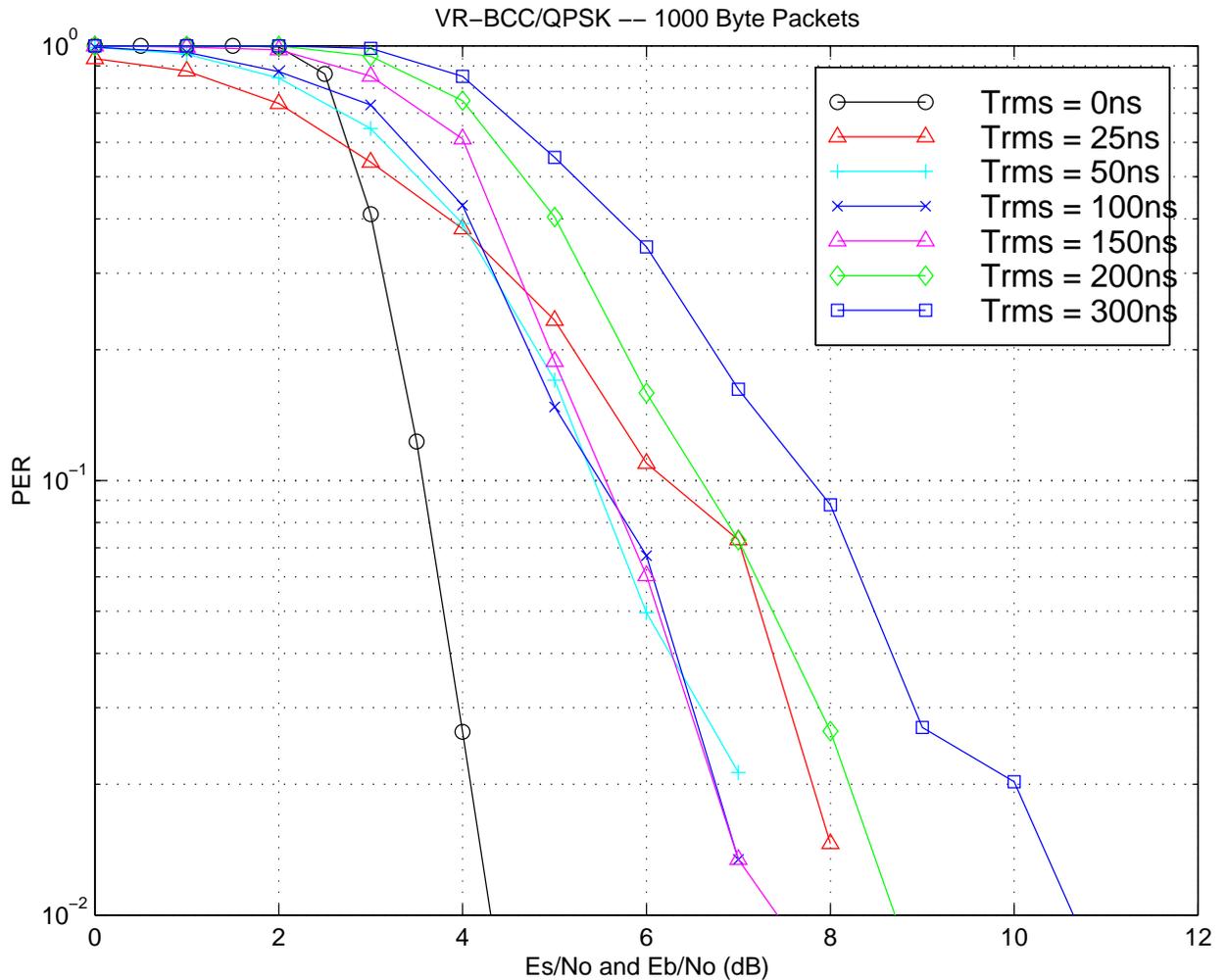


Figure 2.2

4 Method 1

Support for optional codes can be added to the 802.11 standard by interpreting the encoding of rate information differently than is currently specified in the standard. The current standard specifies that each supported rate is encoded as an octet such that the value in the octet represents a data rate in units of 500 Kb/s. Additional values need to be defined for the high-speed PHY. We propose that the values 0x0b and 0x16 be used to indicate 5.5 and 11 Mb/s respectively using the CCK codes, and 0x0c and 0x18 be used to indicate 5.5 and 11 Mb/s respectively using PBCC.

It should be noted that there is already a precedent in the current standard to allow the actual data rate to differ from the encoded "data rate". In the FH PHY a value of 0x02 represents an actual data rate of 0.97 Mb/s, and a value of 0x04 represents an actual data rate of 1.94 Mb/s. The difference in encoded rates and actual rates is due to the bit stuffing algorithm used by the FH PHY. The PHY parameter MPDUDurationFactor (1.03125 for the FH PHY) compensates for this difference when the encoded rate is used for duration calculation. Our proposal causes the MPDUDurationFactor to be set to 1.1 (11/10) whenever a frame is being received or transmitted using PBCC.

In addition to using the rate information in the MAC for duration calculation this information is passed across the MAC/PHY interface in the RX_VECTOR and TX_VECTOR parameters. The code information can be passed across this interface by transferring the encoded octet and allowing the PHY to translate it to the appropriate PLCP header bits.

4.1 Changes to the 802.11 Standard Document

The following is a list of changes required to the current 802.11 standard document to support encoding the data rate and code information into one octet. Changes are also needed to the state machines to incorporate the text changes described here.

The black text is the original text from the standard document (will need to be updated to incorporate changes by Tgrev). The text marked with revision marks depicts the changes required to support the new rate encoding mechanism.

3.8 basic service set (BSS) basic rate set:

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

7.3.2.2 Supported Rates element

The Supported Rates element specifies the rates/codes in the Operational Rate Set 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.

rate/code. For the FH, DS and IR PHYs a 1 Mb/s rate is encoded as X'02' and a 2 Mb/s rate is encoded as X'04'. All other encodings are PHY specific and are defined in the related clauses. Within Beacon, Probe Response, Association Response, and Reassociation Response management frames, each supported rate/code belonging to the BSSBasicRateSet as defined in 10.3.10.1, is encoded as an octet with the msb (bit 7) set to 1 (e.g., a 1 Mbit/s rate belonging to the BSSBasicRateSet is encoded as X'82'). Rates/codes not belonging to the BSSBasicRateSet are encoded with the msb set to 0 (e.g., a 2 Mbit/s rate not belonging to the BSSBasicRate Set is encoded as X'04'). The msb of each Supported Rate octet in other management frame types is ignored by receiving STAs.

BSSBasicRateSet information in Beacon and Probe Response management frames is used by STAs in order to avoid associating with a BSS if they do not support all the data rates/codes in the BSSBasicRateSet. See Figure 36.

9.2 Distributed Coordination Function

-
-
-

The medium access protocol allows for stations to support different sets of data rates. ~~All STAs shall receive all the data rates in the aBasicRateSet and transmit at one or more of the aBasicRateSet data rates.~~ rates/codes. All STAs shall be able to receive and transmit all the data rates/codes in the aBSSBasicRateSet. 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 aBasicRateSet rates. ~~aBSSBasicRateSet rates/codes.~~ (See Clause 9.6 for a description of multirate/multicode operation).

Data frames sent under the DCF shall use the Frame Type Data and Subtype Data or Null Function. Stations receiving Data Type frames shall only consider the frame body as the basis of a possible indication to LLC.

9.6 Multirate/Multicode support

Some PHYs have multiple data transfer rate/code capabilities that allow implementations to perform dynamic rate/code switching with the objective of improving performance. The algorithm for performing rate/code switching is beyond the scope of this standard, but in order to ensure coexistence and interoperability on multirate-capable/multirate/multicode 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/codes in the BSSBasicRateSet (see 10.3.10.1), ~~or at one of the rates in the PHY mandatory rate set so~~10.3.10.1) 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 rates/codes in the BSSBasicRateSet, regardless of their type or subtype.

