Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Algebraic Low-Density Parity-Check Codes for OFDMA PHY Layer**
Date Submitted	**2004-05-16**
Source(s):	Aleksandar Purkovic, Sergey Sukobok, Nina Burns, Brian Johnson Nortel Networks 20410 Century Blvd Gaithersburg, MD 20874 USA Voice: 1-301-528-3230 Fax: 1-301-528-3298 apurkovi@nortelnetworks.com
Re:	Contribution in conjunction with comments to IEEE 802.16e
Abstract	This contribution describes construction and encoding of optional LDPC codes for 802.16e OFDMA
Purpose	Support of the comments
Notice	
Release	
Patent Policy and Procedures	

Algebraic Low-Density Parity-Check Codes for OFDMA PHY Layer

1 Introduction

This contribution describes a particular algebraic construction of the parity check matrix for Low-Density Parity-Check (LDPC) codes and outlines the encoding procedure for such codes. It also presents some Packet Error Rate (PER) results for AWGN and fading channels. These results were obtained through the simulation of the encoder and the decoder of these LDPC codes in an OFDM based WLAN transceiver. In order to establish a reference for comparison purposes we also simulated convolutional codes as they are defined in 802.11a/g WLAN standards.

2 LDPC Parity Check Matrix Construction

LDPC parity check matrix is generated in two steps, based on the information block size (K) and desired code rate, R=K/N, where N equals the codeword length. In the first step the base matrix is generated whereas in the second step the base matrix is expanded, if necessary.

2.1 Base Parity Check Matrix Construction

The base LDPC parity check matrix of the size MxN (M – number of parity check bits, M=N-K), constructed this way can be represented in the following general form:

$$H = [H^p \mid H_1^d \mid H_2^d \mid \ldots \mid H_q^d]$$

All of the component sub-matrices, $H^p, H_1^d, H_2^d, \ldots, H_q^d$ are square matrices of the size MxM = 4*m*x4*m*, where parameter *m* depends on the information block length, K and the code rate R:

$$m = \frac{K(1-R)}{4R}$$

There is always only one sub-matrix H^p in the parity check matrix *H*, corresponding to the parity check bits of the codeword. Number of other component sub-matrices $(H_1^d, H_2^d, \ldots, H_q^d)$, q is related to the desired code rate, as follows:

$$R = \frac{q}{q+1}$$

In fact, parity check matrix in this form, if created by using a special carefully specified procedure, can be used for generating codes with the series of code rates: 1/2, 2/3, 3/4, up to and including q/(q+1). This is illustrated below, where only the used part of the whole matrix is outlined in bold letters:

R = 1/2: $\boldsymbol{H} = [\boldsymbol{H}^{\boldsymbol{p}} | \boldsymbol{H}_{\boldsymbol{1}}^{\boldsymbol{d}} \mid H_2^d \mid ... \mid H_q^d]$

R = 2/3: $\boldsymbol{H} = [\boldsymbol{H}^{\boldsymbol{p}} | \boldsymbol{H}_{\boldsymbol{1}}^{\boldsymbol{d}} \mid \boldsymbol{H}_{\boldsymbol{2}}^{\boldsymbol{d}} \mid ... \mid H_q^d]$

...

R = q/(q+1): $\boldsymbol{H} = [\boldsymbol{H}^{\boldsymbol{p}} | \boldsymbol{H}_{\boldsymbol{1}}^{\boldsymbol{d}} \mid \boldsymbol{H}_{\boldsymbol{2}}^{\boldsymbol{d}} \mid ... \mid \boldsymbol{H}_{\boldsymbol{q}}^{\boldsymbol{d}}]$

Matrix H^p is in a “dual diagonal” form. An example H^p matrix for *m* = 3 is shown in Figure 1.

