Project	**IEEE 802.16j Mobile Multihop Relay Task Group**
Title	**Rate Compatibility and Incremental Redundancy HARQ for 802.16j LDPC**
Date	**2006-11-06**
Source(s)	Wataru Matsumoto Toshiyuki Kuze Mitsubishi Electric Corp 5-1-1 Ofuna Kamakura, Kanagawa 2478501, Japan Voice: +81-467-41-2074 Fax: +81-467-41-2136 Matsumoto.Wataru@aj.MitsubishiElectric.co.jp Kuze.Toshiyuki@ah.MitsubishiElectric.co.jp Koon Hoo Teo, Jinyun Zhang Mitsubishi Electric Research Lab 201 Broadway Cambridge, MA 02139 USA Voice: 617-621-7527, 617-621-7595 Fax: 617-621-7550 {teo, jzhang}@merl.com
Re:	Response to the call for proposal of the 802.16j relay TG (i.e., IEEE 802.16j-06/027, “Call for Technical Proposals regarding IEEE Project P802.16j”, October 15, 2006).
Abstract	This contribution describes an enhancement to the current IEEE 802.16e LDPC.
Purpose	To enhance the current IEEE 802.16e LDPC as suggested in this contribution.
Notice	
Release	
Patent Policy and Procedures	

Rate Compatibility and Incremental Redundancy HARQ for 802.16j LDPC

***Wataru Matsumoto** ,Toshiyuuki Kuze*
Mitsubishi Electric Corp
5-1-1 Ofuna Kamakura, Kanagawa
2478501, Japan

Koon Hoo Teo, Jinyun Zhang
Mitsubishi Electric Research Lab
201 Broadway
Cambridge, MA 02139 USA

Table of contents

Rate Compatibility and Incremental Redundancy HARQ for 802.16j LDPC

Wataru Matsumoto *,Toshiyuuki Kuze*
Mitsubishi Electric Corp
5-1-1 Ofuna Kamakura, Kanagawa
2478501, Japan

Koon Hoo Teo, Jinyun Zhang
Mitsubishi Electric Research Lab
201 Broadway
Cambridge, MA 02139 USA

1. Introduction

One of the most important advantages of LDPC is its higher decoding efficiency compared to Turbo codes. Therefore, a high throughput link such as BS-RS or RS-RS could be achieved with lower hardware cost. Unfortunately, current 802.16e LDPC is not rate compatible and does not support 1/3 code rate. In addition, the current LDPC only support HARQ with chase combining but not incremental redundancy (IR).

1/3 code rate and IR HARQ are known to improve reliability or robustness in links with hostile channel conditions.

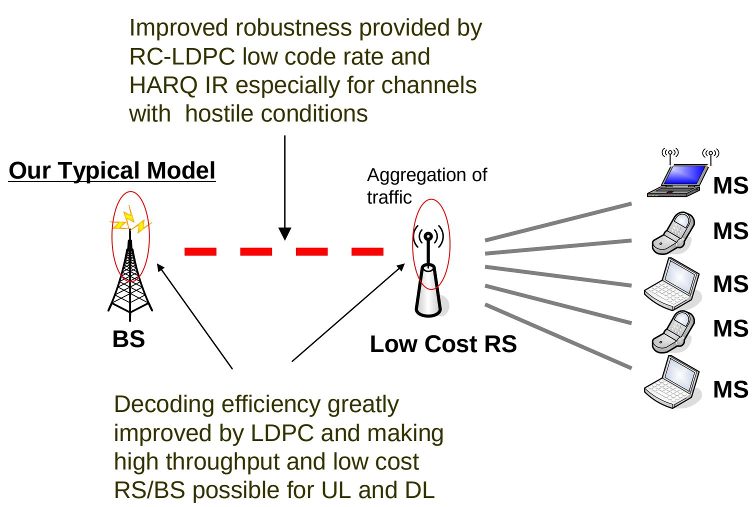

Figure 1: The merits of LDPC and RC-LDPC

We proposed an enhanced version of 802.16 LDPC using the current 802.16e LDPC as the baseline. This nhanced LDPC shown in Figure 1, which we called rate compatible RC-LDPC will support 1/3 rate code and also provide the mean to support HARQ with incremental redundancy to provide the BS-RS and RS-RS links operating in hostile channel condition, with high throughput, improved robustness and increased decoding efficiency.

