Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**High girth LPDC coding for OFDMA PHY**
Date Submitted	**2004-11-03**
Source(s)	Robert Xu, Liuqing Yuan, Yougang Zhang, Li Zeng ZTE Corp. 3/F, Bldg.711, Pengji Industrial Park, Liantang, Shenzhen, 518004, P.R.China Voice: +86 755 26773000 6574 Fax: +86 755 26773000 6616 mailto: xu.jun2@zte.com.cn mailto: zhang.yougang@zte.com.cn
Re:	IEEE P802.16e/D5-2004
Abstract	High girth LDPC codes design technologies are introduced into the contribution for the first time, to overcome the "error floor " phenomenon, and enhance the BER curve speed when SNR is high. Also it has a performance improvement over the LDPC codes proposed in the previous contribution "IEEE C802.16e-04/373r1", and the latter is supported by 6 of the 8 companies that provided LDPC proposals to an informal LDPC group.
Purpose	To incorporate the text modification proposed in this contribution into IEEE 802.16 standard
Notice	
Release	
Patent Policy and Procedures	

High girth LPDC coding for OFDMA PHY

Robert Xu , LiuQing Yuan, Yougang Zhang, Li Zeng
ZTE Corp.

1. Overview

Many excellent code designs have been submitted. The codes have been qualitatively and quantitatively characterized, and it is clear that a LDPC code with excellent flexibility and performance, as well as low encoding and decoding complexity, can be defined for 802.16e. As we all know, the code rate , codeword length and degree distribution decide the performance of LDPC codes.

An informal LDPC group has been working on the goal of achieving consensus on a proposed LDPC code design as an optional advanced code for the OFDMA PHY. We would like to support their work, so we apply our technologies to the contribution “IEEE C802.16e-04/373r1” produced by the group, then this contribution has been created.

Based on the contribution “IEEE C802.16e-04/373r1” which contains a harmonized LDPC code definition agreed to by 6 of the 8 companies that provided LDPC proposals to an informal LDPC group, we design high girth LDPC codes . The base matrix ‘s dimension and the degree distribution of these LDPC codes is the same to this contribution “IEEE C802.16e-04/373r1”. That is, the part of check matrix corresponding to systematic bits has a regular structure, row weight and column weight of the part all are 4 ,the part of check matrix corresponding to parity bits has a dual-diagonal structure.

As we all know, code rate, codeword length and degree distribution decide the performance of LDPC codes. When Message Passing algorithm is used, the short cycles in the bipartite of LDPC codes will obviously degrade the performance of the LDPC codes, especially when SNR is high. Girth was defined as the length of the shortest cycle of the bipartite of LDPC codes, and it has become a criterion on the performance of LDPC codes. During the decoding iteration process, the extrinsic information from one variable node always returns to itself. Some variable nodes are dependent on each other. The higher the girth of one LDPC code is, the more iteration times that the extrinsic information from one variable node return to itself needs, and the less extrinsic information from one variable node returns to itself, so more independent the variable nodes will be. Thus it is very important to construct high girth LDPC codes to satisfy the requirement of Message passing algorithm.

High girth LDPC codes try to overcome the “error floor” phenomenon, and the BER curve of them will descend more steeply. However, normal LDPC codes have the “error floor” phenomenon, BER curve descends more and more slow. It is always difficult to arrive at the point BER = 10e-6, which efficient data communication needs. So high girth LDPC codes are suitable for the situation where low BER is needed.

By changing base matrix for each specific code rate to design high girth LDPC codes, performance improvement is obtained. Our method not only can be used to design regular LDPC codes, but also can be used to design irregular LDPC codes. In addition, any matrix structure with given code rate, codeword length and degree distribution can be obtained by our method, not only those given in [1], which largely increases the feasibilities of our method.

2. Recommended Text Changes:

Add/Modify the following text to 802.16e_D4, adjusting the numbering as required:

8.4.9.2 Encoding

<add text to the end of the 'Concatenation' paragraph starting at line 39>, and for the LDPC encoding scheme (see 8.4.9.2.5) the concatenation rule is defined in 8.4.2.9.5.4.

8.4.9.2.5 Low Density Parity Check Code (optional)

8.4.9.2.5.1 Code Description

The LDPC code is based on a set of one or more fundamental LDPC codes. Each of the fundamental codes is a systematic linear block code. Using the described methods of scaling and shortening in 8.4.9.2.5.3 Code Rate and Block Size Adjustment, the fundamental codes can accommodate various code rates and packet sizes. The code set can be applied to packets from [40] bytes up to ~200 bytes.

Each LDPC code in the set of LDPC codes is defined by a matrix **H** of size *m*-by-*n*, where *n* is the length of the code and *m* is the number of parity check bits in the code. The number of systematic bits is *k*=*n*-*m*.

The matrix **H** is defined as

$$\mathbf{H}=\begin{bmatrix} \mathbf{P}_{0,0} & \mathbf{P}_{0,1} & \mathbf{P}_{0,2} & \cdots & \mathbf{P}_{0,n_b-2} & \mathbf{P}_{0,n_b-1} \\ \mathbf{P}_{1,0} & \mathbf{P}_{1,1} & \mathbf{P}_{1,2} & \cdots & \mathbf{P}_{1,n_b-2} & \mathbf{P}_{1,n_b-1} \\ \mathbf{P}_{2,0} & \mathbf{P}_{2,1} & \mathbf{P}_{2,2} & \cdots & \mathbf{P}_{2,n_b-2} & \mathbf{P}_{2,n_b-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \mathbf{P}_{m_b-1,0} & \mathbf{P}_{m_b-1,1} & \mathbf{P}_{m_b-1,2} & \cdots & \mathbf{P}_{m_b-1,n_b-2} & \mathbf{P}_{m_b-1,n_b-1} \end{bmatrix}=\mathbf{P}^{H_b}$$

where $\mathbf{P}_{i,j}$ is one of a set of *z*-by-*z* permutation matrices or a *z*-by-*z* zero matrix. The matrix **H** is expanded from a binary base matrix $\mathbf{H}_b$ of size m_b-by-n_b, where $n = z \cdot n_b$ and $m = z \cdot m_b$, with *z* an integer ≥ 1. The base matrix is expanded by replacing each 1 in the base matrix with a *z*-by-*z* permutation matrix, and each 0 with a *z*-by-*z* zero matrix. The base matrix n_b is an integer is an integer multiple of 24.