$$H^p = \begin{bmatrix} 1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$$

Figure 1. Example H^p matrix

Matrices H_i^d, $i = 1,2,...,q$ are constructed by using modified π-rotation technique, originally described in [1]. Each of the matrices H_i^d can be represented in the following form:

$$H_i^d = \begin{bmatrix} \pi_{A,i}\pi_{B,i}\pi_{C,i}\pi_{D,i} \\ \pi_{B,i}\pi_{C,i}\pi_{D,i}\pi_{A,i} \\ \pi_{C,i}\pi_{D,i}\pi_{A,i}\pi_{B,i} \\ \pi_{D,i}\pi_{A,i}\pi_{B,i}\pi_{C,i} \end{bmatrix}, \ i = 1,2,...,q$$

Figure 2. Structure of the matrix H_i^d

$\pi_{A,i}$ is an mxm permutation matrix, whereas $\pi_{B,i}$, $\pi_{C,i}$, and $\pi_{D,i}$ are obtained by 90°, 180°, and 270° counterclockwise rotations of the permutation matrix $\pi_{A,i}$. Permutation matrix is defined here as the matrix in which each row and column have weight equal to one (row/column weight are defined as the number of 1's in a row/column, respectively). An example of such permutation matrices is shown in Figure 3. for m = 6.

$$\pi_{A,i}=\begin{bmatrix}0&0&0&0&1&0\\0&0&0&0&0&1\\1&0&0&0&0&0\\0&0&1&0&0&0\\0&1&0&0&0&0\\0&0&0&1&0&0\end{bmatrix}\quad \pi_{D,i}=\begin{bmatrix}0&0&0&1&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\1&0&0&0&0&0\\0&0&0&0&0&1\\0&0&0&0&1&0\end{bmatrix}$$

$$\pi_{B,i}=\begin{bmatrix}0&1&0&0&0&0\\1&0&0&0&0&0\\0&0&0&0&0&1\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&1&0&0&0\end{bmatrix}\quad \pi_{C,i}=\begin{bmatrix}0&0&1&0&0&0\\0&0&0&0&1&0\\0&0&0&1&0&0\\0&0&0&0&0&1\\1&0&0&0&0&0\\0&1&0&0&0&0\end{bmatrix}$$

Figure 3. Example of the permutation matrix $\pi_{A,i}$ and its three rotations

Each component permutation matrix $\pi_{A,i}$ is completely specified by a pair of small seed numbers (a_i,b_i), such that $a_i<m$ and $b_i<m$, and a simple algorithm which defines positions of 1's in $\pi_{A,i}$, based on the selected pair (a_i,b_i). This way a base matrix designed for rate R=q/(q+1) is completely specified by 2q+1 numbers only:

$m, a_1, b_1, a_2, b_2, \ldots, a_q, b_q$

In terms of the row and column weights, this base parity check matrix has the following properties. First (leftmost) column has weight 1, columns 2 through 4m have weight 2, and columns 4m+1 through 4m(q+1) have weight 4. Top 4m-1 rows have weight 4q+2, whereas the last (bottom) row has weight 4q+1.

2.2 Expansion of the Base Parity Check Matrix

If there is a need to accommodate a larger block size, base parity check matrix may be expanded by an integer factor L, from a 4mx4m(q+1) matrix to a 4mLx4mL(q+1) matrix. Factor L should be chosen to be a small single digit number in order to keep the hardware complexity at a reasonable level. The expansion of H^p and H_i^d component sub-matrices, is performed differently, however.

Expansion of H^p is done by replacing each "0" element by an LxL zero matrix, $\mathbf{0}_{LxL}$, and each "1" element by an LxL identity matrix, $\mathbf{I}_{LxL}$.

Expansion of H_i^d is done by replacing each "0" element by an LxL zero matrix, $\mathbf{0}_{LxL}$, and each "1" element by a rotated version of an LxL identity matrix, $\mathbf{I}_{LxL}$. The rotation order, s (number of circular shifts to the right, for example) can be determined by using a simple rule: $s = (r + c)_{\text{mod L}}$, where r and c correspond to the row and column index of the particular "1" element, respectively.