2. Summary of Proposal

We proposed the construction of an extended parity-check matrix of the RC (Rate-Compatible)-LDPC codes to achieve good performance for wide range of code word length and code rate. The parity-check matrix construction method can support code rates ranging from 1/3 to 4/5 and codeword lengths ranging from 288 bits to 2304bits. The RC-LDPC codes are constructed using a code-rate 1/2 parity-check matrix for code-rate greater than or equal to 1/2. For code rate smaller than ½ an additional parity check matrix is being extended to the ½ parity-check matrix.

The proposed RC-LDPC uses the current 802.16 LDPC as the baseline matrix. It also uses the same four sub-packets protocol for its HARQ operation. The proposal provide some small additions and require no change to the current 802.16e LDPC.

3. Performance Results

	LDPC	TC	Complexity of LDPC Complexity of TC
Algorithm	LBP Min-Sum+Offset	Max Log Map +extrinsic scaling	
Number Iterations	20	8	
Total cost (R=1/3)	38.5K x 20 = 770K	171K x 8 = 1368K	56%
Total cost (R=1/2)	28.8K x 20 = 576K	171K x 8 = 1368K	42%
Total cost (R=3/4)	20.6K x 20 = 412K	171K x 8 = 1368K	30%

Table 1:Operations count comparison of sub-optimal decoders LDPC and TC decoders

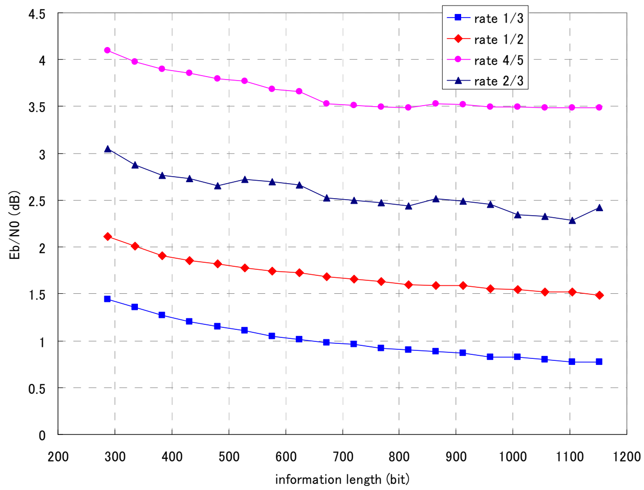

Figure 2: Performance for RC LDPC codes based on the 16e LDPC codes

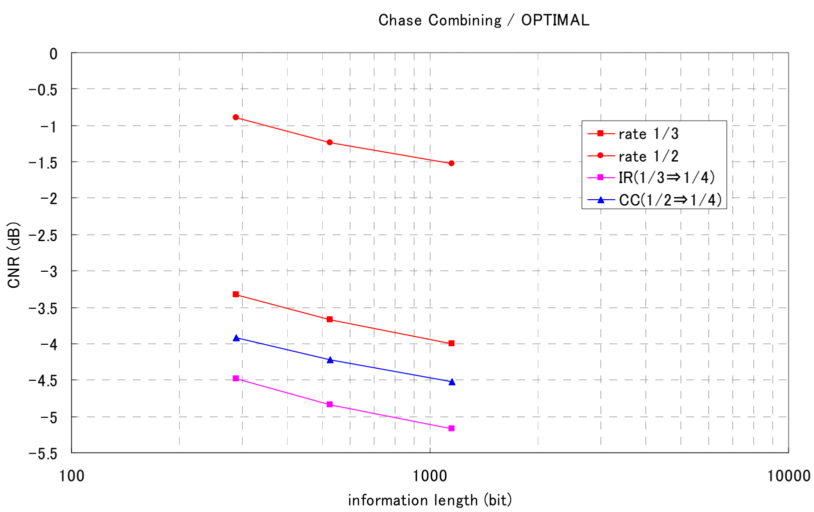

Figure 3: Performance of RC LDPC codes based on the 16e LDPC

4. Proposed Text Additions

6.3.2.3.43.7.9 MIMO Compact UL

Insert into *Table 109a – MIMO Comapct UL-MAP IE format*

Syntax	*Size*	*Notes*
If (HARQ Mode=CTC Incremental Redundany) { *If (HARQ Mode=LDPC Incremental Redundancy*	-	*HARQ Mode is specified in the HARQ compact UL_MAP IE format for Switch HARQ Mode hardware*

