Project	**IEEE 802.20 Working Group on Mobile Broadband Wireless Access** **<http://grouper.ieee.org/groups/802/20/>**	
Title	**LDPC Code Proposal – Technology Overview**	
Date Submitted	**2007-03-05 (March 5, 2007)**	
Sources	Sung-Eun Park, Seunghoon Choi Samsung Electronics, Suwon, Korea	Email : {se.park, seunghoon.choi} @samsung.com
	Thierry Lestable Samsung Electronics Research Institute, UK	Email : thierry.lestable@samsung.com
	Anna Tee Samsung Telecommunications America	Voice: 1 (972) 761-7437 Email: atee@sta.samsung.com
Re:	IEEE 802.20 Call for Proposal	
Abstract	This document proposes an LDPC coding scheme for Mobile Broadband Wireless Access Systems.	
Purpose	For consideration and adoption as a feature supported by 802.20 standard	
Notice	This document has been prepared to assist the IEEE 802.20 Working Group. 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 802.20.	
Patent Policy	The contributor is familiar with IEEE patent policy, as outlined in Section 6.3 of the IEEE-SA Standards Board Operations Manual <http://standards.ieee.org/guides/opman/sect6.html#6.3> and in *Understanding Patent Issues During IEEE Standards Development* <http://standards.ieee.org/board/pat/guide.html>.	

1. Introduction

The current version of 802.20 air interface standard draft [1] supports two basic channel coding schemes, namely, a rate 1/5 Parallel Concatenated Convolutional Code (PCCC) i.e. Turbo codes and a rate 1/3 Convolutional Code (CC). The rate 1/5 Turbo codes shall be used for number of information bits *k* larger than 128, while the rate 1/3 Convolutional Code shall be used for values of *k* less than or equal to 128. We propose hereafter to include an optional Low-Density Parity-Check (LDPC) coding scheme for high data rates. The proposed LDPC codes offer both efficient support of Type II HARQ (Incremental Redundancy) together with similar or better performances than Turbo codes through all HARQ retransmissions. Besides, this proposed code structure enables highly parallelizable decoder architectures, thus resulting in high-throughput decoder implementations.
LDPC codes are fully defined by their sparse parity-check matrices, and can also be represented by a Tanner Graph, a bipartite graph illustrating the constraints of the code.. Such graph consists first of two types of nodes namely variable nodes and check nodes, then of edges which connect those two types of nodes. Variable nodes represent bits in the codeword, but they can also contain some punctured bits that are not transmitted on the channel. Check nodes represent the constraints that define the code: the set of variable nodes connected to any given check node is constrained to sum to zero.
LDPC codes can be decoded by Message-Passing based algorithms. One of them is the Pearl's Belief-Propagation (BP) algorithm which passes beliefs in the form of Log-Likelihood Ratios (LLR) along the edges of the bipartite graph. Although optimal only for tree structure codes (cycle free), near optimal performances are usually obtained. Moreover, the complexity of BP algorithm is proportional to the number of edges in the graph. Due to the sparseness of the parity-check matrix, and thus of the corresponding bipartite graph, the resulting decoding complexity is quite affordable.

2. The proposed LDPC code structure

a. Structured LDPC codes

We propose hereafter a class of structured LDPC codes, also called block-type LDPC codes (BLDPC). The parity-check matrix of such structured LDPC codes is lifted from a small base parity-check matrix of size *m* x *n*, where m is the number of check (constraint) nodes, n is the number of variable nodes or length of the codeword. Each non-zero element in the base parity-check matrix is replaced by an L x L permutation matrix. As a result of lifting procedure, a parity-check matrix of size mL x nL is obtained. This lifting process not only enables the implementation of parallel encoding and decoding algorithms but also reduces the storage of large parity-check matrices through easy parameterization. A simple and common choice for an L x L permutation matrix is a cyclic permutation matrix which can be expressed by the powers of a matrix P defined by

$$P = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

As an illustration, the generic form of parity-check matrix for the structured LDPC codes is shown in Figure 1 below.

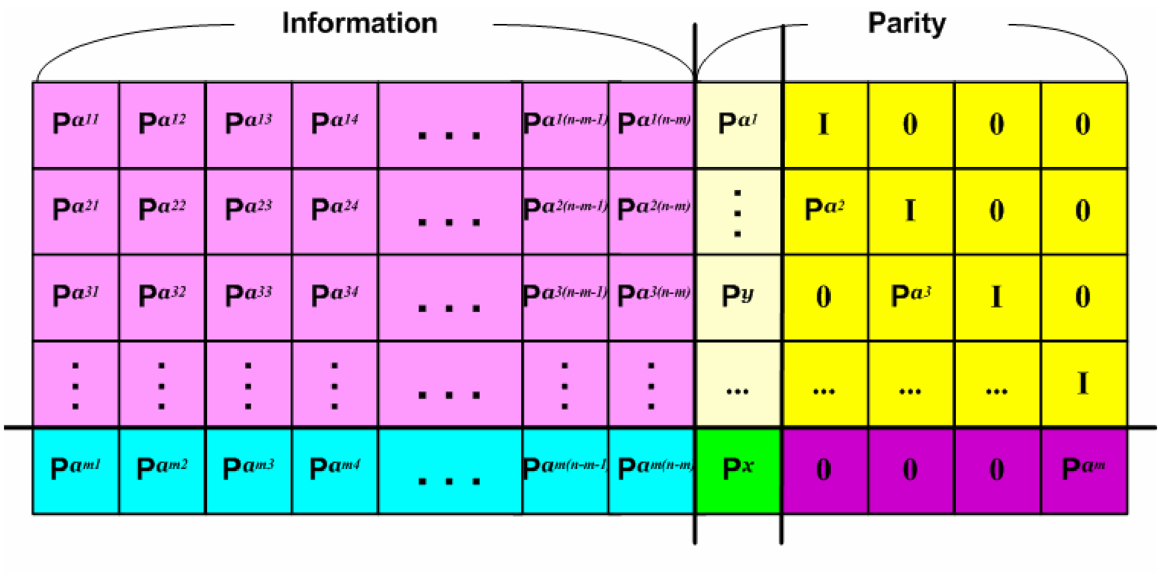

Information
Parity
$P^{a_{11}}$ $P^{a_{12}}$ $P^{a_{13}}$ $P^{a_{14}}$ . . . $P^{a_{1(n-m-1)}}$ $P^{a_{1(n-m)}}$ P^{a_1} I 0 0 0
$P^{a_{21}}$ $P^{a_{22}}$ $P^{a_{23}}$ $P^{a_{24}}$ . . . $P^{a_{2(n-m-1)}}$ $P^{a_{2(n-m)}}$ P^{a_2} I 0 0
$P^{a_{31}}$ $P^{a_{32}}$ $P^{a_{33}}$ $P^{a_{34}}$ . . . $P^{a_{3(n-m-1)}}$ $P^{a_{3(n-m)}}$ P^{y} 0 P^{a_3} I 0
... I
$P^{a_{m1}}$ $P^{a_{m2}}$ $P^{a_{m3}}$ $P^{a_{m4}}$ . . . $P^{a_{m(n-m-1)}}$ $P^{a_{m(n-m)}}$ P^{x} 0 0 0 P^{a_m}

Figure 1. Example parity-check matrix of structured LDPC code

One of the major strengths of such class of LDPC codes compared with Turbo codes is its high decoding throughput inherited from the highly parallelizable decoder architectures. The first ASIC implementation [2] of an LDPC decoder adopted full parallel architecture. Although such fully parallel architecture offers the highest throughput, its hardware complexity is prohibitive. Therefore partially parallel decoder is considered for structured LDPC codes, offering affordable complexity. Assuming that the size of the structured

LDPC codes is mL x nL, there are then two possible partial parallel decoder approach: first an edge parallel decoder, then the node parallel decoder. On the one hand, in edge parallel decoder implementation, edges in the base graph are processed in a serial fashion, and parallelism is achieved by simultaneously processing L copies of the same edge. The basic parallelism factor of such edge parallel architecture is thus L. (N.B: $L/2$, $L/4$,…, $L/2^p$, etc. can also be implemented where p is an natural number). On the other hand, in node parallel decoder implementation, different copies of the base graph are processed in a serial fashion and parallelism is achieved by simultaneously processing different nodes in the base graph. As a result, the basic parallelism factor of such node parallel decoder is (m, n). It should be noted that multiples of (m, n) are also possible.

