Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**	
Title	**LDPC coding for OFDMA PHY**	
Date Submitted	**2004-06-25**	
Source(s)	Eric Jacobsen Bo Xia Val Rhodes Intel Corporation	eric.a.jacobsen@intel.com bo.xia@intel.com valentine.j.rhodes@intel.com
	Brian Johnson Alek Purkovic Nortel Networks	brjohnso@nortelnetworks.com apurkovi@nortelnetworks.com
	Brian Classon Yufei Blankenship Motorola	brian.classon@motorola.com yufei.blankenship@motorola.com
	Jerry Kim Panyuh Joo Gyubum Kyung Hongsil Jeong Sang-Hyo Kim Hanju Kim DS Park Jaeho Jeon Samsung Electronics	kimjy@samsung.com panyuh@samsung.com gyubum.kyung@samsung.com hongsil.jeong@samsung.com sanghyo7.kim@samsung.com hanju.kim@samsung.com dspark@samsung.com jhjeon@samsung.com
	Eli Sasha Runcom Technologies	shasha@runcom.co.il
	Kyuhyuk Chung LG Electronics	kyuhyuk@lge.com
	Ron Murias Wi-LAN	rmurias@Wi-LAN.com
Re:	802.16REVe Sponsor Ballot Recirculation comment	
Abstract	This contribution describes the status of the activities of the LDPC interested parties towards submitting a final LDPC coding scheme and also describes current consensus text for the LDPC coding scheme	
Purpose	Proposal to include the LDPC FEC scheme as an option for OFDMA	

Notice	This document has been prepared to assist IEEE 802.16. 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 grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:chair@wirelessman.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.

Progress of LDPC discussions

Based on the multiple LDPC contributions and the large number of interested parties, an informal 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. First steps are to characterize the proposed codes, both qualitatively (features, structure) and quantitatively (performance comparisons and complexity analysis). The results of the characterization can be used to explore desirable code features for harmonization and consensus building. This contribution outlines the progress of the discussions, qualitative comparison of proposals, the evaluation criteria, the design issues, the current consensus, and the currently open issues. Finally this contribution outlines the current consensus text for 802.16e.

Qualitative Comparison of Proposals

The four LDPC proposals have been studied to determine desired features of each proposal and commonality between proposals. The goal is to create a harmonized proposal better than any individual proposal by considering all desired features. The table below is based primarily upon the original four proposals submitted by Intel, Samsung, Nortel, and Motorola. The comparison shows that there is a fair amount of commonality between the original designs.

Note that some consensus building and harmonization has already occurred during the ad hoc process, which have been included in the table. Consensus has helped streamline the table (e.g., no entry for ‘supports 6 byte info packet size’). The harmonization efforts include a (2400,1920) version of the Motorola design that could be used to achieve a feature of exact code rates that some participants desire, a R=4/5 code from Samsung better designed for shortening through the use of a column weight interlacing (similar to the Motorola design), and entries for an LG LDPC proposal.

Features	Intel	Samsung	Nortel	Motorola	LG	Notes
complexity						
high % wt 2 in H2 (parity portion)	~100%	~90%	~100%	~100%	~100%	
wt 2 columns dual diagonal	Y	Y	Y	Y	Y	Samsung matrix descriptor is dual-diagonal, but resulting matrices are not
simple recursive encoding using H2 (parity portion)	Y	Y	Y	Y	Y	Samsung uses more complex (but still linear time) Richardson encoding
low row wt	18	20	18	20	18	Based on (2000,1600) code
designed such that shortening can produce multiple codes from one H matrix	Y	Y & N	Y	Y	Y	Intel, Motorola designed to shorten by any number of info bits; Nortel in discrete chunks of size N-K New Samsung scheme has flexibility with shortening, original scheme requires unique parity check descriptor per code rate.
code rates per H matrix	16	16 or 1	16	16	16	Based on (2000,1600) code. New Samsung scheme (shortening incorporated) has 16 code rates per H; original scheme (no shortening incorporated) number of code rates per H is one. Note that any H matrix can produce multiple rates for a single block size.
designed to use structured decoder	N	Y	Y	N	N	Generic non-company-specific decoding algorithms may be preferred. Structured decoder can reduce complexity for supporting multiple block sizes for a given code rate All could use generic (sum-product and min-sum decoding possible).
performance						
irregular H1 (info portion)	N	Y	N	Y	N	Irregular improves performance (for longer codewords)
performs well under shortening	Y	Y	TBD	Y	TBD	Motorola irregular design has good wt dist for each shortened code. Intel code performs well down to R=2/3.
designed such that rate-compatible parity can be added (for IR)	N	TBD	N	Y	Y	Motorola contrib 101 IR shows ~1dB gain over Chase HARQ for 288 info bit, initial code rate 3/4, rate 1/2 after combining.
misc						
descriptor or algorithm to produce H matrix for a given block size and code rate	-	D	D	A	A?	Samsung has one descriptor per code rate. Note that the memory requirement for a descriptor or algorithm is typically much less than the memory to store the full matrix.
does not contain wt 1 columns	N	Y	N	Y	N	wt 1 can hurt performance or cause error floor, esp in fixed pt implementation of generic SPA decoder (impact for blocks under consideration TBD)
synergy with other IEEE 802 standards	802.11	802.11	802.11	802.11 or UWB	802.11	(2000,1600) fits 802.11/(2400,1920) fits UWB

