**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |
| --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)  |
| Title | Text proposal of LDPC |
| Date Submitted | 06/09/2018 |
| Source | Nabil Loghin, Dana Ciochina (Sony European Technology Center, Stuttgart, Germany), Seiji Kobayashi and Ryoji Ikegaya (Sony Semiconductor Solutions Corporation) | Voice: [+497115858734]E-mail: [Nabil.Loghin@sony.com] |
| Re: | P802-15-18-0147-00-004w-call-for-proposals.docx 8 (08-Mar-2018) |
| Abstract | [Proposal of LDPC]  |
| Purpose | Text proposal of LDPC as FEC |
| 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. |

**24.3.4 FEC**

The use of FEC is controlled by the PIB attribute *phyLecimFecEnabled*, as defined in 11.3.

When used, FEC shall employ for *phyLecimFecScheme* (TBD in 11.3, Table 11-2 (PHY PIB attributes)) being TRUE a rate 1/2 convolutional coding and if *phyLecimFecScheme* (TBD in 11.3, Table 11-2 (PHY PIB attributes)) being FALSE a low density parity check (LDPC) code of rate 1/4.

**24.3.4.1 Convolutional encoding**

When convolutional code is used, it shall have constraint length *K* = 7 using the following generator polynomials:



The encoder is shown in Figure 24-6, where ⊕ denotes modulo-2 addition.

Prior to the convolutional encoding of the PHR bits, as described in 24.2.2, the initial encoder state at *k* = 0

shall be set as follows:

…

(keep text for conv. code as is until: )



The sequence shown in Figure 24-7 shall be passed to the convolutional encoder.

**24.3.4.2 LDPC encoding**

When LDPC code is used, PHR bits and PSDU bits are concatenated and extended by a sequence of pad bits as shown in Figure 24.7.

pad bits

PSDU bits bits

PHR bits bits

Figure 24-7 – PHR and PSDU extension prior to encoding

The pad bits shall be set to zero, and guarantee that the total number of input bits for the LDPC encoder are a multiple of 184 bits. This sequence is then split into information sequences of length *Kldpc* = 184 bits (23 octets).

A systematic binary LDPC code of rate 1/4 appends *Mldpc* = 552 parity bits to each information sequence. The LDPC has quasi-cyclic structure (information part) and dual staircase (parity part), i.e., parities shall be accumulated (see below). Encoding is as follows:

* first *Kldpc* = 184 code bits shall equal information bits, the remaining *Mldpc* = 552 parity bits are *p0 = p1 = p2 = ... = pMldpc −1* are described below
* initialize *p0 = p1 = p2 = ... = pMldpc −1* = 0
* accumulate the first information bit, *i*0, at parity bit addresses specified in the first row of table 24-3. For example, (all additions are in GF(2)):

* for the next 7 information bits, *im*, *m* =1, 2, ..., 7, accumulate *im* at parity bit addresses {*x* + (*m* mod 8)×*Qldpc*} mod *Mldpc*, where x denotes the address of the parity bit accumulator corresponding to the first bit *i*0, and *Qldpc* = 69. So for example for information bit *i*1, the following operations are performed:

* for the 9th information bit *i*8, the addresses of the parity bit accumulators are given in the second row of table 24-3. In a similar manner the addresses of the parity bit accumulators for the following 7 information bits *im, m* = 8, 9, ..., 15 are obtained using the formula {x + (*m* mod 8)×*Qldpc*} mod *Mldpc*, where x denotes the address of the parity bit accumulator corresponding to the information bit *i*15 , i.e. the entries in the second row of table 24-3.
* In a similar manner, for every group of 8 new information bits, a new row from the table 24-3 is used to find the addresses of the parity bit accumulators.

After all of the information bits are exhausted, the final parity bits are obtained by accumulation as follows:

* Sequentially perform the following operations starting with *i* = 1:

*Mldpc* −1

* Final content of *pi* , *i* = 0, 1,.., *Mldpc* −1 is equal to the parity bit *pi.*

Table 24-3: Parity bit addresses for LDPC code

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 7 | 90 | 172 | 209 | 359 | 401 | 420 | 483 | 487 |
| 57 | 164 | 192 | 197 | 284 | 307 | 174 | 356 | 408 | 425 |
| 22 | 50 | 191 | 379 | 385 | 396 | 427 | 445 | 480 | 543 |
| 32 | 49 | 71 | 234 | 255 | 286 | 297 | 312 | 537 | 550 |
| 30 | 70 | 88 | 111 | 176 | 201 | 283 | 322 | 419 | 499 |
| 86 | 94 | 177 | 193 | 266 | 368 | 373 | 389 | 475 | 529 |
| 134 | 223 | 242 | 254 | 285 | 319 | 403 | 496 | 503 | 534 |
| 18 | 84 | 106 | 165 | 170 | 199 | 321 | 355 | 386 | 410 |
| 129 | 158 | 226 | 269 | 288 | 316 | 397 | 413 | 444 | 549 |
| 33 | 113 | 133 | 194 | 256 | 305 | 318 | 380 | 507 |  |
| 317 | 354 | 402 |  |  |  |  |  |  |  |
| 53 | 64 | 374 |  |  |  |  |  |  |  |
| 83 | 314 | 378 |  |  |  |  |  |  |  |
| 162 | 259 | 280 |  |  |  |  |  |  |  |
| 166 | 281 | 486 |  |  |  |  |  |  |  |
| 185 | 439 | 489 |  |  |  |  |  |  |  |
| 119 | 156 | 224 |  |  |  |  |  |  |  |
| 26 | 62 | 244 |  |  |  |  |  |  |  |
| 8 | 246 | 482 |  |  |  |  |  |  |  |
| 15 | 72 | 91 |  |  |  |  |  |  |  |
| 43 | 69 | 390 |  |  |  |  |  |  |  |
| 127 | 186 | 506 |  |  |  |  |  |  |  |
| 55 | 81 | 412 |  |  |  |  |  |  |  |

**24.3.4.3 Encoding output sequence**

The corresponding output sequence of code-bits, *z,* shall be generated as follows:

…

(keep last section of current 23.3.4 section)