A Novel Hybrid ARQ Scheme Based on LDPC Code Extension and Feedback

Hamid Saber and Ian Marsland
Department of System and Computer Engineering, Carleton University, Ottawa, ON, Canada
{hsaber,ianm}@sce.carleton.ca

***Abstract*—The design of an efficient hybrid automatic repeat request (ARQ) scheme based on rate compatible low density parity check (LDPC) codes is considered. It has been shown that extending as well as puncturing of LDPC codes can produce good rate compatible LDPC codes for the additive white Gaussian noise channel. One issue with the traditional LDPC-based hybrid ARQ methods is that the throughput drops off significantly at low signal-to-noise ratios (SNRs). In this paper we introduce a coding scheme which is capable of using puncturing, extending and feedback at the same time to address this issue. Appropriate choice of feedback functions along with optimum combining of received signals for the belief propagation decoder mitigates the throughput drop-off issue at low SNRs, while having a small feedback overhead from the receiver. A powerful mother code is generated via the progressive edge growth algorithm and is used for puncturing and extending in the proposed scheme. Clustering the codewords of the longest codebook is used to decrease the overhead of the feedback connection. Simulation analysis of the throughput shows that our scheme could get as close as 0.5 dB to the Shannon limit while having up to 2 dB gain compared to previous works at low SNRs.**

***Index Terms*– Hybrid ARQ, LDPC codes, Rate Compatible Codes, Incremental Redundancy, Belief Propagation, Tanner Graph**

I. Introduction

Channel state variations in digital communication through wireless media urged the need for coding schemes with rate flexibility to improve resource utilization [1]. Rate compatible (RC) codes using convolutional codes have been proposed to address this issue [2] and [3]. RC codes have some important advantages over fixed rate codes including their ability to implement efficient error control using automatic repeat request (ARQ) protocols. Also RC codes can be implemented via a single encoder and decoder thus become an important tool for flexible-rate error control coding. RC convolutional codes using puncturing have been widely studied, showing good performance over a range of SNRs [2] and [3]. First introduced by Gallager [4], low density parity check (LDPC) codes along with their low complexity message-passing decoders are amongst the most promising codes nowadays.

There has been a lot of research on the construction of LDPC codes that approach the capacity of a wide variety of channels, including the additive white Gaussian noise (AWGN) channel. These construction methods generally fall into two different categories, random or pseudorandom methods [5], [6], [7] and algebraic methods [8]. Progressive edge growth (PEG) construction of LDPC codes was introduced in [7] to maximize the girth of the parity check matrix with a given node degree distribution. Great successes in designing promising fixed rate LDPC codes for both medium and long code lengths motivated the development of RC-LDPC codes. Ha et al. considered optimal puncturing, in the sense of maximizing the threshold of the belief propagation (BP) decoder, of LDPC codes to obtain higher-rate codes from a lower-rate LDPC mother code [9]. Although puncturing is a good method for constructing RC-LDPC codes, the authors in [10] showed that it is not capable of yielding very good high-rate codes from a low-rate mother code, when a large number of code bits must be punctured. They proposed an alternative approach (extending) in which the parity check matrix of a high-rate LDPC code is embedded in the parity check matrix of a low-rate LDPC code. For the extra parity bits of the extended code to contain the previous code bits, the extended matrix must possess a certain structure, which we will discuss later in this paper. In [11] both extending and puncturing is used to produce RC-LDPC codes.

One issue with hybrid ARQ methods with LDPC codes is that in spite of their effectiveness at high SNRs, their throughput drops off significantly at low SNRs. In this paper we devise a scheme that uses both puncturing and extending to give high throughputs for intermediate and high SNRs and also uses feedback through a reverse channel from the receiver to improve the throughput at low SNRs. It is of crucial importance what kind of information should be fed back from the receiver. Roughly speaking, the receiver gets more benefit from the retransmission of some previously non-well-received bits instead of receiving new parity bits.

The rest of this paper is organized as follows. In Section II preliminaries of LDPC codes and BP decoding on Tanner graph are presented. Section III contains the system model, including the transmitter and receiver structure. The details of the extended matrix structure, the feedback connection and the proposed system structure are explained in this section. Simulation results for performance analysis and comparisons are presented in Section IV. Section V concludes the paper.