There is a critical trade-off between the parallelism factor and the hardware complexity, to be decided by the designer. Indeed, parallelism factor should be decided considering throughput requirements of the targeted system. Even though the structured LDPC codes are well suited for both edge parallel and node parallel decoder approach, the latter is a better solution for the candidate scheme, due to the code length flexibility property. Such reasons will be described in the sequel.

b. Multi-edge-type LDPC codes

Multi-edge-type LDPC codes are generalizations of regular and irregular LDPC codes. They perform better with lower error floors than standard irregular LDPC codes, while requiring lower complexity. In the factor graph of multi-edge-type LDPC codes as shown in Figure 2, there are several types of edges, including those that are connected to punctured information nodes, and others that are connected to degree-one parity nodes. The punctured information nodes give higher noise thresholds with lower degrees, resulting in performance and complexity improvement. The degree-one parity nodes are very convenient for HARQ. The distribution of each type of edge is optimized through Density Evolution (DE) algorithm. Multi-edge-type framework is used for designing base code graph.

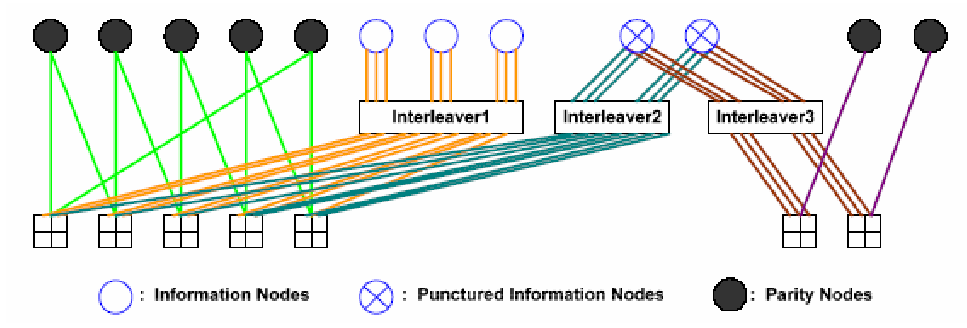


Figure 2. Factor graph of typical multi-edge-type LDPC codes

c. Code length flexibility

The candidate LDPC codes support code length flexibility by increasing or decreasing the size of cyclic permutation matrix *P* as shown in Figure 3. LDPC codes of variable length need to be expressed by only one parity check matrix thus reducing the memory storage requirements. Then, flexibility w.r.t. length is achieved by adopting modulo function on the shift factor of the non-zero sub-matrices in the parity-check matrix. Let's assume that (i, j)-th element in base matrix is non-zero. Then shift factor $p(f, i, j)$ corresponding to the expansion factor L_f is derived from the original shift factor $p(i, j)$ by using a modulo function $p(f, i, j) = \text{mod}\,(p(i, j)\,,\, L_f\,)$.

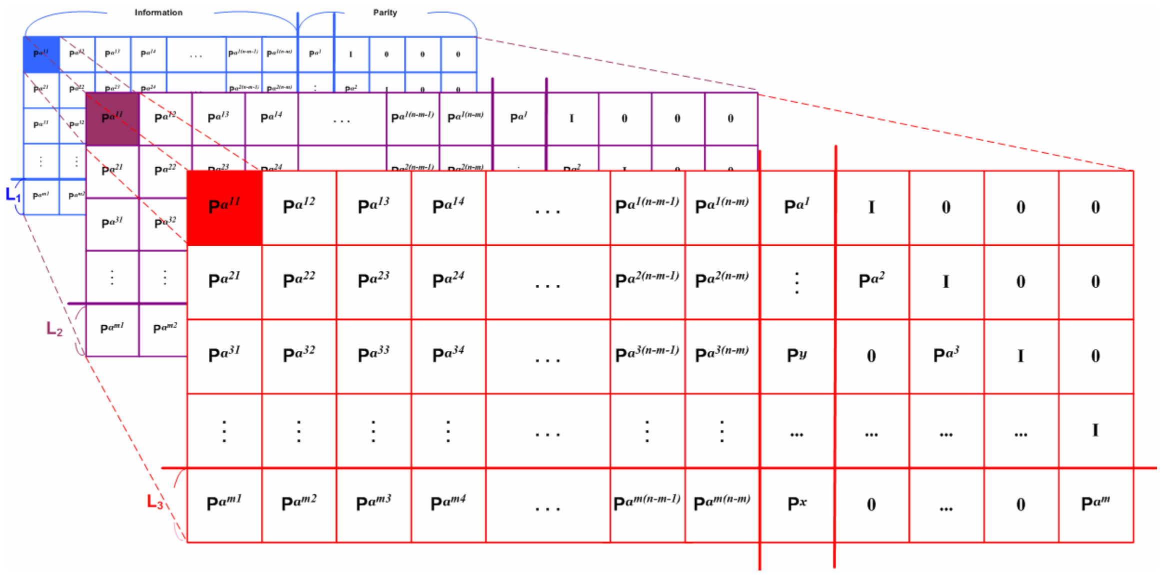


Figure 3. Parity-check matrix supporting code length flexibility

d. HARQ design

First of all, the lowest code-rate multi-edge-type parity-check matrix is designed to support HARQ. Then only a part of codeword is transmitted during each HARQ transmission (Incremental Redundancy). In other words, higher rate codes are achieved by puncturing parity bits from the lower rate codes. As a consequence, such LDPC codes are thus Rate-Compatible Punctured Codes (RCPC), suitable for IR Type II HARQ. Finally, it’s worth mentioning that since decoder doesn’t need to process edges associated with punctured nodes, this contributes to a reduction of the computation power at the receiver. Figure 4 illustrates how HARQ can be supported by our candidate design.

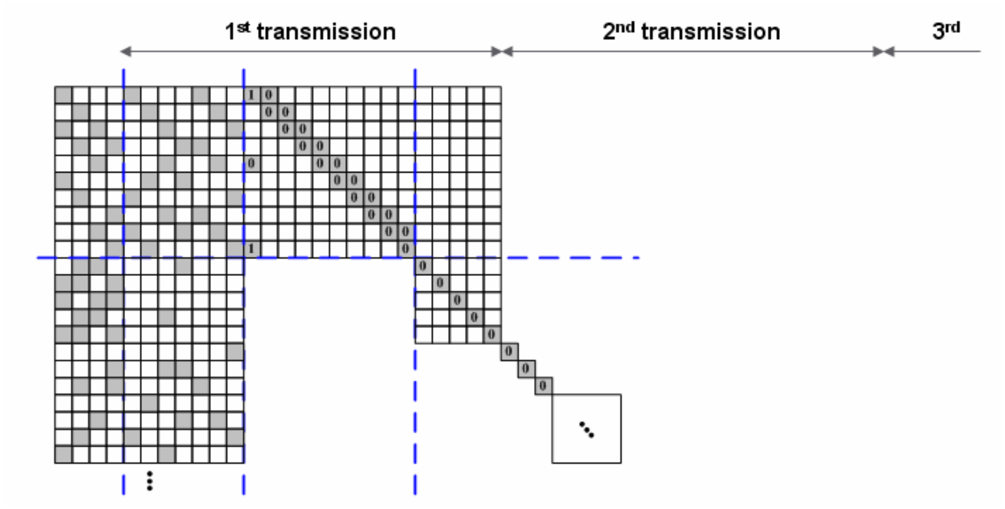


Figure 4. Parity-check matrix supporting HARQ

e. Encoding Algorithm

Encoding procedure for the proposed LDPC codes is accomplished by two steps as depicted in Figure 5 below. The first part of the parity-check matrix is encoded following Richardson & Urbanke’s encoding algorithm [3]. The second part is encoded by simple single parity-check coding.