The permutations used are circular right shifts, and the set of permutation matrices contains the $z \times z$ identity matrix and circular right shifted versions of the identity matrix. Because each permutation matrix is specified by a single circular right shift, the binary base matrix information and permutation replacement information can be combined into a single compact model matrix $\mathbf{H}_{bm}$. The model matrix $\mathbf{H}_{bm}$ is the same size as the binary base matrix $\mathbf{H}_b$, with each binary entry (i,j) of the base matrix $\mathbf{H}_b$ replaced to create the model matrix $\mathbf{H}_{bm}$. Each 0 in $\mathbf{H}_b$ is replaced by a blank or negative value (e.g., by ¨C1) to denote a $z \times z$ all-zero matrix, and each 1 in $\mathbf{H}_b$ is replaced by a circular shift size $p(i,j) \geq 0$. The model matrix $\mathbf{H}_{bm}$ can then be directly expanded to **H**. $\mathbf{H}_b$ is partitioned into two sections, where $\mathbf{H}_{b1}$ corresponds to the systematic bits and $\mathbf{H}_{b2}$ corresponds to the parity-check bits, such that $\mathbf{H}_b = \left[(\mathbf{H}_{b1})_{m_b \times k_b} \;\vdots\; (\mathbf{H}_{b2})_{m_b \times m_b} \right]$. Section $\mathbf{H}_{b2}$ is further partitioned into two sections, where vector $\mathbf{h}_b$ has odd weight, and $\mathbf{H}'_{b2}$ has a dual-diagonal structure with matrix elements at row *i*, column *j* equal to 1 for *i*=*j*, 1 for *i*=*j*+1, and 0 elsewhere:

$$\mathbf{H}_{b2} = \left[\mathbf{h}_b \;\vdots\; \mathbf{H}'_{b2}\right] = \left[\begin{array}{c|cccccc} h_b(0) & 1 & & & & & \\ h_b(1) & 1 & 1 & & 0 & & \\ \cdot & & 1 & \ddots & & & \\ \cdot & & & \ddots & 1 & & \\ \cdot & & 0 & & 1 & 1 & \\ h_b(m_b-1) & & & & & 1 & \end{array}\right].$$

The base matrix has $h_b(0)$=1, $h_b(m$-1)=1, and a third value $h_b(j)$, $0<j<(m_b$-1) equal to 1. The base matrix structure avoids having multiple weight-1 columns in the expanded matrix. In particular, the non-zero submatrices are circularly right shifted by a particular circular shift value. Each 1 in $\mathbf{H}'_{b2}$ is assigned a shift size of 0, and is replaced by a $z \times z$ identity matrix when expanding to **H**. The two 1s located at the top and the bottom of $\mathbf{h}_b$ are assigned equal shift sizes, and the third 1 in the middle of $\mathbf{h}_b$ is given an unpaired shift size.

Model Matrix Set

Three block semi-regular model matrices $\mathbf{H}_{bm}$ are defined, one each for rates 1/2, 2/3, and 3/4. For each code rate, $\mathbf{H}_{bm}$ will be changed , and j0,j1,j2,j3 ⋯ indicate the variable values in the base matrix $\mathbf{H}_{bm}$. For each codeword length, j0, j1, j2, j3 ⋯ will have different values, in order to construct high girth LDPC codes. For rate 1/2, when $z < 80$, designed LDPC codes' girth all are 8; when $z \geq 80$, designed LDPC codes' girth all are 10.
For rate 2/3 and 3/4 , designed LDPC codes' girth all are 6.

Rate 1/2:

$\mathbf{H}_{bm} =$

12	-1	-1	11	-1	-1	j8	-1	j16	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
4	-1	-1	-1	j0	-1	-1	-1	-1	-1	j24	j28	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	14	19	-1	-1	-1	-1	j12	j17	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	15	-1	j4	-1	j13	-1	-1	j25	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	j5	-1	-1	-1	j20	-1	j29	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1
-1	-1	4	-1	j1	-1	j9	-1	-1	j21	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1
-1	2	-1	-1	j2	-1	-1	j14	-1	-1	-1	j30	0	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1
3	-1	-1	-1	-1	j6	-1	j15	-1	j22	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1
-1	-1	20	-1	-1	j7	j10	-1	-1	-1	-1	j31	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1
-1	-1	-1	23	j3	-1	j11	-1	-1	-1	j26	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1
-1	-1	0	5	-1	-1	-1	-1	j18	j23	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0
2	7	-1	-1	-1	-1	-1	-1	j19	-1	j27	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0