6.3.2.3.43.7.10 SDMA Compact UL-MAP IE format

Insert into *Table 109b – SDMA compact UL-MAP IE format*

Syntax	*Size*	*Notes*
If (HARQ Mode=CTC Incremental Redundany) { *If (HARQ Mode=LDPC Incremental Redundancy*	-	*HARQ Mode is specified in the HARQ compact UL_MAP IE format for Switch HARQ Mode hardware*

6.3.17 *No change necessary*

6.3.17.1 No change necessary

6.3.17.2 No change necessary

6.3.17.3 No change necessary

8.4.5.3.21 HARQ DL MAP IE

Insert:

d) Incremental redundancy HARQ for LDPC (HARQ LDPC-IR)

Insert into *Table 2861 – HARQ DL MAP IE format*

Syntax	*Size*	*Notes*
Mode	*4 bits*	*0b0111=Incremental redundancy HARQ for LDPC*

8.4.5.4.24 HARQ UL MAP IE

Insert:

4. Incremental redundancy HARQ for LDPC (HARQ LDPC-IR)

Insert into *Table 302j – HARQ UL MAP IE*

Syntax	*Size*	*Notes*
Mode	*3bits*	*0b111=Incremental redundancy HARQ for LDPC*

8.4.9.2.5.1 Code description

Insert new text at end of the subsection.

We propose the parity check matrix for rate-compatible LDPC(RC-LDPC) codes using the parity-check matrix for rate-1/2 specified in the 802.16e as the following Figure 4.

Extend for Code Rate=1/3

−1	94	73	−1	−1	−1	−1	−1	55	83	−1	−1	7	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	27	−1	−1	−1	22	79	9	−1	−1	−1	12	−1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	−1	24	22	81	−1	33	−1	−1	−1	0	−1	−1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
61	−1	47	−1	−1	−1	−1	−1	65	25	−1	−1	−1	−1	−1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	39	−1	−1	−1	84	−1	−1	41	72	−1	−1	−1	−1	−1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	−1	−1	46	40	−1	82	−1	−1	−1	79	0	−1	−1	−1	−1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	95	53	−1	−1	−1	−1	−1	14	18	−1	−1	−1	−1	−1	−1	−1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	11	73	−1	−1	−1	2	−1	−1	47	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
12	−1	−1	−1	83	24	−1	43	−1	−1	−1	51	−1	−1	−1	−1	−1	−1	−1	−1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	−1	−1	−1	94	−1	59	−1	−1	70	72	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	7	65	−1	−1	−1	−1	39	49	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
43	−1	−1	−1	−1	66	−1	41	−1	−1	−1	26	7	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	−1	−1	−1	17	−1	9	−1	−1	−1	20	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	36	−1	−1	−1	−1	−1	−1	7	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	5	−1	−1	−1	−1	−1	−1	5	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	−1	−1	−1	23	−1	11	−1	−1	−1	23	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1
−1	−1	−1	−1	−1	5	−1	32	−1	−1	−1	38	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1
−1	−1	7	−1	−1	−1	−1	−1	−1	19	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1
−1	−1	−1	−1	−1	31	−1	23	−1	−1	−1	48	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1
−1	−1	−1	−1	−1	29	−1	37	−1	−1	−1	7	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1
−1	−1	6	−1	−1	−1	−1	−1	−1	3	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1
−1	−1	1	−1	−1	−1	−1	−1	−1	2	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1
−1	−1	−1	−1	−1	37	−1	26	−1	−1	−1	8	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1
−1	−1	3	−1	−1	−1	−1	−1	−1	6	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	0

Figure 4 Parity check matrix for rate compatible LDPC codes

Let $\mathbf{u}$ *and* $\mathbf{r}$ *be information bits as* $\mathbf{u}=(u_1 \quad u_2 \quad \cdots \quad u_K)$ *and parity bits as* $\mathbf{r}=(r_1 \quad r_2 \quad \cdots \quad r_M)$, *respectively. And Let* $\mathbf{v}$ *be a systematic codeword such that:*
$\mathbf{v}:=(u_1 \quad u_2 \quad \cdots \quad u_K \quad r_1 \quad r_2 \quad \cdots \quad r_M)$. *satisfying* $\mathbf{H}_M \cdot \mathbf{v}^T = \mathbf{0}$.