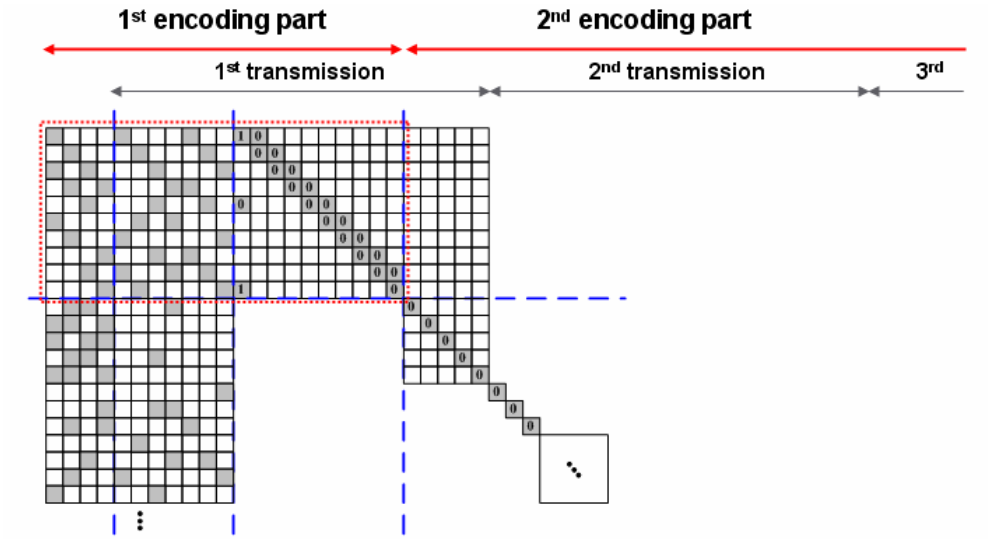


Figure 5. Parity-check matrix supporting efficient encoding

Figure 6 shows the partitioning of the parity-check matrix for Richardson & Urbanke's encoding algorithm on the structured LDPC codes. In figure 6, dimensions for submatrix *A* is (*m*-1)*L* x *kL*, *B* is (*m*-1)*L* x *L*, *T* is (*m*-1)*L* x (*m*-1)*L*, *C* is $L \times kL$, *D* is *L* x *L*, *E* is *L* x (*m*-1)*L*.

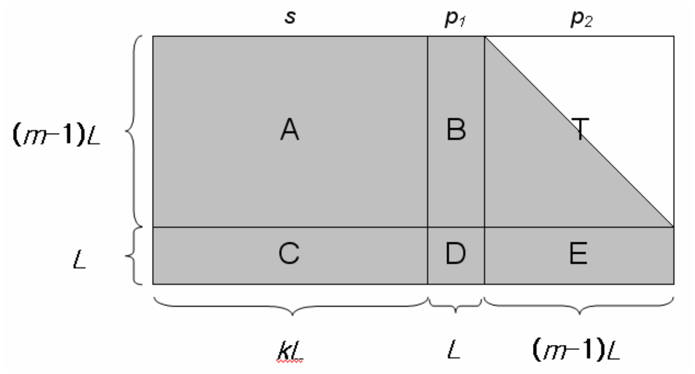


Figure 6. Parity-check matrix partitioning for R&U encoding algorithm

Let $c = (s, p_1, p_2)$ be a codeword where s denotes the systematic part, p_1 and p_2 denote the parity parts of length L and $(m-1)L$, respectively. Then the following equations should be satisfied:

$$As^T + Bp_1^T + Tp_2^T = 0 \qquad \textbf{Equation 1}$$

$$(ET^{-1}A + C)s^T + (ET^{-1}B + D)p_1^T = 0 \qquad \textbf{Equation 2}$$

Thus,
$$p_1^T = \phi^{-1}(ET^{-1}A + C)s^T, \ \phi := ET^{-1}B + D$$

We assume for the moment that ϕ is non-singular. Based on the above equations, the encoding procedure can be summarized as follows:

Step 1) Compute As^T and Cs^T.

Step 2) Compute $T^{-1}As^T$ by back-substitution .

Step 3) Compute $E(T^{-1}As^T)$ and $E(T^{-1}As^T)+Cs^T$.

Step 4) Compute $\phi = ET^{-1}B + D$ and ϕ^{-1}.

Step 5) Compute $p_1^T = \phi^{-1}(ET^{-1}As^T + Cs^T)$

Step 6) Compute p_2^T using $As^T + Bp_1^T + Tp_2^T = 0$ by back-substitution.

The computational complexity of such encoding procedure is of order $O(N)+O(L^2)$. The second term comes from multiplying by ϕ^{-1} in Step 5). In general, since the matrix ϕ^{-1} is not sparse, then multiplying by ϕ^{-1} is a main source of complexity increase within the encoding procedure. Therefore if we can make ϕ an identity matrix, we can skip the Step 5) of the multiplication by ϕ^{-1} and thus reduce greatly the overall encoding complexity.

We have identified [4] hereafter, several constraints on the sub-matrices (B, T, D, E) of the parity-check matrix guaranteeing that ϕ is always an identity matrix.

Constraints on $\phi = I$

- **B**: two non-zero element.
    - Position: 1st and arbitrary.
    - Shift factor: A (arbitrary number) and zero
- **T**: dual diagonal structure (accumulate chain)
    - Shift factor: all zero
- **D**: 1x1
    - Shift factor: A (same as 1st non-zero element in **B**)
- **E**: one non-zero element.
    - Position: right most.
    - Shift factor: zero

The resulting format of the constrained parity-check matrix is highlighted in Figure 7 below.

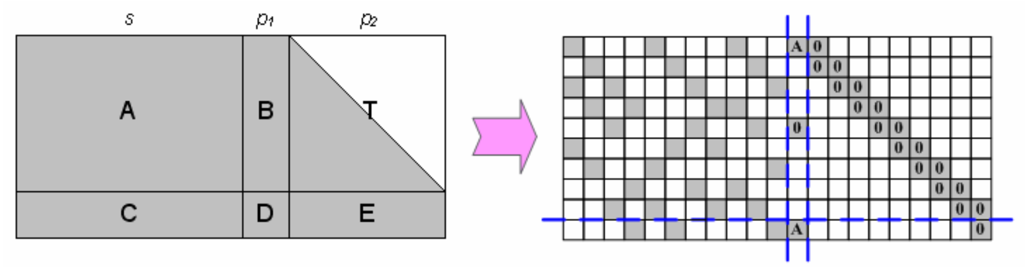


Figure 7. Parity part form of parity-check matrix for efficient encoding

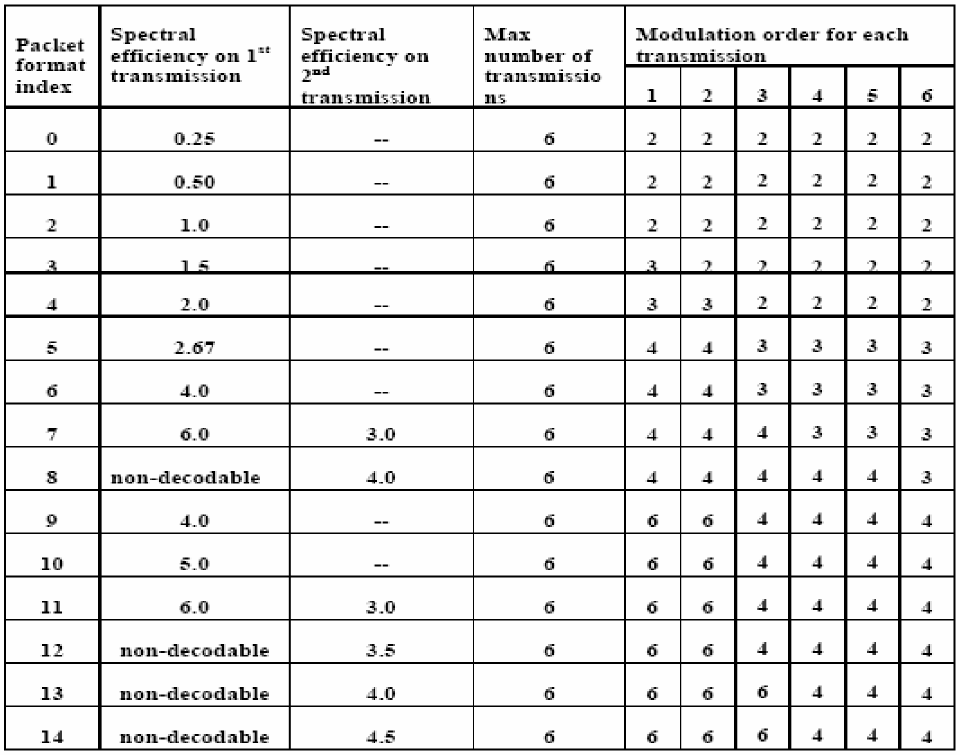
3. Packet format

The design of the parity-check matrices for the proposed LDPC codes follows the packet formats defined in [1] which are shown in Table 1 and Table 2.