z	j0,j1,j2,j3,⋯,j31
24	14,3,1,9,14,11,6,10,2,12,5,7,12,13,6,23,16,9,9,1,19,10,16, 4,19,1,19,20,0,5,21,20
28	16,25,23,3,2,13,16,8,4,12,27,5,2,26,25,2,13,3,23,17,18,15, 23,13,11,27,16,6,17,0,19,0
32	22,4,18,19,14,23,14,29,0,1,27,14,29,31,24,20,7,5,15,23,7, 10,25,0,19,9,15,30,4,11,5,28
36	21,28,17,21,14,0,3,2,9,26,12,7,14,3,34,18,25,10,9,14,22,2 9,27,19,25,13,14,31,22,1,24,14
40	33,8,34,36,17,29,33,2,16,19,1,14,32,3,27,6,1,11,25,16,32, 28,13,37,2,24,39,38,5,9,2,28
44	8,31,24,23,16,25,38,36,37,11,18,35,8,36,41,38,18,11,40,3 1,3,21,33,40,39,21,24,1,26,21,0,30
48	38,20,6,21,25,46,36,25,11,31,35,0,19,47,25,32,24,45,19,3 2,12,29,11,35,37,40,28,3,36,22,43,5
52	4,20,30,17,16,22,45,20,8,47,25,40,35,21,48,32,32,42,21,1 3,43,35,4,15,39,50,30,45,50,19,3,36
56	30,24,1,35,23,15,41,5,50,38,37,29,54,50,8,6,3,28,17,48,4,

	47,23,44,50,53,18,7,7,24,18,47
60	52,55,18,24,6,3,35,24,30,42,4,7,14,29,28,6,15,0,12,49,14,33,50,9,16,56,59,13,38,27,40,1
64	10,58,22,0,32,36,12,6,1,25,4,14,3,31,34,14,29,23,38,7,62,29,3,31,51,28,10,0,22,59,19,60
68	43,18,8,47,21,33,5,6,63,22,26,0,44,58,34,12,35,61,64,66,51,43,37,46,3,45,47,17,49,60,43,2
72	37,61,38,42,52,29,19,71,24,44,66,39,44,45,5,20,52,70,34,13,3,14,27,29,42,38,71,61,2,17,42,47
76	57,72,32,16,50,71,67,65,73,52,60,38,14,36,39,12,41,61,23,24,35,52,23,61,8,73,48,46,31,50,58,23
80	52,18,68,70,24,0,71,23,9,58,48,7,53,61,2,26,15,48,52,17,60,72,20,10,64,2,69,19,21,43,22,21
84	2,34,45,23,21,61,86,48,6,9,10,82,58,12,15,18,30,3,63,7,78,51,64,4,2,30,16,63,30,81,36,15
88	7,49,35,4,74,6,41,29,65,39,49,48,7,65,14,18,55,53,84,57,73,28,17,65,1,84,79,16,42,59,34,72
92	22,55,78,50,66,72,7,10,59,42,28,5,55,41,25,87,8,43,56,86,14,51,79,53,85,69,78,42,8,60,9,32
96	23,20,27,19,83,55,41,95,27,87,42,44,12,68,45,27,85,18,86,40,25,89,56,71,31,24,26,41,62,81,22,44

Rate 2/3:

$\mathbf{H}_{bm} = [\mathbf{H}_{bm1}\ \mathbf{H}_{bm2}]$

$\mathbf{H}_{bm1} =$

```
18  14  -1  -1  j0  -1  j8 j12  -1  -1 j24  -1 j32 j36  -1  -1
-1   3  13   4  -1  -1  -1 j13 j16 j20  -1  -1 j33  -1 j40  -1
 3  -1  -1   3  j1  -1  j9  -1 j17  -1  -1 j28  -1 j37  -1 j44
17  21   8  -1  -1  j4  -1  -1 j18  -1 j25 j29 j34  -1  -1  -1
-1  -1  -1  11  j2  j5  -1 j14  -1 j21  -1 j30 j35  -1  -1 j45
22  -1  11  -1  -1  j6 j10 j15  -1  -1  -1 j31  -1  -1 j41 j46
-1   9  -1  21  -1  j7 j11  -1  -1 j22 j26  -1  -1 j38 j42  -1
-1  -1   5  -1  j3  -1  -1  -1 j19 j23 j27  -1  -1 j39 j43 j47
```

$\mathbf{H}_{bm2} =$

```
 0   0  -1  -1  -1  -1
-1   0   0  -1  -1  -1
-1  -1   0   0  -1  -1
-1  -1  -1   0   0  -1
 0  -1  -1  -1   0   0
-1  -1  -1  -1  -1   0
-1  -1  -1  -1  -1  -1
 0  -1  -1  -1  -1  -1
```

z	j0,j1,j2,j3,⋯,j47
24	9,13,7,22,16,1,3,9,7,2,4,21,0,7,8,0,0,1,4,0,7,22,4,16,0,12,23,19,3,4,1,14,6,11,17,3,4,11,20,18,9,9,16,17,19,16,4,17
28	13,11,15,21,18,1,0,18,20,1,12,13,0,11,20,18,2,15,5,18,6,11,8,13,26,9,1,16,1,12,4,24,16,1,17,27,14,1,26,0,12,3,25,14,23,4,26,7
32	18,14,7,15,20,3,15,18,14,25,31,8,28,27,29,2,27,3,15,22,3,29,16,24,3,0,22,25,3,9,10,27,3,16,28,5,25,22,0,13,17,20,26,3,4,20,15,26
36	9,32,6,35,5,11,28,33,1,7,9,21,1,31,20,30,2,7,10,33,2,6,34,7,3,19,1,30,11,16,31,28,3,17,1,17,9,14,18,23,11,28,30,31,7,12,35,24
40	32,16,33,5,29,36,28,3,3,34,10,32,22,39,33,10,39,0,20,30,30,33,30,27,18,11,3,15,16,7,5,29,18,13,32,6,18,16,1,0,36,16,4,2,23,11,16,1
44	38,2,6,8,14,5,34,14,43,17,8,15,16,43,19,12,28,5,26,7,6,41,9,30,12,8,3,29,32,43,8,20,11,14,43,25,6,43,26,40,21,26,13,42,14,5,12,24
48	10,8,37,15,5,11,23,32,40,21,12,25,21,40,15,4,22,13,0,44,21,37,21,9,14,26,21,12,34,26,10,28,44,13,6,25,44,39,26,47,21,20,41,7,39,3,38,6