The RC-LDPC encoder consists of a common LDPC encoder and a puncturing device. The decoder for RC-LDPC codes is the same as an ordinary LDPC decoding algorithm with received LLR=0 for puncturing bits.
A set of code rates and un-puncturing bits set $\hat{\mathbf{r}}$ *for RC-LDPC codes can be represented by:*

For the number of un-puncturing bits $i(\le M/2)$, $\hat{\mathbf{r}}=\{\hat{r}_l\}$,

$\boldsymbol{q}=\{q_y : 8,4,6,2,7,3,5,1\}$;

$x=q_0$, $y=0$;

For $l=1$ *to* i

If $x \le M/2$ *then* $\hat{r}_l = r_x$, $x = x+8$;

Else $x=q_y$, $y=y+1$;

Endif

End for

For the number of un-puncturing bits $i(> M/2)$, $\hat{\boldsymbol{r}}=\{\boldsymbol{r}_l\}$, *for* $M/2 \le l \le i$.

8.4.9.2.5.2 Code rate and block size adjustment

The LDPC code flexibly supports different block sizes for each code rate through the use of an expansion factor.

Each base model matrix has n_b=24 columns, and the expansion factor (z factor) is equal to *n*/24 for code length *n*. In each case, the number of information bits is equal to the code rate times the coded length *n*.

Insert into Table 33a – RC-LDPC block Sizes and code rates

n	*n*	~~*z*~~	*k*(bytes)					Number of subchannels		
(bits)	(bytes)	~~factor~~	R=1/3	R=1/2	R=2/3	R=3/4	R=5/6	QPSK	16QAM	64QAM
576	72	~~24~~	24	36	48	54	60	6	3	2
672	84	~~28~~	28	42	56	63	70	7	-	-
768	96	~~32~~	32	48	64	72	80	8	4	-
864	108	~~36~~	36	54	72	81	90	9	-	3
960	120	~~40~~	40	60	80	90	100	10	5	-
1056	132	~~44~~	44	66	88	99	110	11	-	-
1152	144	~~48~~	48	72	96	108	120	12	6	4
1248	156	~~52~~	52	78	104	117	130	13	-	-
1344	168	~~56~~	56	84	112	126	140	14	7	-
1440	180	~~60~~	60	90	120	135	150	15	-	5
1536	192	~~64~~	64	96	128	144	160	16	8	-
1632	204	~~68~~	68	102	136	153	170	17	-	-
1728	216	~~72~~	72	108	144	162	180	18	9	6
1824	228	~~76~~	76	114	152	171	190	19	-	-
1920	240	~~80~~	80	120	160	180	200	20	10	-
2016	252	~~84~~	84	126	168	189	210	21	-	7
2112	264	~~88~~	88	132	176	198	220	22	11	-
2208	276	~~92~~	92	138	184	207	230	23	-	-
2304	288	~~96~~	96	144	192	216	240	24	12	8

8.4.9.2.5.3.1 Packet encoding for HARQ

Insert.

The reordered codeword for RC-LDPC codes are defined such as

$$\bar{\mathbf{v}} := (\bar{v}_1 \quad \bar{v}_2 \quad \cdots \quad \bar{v}_{K+M}) = (u_1 \quad u_2 \quad \cdots \quad u_K \quad \hat{r}_1 \quad \hat{r}_2 \quad \cdots \quad \hat{r}_M).$$

Insert this subclause

8.4.9.2.5.4 Subpacket generation for LDPC

Proposed FEC structure punctures the reordered codeword to generate a subpacket with various coding rates. The subpacket is also used as HARQ packet transmission. Figure 5 shows a block diagram of subpacket generation. 1/3 LDPC encoded codeword puncturing is performed. The puncturing is performed to select the consecutive bit sequence that starts at any point of the whole reordered codeword. For the first transmission, the subpacket is generated to select the consecutive bit sequence that starts from the first bit of the systematic part of the reordered codeword. The length of the subpacket is chosen according to the needed coding rate reflecting the channel condition. The first subpacket can also be used as a codeword with the needed coding rate for a burst where HARQ is not applied.