Table 1. FL Packet formats

Packet Format Index	Spectral efficiency on 1st trans-mission	Spectral efficiency on 2nd trans-mission	Max number of trans-missions	Modulation order for each transmission					
				1	2	3	4	5	6
0	0.2	--	6	2	2	2	2	2	2
1	0.5	--	6	2	2	2	2	2	2
2	1.0	--	6	2	2	2	2	2	2
3	1.5	--	6	3	2	2	2	2	2
4	2.0	--	6	4	3	3	3	3	3
5	2.5	--	6	6	4	4	4	4	4
6	3.0	--	6	6	4	4	4	4	4
7	4.0	--	6	6	6	4	4	4	4
8	5.0	--	6	6	6	4	4	4	4
9	6.0	3.0	6	6	6	4	4	4	4
10	non-decodable	3.5	6	6	6	4	4	4	4
11	non-decodable	4.0	6	6	6	6	4	4	4
12	non-decodable	4.5	6	6	6	6	4	4	4
13	non-decodable	5.0	6	6	6	6	6	4	4
14	non-decodable	5.5	6	6	6	6	6	4	4
15	NULL	NULL							

Table 2. RL Packet Formats



Packet format index	Spectral efficiency on 1st transmission	Spectral efficiency on 2nd transmission	Max number of transmissions	Modulation order for each transmission					
				1	2	3	4	5	6
0	0.25	--	6	2	2	2	2	2	2
1	0.50	--	6	2	2	2	2	2	2
2	1.0	--	6	2	2	2	2	2	2
3	1.5	--	6	3	2	2	2	2	2
4	2.0	--	6	3	3	2	2	2	2
5	2.67	--	6	4	4	3	3	3	3
6	4.0	--	6	4	4	3	3	3	3
7	6.0	3.0	6	4	4	4	3	3	3
8	non-decodable	4.0	6	4	4	4	4	4	3
9	4.0	--	6	6	6	4	4	4	4
10	5.0	--	6	6	6	4	4	4	4
11	6.0	3.0	6	6	6	4	4	4	4
12	non-decodable	3.5	6	6	6	4	4	4	4
13	non-decodable	4.0	6	6	6	6	4	4	4
14	non-decodable	4.5	6	6	6	6	4	4	4

4. Simulation Results

We provide in this part several simulation results to compare the performance between LDPC codes and Turbo codes in terms of Frame Error Rate (FER) over AWGN Channel. It is assumed that 440 modulated symbols are transmitted for each HARQ transmission. Decoding for LDPC codes is performed by standard Pearl's Belief-Propagation (BP) algorithm with respectively 25, 50 and 100 maximum number of iterations under flooding scheduling. On the other hand, for Turbo codes, Log-MAP algorithm is performed with a maximum number of 12 iterations.