Data and/or management MPDUs with a unicast immediate address shall be sent ~~on any supported data rate selected by the rate switching mechanism (whose output is an internal MAC variable called MACCurrentRate, defined in units of 500 kbit/s, which is used for calculating the Duration/ID field of each frame) at the rate/code specified by MACCurrentRate. MACCurrentRate is an internal MAC variables whose values are selected by the rate/code switching mechanism from the ones specified in the OperationalRateSet, a parameter of the MLME-JOIN.request primitive.~~ A STA shall not transmit at a rate/code 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~~Under no circumstances shall a STA initiate transmission of a data or management frame at a data rate higher than the greatest rate in the OperationalRateSet, a parameter of the MLME-JOIN.request primitive.

rate/code used 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 allow the transmitting STA to calculate the contents of the Duration/ID field, the responding STA shall transmit its Control Response frame (either CTS or ACK) at ~~the same rate as a rate/code which has the highest encoded value in the BSSBasicRateSet that is less than or equal to the encoded value of the rate/code of 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 9.7).~~ The time required to transmit a frame, for use in the Duration/ID field, can be calculated using the following equation:

~~the BSSBasicRateSet.~~

$$\text{Frame duration} = \text{aPreambleLength} + \text{aPLCPHeaderLength} + \frac{8 \times \text{MPDU length} \times \text{duration factor}}{(\text{MACCurrentRate} \times 32768)}$$

Where MACCurrentRate is the encoded value, from the OperationalRateSet, of the rate and code used to transmit this frame.

Where the duration factor is the value in the MPDUDurationFactor set corresponding to the MACCurrentRate value from the OperationalRateSet.

10.3.2.2.2 Semantics of the service primitive

-
-
-

Each BSSDescription consists of the following elements:

| Name | Type | Valid Range | Description |
|---------|--------------|-----------------------------|----------------------------|
| BSSID | MACAddress | N/A | The BSSID of the found BSS |
| SSID | octet string | 1 - 32 octets | The SSID of the found BSS |
| BSSType | Enumeration | INFRASTRUCTURE, INDEPENDENT | The type of the found BSS |

| | | | |
|---------------------------------|---------------------------------|---|---|
| Beacon Period | integer | N/A | The Beacon period of the found BSS (in Kμs) |
| DTIM Period | integer | As defined in Frame Format | The DTIM Period of the BSS (in Beacon Periods) |
| Timestamp | integer | N/A | The timestamp of the received frame (probe response/beacon) from the found BSS |
| Local Time | integer | N/A | The value of the station's TSF timer at the start of reception of the first octet of the timestamp field of the received frame (probe response or beacon) from the found BSS. |
| PHY parameter set | As defined in Frame Format | As defined in Frame Format | The parameter set relevant to the PHY |
| CF parameter set | As defined in Frame Format | As defined in Frame Format | The parameter set for the CF periods, if found BSS supports CF mode. |
| IBSS parameter set | As defined in Frame Format | As defined in Frame Format | The parameter set for the IBSS, if found BSS is an IBSS. |
| CapabilityInformation | As defined in Frame Format | As defined in Frame Format | The advertised capabilities of the BSS. |
| BSSBasicRateSet | set of integers | 1 through 127 inclusive (for each integer in the set) | The set of data rates (in units of 500kbit/s) that must be supported by all STAs that desire to join this BSS. The STAs must be able to receive at each of the data rates listed in the set. |
| BSSBasicRateSet | set of integers | 1 through 127 inclusive (for each integer in the set) | The set of data rates/codes 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/codes listed in the set. For the FH, DS and IR PHYs a 1 Mb/s rate is encoded as X'02' and a 2 Mb/s rate is encoded as X'04'. All other encodings are PHY specific and are defined in the related clauses. |

10.3.3.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```

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

```

| Name | Type | Valid Range | Description |
|----------------|----------------|-------------|--|
| BSSDescription | BSSDescription | N/A | The BSSDescription of the BSS to join. The BSSDescription is a member of the set of descriptions that was returned as a result of a MLME-SCAN.request. |

| | | | |
|--------------------|-----------------|---|--|
| JoinFailureTimeout | integer | 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 μ s) 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 is a superset of the BSSBasicRateSet advertised by the BSS. |
| OperationalRateSet | set of integers | 1 through 127 inclusive (for each integer in the set) | The set of data rates/codes that the STA desires to use for communication within the BSS. The STA must be able to receive at each of the data rates/codes listed in the set. The OperationalRateSet is a superset of the BSSBasicRateSet advertised by the BSS. For the FH, DS and IR PHYs a 1 Mb/s rate is encoded as X'02' and a 2 Mb/s rate is encoded as X'04'. All other encodings are PHY specific and are defined in the related clauses |

10.3.10.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```

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

| Name | Type | Valid Range | Description |
|-------------------|----------------------------|------------------------------|---|
| SSID | octet string | 1 - 32 octets | The SSID of the BSS. |
| BSSType | Enumeration | INFRA-STRUCTURE, INDEPENDENT | The type of the BSS. |
| Beacon Period | integer | greater than or equal to 1 | The Beacon period of the BSS (in K μ s). |
| DTIM Period | integer | As defined in Frame Format | The DTIM Period of the BSS (in Beacon Periods) |
| CF parameter set | As defined in Frame Format | As defined in Frame Format | The parameter set for CF periods, if the BSS supports CF mode. aCFPPeriod is modified as a side effect of the issuance of a MLME-START.request primitive. |
| PHY parameter set | As defined in | As defined in | The parameter set relevant to the PHY. |

| | Frame Format | Frame Format | |
|------------------------------------|---------------------------------|---|--|
| IBSS parameter set | As defined in Frame Format | As defined in Frame Format | The parameter set for the IBSS, if BSS is an IBSS. |
| ProbeDelay | integer | N/A | Delay (in μ s) to be used prior to transmitting a Probe frame during active scanning |
| CapabilityInformation | As defined in Frame Format | As defined in Frame Format | The capabilities to be advertised for the BSS. |
| BSSBasicRateSet | set of integers | 1 through 127 inclusive (for each integer in the set) | The set of data rates (in units of 500 kbit/s) that must be supported by all STAs that desire to join this BSS. The STA that is creating the BSS must be able to receive at each of the data rates listed in the set. |
| BSSBasicRateSet | set of integers | 1 through 127 inclusive (for each integer in the set) | The set of data rates/codes 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/codes listed in the set. For the FH, DS and IR PHYs a 1 Mb/s rate is encoded as X'02' and a 2 Mb/s rate is encoded as X'04'. All other encodings are PHY specific and are defined in the related clauses |
| 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 is a superset of the BSSBasicRateSet advertised by the BSS. |
| OperationalRateSet | set of integers | 1 through 127 inclusive (for each integer in the set) | The set of data rates/codes that the STA desires to use for communication within the BSS. The STA must be able to receive at each of the data rates/codes listed in the set. The OperationalRateSet is a superset of the BSSBasicRateSet advertised by the BSS. For the FH, DS and IR PHYs a 1 Mb/s rate is encoded as X'02' and a 2 Mb/s rate is encoded as X'04'. All other encodings are PHY specific and are defined in the related clauses |

10.4.3.2 Semantics of the service primitive

The primitive provides the following parameters:

```

PLME-CHARACTERISTICS.confirm(
    aSlotTime,
    aSIFSTime,
    aCCATime,
    aRxTxTurnaroundTime,
    aTxPLCPDelay,
    aRxPLCPDelay,
    aRxTxSwitchTime,
    aTxRampOnTime,
    aTxRampOffTime,
    aTxRRFDelay,
    aRxRRFDelay,

```

aAirPropagationTime,
 aMACProcessingDelay,
 aPreambleLength,
 aPLCPHeaderLength,
 aMPDUDurationFactor,
 aMPDUMaxLength,
 aCWmin,
 aCWmax
)

-
-
-

| | | | |
|---|--|---------------------------------------|--|
| <p>MPDUDurationFactor</p> | <p>Integer for FH, DS and IR PHYs. A set of integer pairs.</p> | <p>PHY dependent.</p> | <p>For backward compatibility this field is an integer for the FH, DS and IR PHYs. For all other PHYs this is a set of integer pairs where each pair contains a rate/code encoding, and a corresponding duration factor which is used to calculate the duration of a frame given the rate/code used to transmit the frame. The duration factor is normalized such that for the DS PHY its value is 65536.</p> <p>The time required to transmit a frame, for use in the Duration/ID field, can be calculated using the following equation:</p> $\text{Frame duration} = \frac{\text{aPreambleLength} + \text{aPLCPHeaderLength} + (8 \times \text{MPDU length} \times \text{duration factor})}{(\text{MACCurrentRate} \times 32768)}$ <p>Where MACCurrentRate is the encoded value, from the OperationalRateSet, of the rate and code used to transmit this frame.</p> <p>Where the duration factor is the value in the MPDUDurationFactor set corresponding to the MACCurrentRate value from the OperationalRateSet.</p> |
|---|--|---------------------------------------|--|

-
-
-

12.3.4.4 Vector Descriptions

Several service primitives include a parameter vector. This vector is a list of parameters which may vary depending on the PHY type. The table below lists the parameter values required by the MAC or PHY in each of the parameter vectors. Parameters in the vectors which are Management rather than medium access control may be specific to the PHY and are listed in the clause covering that PHY.

| Parameter | Associate Vector | Value |
|-----------|------------------|-------|
|-----------|------------------|-------|

| | | |
|----------|--------------------|--|
| DATARATE | TXVECTOR, RXVECTOR | PHY dependent. The name of the field used to specify the Tx data rate and report the Rx data rate may vary for different PHYs. |
| Code | TXVECTOR, RXVECTOR | PHY dependent. The name of the field used to specify the Tx code and report the Rx code may vary for different PHYs. |
| LENGTH | TXVECTOR, RXVECTOR | PHY dependent |

Table xx, Vector Descriptions

4.2 Text Changes for the PHY Clause

The following changes to the PHY clause for the high-speed PHY are based on the description of the DS PHY in the current standard.

15.2.3.3 PLCP 802.11 Signal Field (SIGNAL)

The 8 bit 802.11 Signal Field indicates to the PHY the modulation which shall be used for transmission (and reception) of the MPDU. The data rate shall be equal to the Signal Field value multiplied by 100kbit/s. The DSSS PHY currently supports two mandatory modulation services given by the following 8 bit words, where the LSB shall be transmitted first in time:

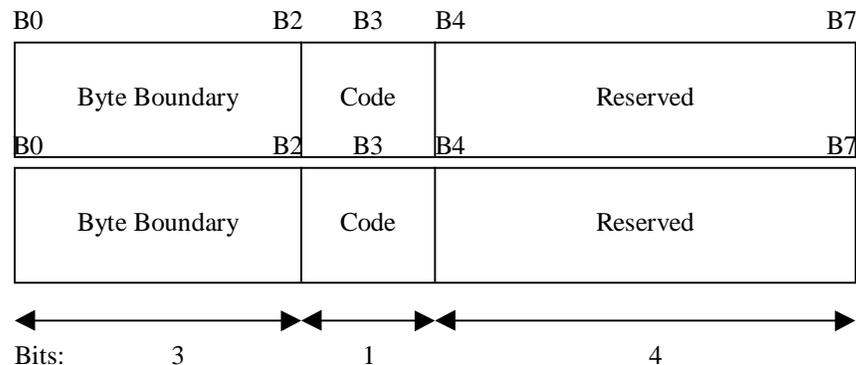
- a) 0Ah (MSB to LSB) for 1 Mbit/s DBPSK
- b) 14h (MSB to LSB) for 2 Mbit/s DQPSK
- c) [37h \(MSB to LSB\) for 5.5 Mbit/s CCK or PBCC with DBPSK](#)
- d) [6eh \(MSB to LSB\) for 11 Mbit/s CCK or PBCC with DQPSK](#)

The DSSS PHY rate change capability is described in clause xx. This field shall be protected by the CCITT CRC-16 frame check sequence described in clause xx.

15.2.3.4 PLCP 802.11 Service Field (SERVICE)

[The service field is used to specify the exact length of the MPDU and the code used to encode it. This field is divided into 3 subfields: Byte Boundary, Code, and Reserved. The format of the service field is illustrated in Figure xx.](#)

[The 8 bit 802.11 service field shall be reserved for future use. The value of 00h signifies 802.11 device compliance.](#)



[The LSB shall be transmitted first in time.](#) This field shall be protected by the CCITT CRC-16 frame check sequence described in clause xx.

15.2.3.4.1 Byte Boundary (BOUNDARY)

[This 3 bit field specifies the number to be subtracted from the maximum number of octets that will fit into the time specified by the length field at the rate specified by the signal field, i.e.](#)

$$\text{MPDU length (in octets)} = \lfloor (\text{LENGTH} * \text{SIGNAL}) / (10 * 8) \rfloor - \text{BOUNDARY}$$

15.2.3.4.2 Code

[This 1 bit field specifies the code/modulation used to transmit the MPDU. A value of 0 indicates that the PSDU is transmitted using the default code \(Barker code when signal field is set to 0ah or 14h, CCK when signal field is set to 37h or 6eh\). A value of 1 indicates that the PSDU is transmitted using the optional code \(PBCC when the signal field value is set to 37h or 6eh\). This field shall always be set to 0 when the signal field is set to 0ah or 14h.](#)

15.2.3.5 PLCP Length Field (LENGTH)

The PLCP length field shall be an unsigned 16 bit integer which indicates the number of microseconds (16 to $2^{16}-1$ as defined by `aMPDUMaxLngth`) required to transmit the MPDU. The transmitted value shall be determined from the LENGTH parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in clause xx. The length field provided in the TXVECTOR is in bytes and is converted to microseconds for inclusion in the PLCP LENGTH field. 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 clause xx.

15.2.4 PLCP / **DSSSPBCC** PHY Data Scrambler and Descrambler

The polynomial $G(z) = z^{-7} + z^{-4} + 1$ shall be used to scramble ALL bits transmitted by the DSSS PHY. 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 1 and Figure 2 show typical implementations of the data scrambler and descrambler. Other implementations are possible.

The scrambler should be initialized to any state except all ones when transmitting.

Scrambler Polynomial; $G(z) = Z^{-7} + Z^{-4} + 1$

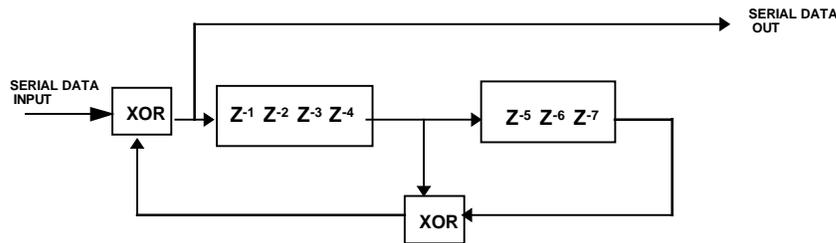


Figure 1, Data Scrambler

Descrambler Polynomial; $G(z) = Z^{-7} + Z^{-4} + 1$

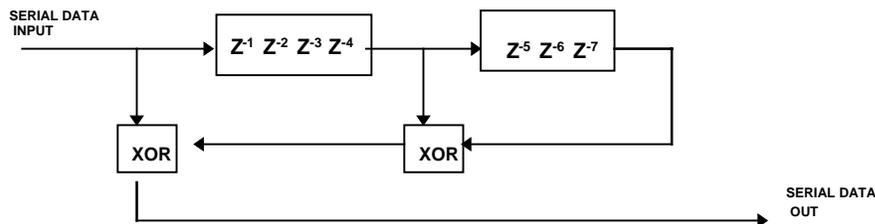


Figure 2, Data Descrambler

15.2.5 PLCP/**PBCC** Data Modulation and Modulation Rate Change

[This coding scheme uses a binary convolutional coding scheme with a 64-state binary convolutional code and a cover sequence. The output of the BCC is encoded jointly onto the I and Q channels, as further documented below. This provides enhanced multi-path performance and reduced complexity over using two generators and encoding the I and Q channel independently. The cover sequence also provides added multi-path immunity.](#)