52	7,9,49,4,30,11,13,20,3,24,4,27,20,49,42,38,24,10,0,11,2,27,13,28,45,22,6,7,24,30,26,38,7,2,37,48,43,32,13,46,2,15,16,47,10,38,0,6
56	48,54,0,30,18,37,4,18,53,6,19,30,8,48,41,19,55,33,18,3,7,22,7,29,15,1,3,6,50,1,14,30,12,47,29,11,26,36,34,27,3,8,41,13,24,33,52,23
60	49,54,31,38,9,43,11,46,9,13,26,37,6,15,49,45,3,1,8,31,41,3,21,49,7,5,38,23,3,32,14,21,42,5,21,43,5,59,38,32,4,23,37,1,54,14,0,19
64	34,57,21,9,2,28,16,56,17,48,11,16,18,43,29,42,51,38,41,52,41,12,14,13,0,24,47,34,3,52,8,28,15,52,44,17,34,2,18,21,43,39,17,5,2,52,3,23
68	19,10,66,44,42,11,28,58,3,27,21,65,67,9,18,10,67,65,21,44,59,7,12,67,50,66,26,64,66,26,65,61,21,51,54,2,0,3,57,29,51,17,40,32,59,23,25,4
72	3,34,19,59,68,2,20,44,30,8,3,59,67,5,19,14,16,71,19,13,65,14,67,48,9,0,44,56,7,68,17,53,69,68,19,67,0,34,21,5,0,59,5,31,60,30,46,57
76	2,48,39,69,48,16,54,69,14,70,5,37,2,16,50,19,65,1,29,24,63,36,40,25,33,0,14,48,75,8,50,24,70,16,62,26,9,28,36,6,55,14,18,9,11,58,5,73
80	65,45,26,10,75,22,19,11,31,29,6,3,0,13,37,9,31,67,34,68,21,76,15,23,54,17,44,47,15,2,73,68,39,77,1,20,39,13,2,7,35,19,6,39,3,37,19,68
84	6,4,78,20,22,51,18,56,34,25,78,16,1,51,9,79,1,79,62,73,23,28,68,64,10,12,12,32,2,71,73,17,26,5,1,15,50,12,5,13,79,27,57,7,2,82,44,5
88	20,70,25,74,20,10,43,67,64,70,43,23,26,84,68,18,8,9,11,18,40,27,75,0,42,83,25,43,19,76,72,33,9,80,0,15,46,60,1,33,80,81,51,47,80,81,25,45
92	18,15,78,37,37,61,83,24,18,55,31,69,54,55,19,85,90,66,47,0,43,58,0,85,3,78,43,89,23,5,15,14,11,57,24,89,21,74,78,6,12,84,88,26,87,31,33,56
96	40,38,22,24,33,10,11,33,38,16,5,39,6,17,20,19,44,28,2,35,92,29,46,80,94,40,41,8,83,29,17,56,85,16,27,73,30,53,90,8,71,10,64,90,4,82,56,14

Rate 3/4:

$\mathbf{H}_{bm} = [\mathbf{H}_{bm1}\ \mathbf{H}_{bm2}]$

$\mathbf{H}_{bm1} =$

```
-1   0  13  12  -1  -1  j0  -1  j8 j12  -1  -1 j24 j28  -1  -1 j40  -1  -1 j52 j56  -1 j64  -1
13  -1   1  -1  12  -1  j1  -1  j9  -1 j16  -1  -1  -1 j32 j36  -1 j44 j48 j53 j57  -1  -1  -1
-1   1  -1   1   0  -1  -1  j4 j10  -1  -1 j20 j25  -1 j33  -1 j41 j45  -1  -1  -1 j60 j65  -1
 4   8   5  -1  -1   4  -1  -1  -1 j13 j17 j21  -1 j29 j34  -1  -1  -1  -1 j54  -1 j61 j66  -1
-1  -1  -1   7   8   2  j2  -1  -1  -1 j18 j22  -1  -1  -1 j37 j42  -1 j49 j55  -1 j62  -1 j68
 1  -1  12   3  -1  -1  -1  j5  -1 j14  -1 j23 j26  -1  -1 j38  -1 j46 j50  -1 j58  -1  -1 j69
-1   2  -1  -1  16   0  j3  j6  -1 j15  -1  -1  -1 j30 j35  -1 j43  -1 j51  -1 j59  -1  -1 j70
15  -1  -1  -1  -1   6  -1  j7 j11  -1 j19  -1 j27 j31  -1 j39  -1 j47  -1  -1  -1 j63 j67 j71
```

$\mathbf{H}_{bm2} =$

```
 0   0  -1  -1  -1  -1  -1  -1
-1   0   0  -1  -1  -1  -1  -1
-1  -1   0   0  -1  -1  -1  -1
-1  -1  -1   0   0  -1  -1  -1
 0  -1  -1  -1   0   0  -1  -1
-1  -1  -1  -1  -1   0   0  -1
-1  -1  -1  -1  -1  -1   0   0
 0  -1  -1  -1  -1  -1  -1   0
```