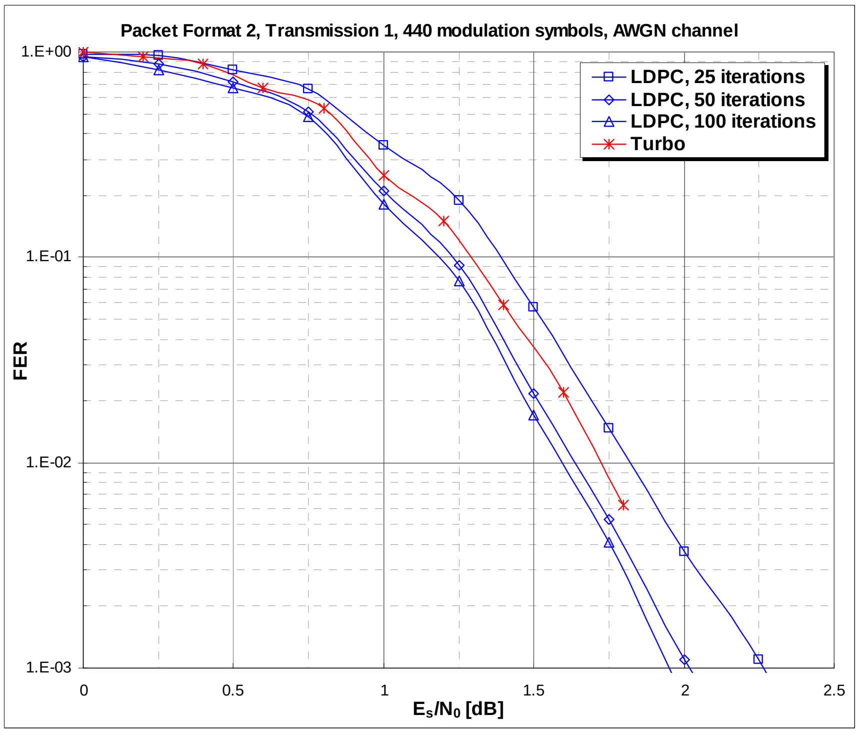


Figure 8. AWGN Performance, FL PF 2, Transmission 1

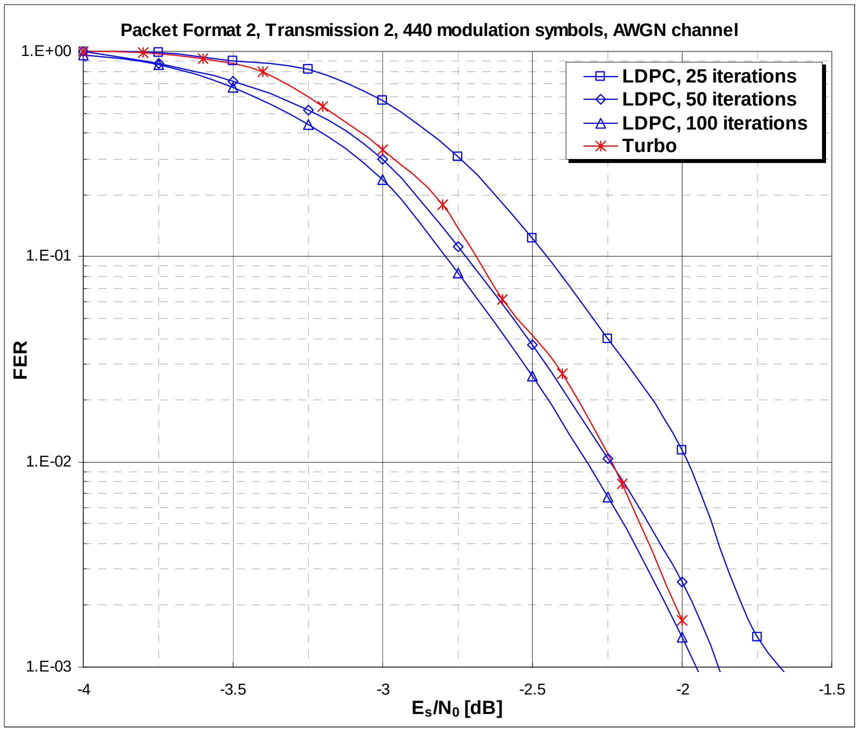


Figure 9. AWGN Performance, FL PF 2, Transmission 2

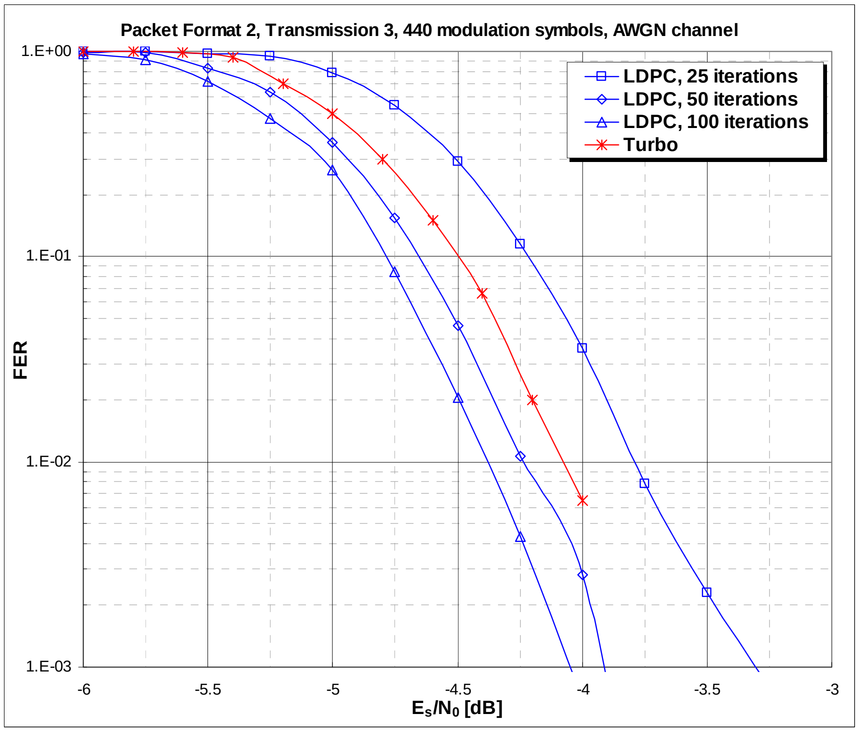


Figure 10. AWGN Performance, FL PF 2, Transmission 3

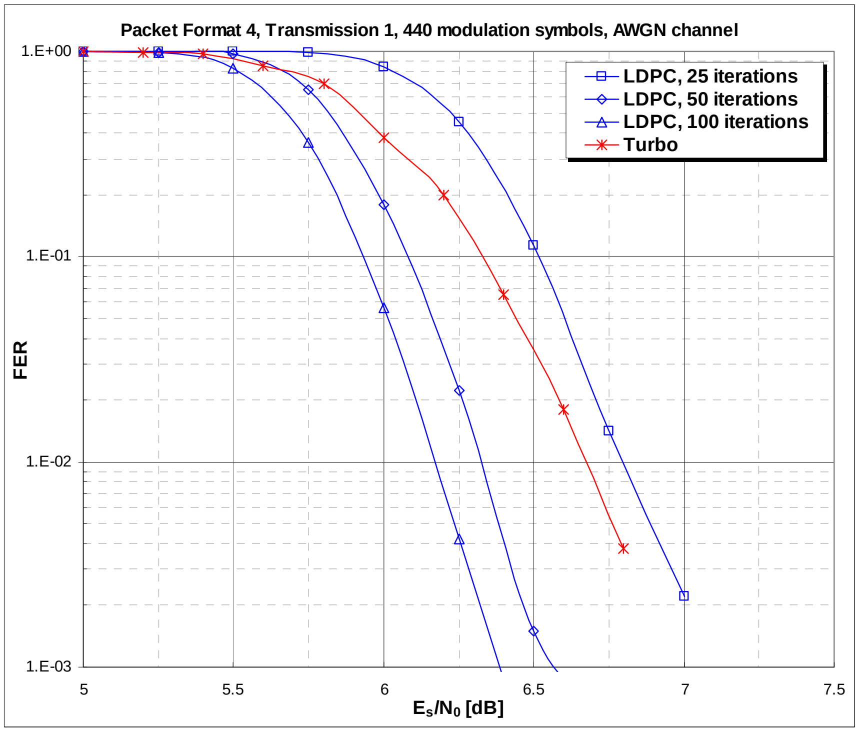


Figure 11. AWGN Performance, FL PF 4, Transmission 1

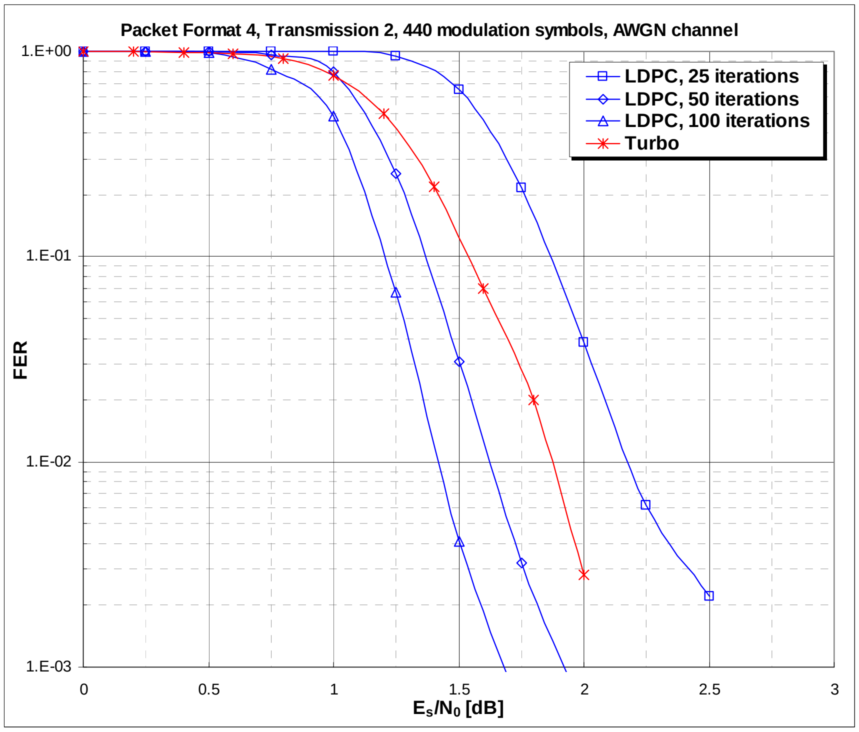


Figure 12. AWGN Performance, FL PF 4, Transmission 2

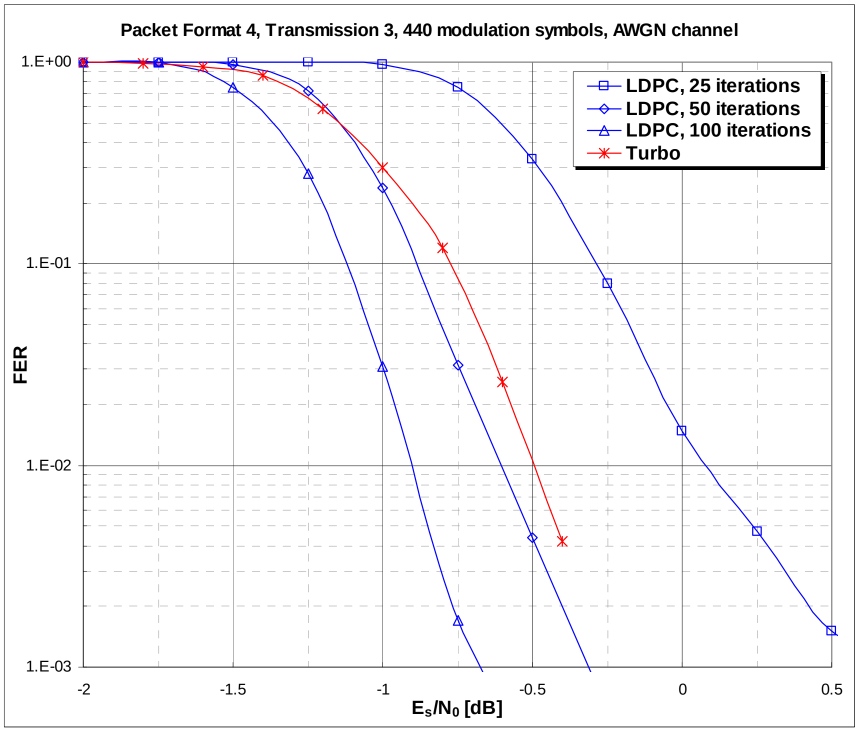


Figure 13. AWGN Performance, FL PF 4, Transmission 3

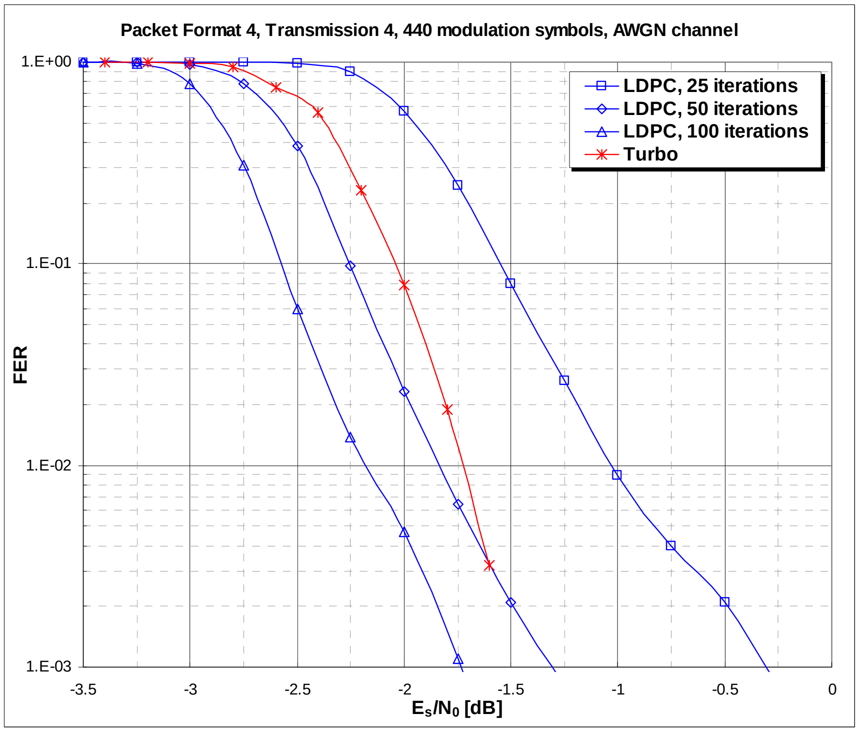


Figure 14. AWGN Performance, FL PF 4, Transmission 4