The encoder for this scheme is shown in Figure AA. Incoming data is first encoded with a binary convolutional code that is well suited for difficult channels such as wireless communications channels. The encoded data is then scrambled before transmission through the channel.

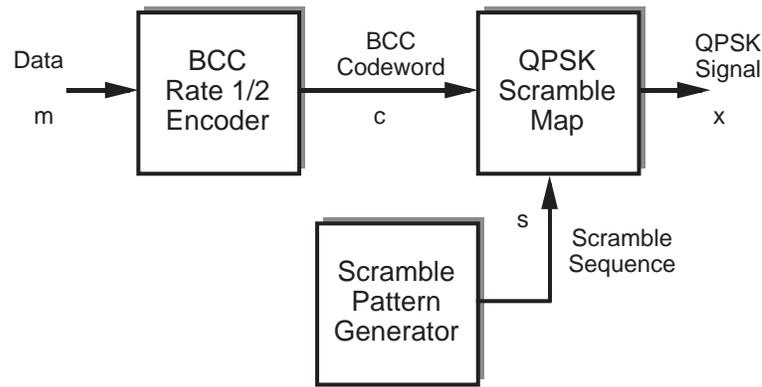


Figure AA

The binary convolutional code that is used is a 64-state, rate 1/2 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 $G = [133, \quad 175]$. This code provides a good trade-off between additive white Gaussian noise (AWGN) performance and performance in multi-path environments.

Since the system is packet based, the encoder is defined as being in state zero, i.e. all memory elements contain zero, at the beginning of each packet. The encoder must also be placed in a known state at the end of each packet to prevent the data bits near the end of the packet from being substantially less reliable than those early on in the packet. To place the encoder in a known state at the end of a packet, six deterministic bits are input immediately following the last data bit input to the convolutional encoder. These bits are all zero, which places the encoder in the zero state.

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

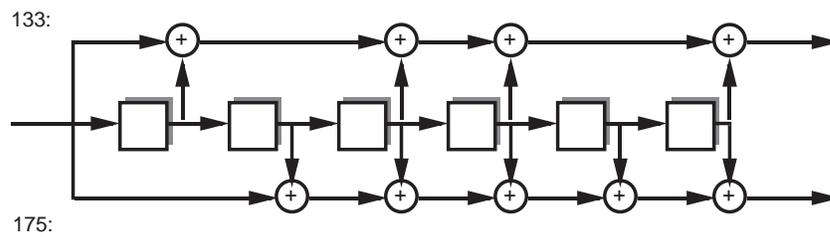


Figure BB

The output of the binary convolutional code described in above is mapped a constellation using one of two possible modes. One mode uses BPSK and the other 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 and used to produce two PSK symbols. This yields a throughput of one bit per symbol in QPSK mode and one-half a bit per second in BPSK mode.

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 CC.

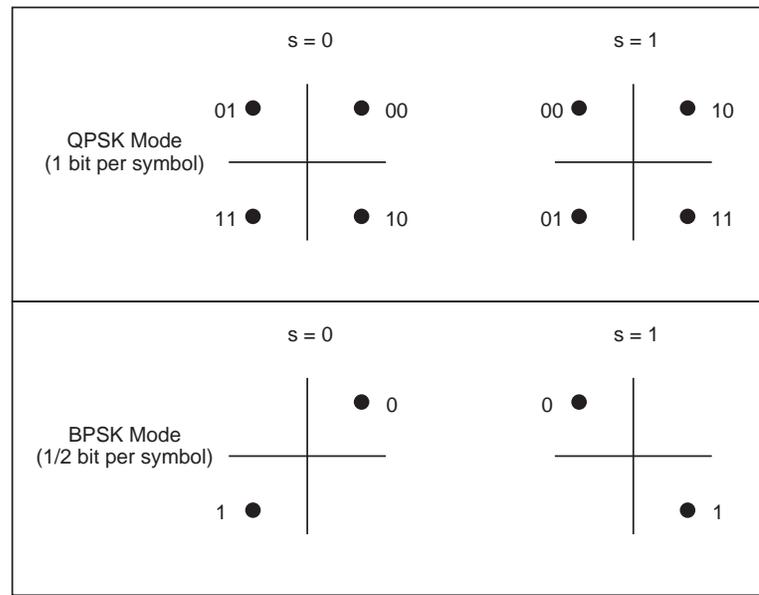


Figure CC

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 2.2.1.

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 c_i is the i th bit of the seed sequence, where $0 \leq i \leq 15$, then the sequence that is used to scramble 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
...
c10 c11 c12 c13 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 packet based systems with more than 256 bits and continuous systems this sequence of 256 bits is simply repeated.

15.2.6 PLCP Transmit Procedure

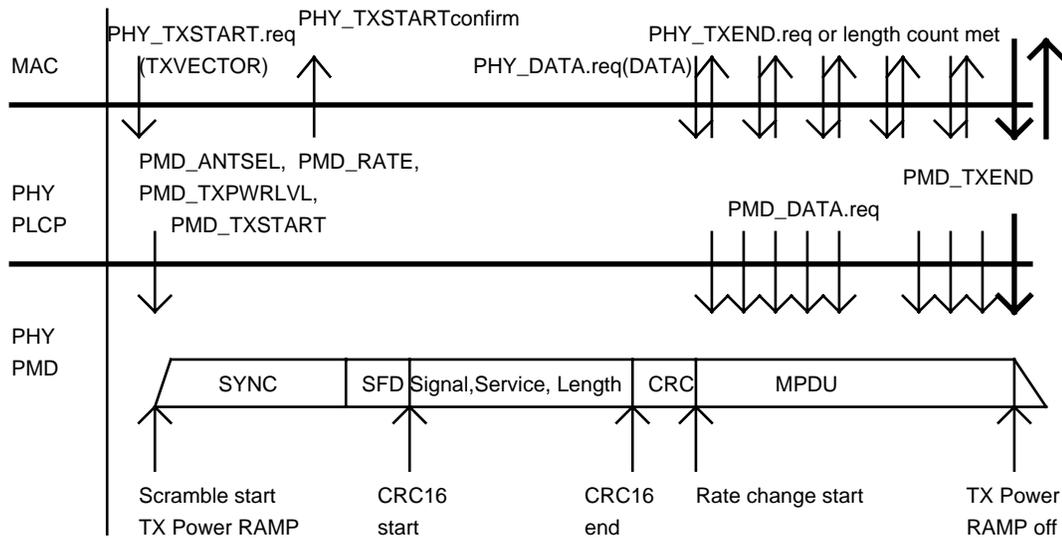
The PLCP transmit procedure is shown in Figure xx.

In order to transmit data, PHY-TXSTART.request shall be enabled so that the PHY entity shall be in the transmit state. Further, the PHY shall be set to operate at the appropriate channel through Station Management via the PLME. Other transmit parameters such as DATARATE, Code, TX antenna, and TX power are set via the PHY-SAP with the PHY-PHY-TXSTART.request(TXVECTOR) as described in clause xx.

Based on the status of CCA indicated by PHY-CCA.indicate, the MAC will assess that the channel is clear. A clear channel shall be indicated by PHY-CCA.indicate(IDLE). If the channel is clear, transmission of the PPDU shall be initiated by issuing the PHY-TXSTART.request (TXVECTOR) primitive. The TXVECTOR elements for the PHY-TXSTART.request are the PLCP header parameters SIGNAL (DATARATE), SERVICE (Code and LENGTH) and LENGTH and the PMD parameters of TX_ANTENNA and TXPWR_LEVEL. The PLCP header parameter

LENGTH is calculated from the TXVECTOR element by multiplying by 8 for 1 Mbit/s and by 4 for 2 Mbit/s, and then dividing by the rate (in Mbit/s).

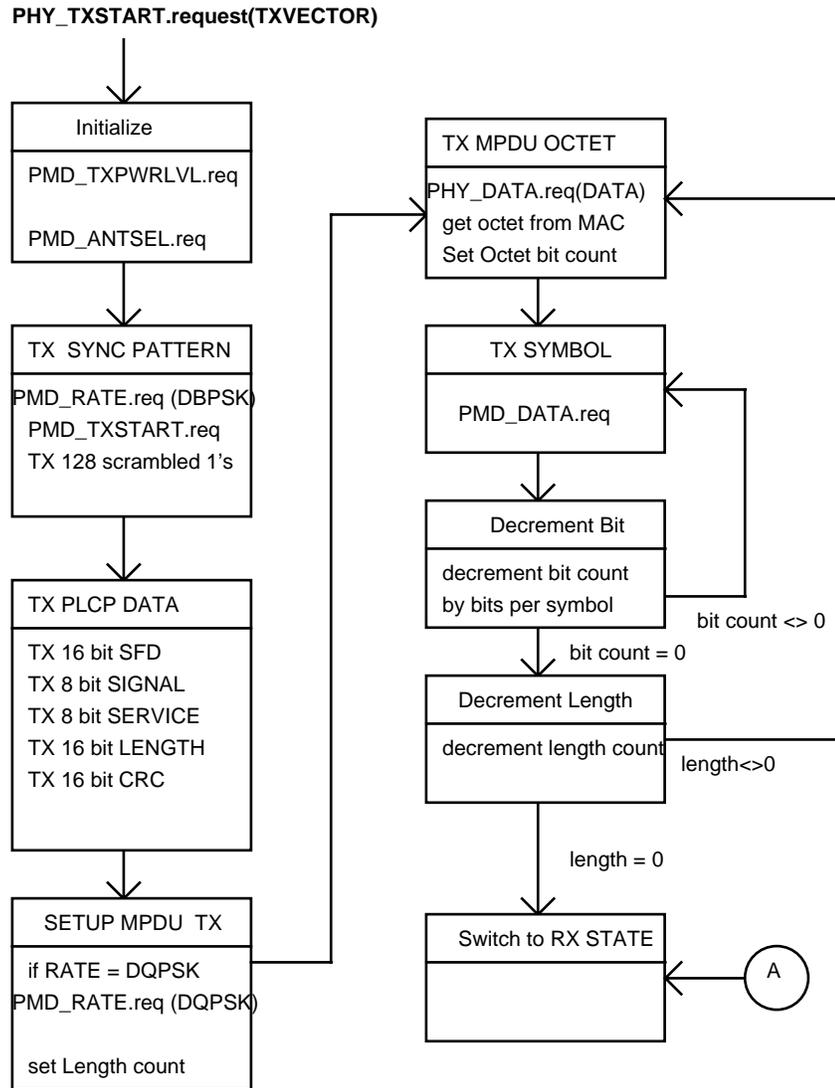
The PLCP shall issue PMD_ANTSEL, PMD_RATE, and PMD_TXPWRLVL primitives to configure the PHY. The PLCP shall then issue a PMD_TXSTART.request and the PHY entity shall immediately initiate data scrambling and transmission of the PLCP preamble based on the parameters passed in the PHY-TXSTART.request primitive. The time required for TX power on ramp described in clause xx 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 rate and code change, if any, shall be initiated with the first data symbol of the MPDU as described in clause xx. The PHY proceeds with MPDU 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 MPDU octet according to the number supplied in the DSSS PHY preamble LENGTH field. The packet transmission shall be completed and the PHY entity shall enter the receive state (i.e. PHY-TXSTART shall be disabled). It is recommended that chipping continue during power down. Each PHY-TXEND.request is acknowledged with a PHY-TXEND.confirm primitive from the PHY.



[NEED TO EDIT THIS FIGURE!](#)

Figure xx, PLCP Transmit Procedure

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



(A) At any stage in the above flow diagram, if a PHY_TXEND.request is received

[NEED TO EDIT THIS FIGURE!](#)

Figure xx, PLCP Transmit State Machine

15.3.2 DSSS Physical Layer Management Information Base

All DSSS Physical Layer Management Information Base attributes are defined in clause **Error! Reference source not found.** with specific values defined in Table xx.

| Managed Object | Default Value / Range | Operational Semantics |
|----------------------------|--------------------------|-----------------------|
| agPhyOperationGroup | | |
| aPHYType | DSSS-2.4 (02) | Static |
| aPHYType | High-Speed-2.4 (xx) | Static |
| aTempType | implementation dependent | Static |
| aCWmin | 31 | Static |
| aCWmax | 1023 | Static |
| aRegDomainsSupported | implementation dependent | Static |
| aCurrentRegDomain | implementation dependent | Static |

| | | |
|-----------------------------|------------------------------|---------|
| aSlotTime | 20 μ s | Static |
| aCCATime | $\leq 15 \mu$ s | Static |
| aRxTxTurnaroundTime | $\leq 5 \mu$ s | Static |
| aTxPLCPDelay | implementation dependent | Static |
| aRxTxSwitchTime | $\leq 5 \mu$ s | Static |
| aTxRampOnTime | implementation dependent | Static |
| aTxRFDelay | implementation dependent | Static |
| aSIFSTime | 10 μ s | Static |
| aRxRFDelay | implementation dependent | Static |
| aRxPLCPDelay | implementation dependent | Static |
| aMACProcessingDelay | not applicable | n/a |
| aTxRampOffTime | implementation dependent | Static |
| aPreambleLength | 144 bits | Static |
| aPLCPHeaderLength | 48 bits | Static |
| | | |
| agPhyRateGroup | | |
| aSupportedDataRatesTx | 02h, 04h | Static |
| aSupportedDataRatesRx | 02h, 04h | Static |
| aMPDUMaxLength | $4 \leq x \leq (2^{13} - 1)$ | Static |
| | | |
| agPhyAntennaGroup | | |
| aCurrentTxAntenna | implementation dependent | Dynamic |
| aDiversitySupport | implementation dependent | Static |
| | | |
| agPhyTxPowerGroup | | |
