IEEE P802.15
Wireless Personal Area Networks

Project	IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)	
Title	**AV PHY updates**	
Date Submitted	[6 May, 2008	
Source	[James P. K. Gilb] [SiBEAM] [555 N. Mathilda, Suite 100, Sunnyvale, CA 94085]	Voice: [408-245-3120] Fax: [408-245-3120] E-mail: [last name at ieee dot org]
Re:	[]	
Abstract	[Suggested updatds for AV PHY.]	
Purpose	[To assist in comment resolution.]	
Notice	This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15.	

1. Technical updates

1.1 Updates for MAC to select/detect PHY mode of piconet

Add the following row to Table 3c in 6.3.2 as shown:

Table 3c—Elements of PiconetDescription

Name	Type	Valid range	Description
PHYMode	Enumeration	2.4_GHZ, SC_MMWAVE, HSI_MMWAVE, AV_MMWAVE	The PHY that is being used in the piconet that was found.

Add the following row to Table 3e in 6.3.3 as shown:

Table 3c—MLME-START primitive parameters

Name	Type	Valid range	Description
PHYMode	Enumeration	2.4_GHZ, SC_MMWAVE, HSI_MMWAVE, AV_MMWAVE	The PHY that will be used in the piconet to be started.

6.3.3.1 MLME-START.request

Change the primitive definition in 6.3.3.1 as shown

```
MLME-START.request              (
                                BSIDLength,
                                BSID,
                                SECMode,
                                MinDepSuperframePercent,
                                DesiredDepSuperframePercent,
                                PHYMode
                                )
```

1.2 Channel probing information in directional ACK

Add the following to the acronyms

ROUS	rise over useable sensitivity
SINR	signal to noise and interference ratio

Add the following to 12.1.7.3

The Receive Status field shall be formatted as illustrated in Figure 1.

Bits: 7	2	4	2	4	3	1
Reserved	Current MCS status	FER	Suggested CES length	SINR	ROUS	Valid

Figure 1—Receive status field format for mmWave PHY

The Valid bit shall be set to one if the information in the Receive Status field is valid and shall be set to zero otherwise.

The ROUS field contains the amount that the recieved frame was above the sensitivity of the MCS used. The range of the ROUS field is from 0 dB to 28 dB in 2 dB steps with 0b0000 corresponding to less than or equal to 0 dB,0b1110 corresponding to greater than 28 dB, and 0b1111 is reserved. For example, a ROUS value that is greater than or equal to 8 B but less than 10 dB would be encoded as 0b0100.

The SINR field contains the estimated signal to noise and interference ratio of the received frame. The range of the SINR field is from 0 dB to 28 dB in 2 dB steps with 0b0000 corresponding to less than or equal to 0 dB, 0b1110 corresponding to greater than 28 dB, and 0b1111 reserved. For example, an SINR value that is greater than or equal to 18 dB but less than 20 dB would be encoded as 0b1001.

Valid values of the Suggested CES Length field are:

- 0b00 → Suggest long CES length
- 0b01 → Suggest short CES length
- 0b10–0b11 → Reserved

The FER field contains the exponent of the estimate of the FER ranging from 10^{-1} to 10^{-10} in steps of -1 with 0b0000 corresponding to an FER exponent of less than or equal to -1 (i.e., an FER greater than or equal to 10^{-1}), 0b1001 corresponding to an FER exponent of greater than -10 (i.e., an FER of less than 10^{-10}), and 0b1010-0b111 Reserved. For example, an FER field value of 0b0110 indicates an exponent of -2.2 and that the FER was less than or equal to 10^{-2} but was greater than $10^{-2.2}$.

Valid values of the Current MCS Status fied are:

- 0b00 → Current MCS is inadequate
- 0b01 → Current MCS is adequate
- 0b10 → Current MCS is more than adequate, a lower MCS would be adequate
- 0b11 → Reserved

Add the following to subclause 12.4.3.8.

