**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |
| --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) |
| Title | **Kookmin Comments on NTU’s RS-FSK modes** |
| Date Submitted | [October 2016] |
| Source | Trang Nguyen, and Yeong Min Jang (Kookmin University) |
| Re: | Combined Sorted D0 Comments (477r10) |
| Abstract | Some technical comments on NTU’s RS-FSK modes before merging with Kookmin M-FSK modes. |
| Purpose | Discuss the suggested resolution on D0 comments |
| 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. |

A brief comparison between Kookmin M-FSK modulation and NTU RS-FSK modulation:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Kookmin M-FSK** | **NTU RS-FSK** | **Merging and conflicting** |
| Preamble | fpreamble\_1 = constant  fpreamble\_2 = variable to rescale BW | fpreamble = constant | Conflicting.  But optional to merge. |
| Data frequency symbol | fpreamble\_2 upper-limits the BW for data frequencies | No relationship between fpreamble and data frequencies? | Conflicting.  But optional to merge. |
| **Frequency separation** | Δf = constant | Δf is variable | **Critical conflicting.**  **But need to be merged.** |
| **Splitter symbol** | Optional for PSDU. No splitter in sending PHY header subfield.  \* We usually do not use the guard symbol in between data symbols.  \* We optionally use 10kHz signal in between two data symbols for dual purposes: guard time and PD mode. (Please refer to our slide# 30 – doc. 15/16-0014r1). | Mandatory for PSDU (?). No splitter in sending PHY header subfields.    Splitter frequency is 7/18 aPF = 868 Hz(?). | PHY header must be supported. However, there is conflicting in selecting frequencies to send PHY header subfields.  PHY header must be sent at the lowest optical clock rate (also means data rate) among supported operating modes.  Also, change it into optional for PSDU. |
| **Frame rate variation support/**  **Asynchronous communication** | Using clock information embedded into each data symbol. A single clock information bit (Ab) leads to the division of BW into two sub-bands. Likewise, a pair of Ab bits leads to the division of BW into four sub-bands. | Division of BW into three sub-bands. This is equivalent to embedding clock information. | No conflict (supporting each other).  Can be merged. |
| **Bandwidth** | Unique BW for data bits-to-symbol mapping.  PHY header fields are sent at the same BW as PSDU. | Low BW for data bits-to-symbol mapping.  Another BW (higher BW) for PIB attributes (actually PHY header field) notification. | **Critical conflicting.**  **PHY header must be sent at the lowest data rate supported in PSDU instead of using another BW and undefined frequencies.** |
| **End symbol** | No | Yes | NTU should change the end symbol to an optional sub-field of PSDU to be compatible with existing standard. Refer to Figure 123 – IEEE 802.15.7-2011. |
| Superframe | No | Yes |  |
| MAC frame format | 6 bits overhead | 2 bytes overhead mandatory | Let’s discuss the common MAC frame format |

**Comments on FSK based methodology:**

Comment 1 (on doc 0189r2): Selection of fbase and frequency band for data mapping

What is the value of fbase? Is this a constant?

Is there any relationship between fbase and the preamble frequency?

Comment 2 (on doc 0189r2): Frequency separation – the fundamental of FSK

The original idea of FSK is that we always fix the frequency separation. It means (fi+1 –fi) = Δf is a constant where fi and fi+1 are two adjacent frequencies on the set of selected frequencies to map data.

This is not true for NTU’s mapping tables from bit to frequency-symbol (tables 3 and table 4). This is a fundamental problem to merge with Kookmin M-FSK because Kookmin has a fixed frequency separation.