Evaluation Criteria and Methods

Currently there is agreement that there should be a set of common evaluation criteria and methods to generate comparable results for various proposed designs. The evaluation criteria and methods have been discussed and there is consensus on a range of criteria while there may remain open issue with other criteria.

Consensus on the following criteria and methods:

- Performance: BLER performance of each code design in AWGN channel for the complete range of block sizes and code rates deemed appropriate to target for the 802.16e context. The appropriate range of block sizes and code rates was to be determined. The method was to generate results for floating point decoding.
- Complexity: The exact method for evaluation of complexity remains open
- There was consensus that performance was the primary metric, since complexity concerns will decrease with time.

Open issues on evaluation criteria and methods:

- In addition to BLER in AWGN, there is a desire (time permitting) to test proposals in fading channels with Doppler. (Default is AWGN)
- There is an open issue if we should compare performance for specific number, or range of iterations in the decoder. One suggestion was a default of 50 iterations for comparison. Another suggestion was to report performance results using a set of iterations such as 8, 12, 16 and 20, in order to evaluate and compare code performance vs. number of iterations. The exact number of iterations will be implementation specific.
- Complexity analysis: The exact method for evaluation of complexity remains open. Suggestions of things to be included in the complexity analysis include:
    - Citing the amount of RAM and ROM required for encoder and decoder operations
    - Citing the number of arithmetic operations required for encoder and decoder operations

  - Citing the number of H matrices required as a qualitative indicator
  - Citing and explaining any matrix structure as a qualitative indicator
- Other open issues w/evaluation criteria and methods?

The primary gating factor in the initial comparison has been the determination of the appropriate range of code rates and block sizes to target for overall good code performance in the 802.16e system context.

Design Issues

There are several design issues that the group is currently discussing.

- Appropriate range of code rates and block sizes to target for overall good code performance in 802.16e system context.
- Appropriate number of H matrices needed to achieve good performance over the target block size and code rate coverage area.
- Flexibility methods for generating codes for different code rate and block sizes based on the base H matrices.

Code Rate and Block Size Range

The code rate and block size range has been discussed. The current consensus is that the LDPC codes should cover block sizes at least as low as 40 bytes, and that the upper bound should be somewhere close to 2000 bits for code word length before resorting to concatenation methods to support higher block sizes. For code rates the consensus is that the code set should cover code rates at least as low as _ before resorting to repetition and at least as high as _.

The open issues include:

- Do we need support for block sizes smaller than 40 bytes? Open issue
- Block size coverage, should it overlap with the existing codes?
- Should we extend the code rate range higher than _ up to 4/5 and lower than _ down to 1/3?
- How low should we support in code rate with repetition? Should we go as low as 1/8 with repetition?

The following tables show examples of Code Rate and Block Size combinations that could be achieved with codes that are being considered.