| aNumberSupportedPowerLevels | implementation dependent | Static |
| aTxPowerLevel1 | implementation dependent | Static |
| aTxPowerLevel2 | implementation dependent | Static |
| aTxPowerLevel3 | implementation dependent | Static |
| aTxPowerLevel4 | implementation dependent | Static |
| aTxPowerLevel5 | implementation dependent | Static |
| aTxPowerLevel6 | implementation dependent | Static |
| aTxPowerLevel7 | implementation dependent | Static |
| aTxPowerLevel8 | implementation dependent | Static |
| aCurrentTxPowerLevel | implementation dependent | Dynamic |
| | | |
| agPhyStatusGroup | | |
| aSynthesizerLocked | implementation dependent | Dynamic |
| | | |
| agPhyDSSSGroup | | |
| aCurrentChannel | implementation dependent | Dynamic |
| aCCAModeSupport | implementation dependent | Static |
| aCurrentCCAMode | implementation dependent | Dynamic |
| aEDThreshold | implementation dependent | Dynamic |
| | | |
| agPhyPwrSavingGroup | | |
| aDozeTurnonTime | implementation dependent | Static |
| aCurrentPowerState | implementation dependent | Dynamic |
| | | |
| agAntennasListGroup | | |
| aSupportTxAntennas | implementation dependent | Static |
| aSupportRxAntennas | implementation dependent | Static |
| aDiversitySelectRx | implementation dependent | Dynamic |
| | | |

Table xx, MIB Attribute Default Values / Ranges

Notes: 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 entity.

15.4.4.2 PMD_SAP Peer-to-Peer Service Primitive Parameters

Several service primitives include a parameter vector. This vector shall be actually a list of parameters which may vary depending on PHY type. Table xx indicates the parameters required by the MAC or [DSSSHigh-Speed](#) PHY in each of the parameter vectors used for peer-to-peer interactions.

| Parameter | Associated Primitive | Value |
|-------------|------------------------------------|-------------------------------|
| LENGTH | RXVECTOR, TXVECTOR | 4 to 2 ¹⁶ -1 |
| DATARATE | RXVECTOR, TXVECTOR | PHY dependent |
| Code | RXVECTOR, TXVECTOR | PHY dependent |
| SERVICE | RXVECTOR, TXVECTOR | PHY dependent |
| TXPWR_LEVEL | TXVECTOR | PHY dependent |
| TX_ANTENNA | TXVECTOR | PHY dependent |
| RSSI | RXVECTOR | PHY dependent |
| SQ | RXVECTOR | PHY dependent |
| RX_ANTENNA | RXVECTOR | PHY dependent |

Table xx, [DSSSHigh-Speed](#) PMD_SAP Peer-to-Peer Service Primitives

15.4.4.3 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_RATE | X | X | | |
| PMD_CPDE | X | X | | |
| PMD_RSSI | | X | | |
| PMD_SQ | | X | | |
| PMD_CS | | X | | |
| PMD_ED | X | X | | |

Table xx, PMD_SAP Sublayer-to-Sublayer Service Primitives

15.4.4.4 PMD_SAP Service Primitive Parameters

| Parameter | Associate Primitive | Value |
|-----------|---------------------------------------|---|
| DATA | PHY-DATA.request PHY-DATA.indicate | octet value: 00h-FFh |
| TXVECTOR | PHY-DATA.request | a set of parameters |
| RXVECTOR | PHY-DATA.indicate | a set of parameters |
| TXD_UNIT | PMD_DATA.request | One(1), Zero(0): DBPSK di bit combinations 00,01,11,10: DQPSK |
| RXD_UNIT | PMD_DATA.indicate | One(1), Zero(0): DBPSK di bit combinations 00,01,11,10: DQPSK |

| | | |
|-------------|---|---|
| RF_STATE | PMD_TXE.request | Receive, Transmit |
| ANT_STATE | PMD_ANTSEL.indicate PMD_ANTSEL.request | 1 to 256 |
| TXPWR_LEVEL | PHY-TXSTART | 0,1,2,3 (max of 4 levels) |
| RATE | PMD_RATE.indicate PMD_RATE.request | 0Ah for 1 Mbit/s DBPSK 14h for 2 Mbit/s DQPSK |
| RATE | PMD_RATE.indicate PMD_RATE.request | 0Ah for 1 Mbit/s DBPSK 14h for 2 Mbit/s DQPSK 37h (MSB to LSB) for 5.5 Mbit/s CCK or PBCC with DBPSK 6eh (MSB to LSB) for 11 Mbit/s CCK or PBCC with DQPSK |
| RSSI | PMD_RSSI.indicate | 0-8 bits of RSSI |
| SQ | PMD_SQ.indicate | 0-8 bits of Signal Quality |

Table xx, List of Parameters for the PMD Primitives

15.4.5.xx PMD_CODE.request

Function

This primitive, generated by the PHY PLCP sublayer, selects the code which shall be used by the high-speed PHY for transmission.

Semantic of the Service Primitive

The primitive shall provide the following parameters:

PMD_CODE.request(CODE)

CODE selects which of the high-speed PHY codes shall be used for MPDU transmission. Clause xx provides further information on the high-speed PHY codes. The high-speed PHY rate and code change capability is fully described in clause xx.

When Generated

This primitive shall be generated by the PLCP sublayer to change or set the current high-speed PHY modulation rate used for the MPDU portion of a PPDU.

Effect of Receipt

The receipt of PMD_CODE selects the code which shall be used for all subsequent MPDU transmissions. This code shall be used for transmission only. The high-speed PHY shall still be capable of receiving the required high-speed PHY code.

15.4.5.xx PMD_CODE.indicate

Function

This primitive, generated by the PMD sublayer, indicates which code was used to receive the MPDU portion of the PPDU. The code shall be indicated in the PLCP preamble 802.11 SERVICE field.

Semantic of the Service Primitive

The primitive shall provide the following parameters:

PMD_CODE.indicate(CODE)

In receive mode, the CODE parameter informs the PLCP layer which of the high-speed PHY codes was used to process the MPDU portion of the PPDU. Clause xx provides further information on the high-speed PHY codes. The high-speed PHY rate and code change capability is fully described in clause xx.

When Generated

This primitive shall be generated by the PMD sublayer when the PLCP preamble 802.11 SERVICE field has been properly detected.

Effect of Receipt

This parameter shall be provided to the PLCP layer for information only.

5 Method 2

This method for providing support for the use of optional codes is similar to the multirate mechanism already in the standard to support optional rates. It requires the addition of an information element called "Supported Codes" to frames that contain the "Supported Rates" element. It also requires some changes to the interface between the MAC and the PHY to transfer the supported codes information from the PHY to the MAC. This method provides a generic mechanism for supporting optional codes.

5.1 Changes to the 802.11 Standard Document

The following is a list of changes required to the current 802.11 standard document to support the new information element called the "Supported Codes". The addition of this element has implications on various portions of the MAC, MAC management and PHY management functionality. The changes required due to these implications are also listed here. Changes are also needed to the state machines to incorporate the text changes described here.