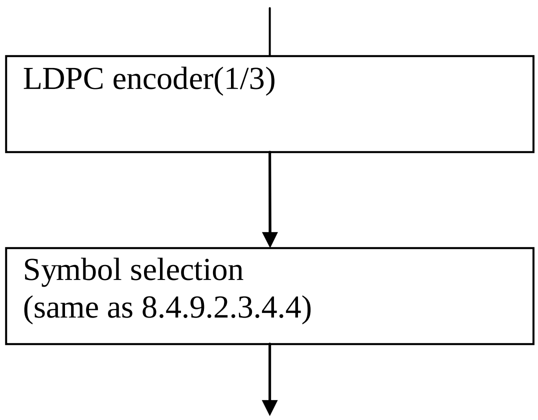

Figure 5: Block diagram of subpacket generation

Insert this subclause

8.4.9.2.5.4.1 Symbol separation

Insert.

All of the encoded symbols shall be demultiplexed into three subblocks denoted A, Y, and W as shown in Figure 6.

A, Y, and W subblocks are defined as

$$\mathbf{A} := (u_1 \quad u_2 \quad \cdots \quad u_K), \mathbf{Y} := (\hat{r}_1 \quad \hat{r}_2 \quad \cdots \quad \hat{r}_{M/2}), \text{and } \mathbf{W} := (\hat{r}_{M/2+1} \quad \hat{r}_{M/2+2} \quad \cdots \quad \hat{r}_M).$$

The encoder output symbols shall be sequentially distributed into 3 subblocks with the first encoded output symbols going to the A subblock, the second encoder output going to the Y subblock and the third to the W subblock, etc.

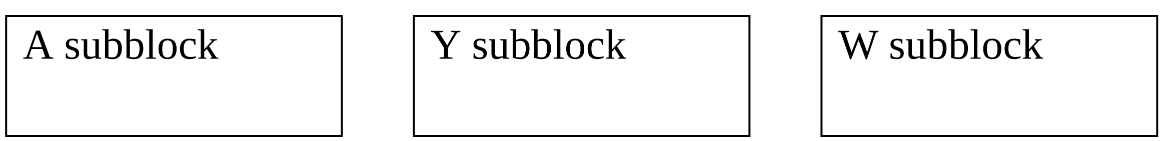

Figure 6: Subblocks for RC-LDPC codes

Insert this subclause

8.4.9.2.5.4.2 Symbol selection

Lastly, symbol selection is performed to generate the subpacket. The puncturing block is referred as symbols selection in the viewpoint of subpacket generation.

The reordered code is transmitted with one of the subpackets. The symbols in a subpacket are formed by selecting specific sequences of symbols from the LDPC encoder output sequence. The resulting subpacket sequence is a binary sequence of symbols for the modulator.

Let

k *be the subpacket index when HARQ is enabled k=0 for the first transmission and increases by one for the next subpacket. k=0 when HARQ is not used*

N_{EP} *be the number of bits in the encoder packet (before encoding)*

NSCHk be the number of subchannel(s) allocated for the k-th subpacket

m_k *be the modulation order for the kth subpacket (m_k=2 for QPSK , 4 for 16-QAM, and 6 for 64-QAM)*

S_{PIDk} *be the subpacket ID for the k-th subpacket, (for the first subpacket, $SPID_{k=0}$=0)*

Also, let the scrambled and selected symbols be numbered from zero with the 0^{th} symbole being the first symbok in the sequence. Then, the index of the i-th symbol of the kth subpacket shall be:

$S_{k,i} = (F_{k+i}) mod(3 . N_{EP})$

where

$T=0\dots L_{k-1}$,

$L_k = 48.N_{SCHk}.m_k$,

$F_k=(S_{PIDk}.L_k) mod(3.N_{EP})$

The N_{EP}, N_{SCHk}, m_k, and S_{PID} values are determined by the BS and can be inferred by the SS through the allocation size in the DL-MAP and UL-MAP. The above symbol selection makes the following possible:

1. The first transmission includes the systematic part of the reordered code. Thus it can be used as the codeword for a burst where the HARQ is not applied.
2. The location of the subpacket can be determined by the SPID itself without the knowledge of previous subpacket. It is very important property for HARQ retransmission.

8.4.9.7 Multiple HARQ (optional)

Insert into *Table 333c – HARQ Modes definition*

HARQ Mode	*Definition*
3	*LDPC Incremental Redundancy*

Insert this subclause.