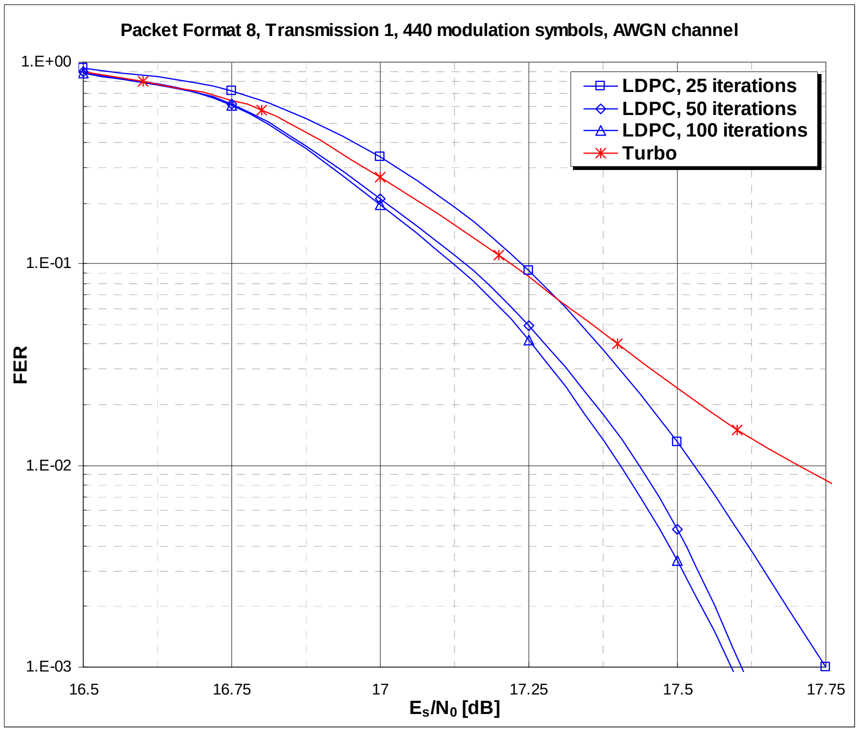


Figure 15. AWGN Performance, FL PF 8, Transmission 1

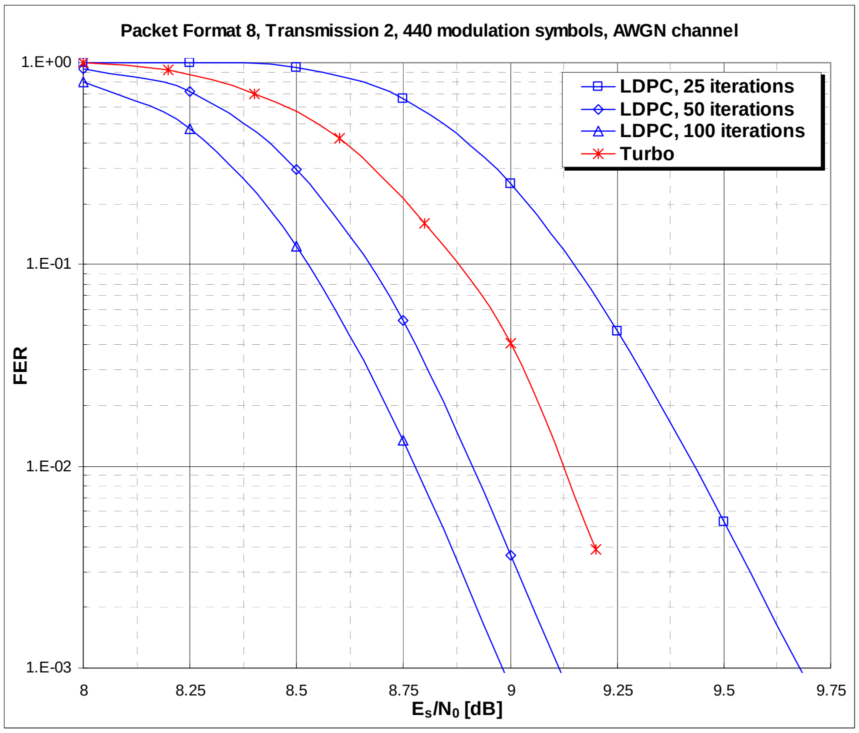


Figure 16. AWGN Performance, FL PF 8, Transmission 2

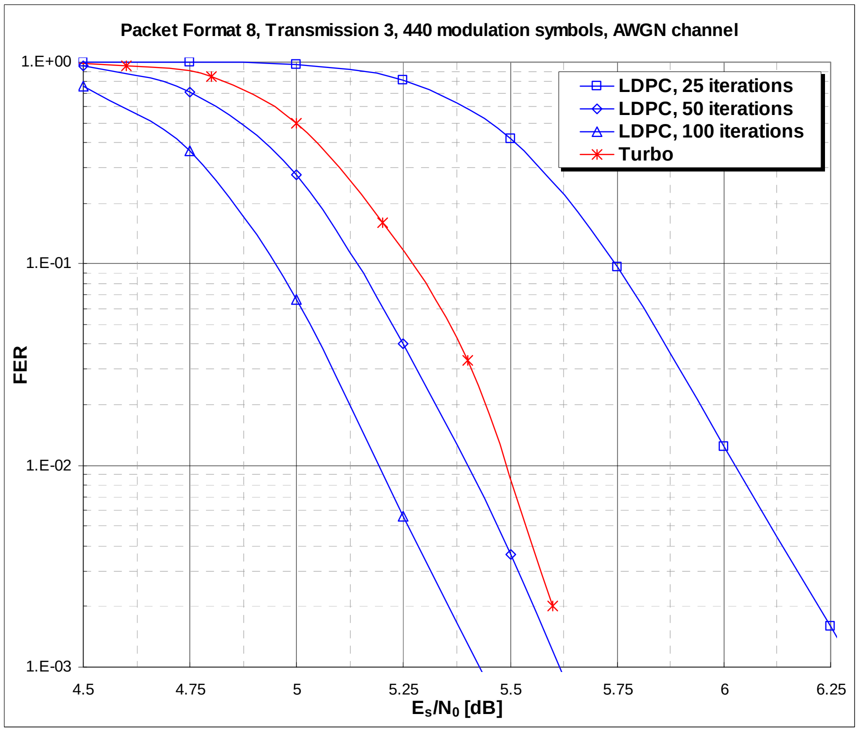


Figure 17. AWGN Performance, FL PF 8, Transmission 3

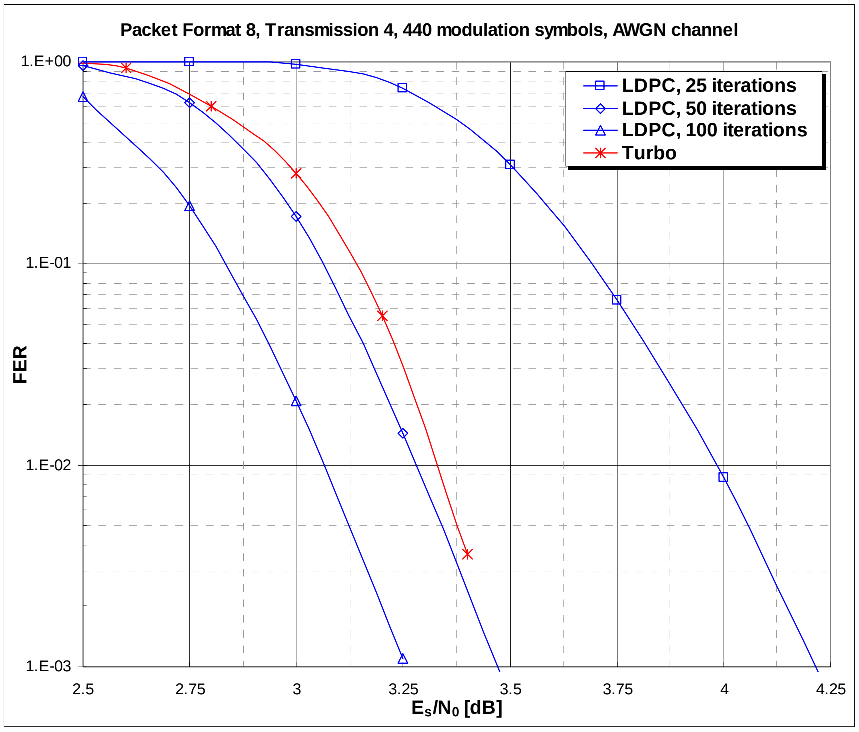


Figure 18. AWGN Performance, FL PF 8, Transmission 4

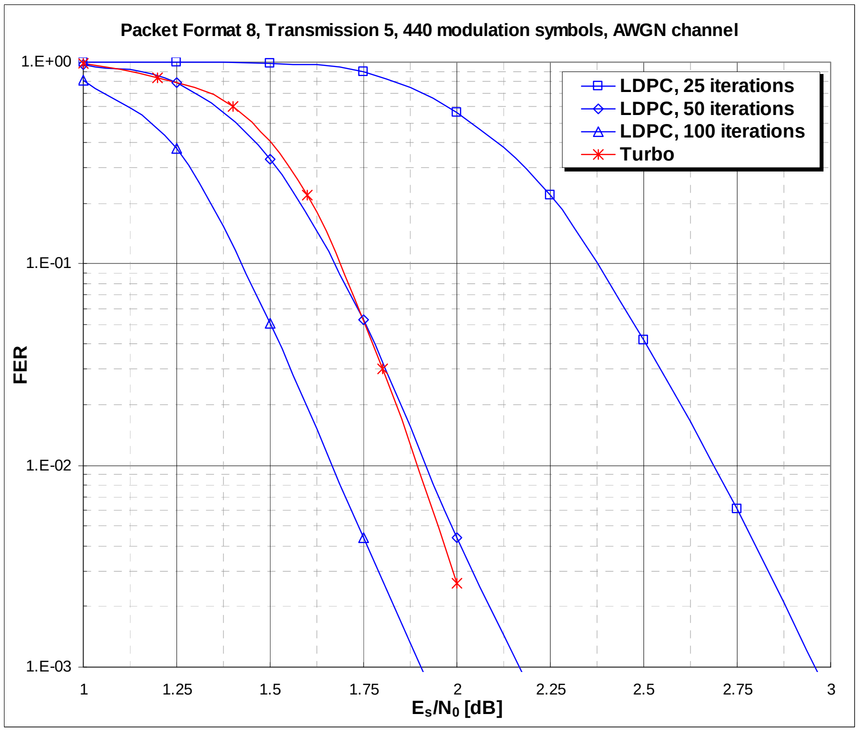


Figure 19. AWGN Performance, FL PF 8, Transmission 5

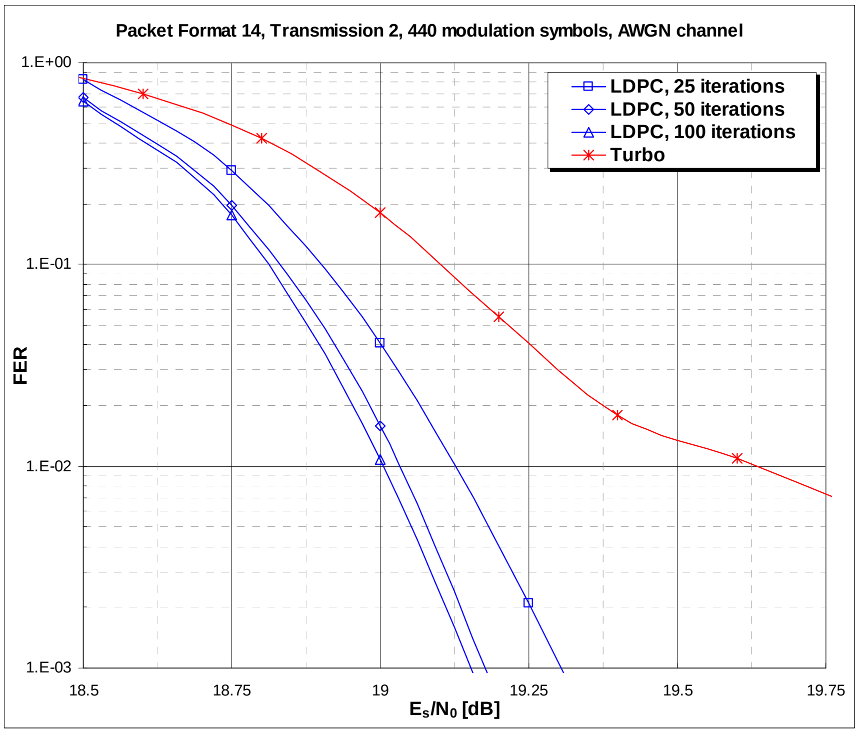


Figure 20. AWGN Performance, FL PF 14, Transmission 2

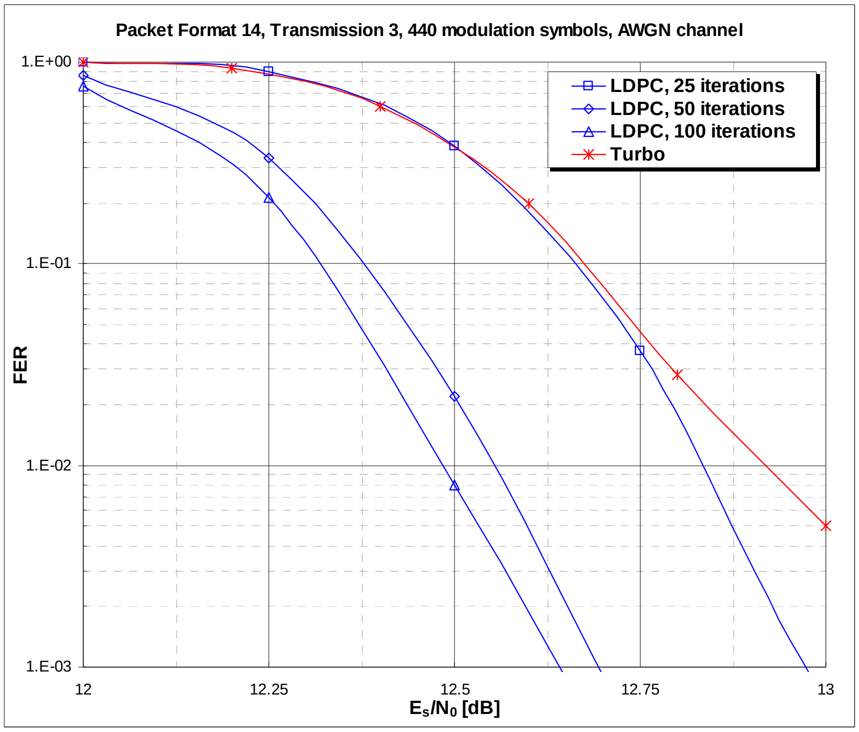


Figure 21. AWGN Performance, FL PF 14, Transmission 3