The black text is the original text from the standard document (will need to be updated to incorporate changes by Tgrev). The text marked with revision marks depicts the changes required to support the new information element.

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 from/to the wireless medium (WM). ~~frames from the wireless medium (WM).~~ The BSS basic rate set data rates are preset for all stations in the BSS.

3.x basic service set (BSS) basic code set:

The set of codes that all the stations in a BSS will be capable of using to receive frames from the wireless medium (WM). The BSS basic code set codes are preset for all stations in the BSS.

7.2.3.1 Beacon Frame Format

The Frame Body of a Management frame of Subtype Beacon contains the following information:

| Order | Information | Note |
|-------|------------------------------------|----------|
| 1 | Timestamp | |
| 2 | Beacon Interval | |
| 3 | Capability Information | |
| 4 | SSID | |
| 5 | Supported Rates | |
| 6 | FH Parameter Set | <u>1</u> |
| 6 | Supported Codes | <u>6</u> |
| 7 | FH Parameter Set | 1 |
| 7 | DS Parameter Set | 2 |
| 8 | DS Parameter Set | <u>2</u> |
| 8 | CF Parameter Set | 3 |
| 9 | CF Parameter Set | 3 |
| 9 | IBSS Parameter Set | 4 |
| 10 | IBSS Parameter Set | 4 |
| 10 | TIM | 5 |
| 11 | TIM | 5 |

Table xx, Beacon Frame Body

Notes:

6. [The Supported Codes Set information element is only present within Association Request Frames generated by STAs using High-Rate PHYs.](#)

7.2.3.4 Association Request Frame Format

The Frame Body of a Management frame of Subtype Association Request contains the following information:

| Order | Information | Note |
|-------|---------------------------------|----------|
| 1 | Capability Information | |
| 2 | Listen Interval | |
| 3 | SSID | |
| 4 | Supported Rates | |
| 5 | Supported Codes | <u>1</u> |

Table xx, Association Request Frame Body

Notes:

1. [The Supported Codes Set information element is only present within Association Request Frames generated by STAs using High-Rate PHYs.](#)

7.2.3.5 Association Response Frame Format

The Frame Body of a Management frame of Subtype Association Response contains the following information:

| Order | Information | Note |
|-------|---------------------------------|----------|
| 1 | Capability Information | |
| 2 | Status Code | |
| 3 | Association ID (AID) | |
| 4 | Supported Rates | |
| 5 | Supported Codes | <u>1</u> |

Table xx, Association Response Frame Body

Notes:

1. [The Supported Codes Set information element is only present within Association Response Frames generated by STAs using High-Rate PHYs.](#)

7.2.3.6 Reassociation Request Frame Format

The Frame Body of a Management frame of Subtype Reassociation Request contains the following information:

| Order | Information | Note |
|-------|---------------------------------|-------------------|
| 1 | Capability Information | |
| 2 | Listen Interval | |
| 3 | Current AP Address | |
| 4 | SSID | |
| 5 | Supported Rates | |
| 6 | Supported Codes | 1 |

Table xx, Reassociation Request Frame Body

Notes:

- [1.](#) The Supported Codes Set information element is only present within Association Request Frames generated by STAs using High-Rate PHYs.

7.2.3.7 Reassociation Response Frame Format

The Frame Body of a Management frame of Subtype Reassociation Response contains the following information:

| Order | Information | Note |
|-------|---------------------------------|-------------------|
| 1 | Capability Information | |
| 2 | Status Code | |
| 3 | Association ID (AID) | |
| 4 | Supported Rates | |
| 5 | Supported Codes | 1 |

Table xx, Reassociation Response Frame Body

Notes:

- [1.](#) The Supported Codes Set information element is only present within Association Request Frames generated by STAs using High-Rate PHYs.

7.2.3.8 Probe Request Frame Format

The Frame Body of a Management frame of Subtype Probe Request contains the following information:

| Order | Information | Note |
|-------|---------------------------------|-------------------|
| 1 | SSID | |
| 2 | Supported Rates | |
| 3 | Supported Codes | 1 |

Table xx, Probe Request Frame Body

Notes:

- [1.](#) The Supported Codes Set information element is only present within Association Request Frames generated by STAs using High-Rate PHYs.

7.2.3.9 Probe Response Frame Format

The Frame Body of a Management frame of Subtype Probe Response contains the following information:

| Order | Information | Note |
|-------|------------------------------------|-------------------|
| 1 | Timestamp | |
| 2 | Beacon Interval | |
| 3 | Capability Information | |
| 4 | SSID | |
| 5 | Supported Rates | |
| 6 | FH Parameter Set | 1 |
| 6 | Supported Codes | 5 |
| 7 | FH Parameter Set | 1 |
| 7 | DS Parameter Set | 2 |
| 8 | DS Parameter Set | 2 |
| 8 | CF Parameter Set | 3 |
| 9 | CF Parameter Set | 3 |
| 9 | IBSS Parameter Set | 4 |
| 10 | IBSS Parameter Set | 4 |

Table xx, Probe Response Frame Body

Notes:

5. [The Supported Codes Set information element is only present within Association Request Frames generated by STAs using High-Rate PHYs.](#)

7.3.1.9 Status Code

The following failure cause codes are defined:

| Status Code | Meaning |
|--------------------------|--|
| 0 | Successful |
| 1 | Unspecified Failure |
| 2-9 | Reserved |
| 10 | Cannot support all requested capabilities in the Capability Information Field |
| 11 | Reassociation denied due to inability to confirm that Association exists |
| 12 | Association denied due to reason outside the scope of this standard |
| 13 | Responding station does not support the specified Authentication Algorithm |
| 14 | Received an Authentication Frame with Authentication Transaction Sequence Number out of expected sequence |
| 15 | Authentication rejected because of challenge failure |
| 16 | Authentication rejected due to timeout waiting for next frame in sequence |
| 17 | Association denied because AP is unable to handle additional associated stations |
| 18 | Association denied due to requesting station not supporting all of the data rates in the BSSBasicRateSet parameter |
| 19-65535 | Reserved |
| 19 | Association denied due to requesting station not supporting all of the data codes in the BSSBasicCodeSet parameter |
| 20-65535 | Reserved |

Table xx, Status Codes

7.3.2 Information Elements

The set of valid elements is defined in Table xx.

| Information Element | Element ID |
|---------------------------------------|------------|
| SSID | 0 |
| Supported Rates | 1 |
| FH Parameter Set | 2 |
| DS Parameter Set | 3 |
| CF Parameter Set | 4 |
| TIM | 5 |
| IBSS Parameter Set | 6 |
| Reserved | 7-15 |