3 Encoding

The described structure of the LDPC parity check matrix enables a simple encoding algorithm. In general, by the definition of the parity check matrix, the following holds:

$\mathbf{H}_{\text{MxN}}\mathbf{c}_{\text{Nx1}}=\mathbf{0}_{\text{Mx1}}$,

where $\mathbf{H}$ is the parity check matrix, $\mathbf{0}_{\text{Mx1}}$, an all-zero Mx1 vector, and $\mathbf{c}$ is a codeword vector given by:

$\mathbf{c} = [p_1,p_2,\ldots,p_M,d_1,d_2,\ldots,d_K]^{\text{T}}$

In the expression above, $p_1,p_2,\ldots,p_M$ represent parity bits (redundancy part) whereas $d_1,d_2,\ldots,d_K$ represent information bits (systematic part). Due to the dual diagonal structure of the sub-matrix H^p, parity bits can be computed recursively as:

$$p_M = \sum_{n=1}^{4q} h_{M,M+k_n^M} d_{k_n^M}, \quad \text{where } \mathrm{M} + k_n^M \text{ is the index of the column in which row M contains a “1”}$$

$$p_{M-1} = p_M + \sum_{n=1}^{4q} h_{M-1,M+k_n^{M-1}} d_{k_n^{M-1}}, \quad \text{where } \mathrm{M} + k_n^{M-1} \text{ is the index of the column in which row M-1 contains a “1”}$$

…

$$p_1 = p_2 + \sum_{n=1}^{4q} h_{1,M+k_n^1} d_{k_n^1}, \quad \text{where } \mathrm{M} + k_n^1 \text{ is the index of the column in which row 1 contains a “1”}$$

where $k_n^1, k_n^2, ..., k_n^{M-1}, k_n^M \in \{1,2,...,4mq\}$ and $h_{r,c}$ are non zero elements (=1) of the parity check matrix in its part corresponding to H_i^d's. Therefore, the encoding operation can be performed by using only simple AND and XOR circuits.

For the cases when the information block size is above a specified threshold, concatenation of the codewords is employed.

4 Performance Evaluation

In this section we present some of the simulation results and analysis which were included in contributions prepared for the IEEE 802.11 TGn meetings in January and March of 2004, [2] and [3], respectively.

Packet Error Rate (PER) curves were obtained through the simulation of the basic 802.11a transceiver extended with 256-QAM constellation. Simulation scenarios are listed in the following table.

DataRate (Mbits/s)	6	9	12	18	24	36	48	54	64	72
Modulation	BPSK	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM	256QAM	256QAM
CodingRate (R)	1/2	3/4	1/2	3/4	1/2	3/4	2/3	3/4	2/3	3/4

Two channel types were included in the simulation: AWGN channel and the Channel D as specified in the 802.11n Channel Models document, [4]. For simulating of this fading channel a reference Matlab code [5] was used. Only Non-LOS case with zero path loss was considered with Doppler spectrum and shadow fading disabled. In order to evaluate encoder/decoder performance only, all other conditions were assumed ideal: ideal channel estimation, all packets detected, ideal synchronization, zero frequency offset, ideal analog front end and Nyquist sampling frequency. Information block lengths of the packets tested were: 40, 200, 600, and 1500 bytes. Code rates corresponded to those specified in the 802.11a standard document: 1/2, 2/3, and 3/4. Uniform bit loading is applied. Decoders were implemented in floating point: Viterbi decoder for convolutional codes and iterative Sum-Product algorithm (log version) with 50 iterations, for the LDPC codes. In addition, concatenation of the codewords was empoloyed for longer blocks. Simulation results for the information block length of 200 bytes for AWGN channel and fading Channel D are shown in Figures 4. and 5., respectively.