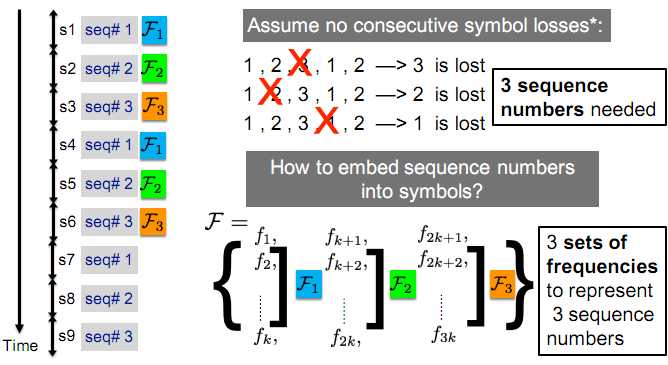
Comment 3 (on doc 0189r2): Unit of symbol time

Please clarify the duration of splitter symbol. If the duration of splitter symbol is less than the others’ symbol time, then consider the symbol duration of the splitter as a time unit. For example:

|  |  |  |  |
| --- | --- | --- | --- |
| DS 1 | SS | DS 2 | SS |
| symbol time: 2 | 1 | 2 | 1 |

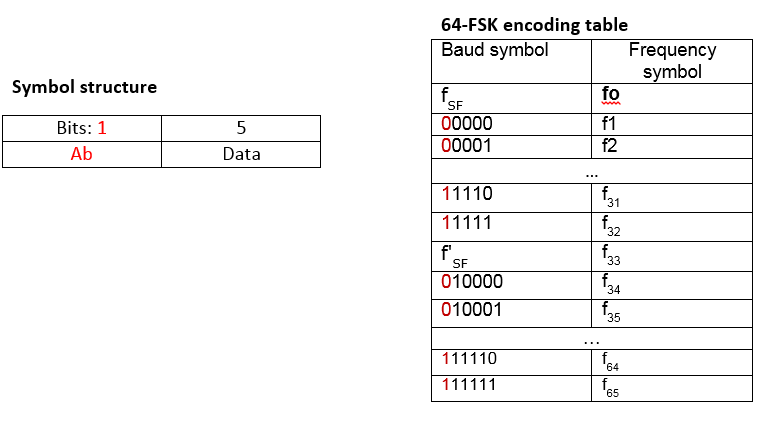
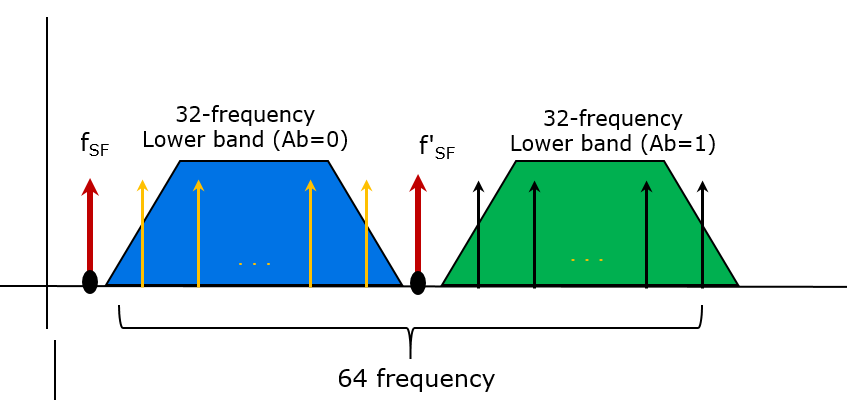
Comment 4 (on doc 0189r2): Frequency Sub-band Division = Clock information embedded to data symbol?

Embedded sequence number (ESN) into data symbol: The idea is that the bandwidth BW is divided into 3 sets of frequency: to map three adjacent data symbols.



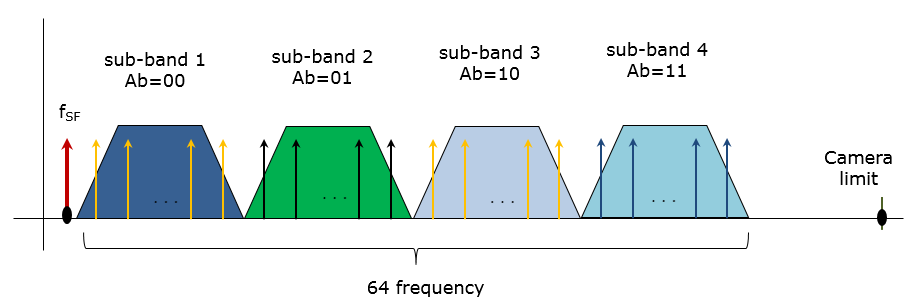
This is the same idea as embedded clock information into data symbol of Kookmin proposal. If so, we can merge this idea as a common method.