The directional LRP payload shall be formatted as illustrated in Figure 2.

octets: 2	3
HCS	Recieve status field

Figure 2—Directional LRP payload format

The HCS field contains the two octet HCS, as defined in <xref 12.2.4.2.1 or 11.2.9>, calculated over the Receive Status Field.

The Receive Status field is defined in 12.1.7.3.

1.3 AV OFDM retransmission procedure

Only one additional rule is required:

Data identified as video by the source DEV may be passed by the destination DEV to the higher layers even if the data was received in error.

All other rules for retransmission are adequately covered by 802.15.3, which says:

“During the CAP, retransmissions shall follow the backoff rules as specified in 8.4.2.

During CTAs within a CFP when an Imm-ACK or Dly-ACK is expected, but is not received during RIFS, the source DEV shall start the retransmission of the frame (or new frame if the failed frame’s retransmission limit has been met) at the end of RIFS as long as there is enough channel time remaining in the CTA for the entire frame exchange.

A DEV determines the number of times a frame is retried before the DEV gives up on that frame and discards it. If the DEV gives up on a fragment of an MSDU/MCDU, the DEV shall discard all MPDUs of that MSDU/MCDU.”

Not only are no other rules required, any other rules, such as those in DF3, are either out of scope (they describe implementation details) or incorrectly restrain the source and destination.

1.4 Aggregation for various PHY modes

Note that the decision to aggregate or not aggregate is up to the transmitter and is an implementation dependent decision that can depend on the application. Thus, no rules need to be defined as to when a transmitter aggregates frames. All that needs to be defined is the format of the frames once they have been aggregated.

1.5 Omni LRP preamble usage

Add the following paragraph to subclause 12.4.3:

The first omni LRP packet in a CTA shall be sent using the short omni LRP preamble. Subsequent frames shall use the the long omni LRP preamble. All omni ACK packets shall use the short preamble. The short omni LRP preamble shall be used for frames sent in a CP. The beacon frame shall use the long omni LRP preamble.”

1.6 AV PHY HCS

The AV PHY HCS is defined in 12.4.1.4 Header check sequence. To clarify, change the paragraph in 12.4.3 as shown:

~~The HRPDU shall~~ The HRP Header field, MAC Header field, and ~~Header Check Sequence (~~HCS~~)~~ field, as defined in <xref 12.4.1.4>,~~s~~ shall be sent using HRP mode 0 modulation, as defined in Table 1.

1.7 Channel 2 mandatory

In order to make sure that AV OFDM DEVs interoperate, it is important that there is at least one common RF channel. For cost considerations, initial implementation may only implement 1 channel. Thus, we selected channel 2 as mandatory.

1.8 ACKs sent at same rate

This rule is OK for omni-ACKs, but doesn't work for HRP and directional ACKs. This also points out that there is no description of HR0, HRRX, HRTX and HRTR in the draft. So we need to add a note to multi-rate support that says

"HRP frames in the AV OFDM PHY shall be ACKed with the directional ACK, as described in 12.4.4.5."

Also, add a description of HR0, HRRX, HRTX and HRTR to the beginning of 12.4 and add two bits to the PHY capabilities field to indicate this.

1.9 Two outer encoders, Figure 187

The first pair should be RS outer encoders and the next outer interleavers. This is a cut-n-paste error.

1.10 Changing RS encoding for unified FEC scheme.

The RS encoding is an integral part of the overall PHY design, including the length of the headers to ensure that the date fits into a minimum number of symbols. The simulations and design were done in combination with the interleaver and inner encoder. One criteria for selecting the encoder was its low overhead, almost half of the RS(255,239) but, according to our simulations and measurements, provides enough protection.

1.11 Interleaver explanation

Replace 12.4.2.7 with the following:

The outer interleaver shall output the octets from $i = 0$, $k = 0$ first to $i = depth-1$, $k = N-1$ last, where *depth* is the depth of the outer interleaver and N is the lenght of RS code. With M parallel convolutional inner encoders for each RS codeword, the outer interleaver shall give the RS octets of b(0,0), …, b(*depth*-1,0) to the first convolutional encoder with lsb first. All octets of $b(i,k \times M+m)$, $i = 0, \ldots, depth-1$, $k = 0, 1, \ldots, N/M-1$, shall be output to the m^{th} convolutional encoder. The number of parallel convolutional encoders is specified in <xref 12.4.2.8>.