Example I

QPSK (N in bits is a multiple of 48x2)					
N(bytes)	N(bits)	Nsch	K(1/3)	K (1/2)	K (3/4)
60	480	5	None	None	45
72	576	6	None	None	54
96	768	8	None	48	72
108	864	9	None	54	81
120	960	10	40	60	90
240	1920	20	80	120	180
360	2880	30	120	180	270
480	3840	40	160	240	320
600	4800	50	200	300	450
720	5760	60	240	360	540
960	7680	80	320	480	720
1080	8640	90	360	540	810

1200	9600	100	400	600	900
16-QAM (N in bits is a multiple of 48x4)					
N(bytes)	N(bits)	Nsch	K(1/3)	K (1/2)	K (3/4)
72	576	3	None	None	54
96	768	4	None	48	72
120	960	5	40	60	90
240	1920	10	80	120	180
480	3840	20	160	240	360
960	7680	40	320	480	720
1200	9600	50	400	600	900
64-QAM (N in bits is a multiple of 48x6)					
N(bytes)	N(bits)	Nsch	K(1/3)	K (1/2)	K (3/4)
72	576	2	None	None	54
108	864	3	None	54	81
180	1440	5	60	90	135
360	2880	10	120	180	270
720	5760	20	240	360	540
1080	8640	30	360	540	810

Example II

Sixteen code rates derived from the single (2000,1600) H matrix via shortening.

QPSK				
Mod	Nsch	N (bits)	K (bits)	Rate
2	5	480	80	0.167
2	6	576	176	0.306
2	7	672	272	0.405
2	8	768	368	0.479
2	9	864	464	0.537
2	10	960	560	0.583
2	11	1056	656	0.621
2	12	1152	752	0.653
2	13	1248	848	0.679
2	14	1344	944	0.702
2	15	1440	1040	0.722
2	16	1536	1136	0.740
2	17	1632	1232	0.755
2	18	1728	1328	0.769
2	19	1824	1424	0.781
2	20	1920	1520	0.792
		Number of rates:		**16**
16QAM				
Mod	Nsch	N (bits)	K (bits)	Rate
4	3	576	176	0.306
4	4	768	368	0.479

4	5	960	560	0.583
4	6	1152	752	0.653
4	7	1344	944	0.702
4	8	1536	1136	0.740
4	9	1728	1328	0.769
4	10	1920	1520	0.792
64QAM				
Mod	Nsch	N (bits)	K (bits)	Rate
6	2	576	176	0.306
6	3	864	464	0.537
6	4	1152	752	0.653
6	5	1440	1040	0.722
6	6	1728	1328	0.769

Example III

Twenty code rates derived from the single (2400,1920) H matrix via shortening.

QPSK				
Mod	Nsch	N (bits)	K (bits)	Rate
2	6	576	96	0.167
2	7	672	192	0.286
2	8	768	288	0.375
2	9	864	384	0.444
2	10	960	480	0.500
2	11	1056	576	0.545
2	12	1152	672	0.583
2	13	1248	768	0.615
2	14	1344	864	0.643
2	15	1440	960	0.667
2	16	1536	1056	0.688
2	17	1632	1152	0.706
2	18	1728	1248	0.722
2	19	1824	1344	0.737
2	20	1920	1440	0.750
2	21	2016	1536	0.762
2	22	2112	1632	0.773
2	23	2208	1728	0.783
2	24	2304	1824	0.792
2	25	2400	1920	0.800
		Number of rates:		**20**
16QAM				
Mod	Nsch	N (bits)	K (bits)	Rate
4	3	576	96	0.167
4	4	768	288	0.375
4	5	960	480	0.500
4	6	1152	672	0.583
4	7	1344	864	0.643

4	8	1536	1056	0.688
4	9	1728	1248	0.722
4	10	1920	1440	0.750
4	11	2112	1632	0.773
4	12	2304	1824	0.792
64QAM				
Mod	Nsch	N (bits)	K (bits)	Rate
6	2	576	96	0.167
6	3	864	384	0.444
6	4	1152	672	0.583
6	5	1440	960	0.667
6	6	1728	1248	0.722
6	7	2016	1536	0.762
6	8	2304	1824	0.792

Number of H matrices

Currently there are proposals that use one H matrix and there are proposals that use multiple. The only consensus we have so far is that one is desired would be the simplest to implement, but that we will likely need more than one to cover the whole range of rates and block sizes. Although there has been some work to study this issue so far, this issue cannot be satisfactorily resolved until we have a comprehensive set of performance data for the agreed upon code rates and block sizes for each design and a better view of the complexity for each design.