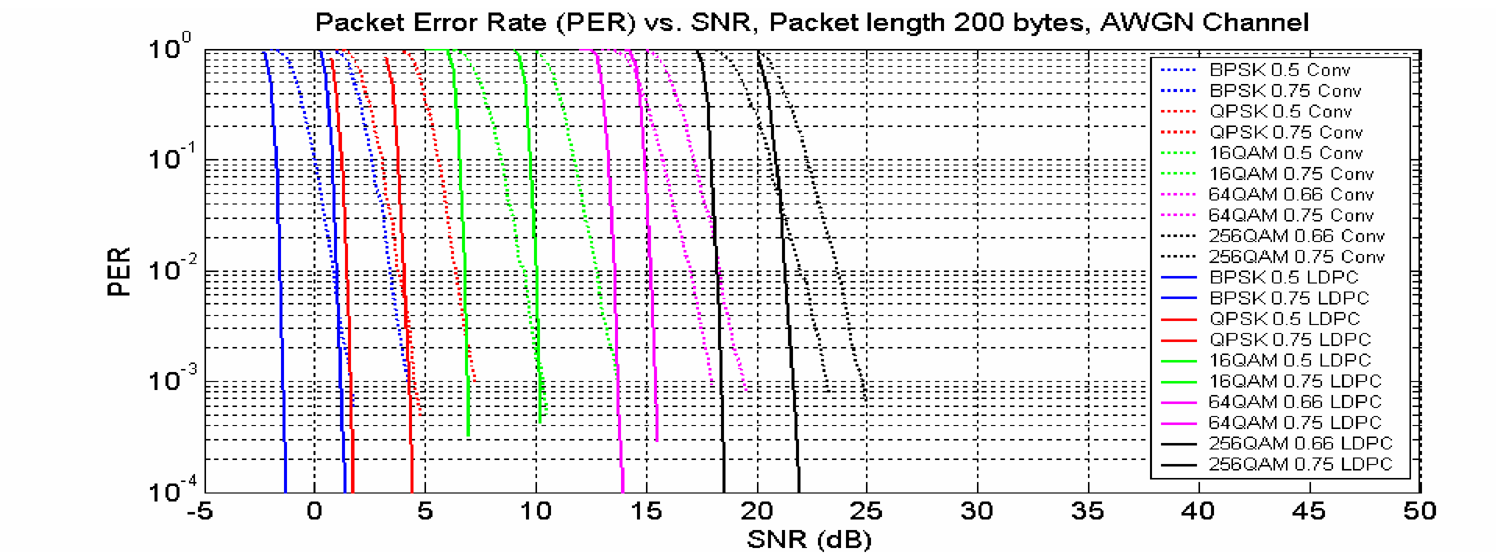

Figure 4. PER vs. SNR, AWGN channel

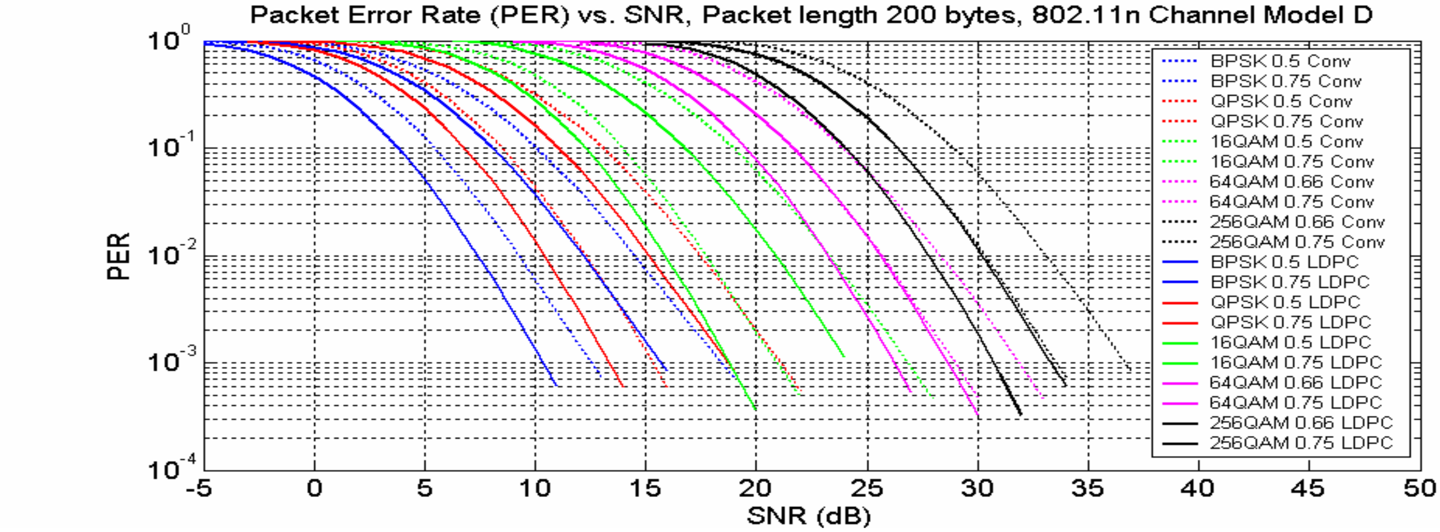

Figure 5. PER vs. SNR, fading Channel D

Solid lines in the figures correspond to LDPC codes, whereas dotted lines correspond to convolutional codes. Coding gain of 2-3dB at PER of 10^{-2} is observed. These results can be translated into different form which shows throughput as a function of SNR. For this purpose throughput is defined as:

Throughput = PHY_data_rate(1 – PER)

Throughput vs. SNR plots for AWGN and fading Channel D cases are shown in Figures 6. and 7., respectively.