The outer block interleaver shall be operated with a $depth = 4$ for HRP data and a $depth = 2$ for the combination of the HRP Header, MAC Header and HCS fields. LRP modes do not use an outer block interleaver.

The combination of HRP header, MAC header and HCS field has 92 octets that is encoded into 112 octets by adding, 16 parity octets for error protection and 4 tail octets to terminate the convolutional code.

For the combination of the HRP Header, MAC Header and HCS fields, the first 48 octets are encoded using RS(56, 48, t=4) while the next 44 octets are encoded using RS(52, 44, t = 4). The second codeword is followed by 4 tail octets set to zero. The transmitted order of those octets, the method to insert the tail octets , and the method of interleaving for the outer interleaver are the same as those used for data.

1.12 Stations on page 119

Editorial mistake, should be "DEVs"

1.13 Table 166, page 122

16-QAM-TCM should be just 16-QAM

1.14 1 bit for UEP MCS

One bit for UEP for all subframes is sufficient because the MCS of each subframe uniquely identifies the type of modulation, including UEP. There are three EEP modes, mode indices 0-2, two UEP mode indices, 3-4, and two msb only retransmissions that EEP modulation, mode indices 5-6.

1.15 Long omni LRP format

Yes, this is a editorial mistake, the editor will fix the text.

1.16 SAP for video

Streaming uncompressed video has unique requirements that are not necesarily supported by the current 802.2 SAP. In particular, the follwing capabilities are required at this interface:

- Start a stream based on application requirements (throughput, latency, frame size, etc.)
- Identify the data as associated with a specific stream
- Indicate the data as one that can be passed up in error
- Specify that unequal error protection is acceptable.
- Idnetify to the receiving layer that the data is in error.

To accomplish this, one new SAP needs to be defined and another SAP modified.

Add the rows shown to Table 36 as shown.

Table 36—MAC-ISOCH-DATA and MAC-ASYNC-DATA primitive parameters

Name	Type	Range	Descripthion
DataType	Enumeration	VIDEO, AUDIO, DATA, UNCOMPRESSED_VIDEO, UNCOMPRESSED_AUDIO	Indicates the type of data that is sent in the stream.
UEPAllowed	Boolean	TRUE, FALSE	Indicates if UEP is allowed for the transmission of the data.
ErrorsFreeData	Boolean	TRUE, FALSE	Indicates if the data that was received contains errors or is free from errors.

Modify the MAC SAP with the following information.

1.16.1 MAC-ISOCH-DATA.request

Change the primitive definition as shown:

MAC-ISOCH-DATA.request (
RequestID,
StreamIndex,
TransmitTimeout,
MaxRetries,
SNAPHeaderPresent,
ACKRequested,
ConfirmRequested,
Length,
Data,
DataType,
UEPAllowed
)

1.16.2 MAC-ISOCH-DATA.indication

Change the primitive definition as shown:

MAC-ISOCH-DATA.indication (
TrgtID,
OrigID,
StreamIndex,
SNAPHeaderPresent,
Length,
Data,
DataType,
ErrorFreeData
)

Insert the following subclause after A.2

A.3 Stream SAP

The Stream SAP is used to transfer time sensitive data streams. Examples of the data intended for this interface include audio and video, particularly uncompressed audio and video. The primitives defined for this SAP are listed in Table 37.

Table 37—Stream SAP primitives

Name	Request	Confirm	Indication	Response
ST_INITIATE	A.3.1.1	A.3.1.2	A.3.1.3	X
ST_MODIFY	A.3.1.4	A.3.1.5	X	X
ST_END	A.3.1.6	A.3.1.7	A.3.1.8	X
ST_DATA	A.3.2.1	A.3.2.2	A.3.2.3	X

A.3.1 Stream creation, modification and deletion

These primitives are used to start, change or stop a stream connection. The paramters for the primitives are defined in Table 38.