z	j0,j1,j2,j3,⋯,j71
18	11,6,16,17,12,16,9,2,16,12,14,9,7,11,7,3,2,0,9,14,16,14,2,5,14,11,2,4,12,17,16,3,4,11,6,1,11,6,16,15,17,11,9,14,6,17,4,2,9,9,3,9,1,11,7,2,4,5,8,11,13,16,6,14,1,4,1,4,8,5,3,5
21	1,17,20,12,19,10,16,17,15,6,14,9,14,4,2,12,4,10,13,14,6,17,10,4,7,5,2,12,14,19,3,11,8,10,9,2,11,0,5,11,14,11,11,10,10,14,7,8,5,3,12,8,8,15,5,9,19,3,11,14,13,18,12,18,2,18,1,11,18,3,0,2
24	5,14,16,9,7,2,1,6,5,15,17,8,5,7,0,20,14,11,21,8,3,16,10,16,5,15,23,23,22,13,15,3,6,4,13,17,6,3,19,4,21,11,4,8,6,20,14,15,8,14,22,2,12,1,17,13,20,21,12,21,21,0,17,10,4,20,18,2,9,14,16,11
27	14,24,0,3,26,20,16,23,11,6,2,22,1,8,20,9,13,3,12,8,18,16,5,21,23,25,24,3,14,24,5,19,6,5,8,11,12,21,1,6,18,6,5,17,24,10,23,6,6,11,3,8,23,5,21,5,13,19,9,1,1,12,11,18,22,12,16,25,11,1,13,17
30	4,1,6,21,26,29,4,9,8,1,6,27,3,14,22,16,14,3,24,4,10,20,2,20,3,25,10,13,0,7,2

	2,21,11,18,29,16,29,2,14,4,27,2,28,17,12,9,15,10,15,5,2,10,15,5,11,5,25,5,13,19,13,1,7,5,9,23,21,1,6,4,15,1
33	28,21,30,29,23,26,16,28,20,3,23,24,23,1,6,32,20,22,3,23,11,22,2,20,17,19,15,3,0,16,21,28,31,30,16,18,11,31,30,5,27,4,23,13,2,21,1,21,4,14,24,18,8,3,2,5,8,4,10,25,5,7,16,11,21,3,9,1,26,21,31,18
36	0,12,17,7,13,28,5,9,0,5,13,16,12,5,30,32,2,8,0,9,35,8,1,32,17,3,1,32,8,12,35,20,34,32,19,1,34,34,24,8,12,14,15,22,29,32,32,5,4,15,17,18,33,35,0,24,12,25,5,25,32,16,35,22,2,14,7,6,17,32,29,7
39	3,11,14,17,35,24,16,26,9,35,29,15,6,15,27,1,26,12,7,34,13,15,36,17,19,31,27,14,6,13,13,2,31,37,9,31,0,22,13,18,1,0,7,29,11,32,23,6,1,10,22,27,19,16,29,6,13,8,10,19,7,34,28,11,4,14,37,33,24,1,16,18
42	38,28,14,11,2,38,2,7,38,16,3,7,39,32,1,20,31,32,8,36,0,18,0,27,24,8,32,6,41,23,8,22,15,16,9,41,14,4,37,13,24,30,4,23,14,3,38,3,14,5,8,5,7,24,9,38,15,1,1,41,17,23,26,39,11,15,37,2,24,4,35,34
45	22,3,8,31,31,36,19,28,26,4,42,40,33,43,23,13,14,17,43,1,17,5,33,8,31,38,10,8,44,24,9,11,27,37,36,19,11,4,14,42,37,27,31,40,16,34,31,25,42,22,7,14,7,1,16,40,11,35,7,41,11,4,36,0,29,42,27,25,36,28,9,38
48	4,26,9,31,1,34,47,22,12,27,14,9,43,12,18,4,8,9,34,32,47,39,12,25,6,22,20,37,18,30,19,29,17,44,9,37,17,19,38,29,6,1,39,26,43,0,24,40,11,8,18,37,18,35,8,40,41,7,21,22,34,44,35,6,17,34,40,34,24,39,7,39
51	1,19,24,12,35,13,8,50,1,20,45,38,38,1,32,15,42,12,29,21,23,41,18,50,8,40,43,20,50,19,6,38,1,19,4,26,29,0,36,49,19,30,40,36,3,20,40,15,18,11,46,31,1,21,17,6,43,34,17,6,15,16,34,3,29,32,22,21,21,49,14,42
54	9,18,46,38,30,17,3,10,4,43,7,26,13,0,47,24,52,53,17,43,5,4,18,48,17,7,41,3,12,7,20,48,13,2,40,25,20,2,51,35,12,31,30,36,20,32,50,37,30,15,39,46,12,20,12,17,12,42,20,35,30,49,20,19,20,20,17,15,42,11,36,9
57	1,37,20,43,37,15,34,28,17,45,32,15,21,46,2,32,41,19,50,18,47,13,6,22,47,21,17,44,51,7,27,26,35,18,4,48,54,39,13,21,6,41,8,3,53,4,48,43,18,5,26,48,20,12,6,51,50,19,17,3,37,25,9,26,6,5,29,21,53,14,12,24
60	11,5,4,22,12,25,7,12,10,46,13,16,13,29,57,22,10,54,47,7,5,16,6,3,5,49,15,53,25,6,1,27,25,12,57,40,5,21,41,30,3,59,18,32,10,50,10,2,10,28,8,40,33,8,47,56,42,25,39,20,36,46,17,15,11,35,21,44,30,52,15,12
63	13,47,15,53,21,5,53,3,26,17,39,44,37,48,24,18,61,26,39,12,40,32,56,22,29,50,59,40,25,28,45,21,21,28,14,28,6,48,1,0,13,36,6,47,2,19,38,3,5,8,53,31,52,18,24,23,25,12,5,12,4,27,43,22,19,56,10,39,55,34,1,39
66	60,15,8,40,52,1,18,12,4,60,65,54,59,2,11,30,60,20,57,34,60,58,11,13,22,20,6,61,22,2,60,15,60,9,10,35,13,16,6,5,2,31,34,30,20,65,42,52,20,0,17,56,20,48,21,36,2,16,47,31,10,42,47,38,2,58,34,50,6,22,33,155
69	8,28,10,50,30,22,27,8,52,42,25,53,48,13,14,33,66,16,56,39,3,5,53,33,38,4,44,19,3,46,12,31,23,21,53,9,8,31,51,47,42,22,10,39,39,7,22,49,31,15,52,41,46,65,2,10,35,44,37,49,1,49,22,35,40,22,8,38,28,3,7,56
72	10,16,62,65,43,48,65,34,39,33,59,64,24,10,16,62,13,16,6,69,67,41,20,7,25,23,15,0,1,41,28,19,8,54,65,30,0,69,32,50,7,6,36,66,6,43,67,49,21,59,57,39,18,62,46,69,10,49,52,58,6,30,35,2,66,40,1,53,2,1,29,54