| Supported Codes | 7 |
| Reserved | 8-15 |
| Challenge Text | 16 |
| Reserved for Challenge Text extension | 17-31 |
| Reserved | 32-255 |

Table xx, Element IDs

7.3.2.x Supported Codes

The Supported Codes element shall specify all the codes that this STA is capable of receiving. The information field is encoded as 1 to 8 octets where each octet defines a single Supported Code as described in XX(The High-Rate PHY clause).

Within Beacon, Probe Response, Association Response and Reassociation Response Management frames, each Supported Code belonging to the aBSSBasicCodeSet as defined in XX, shall be encoded as an octet with the most significant bit (bit 7) set to 1. Codes not belonging to the aBSSBasicCodeSet shall be encoded with the most significant bit set to 0. Receiving STAs shall ignore the most significant bit of each Supported Code octet in other Management frame types.

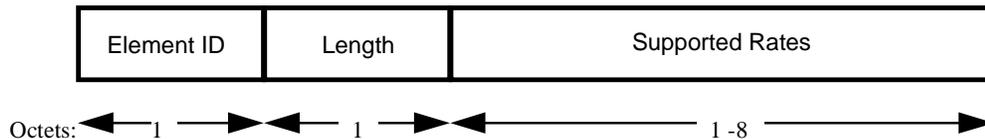


Figure xx, Supported Codes Element Format

9 MAC Sublayer Functional Description

Here, the MAC functional description is presented. Clause **Error! Reference source not found.** introduces the architecture of the MAC sublayer, including the distributed coordination function, the point coordination function and their coexistence in an 802.11 LAN. Clauses 0 and **Error! Reference source not found.** expand on this introduction and provide a complete functional description of each. Clauses **Error! Reference source not found.** and **Error! Reference source not found.** cover fragmentation and defragmentation. Multirate support is addressed in clause **Error! Reference source not found.** Multi-code support is described in Clause XX. Clause **Error! Reference source not found.** lists the allowable frame exchange sequences. Clause **Error! Reference source not found.** describes a number of additional restrictions to limit the cases in which MSDUs are reordered or discarded.

9.2 Distributed Coordination Function (DCF)

-
-

•
 The medium access protocol allows for stations to support different sets of data rates. ~~All STAs shall receive all the All STAs shall be able to receive and transmit at all the~~ data rates in the ~~aBasicRateSet~~~~aBSSBasicRateSet~~. ~~To support the proper operation of the RTS/CTS and the Virtual Carrier Sense mechanism, all STAs must be able to detect the RTS and CTS frames. For this reason the RTS and CTS frames shall be transmitted at one of the aBSSBasicRateSet rates.~~ (See Clause 9.6 for a description of multirate operation).

rates. (See Clause 9.6 for a description of multirate operation).

~~In addition to supporting different sets of data rates the medium access protocol allows for stations to support different sets of codes. All STAs shall receive all the~~ codes in the ~~aBssBasicCodeSet~~ and transmit at one or more of the ~~aBssBasicCodeSet~~ codes. 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 ~~aBssBasicCodeSet~~ codes. (See Clause 9.x for a description of multi-code operation).

Data frames sent under the DCF shall use the Frame Type Data and Subtype Data or Null Function. Stations receiving Data Type frames shall only consider the frame body as the basis of a possible indication to LLC.

9.6 Multirate Support

Some PHYs have multiple data transfer rate capabilities which 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 which shall be followed by all stations.

All Control Frames shall be transmitted at one of the rates in the BSSBasicRateSet, see **Error! Reference source not found.**, ~~or at one of the rates in the PHY mandatory rate set so so that~~ they will be understood by all stations in the BSS.

All frames with Multicast and Broadcast immediate address shall be transmitted at one of the rates included in the BSSBasicRateSet, regardless of their type or subtype.

Data and/or Management MPDUs with a Unicast immediate address shall be sent on any supported data rate selected by the rate switching mechanism (whose output is an internal MAC variable called MACCurrentRate, defined in units of 500 kBit/s, which is used for calculating the Duration/ID field of each frame). A station shall not transmit at a rate that is known not to be supported by the destination station, 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 station and the station to which the ACK is intended.

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

In order to allow the transmitting station to calculate the contents of the Duration/ID field, the responding station shall transmit its Control Response frame (either CTS or ACK) at the same rate as highest rate in the BSSBasicRateSet that is less than or equal to the rate of the immediately previous frame in the frame exchange sequence (as defined in **Error! Reference source not found.**).

9.x Multi-Code Support

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

~~**Error! Reference source not found.**~~, if this rate belongs to ~~All Control Frames shall be transmitted at one of the codes in the BSSBasicCodeSet, see XX, or at one of the codes in~~ the PHY mandatory ~~rates, or else at the highest code set so they will be understood by all stations.~~

possible rate belonging to the PHY rates in the BSSBasicRateSet. All frames with Multicast and Broadcast immediate destination addresses shall be transmitted at one of the codes included in the BSSBasicCodeSet.

Data and/or Management MPDUs with a Unicast immediate destination address shall be sent using a code selected by the code switching mechanism (whose output is an internal MAC variable called MACCurrentCode). A station shall not transmit using a code that is known not to be supported by the destination station, as reported in the Supported Codes Element in the Management Frames. For frames of type Data+CF-ACK, Data+CF-Poll+CF-ACK and CF-Poll+CF-ACK, the code chosen to transmit the frame must be supported by both the addressed recipient station and the station to which the ACK is intended.

Under no circumstances shall a station initiate transmission of a Data or Management using a code other than the ones in the OperationalCodeSet, a parameter of the MLMEJOIN.request primitive.

10.3.2.2.2 Semantics of the service primitive

Each BSSDescription consists of the following elements:

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

| | | | |
|--|--|--|--|
| | | | be able to receive and transmit using each of the codes listed in the set. |
|--|--|--|--|

10.3.3.1.2 Semantics of the service primitive

Function

This primitive requests synchronization with a BSS.

Semantics of the Service Primitive

The primitive parameters are as follows:

```

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

| 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 μ s) 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 is a superset of the BSSBasicRateSet advertised by the BSS. |
| OperationalCodeSet | set of integers | 1 through 127 inclusive (for each integer in the set) | The set of codes that the STA desires to use for communication within the BSS. The STA must be able to receive using each of the codes listed in the set. The OperationalCodeSet is a superset of the BSSBasicCodeSet advertised by the BSS. |

10.3.10.1.2 Semantics of the service primitive

Semantics of the Service Primitive

The primitive parameters are as follows:

```

MLME-START.request (
    SSID,
    BSSType,
    BeaconPeriod,
    DTIMPeriod,
    CF parameter set,
    PHY parameter set,
)
    
```

IBSS parameter set,
 ProbeDelay,
 CapabilityInformation,
 BSSBasicRateSet,
 OperationalRateSet,
[OperationalCodeSet](#)
)

| Name | Type | Valid Range | Description |
|------------------------------------|---------------------------------|---|--|
| SSID | octet string | 1 - 32 octets | The SSID of the BSS. |
| BSSType | Enumeration | INFRA-STRUCTURE, INDEPENDENT | The type of the BSS. |
| Beacon Period | integer | greater than or equal to 1 | The Beacon period of the BSS (in K μ s). |
| DTIM Period | integer | As defined in Frame Format | The DTIM Period of the BSS (in Beacon Periods) |
| CF parameter set | As defined in Frame Format | As defined in Frame Format | The parameter set for CF periods, if the BSS supports CF mode. aCFPPeriod is modified as a side effect of the issuance of a MLME-START.request primitive. |
| PHY parameter set | As defined in Frame Format | As defined in Frame Format | The parameter set relevant to the PHY. |
| IBSS parameter set | As defined in Frame Format | As defined in Frame Format | The parameter set for the IBSS, if BSS is an IBSS. |
| ProbeDelay | integer | N/A | Delay (in μ s) to be used prior to transmitting a Probe frame during active scanning |
| CapabilityInformation | As defined in Frame Format | As defined in Frame Format | The capabilities to be advertised for the BSS. |
| BSSBasicRateSet | set of integers | 1 through 127 inclusive (for each integer in the set) | The set of data rates (in units of 500 kbit/s) that must be supported by all STAs that desire to join this BSS. The STA that is creating the BSS must be able to receive at each of the data rates listed in the set. |
| 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. |
| BSSBasicCodeSet | set of integers | 1 through 127 inclusive (for each integer in the set) | The set of codes that must be supported by all STAs that desire to join this BSS. The STAs must be able to receive and transmit using each of the codes 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 is a superset of the BSSBasicRateSet advertised by the BSS. |
| 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 is a superset of the BSSBasicRateSet advertised by the BSS. |

| | | | |
|------------------------------------|---------------------------------|---|--|
| OperationalCodeSet | set of integers | 1 through 127 inclusive (for each integer in the set) | The set of codes that the STA desires to use for communication within the BSS. The STA must be able to receive using each of the codes listed in the set. The OperationalCodeSet is a superset of the BSSBasicCodeSet advertised by the BSS. |
|------------------------------------|---------------------------------|---|--|

12.3.4.3 PHY SAP Service Primitives Parameters

The following table shows the parameters used by one or more of the PMD SAP Service Primitives.

| Parameter | Associated Primitive | Value |
|-------------------------|---|---|
| DATA | PHY-DATA.request PHY-DATA.indication | Octet value 00-FFh |
| TXVECTOR | PHY-TXSTART.request | A set of parameters. |
| STATUS | PHY-CCA.indication | BUSY, IDLE |
| RXVECTOR | PHY-RXSTART.indication | A set of parameters. |
| RXERROR | PHY-RXEND.indication | NoError, FormatViolation, CarrierLost, UnsupportedRate |
| RXERROR | PHY-RXEND.indication | NoError, FormatViolation, CarrierLost, UnsupportedRate, UnsupportedCode |

Table xx, PHY SAP Service Primitive Parameters

12.3.4.5 Vector Descriptions

Several service primitives include a parameter vector. This vector is a list of parameters which may vary depending on the PHY type. The table below lists the parameter values required by the MAC or PHY in each of the parameter vectors. Parameters in the vectors which are Management rather than medium access control may be specific to the PHY and are listed in the clause covering that PHY.

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

Table xx, Vector Descriptions

12.3.5.12.2 Semantics of the Service Primitive

The primitive provides the following parameters:

PHY-RXEND.indication (RXERROR)

The RXERROR parameter can convey one or more of the following values: NoError, FormatViolation, CarrierLost, [or UnsupportedRate](#).[UnsupportedRate](#), [or UnsupportedCode](#). A number of error conditions may occur

after the PLCP's receive state machine has detected what appeared to be a valid preamble and start frame delimiter. The following describes the parameter returned for each of those error conditions.

NoError. This value is used to indicate that no error occurred during the receive process in the PLCP.

FormatViolation. This value is used to indicate that the format of the received PLCPPDU was in error.

CarrierLost. This value is used to indicate that during the reception of the incoming MPDU, carrier was lost and no further processing of the MPDU can be accomplished.

UnsupportedRate. This value is used to indicate that during the reception of the incoming PLCPPDU, a non supported data rate was detected.

[UnsupportedCode.](#) This value is used to indicate that during the reception of the incoming PLCPPDU, a non supported code was detected.

6 Changes Required to the Standard to Support the High-Speed PHY

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 frames from the wireless medium (WM). The BSS basic rate set data rates are preset for all stations in the BSS.

7 References

- | | |
|---------------------|---|
| IEEE P802.11-98/82 | Proposal for High Data Rate 2.4 GHz PHY With Variable Rate Binary Convolutional Coding on QPSK (VR-BCC) |
| IEEE P802.11-98/83 | Intellectual Property Release for Alantro Communications Proposal for High Rate 2.4 GHz PHY (VR-BCC QPSK) |
| IEEE P802.11-98/84 | FEC is Not Overhead |
| IEEE P802.11-98/246 | Introducing the Harris-Lucent Compromise Proposal for TGb |