Table 38—ST_INITIATE, ST_MODIFY and ST_END primitive paramters

Name	Type	Range	Descripthion
RequestID	Integer	0-255	A unique value created by the originating DEV to match the request primitive to the repsonse primitve
TargetAddress	DEVAddress	Any valid MAC address as defined in 7.1	The address of the target of the primitve.
SourceAddress	DEVAddress	Any valid MAC address as defined in 7.1	The address of the Source of the stream.
StreamIndex	Integer	Any vaild stream index, as defined in 7.2.3.	The index of the stream that was created or the index of the stream to be modified or ended.
MinThroughput	Integer	$(1\text{-}2^{64})\text{-}1$	The minimum required throughput in bits per second at the MAC SAP.
DesiredThroughput	Integer	$(1\text{-}2^{64})\text{-}1$	The desired throughput in bits per second at the MAC SAP.
MaxTransmitDelay	Duration	$(1\text{-}2^{64})\text{-}1$	The maximum delay in microseconds
TypicalFrameSize	Integer	0-pMaxFrameBodySize	The typical size, in octets, of an MSDU that would be presented to the MAC SAP.
SECMode	Boolean	TRUE, FALSE	Indicates if security is applied to the stream.
ReliabiltiyExponent	Integer	0-31	The negative power of 10 that is the that is the maximum FER including retries and frames lost to MaxTransmitDelay. For example, a value of 4 corresponds to an FER < 10^{-4}. If the value of the parameter is zero, then the parameter is ignored.

Table 38—ST_INITIATE, ST_MODIFY and ST_END primitive paramters

Name	Type	Range	Descripthion
TimeOut	Integer	0-65535	The time in milliseconds for the primitive to complete.
AvailableThroughput	Integer	$(1-2^{64})-1$	The estimate of the throughput in bits per second available in the allocated stream
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the request
ReasonCode	Enumeration	REQUEST_TIMEOUT, NOT_ASSOCIATED, TARGET_UNAVAILABLE, TERMINATED_BY_PNC, TERMINATED_BY_DEST, INVALID_STREAM_INDEX, TRANSMIT_DELAY_UNSUPPORTED RESOURCES_UNAVAILABLE, OTHER	The reason for a ResultCode of FAILURE.

A.3.1.1 ST_INITIATE.request

This primitive is used to setup a data stream. The semantics of this primitive are:

```
ST_INITIATE.request          (
                             RequestID,
                             TargetAddress,
                             MinThroughput,
                             DesiredThroughput,
                             MaxTransmitDelay,
                             TypicalFrameSize,
                             SECMode,
                             ReliabilityExponent,
                             TimeOut
                             )
```

The primitive parameters are defined in Table 38.

A.3.1.2 ST_INITIATE.confirm

This primitive is used to report the result of a request to set up a data stream. The semantics of this primitive are:

```
ST_INITIATE.confirm          (
                             RequestID,
                             StreamIndex
                             AvailableDataRate,
                             ResultCode,
                             ReasonCode
                             )
```

The primitive parameters are defined in Table 38.

A.3.1.3 ST_INITIATE.indication

This primitive is used to report that the creation of a stream by another DEV. The semantics of this primitive are:

```
ST_INITIATE.indication       (
                             SourceAddress,
                             StreamIndex
                             )
```

The primitive parameters are defined in Table 38.

A.3.1.4 ST_MODIFY.request

This primitive is used to modify the parameters of an existing data stream. The semantics of this primitive are:

```
ST_MODIFY.request            (
                             RequestID,
                             StreamIndex,
                             MinThroughput,
                             DesiredThroughput,
                             MaxTransmitDelay,
                             TypicalFrameSize,
                             SECMode,
                             ReliabilityExponent,
                             TimeOut
                             )
```

The primitive parameters are defined in Table 38.

A.3.1.5 ST_MODIFY.confirm

This primitive is used to report the result of a request to modify the parameters of an existing data stream. The semantics of this primitive are:

```
ST_MODIFY.confirm          (
                           RequestID,
                           StreamIndex
                           AvailableDataRate,
                           ResultCode,
                           ReasonCode
                           )
```

The primitive parameters are defined in Table 38.