8.4.9.2.5.2 LDPC encoding

The code is flexible in that it can accommodate various code rates as well as packet sizes. Since LDPCs are block-oriented codes, some restrictions are necessary on the combinations of available code rates and codeword sizes in order to control complexity.

The encoding of a packet at the transmitter generates parity-check bits $(p_0, \ldots, p_{m-1})$ based on an information block $s=(s_0, \ldots, s_{k-1})$, and transmits the parity-check bits along with the information block. Because the current symbol set to be encoded and transmitted is contained in the transmitted codeword, the information block is also known as systematic bits. The encoder receives the information block $s=(s_0, \ldots, s_{k-1})$ and uses the matrix $\mathbf{H}_{bm}$ to determine the parity-check bits. The expanded matrix $\mathbf{H}$ is determined from the model matrix

$\mathbf{H}_{bm}$. Since the expanded matrix **H** is a binary matrix, encoding of a packet can be performed with vector or matrix operations conducted over GF(2).

One method of encoding is to determine a generator matrix **G** from **H** such that $\mathbf{G}\,\mathbf{H}^T = \mathbf{0}$. A *k*-bit information block $\mathbf{s}_{1\times k}$ can be encoded by the code generator matrix $\mathbf{G}_{k\times n}$ via the operation **x** = **s G** to become an *n*-bit codeword $\mathbf{x}_{1\times n}$, with codeword $\mathbf{x}=[\mathbf{s}\ \mathbf{p}]=[s_0, s_1, \ldots, s_{k-1}, p_0, p_1, \ldots, p_{m-1}]$, where $p_0, \ldots p_{m-1}$ are the parity-check bits; and $s_0, \ldots s_{k-1}$ are the systematic bits. Encoding an LDPC code from **G** can be quite complex. The LDPC codes are defined such that very low complexity encoding directly from **H** is possible.

Direct Encoding

Encoding is the process of determining the parity sequence **p** given an information sequence **s**. To encode, the information block **s** is divided into $k_b = n_b - m_b$ groups of *z* bits. Let this grouped **s** be denoted **u**,

$$\mathbf{u} = [\mathbf{u}(0) \quad \mathbf{u}(1) \quad \cdots \quad \mathbf{u}(k_b-1)],$$

where each element of **u** is a column vector as follows

$$\mathbf{u}(i) = [s_{iz} \quad s_{iz+1} \quad \cdots \quad s_{(i+1)z-1}]^T$$

Using the model matrix $\mathbf{H}_{bm}$, the parity sequence **p** is determined in groups of *z*. Let the grouped parity sequence **p** by denoted **v**,

$$\mathbf{v} = [\mathbf{v}(0) \quad \mathbf{v}(1) \quad \cdots \quad \mathbf{v}(m_b-1)],$$

where each element of **v** is a column vector as follows

$$\mathbf{v}(i) = [p_{iz} \quad p_{iz+1} \quad \cdots \quad p_{(i+1)z-1}]^T$$

Encoding proceeds in two steps, (a) initialization, which determines **v**(0), and (b) recursion, which determines **v**(*i*+1) from **v**(*i*), $0 \le i \le m_b - 2$. An expression for **v**(0) can be derived by summing over the rows of $\mathbf{H}_{bm}$ to obtain

$$\mathbf{P}_{p(x,k_b)}\mathbf{v}(0) = \sum_{j=0}^{k_b-1}\sum_{i=0}^{m_b-1}\mathbf{P}_{p(i,j)}\mathbf{u}(j) \tag{1}$$

where x, $1 \le x \le m_b - 2$, is the row index of $\mathbf{h}_{bm}$ where the entry is nonnegative and unpaired, and $\mathbf{P}_i$ represents the $z\times z$ identity matrix circularly right shifted by size *i*. Equation (1) is solved for **v**(0) by multiplying by $\mathbf{P}^{-1}_{p(x,k_b)}$, since $p(x,k_b)$ represents a circular shift.

The recursion expressed in Equation (2) can be derived by considering the structure of **H** .

$$\mathbf{v}(1) = \sum_{j=0}^{k_b-1}\mathbf{P}_{p(i,j)}\mathbf{u}(j) + \mathbf{P}_{p(i,k_b)}\mathbf{v}(0), \quad i=0, \tag{2}$$

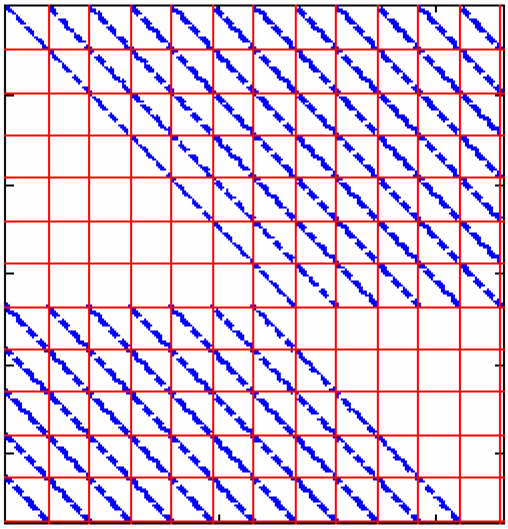
$$\mathbf{v}(i+1) = \mathbf{v}(i) + \sum_{j=0}^{k_b-1}\mathbf{P}_{p(i,j)}\mathbf{u}(j) + \mathbf{P}_{p(i,k_b)}\mathbf{v}(0), \quad i=1,\ldots,m_b-2 \tag{3}$$

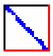
where $\mathbf{P}_{-1} \equiv \mathbf{0}_{z\times z}$.