We use two sets of frequency: to map data two adjacent data symbols. This is an efficient solution to deal with frame rate variation. We assumed that the frame rate of camera Rx is no less than the symbol rate of data in Tx.

**Figure 297– Symbol structure and 64-FSK encoding table (doc 16-0460r0)**

However, we can use the same idea as an extension, the division of multiple sub-bands aims to detect the missed symbol problem (due to the frame rate drop off in Rx). For example, by using two asynchronous bits (equivalent to dividing the BW into four sub-bands), Rx is capable of detecting 2 missed symbols continuously (see Figure A.2 below). Later, the parity code (as a proposal for three sub-bands by NTU) may be used to recover the missed data symbols.

More missed symbols detectable!

**Figure A.1 Modulation with Four sub-bands division for two Ab bits clock embedding**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data symbol 1** | **Data symbol 2** | **Data symbol 3** | **Data symbol 4** | **Data symbol 5** |
| Ab=00 | Ab=01 | Ab=10 | Ab=11 | Ab=00 |
|  |  |  |  |  |
| **Camera frame #1**  **(Ab=00)** | Missed symbol: Ab=01  (detectable) | Missed symbol: Ab=01  (detectable) | **Camera frame #2**  **(Ab=11)** |  |

**Figure A.2 Demodulation with missed symbols detection using two Ab bits**

The use of two (or more) sub-bands division is equivalent to the use of one (or more) clock information bits. So, Kookmin and NTU can merge ideas into a common proposal.

Comment 5 (on doc 0189r2): Frequency Sub-band Division is mandatory?

The division of BW into multiple sub-bands is necessary to support a time-variant frame rate Rx decoding. If the support for varying frame rate Rx is mandatory, then the data mapping tables (table 3 and 4) should be spitted into multiple tables instead of one table.

For example:

Table 3.1 Data mapping using fbase\_1 to 18/11 fbase\_1

Table 3.2 Data mapping using fbase\_2 to 18/11 fbase\_2

Comment 6 (on doc 0189r2): Variable number of frequency for data mapping

As addressed in 1.0 PHY layer operating modes table 1, there is one mode for a variable number of frequency used (RS-FSK-V mode). I wonder how does Rx know how many frequencies are used in Tx to demodulate data if the number of frequencies is variable?

Also, the adaption of the number of frequency is possible. But the PHY operating mode shall be specified how many frequencies are used. RS-FSK-C8 and RS-FSK-C16 are enough to represent RS-FSK-V. The

Comment 7 (on doc 0189r2): Two bandwidths are used: one for data and another for PHY attributes?

In the document, table 3 and table 4 define the set of frequencies to map data symbols. The calculation of this low bandwidth (data frequency band) is based on fbase.

Table 6 defines another set of frequencies (higher band) to map and represent PHY attributes values. The calculation of this higher bandwidth is based on the preamble frequency.

This seems to be unreasonable. Usually, we use only a unique set of frequencies to map with data symbol. The definition of new bandwidth for PHY attributes that is different from the definition of bandwidth for data mapping is wasting resource.

We think that the author wants to introduce the PPDU format as follows (table A.1). Frequency labelling sub-field and PIB attributes sub-field that the author introduced are parts of PHY header (PHR field). IEEE 802.15.7-2011 indicated that “PHY header shall be sent at the lowest data rate among given modes”, and therefore 8-FSK is the proper selection for NTU PHY header (among 8-FSK and 16-FSK modes). No more bandwidth is necessary to transmit the content of PHY header.