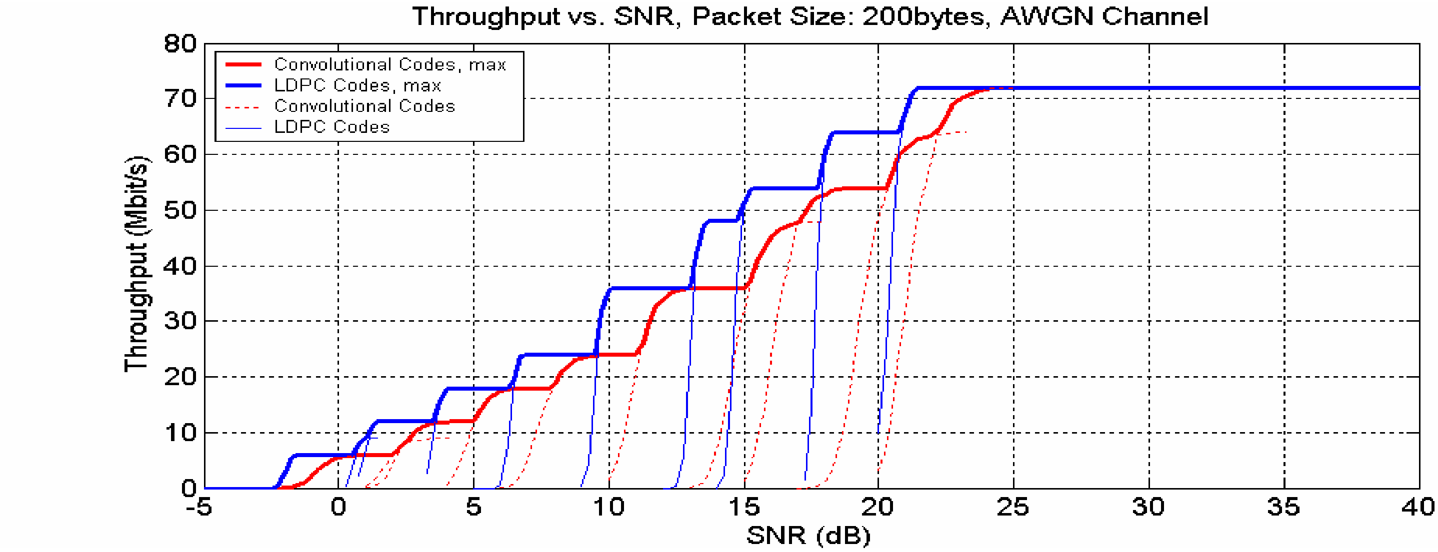

Figure 6. Throughput vs. SNR, AWGN channel

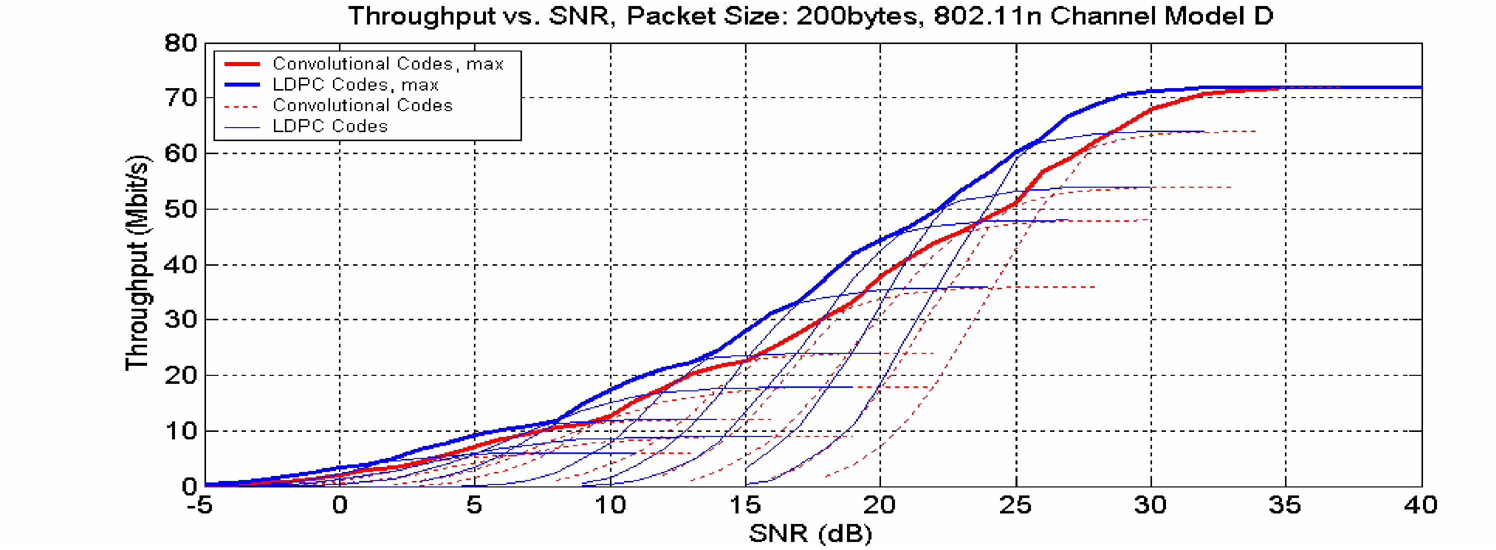

Figure 7. Throughput vs. SNR, fading Channel D

It was noticed in the process of performance evaluation that the coding gain relative to convolutional codes drops as the information block size decreases. Figure 8. illustrates this phenomenon.