Thus all parity bits not in **v**(0) are determined by evaluating Equation (2) for $0 \le i \le m_b - 2$. Equations (1) and (2) completely describe the encoding algorithm. These equations also have a straightforward interpretation in terms of standard digital logic architectures. Since the non-zero elements *p*(*i*,*j*) of $\mathbf{H}_{bm}$ represent circular shift sizes of a vector, all products of the form $\mathbf{P}_{p(i,j)}\mathbf{u}(j)$ can be implemented by a size-*z* barrel shifter.

The above encoding theory can be explained as following.

Let $\mathbf{H}_{b2}$ is extended to $\mathbf{H}_2$, $\mathbf{H}_{b1}$ is extended to $\mathbf{H}_1$, so $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2]$.
It is easy to get $\mathbf{H}_2^{-1}$ as following:



Here  represents a $z \times z$ identity matrix, z can be any expand factor .

Thus, let $\mathbf{v} = \mathbf{H_1} \times \mathbf{s} = \mathbf{H_2} \times \mathbf{p}$, then the parity sequence

$$\mathbf{p} = \mathbf{H_2}^{-1} \times \mathbf{v} \qquad (4)$$

(1)(2)(3) just more efficiently finish the function of (4) .

8.4.9.2.5.3 Code Rate and Block Size Adjustment

Same as in [1], but Table in this section will be changed.

Examples of the *z* expansion factors are given in the tables below. The base matrix n_b is an integer is an integer multiple of 24 when code rate R = 1/2 or 2/3 , The base matrix n_b is an integer is an integer multiple of 32 when code rate R = 3/4.

Table

n (bits)	*n* (bytes)	Z expansion factor			Number of subchannels		
		R=1/2	R=2/3	R=3/4	QPSK	16QAM	64QAM
96	12	4	4	3	1		
192	24	8	8	6	2	1	
288	36	12	12	9	3		1
384	48	16	16	12	4	2	
480	60	20	20	15	5		
576	72	24	24	18	6	3	2
672	84	28	28	21	7		
768	96	32	32	22	8	4	
864	108	36	36	27	9		3
960	120	40	40	30	10	5	
1056	132	44	44	33	11		
1152	144	48	48	36	12	6	4
1248	156	52	52	39	13		
1344	168	56	56	42	14	7	
1440	180	60	60	45	15		5
1536	192	64	64	48	16	8	
1632	204	68	68	51	17		
1728	216	72	72	54	18	9	6
1824	228	76	76	57	19		
1920	240	80	80	60	20	10	
2016	252	84	84	63	21		7
2112	264	88	88	66	22	11	
2208	276	92	92	69	23		
2304	288	96	96	72	24	12	8
	base n-k	12	8	8			
	base n	24	24	32			

8.4.9.2.5.4 Packet Encoding

Same as in [1].

3. Simulation Results

Simulation results for the ZTE high girth code and Motorola code of the rate 1/2,2/3 code families are shown in Figure 1-4. For rate 1/2, code sizes considered are 576,1152,2304. For rate 2/3, code sizes considered are 576, 2304.The simulation conditions are: AWGN channel, BPSK modulation, maximum of 50 iterations, using generic floating-point belief propagation.Our high girth code has the same degree distribution (column weight, row weight) as the Motorola code in [1], base matrices of our contribution have the same dimension. as Motorola code[1].

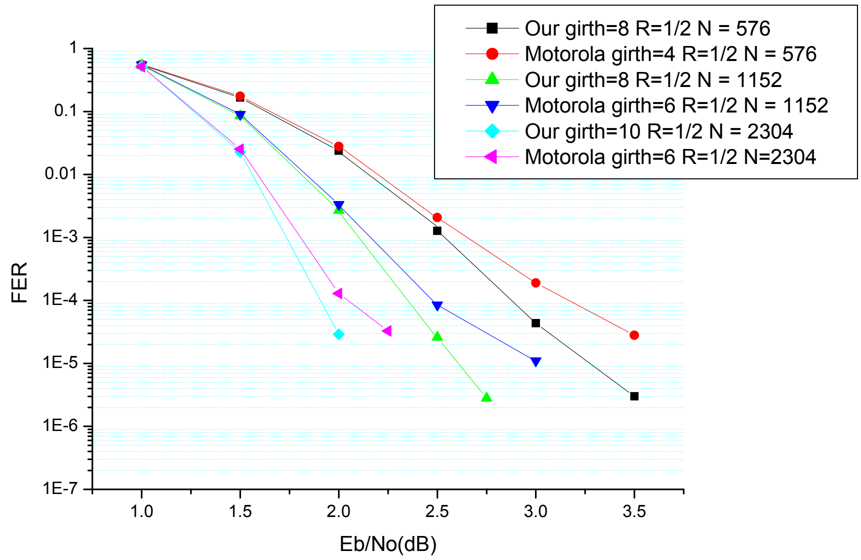


Figure 1. FER performance of R=1/2 structured codes, code size N= 576,1152,2304
Base matrix size: m_b=12,n_b=24,AWGN, BPSK