A.3.1.6 ST_END.request

This primitive is used to setup a data stream. The semantics of this primitive are:

```
ST_END.request             (
                           StreamIndex,
                           TimeOut
                           )
```

The primitive parameters are defined in Table 38.

A.3.1.7 ST_END.confirm

This primitive is used to report the result of a request to set up a data stream. The semantics of this primitive are:

```
ST_END.confirm             (
                           StreamIndex
                           ResultCode,
                           ReasonCode
                           )
```

The primitive parameters are defined in Table 38.

A.3.1.8 ST_END.indication

This primitive is used to report that the creation of a stream by another DEV. The semantics of this primitive are:

```
ST_END.indication          (
                           StreamIndex,
                           ReasonCode
                           )
```

The primitive parameters are defined in Table 38.

A.3.2 Stream data interface

These primitives are used to send stream data and report the reception of stream data. The paramters for the primitives are defined in Table 39.

Table 39—ST_DATA primitive paramters

Name	Type	Range	Descripthion
RequestID	Integer	0-255	A unique value created by the originating DEV to match the request primitive to the repsonse primitve
SourceAddress	DEVAddress	Any valid MAC address as defined in 7.1	The address of the source of the data.
DestinationAddress	DEVAddress	Any valid MAC address as defined in 7.1	The address of the destination of the data.
StreamIndex	Integer	Any vaild stream index, as defined in 7.2.3.	The index of the stream that was created or the index of the stream to be modified or ended.
TransmitTimeout	Integer	(1-2^{32})-1	The maximum allowed delay in microseconds from when the data is presented to the SAP until the frame has finished transmission and the acknowledgement, if required, is received.
SNAPHeaderPresent	Boolean	TRUE, FALSE	TRUE indicates that an LLC/SNAP header is present in the data frame.
ConfirmRequested	Boolean	TRUE, FALSE	Indicates when a confirm primitive is required for the request.
Length	Integer	0-(1-2^{32})-1	The length of the Data in octets.
Data	Octet string		The information to be sent in the stream connectin.
DataType	Enumeration	VIDEO, AUDIO, DATA, UNCOMPRESSED_VIDEO, UNCOMPRESSED_AUDIO	Indicates the type of data that is sent in the stream.
UEPAllowed	Boolean	TRUE, FALSE	Indicates if UEP is allowed for the transmission of the data.
ErrorsFreeData	Boolean	TRUE, FALSE	Indicates if the data that was received contains errors or is free from errors.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the request
ReasonCode	Enumeration	TRANSMIT_TIMEOUT, NOT_ASSOCIATED, TARGET_UNAVAILABLE, INVALID_STREAM_INDEX, OTHER	The reason for a ResultCode of FAILURE.

A.3.2.1 ST_DATA.request

This primitive is used to send stream data to another DEV. The semantics of this primitive are:

```
ST_DATA.request                    (
                                   RequestID,
                                   StreamIndex,
                                   TransmitTimeout,
                                   SNAPHeaderPresent,
                                   ConfirmRequested
                                   Length,
                                   Data,
                                   DataType,
                                   UEPAllowed
                                   )
```

The primitive parameters are defined in Table 39.

A.3.2.2 ST_DATA.confirm

This primitive is used to send stream data to another DEV. The semantics of this primitive are:

```
ST_DATA.request                    (
                                   RequestID,
                                   StreamIndex,
                                   TransmitDelay,
                                   ResultCode,
                                   ReasonCode
                                   )
```

The primitive parameters are defined in Table 39.

A.3.2.3 ST_DATA.indication

This primitive is used to send stream data to another DEV. The semantics of this primitive are:

```
ST_DATA.request                    (
                                   SourceAddress,
                                   DestinationAddress,
                                   StreamIndex,
                                   SNAPHeaderPresent,
                                   Length,
                                   Data,
                                   DataType,
                                   ErrorFreeData
                                   )
```

The primitive parameters are defined in Table 39.