8.4.9.2.5.5 Optional H-ARQ Optional IR HARQ

The procedure of HARQ LDPC subpacket generation is as follows: Padding, CRC addition, Fragmentation, Randomization and LDPC encoding. HARQ implementation is optional.

11.8.3.7.19 HARQ buffer capability

Insert.

Downlink/Uplink HARQ buffering capability indicates the maximal number of data bits the SS is able to store for downlink/uplink HARQ. The buffering capability is separately indicated for NEP/HSCH based incremental redundancy used for CTC and LDPC, and for DIUC/duration based HARQ methods (Chase combining and CC-IR) and separately for uplink and downlink transmissions.

- For incremental redundancy LDPC (NEP based): Number of bits is indicated by NEP code, according to Table 330

The IR-CTC and IR-LDPC HARQ buffer capability shall also be applied to bursts for which ACK channel is not allocated (ACK disable is set).

11.8.3.7.19.1 HARQ incremental redundancy buffer capability

Insert into Table:

Type	*Length*	*Value*	*Scope*
162	*2*	*Bits #0-3;* N_{EP} *value indicating downlink HARQ* *buffering capability for incremental redandancr* *LDPC.* *Bits #4;Aggregation Flag for DL* *Bits #5-7;Reserved* *Bits #8-11;* N_{EP} *value indicating uplink HARQ* *buffering* *capability for incremental redandancr LDPC.* *Bits #12;Aggregation Flag for UL* *Bits #13-15;Reserved*	*SBC-REQ?* *SBC-RSP?*

8.4.15 Optional HARQ support

Insert.

Incremental redundancy for LDPC codes – specified in section 6.3.17 and in 8.4.9.2.3.5

8.4.15.2.2 Optional IR HARQ for LDPC

Insert.

The following optional modes exist for HARQ.

Incremental redundancy for LDPC codes –specified in section 8.4.9.2.5.4

5. Conclusions

LDPC support high throughput with less hardware complexity and lower cost compared to Turbo Codes. Our RC-LDPC is an enhanced version of the current 802.16e LDPC. It uses the 802.16e LDPC as a baseline. RC-LDPC is rate compatible and can provide 1/3 code rate and Hybrid ARQ with incremental redundancy. For operation in very hostile channel conditions as shown in Figure 7, such as those encountered by mobile RS or non-LOS fixed RS and nomadic RS, RC-LDPC would be able to provide improved robustness to the BS-RS or RS-RS links.

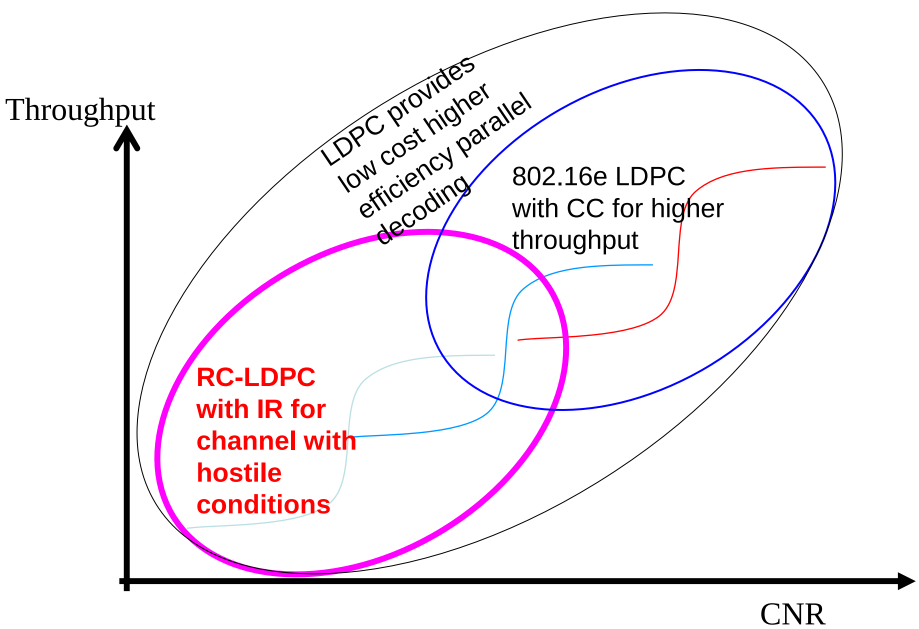

Figure 7: RC-LDPC is best for hostile channel conditions