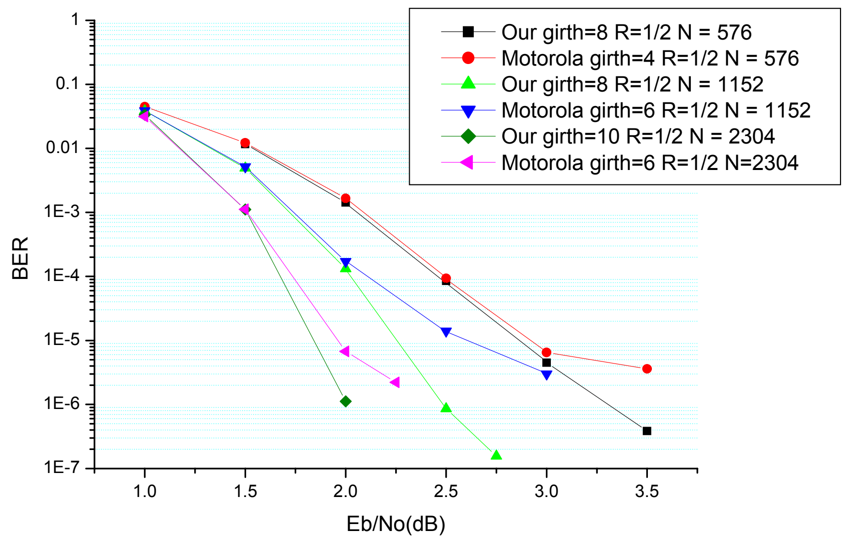


Figure 2. BER performance of R=1/2 structured codes, code size N= 576,1152,2304
Base matrix size: m_b=12,n_b=24,AWGN, BPSK

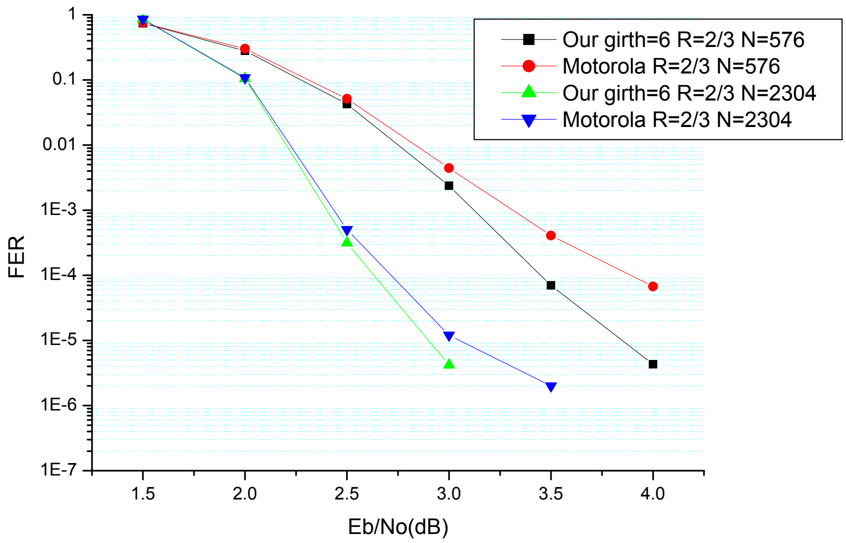


Figure3. FER performance of R=2/3 structured codes, code size N= 576, 2304
Base matrix size: m_b=8,n_b=24,AWGN, BPSK

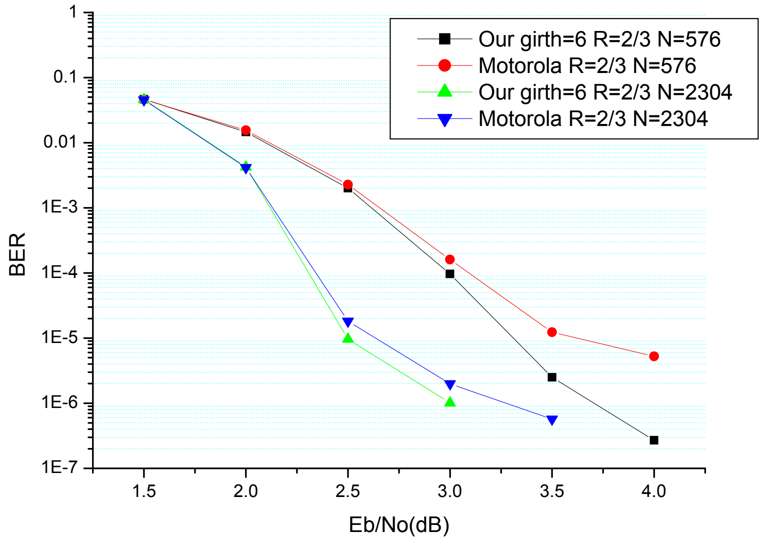


Figure 4. BER performance of R=2/3 structured codes, code size N= 576, 2304
Base matrix size: m_b=8,n_b=24,AWGN, BPSK

From our simulation results, we can find that our codes overcome the “error floor” phenomenon, and the BER curve of them will descend more steeply. When SNR is high, our high girth method obviously improve the performance.

From Figure 1 and 4, normal LDPC BER curve descends more and more slow. It is always difficult to arrive at the point BER = 10e-6, which efficient data communication needs. So high girth LDPC codes are suitable for the situation where low BER is needed.

4. Reference Material

The following documents contain background material and source material from which the group is working. Modifications to this material are being considered as well as new material from Motorola etc ,in order to achieve harmonization on the best possible code for 802.16e.

1 C80216e-04_373r1_LDPC coding for OFDMA PHY Brian Classon, Yufei Blankenship, Motorola

2 C80216e-04_141r2_LDPC coding for OFDMA PHY Eric Jacobsen,Bo Xia,Val Rhodes Intel Corporation