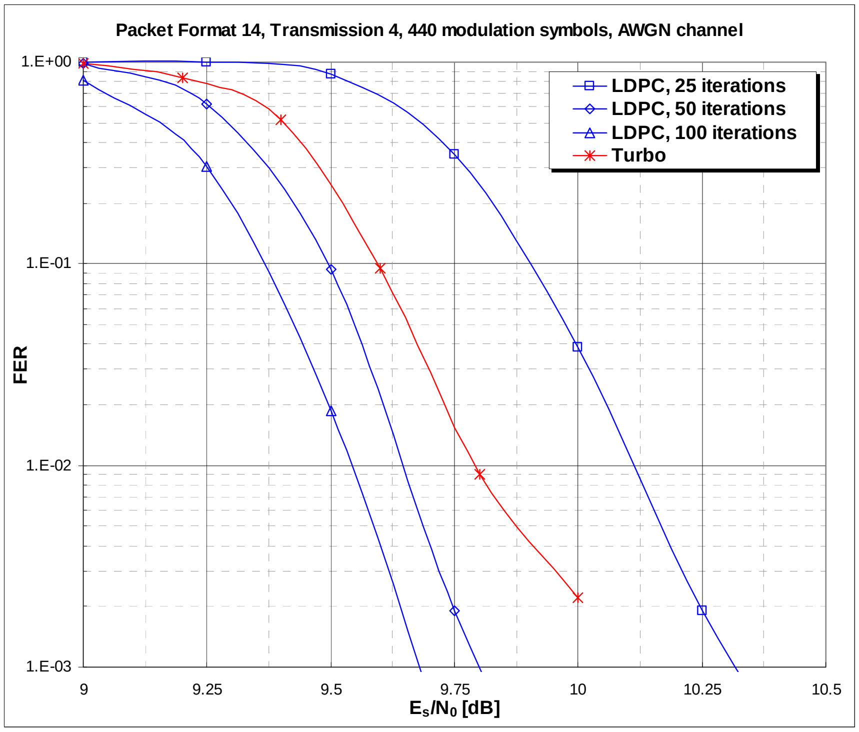


Figure 22. AWGN Performance, FL PF 14, Transmission 4

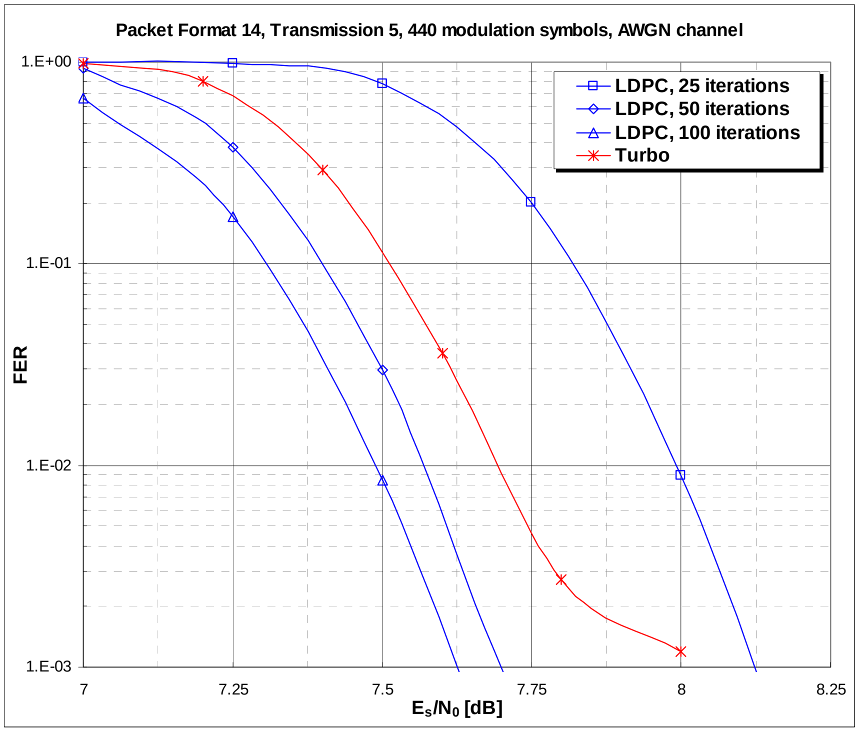


Figure 23. AWGN Performance, FL PF 14, Transmission 5

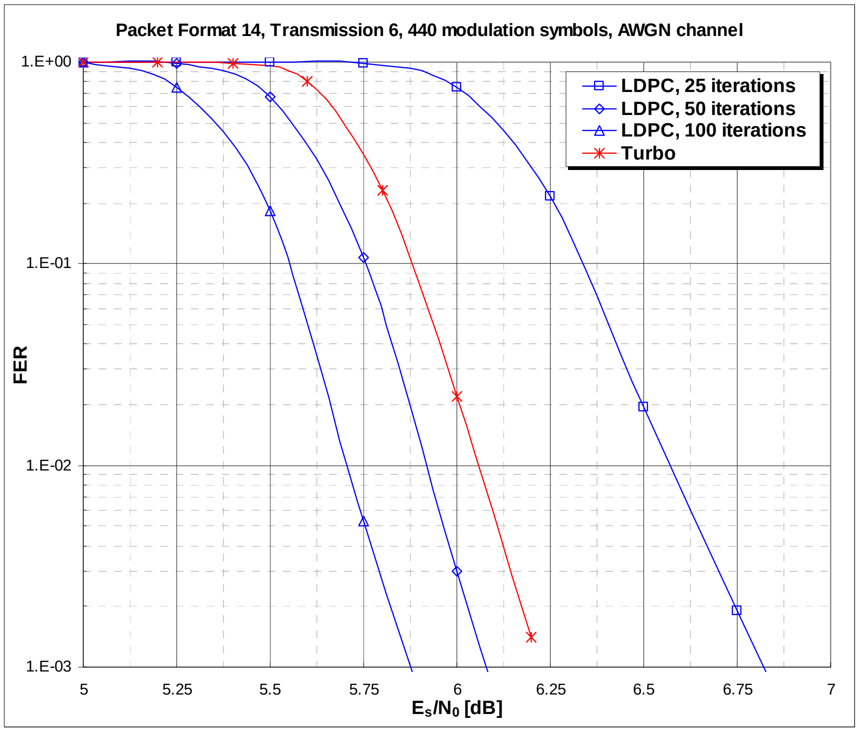


Figure 24. AWGN Performance, FL PF 14, Transmission 6

5. Conclusions

The proposed LDPC codes offer both efficient support of Type II HARQ (Incremental Redundancy) together with similar or better performance than Turbo codes through all HARQ retransmissions. Besides, this proposed code structure enables highly parallelizable decoder architectures, thus resulting in high-throughput decoder implementations.

References

[1] ‘Draft Standard for Local and Metropolitan Area Networks - Standard Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility - Physical and Media Access Control Layer Specification’, IEEE P802.20/D2.1, May 2006.

[2] A. J. Blanksbyand C. J. Howland, “A 690-mW 1-Gb/s 1024-b, Rate-1/2 Low-Density Parity-Check Code Decoder, IEEE Journal of solid-state circuits, vol. 37, no. 3, March 2002.

[3] T. J. Richardson and R. Urbanke, “Efficient encoding of low-density parity-check codes,” IEEE Transactions on Information Theory, vol. 47, no. 2, pp.638-656, Feb. 2001.

[4] S. Myung, K. Yang and J. Kim, “Quasi-Cyclic LDPC Codes for Fast Encoding”, IEEE Trans. on Info. Theory, Vol.51, N.8, Aug. 2005