Flexibility methods for coverage

The methods proposed include shortening, concatenation, expansion, and repetition. There is concern that using a single matrix and only relying on shortening and concatenation will result in both incomplete coverage of the required block lengths as well as poor performance for larger block sizes and lower code rates. The intention is to use the consensus code rate and block sizes and the comparison data to arrive at a good combination of matrix choice and flexibility methods to achieve near best overall performance

Impact of Scalability

Impact of 128 FFT size should also be considered on overall block size support range as this could be smaller than in the other cases.

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 in order to achieve harmonization on the best possible code for 802.16e.

C802.16e-04/78 – Optional B-LDPC coding for OFDMA PHY. Panyuh Joo, Seho Myung, Jaeyeol Kim, Gyubum Kyung, Hongsil Jeong, Kyungcheol Yang, DS Park, Jeho Jeon, Samsung Electronics

C802.16e-04/96 – Draft Text for LDPC coding scheme for OFDMA. Eric Jacobsen, Intel Corp.

C802.16e-04/101r1 – Modified LDPC Matrix providing improved performance. Brian Classon, Yufei Blankenship, Motorola

C802.16e-04/102r1 – Modified LDPC Matrix providing improved performance. Brian Classon, Yufei Blankenship, Motorola

C802.16e-04/104 – Algebraic Low-Density Parity-Check Codes for OFDMA PHY Layer. Aleksandar Purkovic, Sergey Sukobok, Nina Burns, Brian Johnson, Nortel Networks

Recommended Text Changes

Add the following text to 802.16e_D3, adjusting the numbering as required:

8.4.9.2.4 Low Density Parity Check Code (optional)

8.4.9.2.4.1 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, 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.

8.4.9.2.4.2 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.

TBD description of packet encoding.

8.4.9.2.4.3 Code Rate and Block Size Adjustment

The code design will be flexible to support a range of code rates and block sizes through code rate and block size adjustment of the one or more H matrices of the fundamental code set. The exact methods for supporting code rate and block size adjustment will depend on the final design. For each supported rate and block size, there will be some combination of matrix selection, shortening, repetition, matrix expansion, and/or concatenation.

TBD description of code adjustment.

8.4.9.2.4.4 Packet Encoding

After harmonization, this section will be replaced with something equivalent to what is in section 8.2.1.2.4.1 which is the concatenation/packet encoding scheme for the CTC.

Since transported data packets can be any size from typically about 40 bytes up to 12000 bits and larger, the system must be able to encode variable length packets in a consistent manner. This consistency is required to ensure that the receiver always knows how to reconstruct the information field from the encoded transmitted data.

Each packet is encoded as an entity. In other words, the data boundary of a packet is respected by the encoder. Control information and packets smaller than 40 bytes are encoded using convolutional coding (CC) with appropriate code rates and modulation orders, as described in section 8.4.9.2.1.

The length and required rate of the packet that is to be encoded is all that is needed to encode or decode the packet using the following rules:

1. If Length <= N_i bits, then TBD.
2. If Length > N_i bits and <= 2 N_i bits, then TBD
3. If Length > 2 N_i bits, then compute N_r = modulo(Length, N_i) (in bits), then TBD.
4. Concatenation when TBD
5. Combination of shortening and concatenation when TBD

The intent of the above rule set is to provide a means for data transmission without the need for additional information beyond the packet field length. This scheme does so with a simple rule set that reduces the rate of the last codewords in order to reduce the number of iterations (and therefore the latency) that must be performed on the last portion of the data. The length and position of the shortened codewords and erased bits are deterministic when the above rules are followed.

For all packets the codeword bits can be indexed using the corresponding column indices of the H matrix. Using this convention the systematic codeword bits comprise the leftmost bits starting at bit location zero, and fill the codeword to bit k-1. The remaining N-k bits of the codeword, from indices k to N-1 are the parity bits. The codeword systematic bits are filled in an order consistent with the indices, so that the first bits of the packet fill the codeword from the lowest indices linearly to the highest indices. The codeword is then transmitted in a linear fashion starting from the lowest indices so that the systematic bits are transmitted first, followed by the parity bits. For shortened codewords the zeros are padded in the low order bits, so that the final codeword starting at the lowest indices contains first zero-padded bits and then the systematic data bits followed by the parity bits. The zero-padded bits are not typically sent over the channel.