**Table A.1. NTU PPDU format**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SHR** | **PHY header** | | | **PSDU** | |
| Preamble | Gap | Frequency Labelling | PIB Attributes | PSDU | End Symbol |
| symbol time: 1 | 0/1 | 0/1 | 0/1 | variable | 1 |

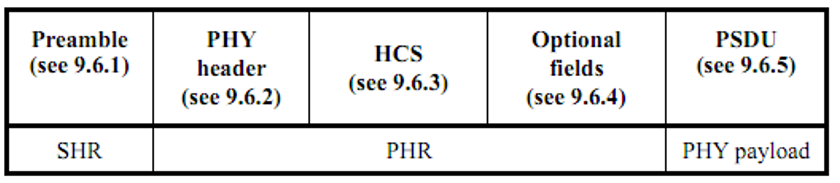
Symbol time=0 means the subfield is optional.

1. **Questions on PHY PPDU frame format:**

Comment 8 (on doc 0189r2): End symbol is not necessary

As seen in table A.1 above, the PPDU format for NTU’s RS-FSK (with end-symbol) was re-formatted compatible with IEEE 802.15.7-2011 PPDU format.

Kookmin does not use end symbol because Rx knows the ending of a packet when it receives the next preamble. No End Symbol (with one more frequency, as well as one more symbol time, are wasted) is required.



**Figure 118 – Format of PPDU (IEEE 802.15.7-2011)**

Figure 118 from IEEE 802.15.7-2011 specification did not use end symbol as a field, it was mentioned as optional tail bits of PSDU field as shown in Fig. 123 PSDU field structure.

To merge Kookmin and NTU proposals, we suggest to change it to the optional field and follow the existing IEEE 802.15.7-2011 PPDU format (Figure 186 and figure 123).

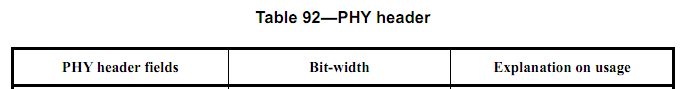
Comment 9 (on doc 0189r2): PIB attributes subfield is an incorrect name?

**Table A.2. NTU PPDU format**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SHR** | **PHY header** | | | **PSDU** | |
| Preamble | Gap | Frequency Labelling | PIB Attributes | PSDU | End Symbol |
| symbol time: 1 | 0/1 | 0/1 | 0/1 | variable | 1 |

Symbol time=0 means the subfield is optional.

Indeed the PIB Attributes subfield that NTU introduced is a subfield of PHY header. We recommend the contributor to follow the existing format from 802.15.7-2011 standard to describe the subfields of PHY header.



(from IEEE 802.15.7-2011 specification)

Comment 10 (on doc 0189r2): PHY header shall be sent at the lowest data rate among supported modes in PSDU

This way is performed in 802.15.7-2011 specification. It means NTU support two modes (RS8-FSK and RS16-FSK) then the PHY header subfields shall be sent by using RS8-FSK.

1. **Question on Superframe Structure**

Comment 11 (on doc 0189r2): Superframe structure is co-existence to the IEEE 802.15.7-2011?



Figure – NTU’s superframe structure (doc 0189r2)

According to existing 2011 specification, the first frame of a superframe is a beacon frame and “The beacons are used to synchronize the attached devices, to identify the VPAN, and to describe the structure of the superframes.”

I think it would be nice if Kookmin and NTU can collaborate to generate a complete superframe structure.

1. **Questions on MAC frame format**

Comment 12 (on doc 0189r2): Too much overhead on MAC frame format

The NTU’s general MAC frame is formatted as follow:



Figure - General MAC frame format (doc 0189r2)

With tens of bps data rate (or less than 200bps), the mandatory of one octet for Frame Control field and one octet for sequence number is too much overhead for MAC of M-FSK modes.

Kookmin MAC frame format has 6 bits overhead. Also, our MAC is designed to be compatible with existing MAC. Let us discuss the common and efficient format for MAC frame.

Comment 13 (on doc 0189r2): Frame type subfield is incompatible with existing IEEE 802.15.7-2011 frame type subfield.

The frame type subfield is mandatory. Therefore, please consider the compatibility with the frame type subfield of IEEE 802.15.7-2011.

|  |  |
| --- | --- |
| **IEEE 802.15.7-2011 frame type subfield** | **NTU frame type subfield** |
|  |  |