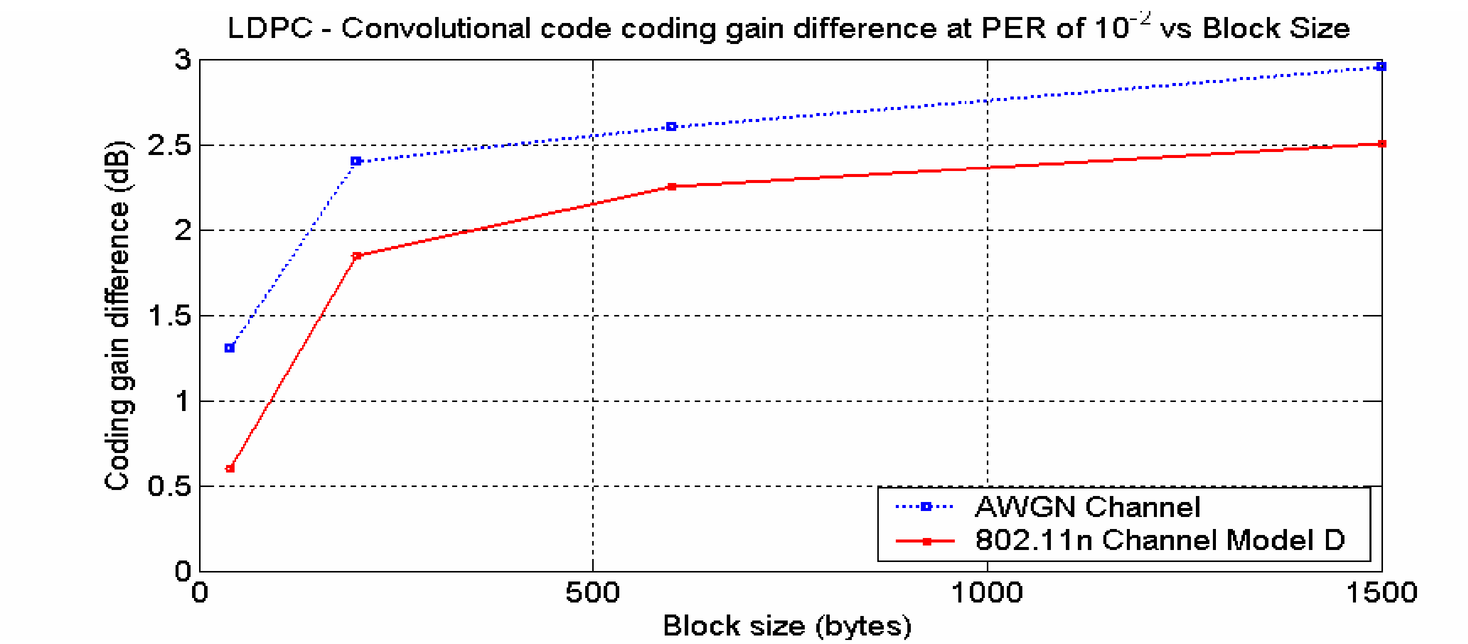

Figure 8. Coding gain as a function of the information block size

In Figure 8. selected modulation was BPSK, code rate was 1/2 and coding gain was recorded at PER of 10^{-2}.

5 Summary

Particular algebraic construction of the parity check matrix for the Low-Density Parity-Check codes is described in this document. Description of encoding algorithm and some results of the performance evaluation effort are included as well. We believe that this code design can provide better overall performance for the complete code set and should be evaluated in comparison to the other LDPC code proposals.

6 References

[1] R. Echard et al, “The π-rotation low-density parity check codes,” In Proc. GLOBECOM 2001, pp. 980-984, Nov. 2001

[2] IEEE 802.11-04/0071r1, “LDPC vs. Convolutional Codes for 802.11n Applications: Performance Comparison,” Aleksandar Purkovic, Nina Burns, Sergey Sukobok, Levent Demirekler, Nortel Networks, January 2004

[3] IEEE 802.11-04/0337r0, “LDPC vs. Convolutional Codes: Performance and Complexity Comparison,” Aleksandar Purkovic, Sergey Sukobok, Nina Burns, Nortel Networks, March 2004

[4] IEEE 802.11-03/940r1, “TGn Channel Models”, TGn Channel Models Special Committee, November 2003.

[5] Laurent Schumacher, “WLAN MIMO Channel Matlab program,” October 2003, version 2.1.