II. Preliminaries

A LDPC code can be either represented by its parity check matrix or the corresponding Tanner graph. A binary parity check matrix $\mathbf{H} = [h_{i,j}]$ of size $M \times N$ represents a LDPC code of design rate $R = 1 - \frac{M}{N}$. The actual rate of the

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

code might be larger than this rate as the matrix may not be full-rank. Associated with every such parity check matrix is a Tanner graph [12]. A LDPC Tanner graph is a bipartite undirected graph consisting of two sets of nodes: the set of check nodes $\mathcal{C} = \{1, 2, \dots, M\}$, and the set of variable nodes $\mathcal{V} = \{1, 2, \dots, N\}$. There is an edge between check node $i \in \mathcal{C}$ and variable node $j \in \mathcal{V}$ iff $h_{i,j} = 1$. A node degree is defined as the number of edges connected to that node. The check degree of an edge is defined as the degree of the check node connected to that edge. Similarly, the variable degree of an edge is the degree of the variable node connected to that edge. $\mathcal{C}_v$ is the set of all check nodes connected to the variable node v and $\mathcal{V}_c$ is the set of all variable nodes connected to the check node c.

The BP decoder is a special kind of message-passing decoder used for decoding LDPC codes on Tanner graphs. Working on the Tanner graph, the BP decoder passes messages along the edges of the graph between the nodes. These messages are beliefs or the probabilities of the values of the corresponding variable nodes on the edges. In case of binary input AWGN (BIAWGN) channels the messages could be as simple as a scalar representing the log-likelihood ratio (LLR) of the variable node values. The process of passing messages between nodes is done iteratively starting from iteration 0. At iteration 0 all variable nodes $v_i \in \mathcal{V}$ send an initial message $m_{v_i}^{(0)} = \frac{4r_i}{N_0}$ to their connected check nodes, where r_i is the received AWGN channel sample corresponding to node v_i and N_0 is the single-sided power spectral density of the AWGN channel. Iteration i starts by sending messages from each check node c to all the variable nodes $v \in \mathcal{V}_c$. This message is denoted by $m_{c,v}^{(i)}$. Afterwards, each variable node v sends a message $m_{v,c}^{(i)}$ to all check nodes $c \in \mathcal{C}_v$. The BP decoder uses the following recursion to update the estimators through iterations and checks if the decoded codeword satisfies parity check equations or not. The iteration of the BP decoder is as follows

$$m_{c,v}^{(i)} = \ln \left(\frac{1 + \prod_{v' \in \mathcal{V}_c \setminus v} \tanh \left(\frac{m_{v',c}^{(i-1)}}{2} \right)}{1 - \prod_{v' \in \mathcal{V}_c \setminus v} \tanh \left(\frac{m_{v',c}^{(i-1)}}{2} \right)} \right) \tag{1}$$

$$m_{v,c}^{(i)} = m_v^{(0)} + \sum_{c' \in \mathcal{C}_v \setminus c} m_{c',v}^{(i)}. \tag{2}$$

A regular LDPC code is a code in which all check nodes are of the same degree and all variable nodes are of the same degree as well. Irregular LDPC codes are characterized by degree distribution pairs (λ, ρ) where $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_i \rho_i x^{i-1}$. λ_i is the fraction of edges in the Tanner graph with variable degree of i and ρ_i is the fraction of edges with a check degree of i. The degree distribution can be also defined in a node perspective approach. The node perspective degree distribution pairs are denoted by $\lambda'(x) = \sum_i \lambda'_i x^{i-1}$ and $\rho'(x) = \sum_i \rho'_i x^{i-1}$ where λ'_i and ρ'_i are the fraction of variable and check nodes with degree i, respectively. These two representations of degree distribution pairs are equivalent. It has been shown that the edge degree distribution is a crucial factor in the performance of LDPC codes [6].

The length of the smallest cycle in the LDPC Tanner graph is called the girth of the code. Beside the degree distribution, the girth plays an important role in LDPC code performance. Roughly speaking, large values of girth are desired. The PEG algorithm in [7] constructs LDPC codes with large girth for a given degree distribution. PEG codes have shown promising performances for AWGN channel applications and are used in this paper as well.

Puncturing is often used to produce higher-rate codes from a mother code. In the case of LDPC codes, the best estimation for the punctured variable node at the decoder is an equiprobable choice between 0 and 1, so the corresponding LLR is zero for the initial message at the punctured variable nodes. This is equivalent to considering the received value for the punctured variable node as 0 or as an erasure. This changes the probability density function (pdf) of the initial message from a pure Gaussian density to a Gaussian density with a Dirac delta function and produces a less powerful code than the mother code. Finding good punctured codes from a low-rate mother code seems unlikely for higher desired rates of the punctured code. Therefore, the idea of extension is introduced to obtain good punctured codes from a relatively high-rate mother code and constructing lower-rate codes by extending the mother code. The details of extending and puncturing are given in the next section.

III. Sytem Model and the Proposed Hybrid ARQ Scheme

The communication channel in this paper is considered to be the AWGN channel with single-sided power spectral density N_0. Binary phase shift keying (BPSK) with antipodal signaling $\{+1, -1\}$ is used. The LDPC mother code for the hybrid ARQ scheme is generated via the PEG algorithm. Note that our proposed method is quite general and can be used for arbitrary code rates and degree distribution pairs. However, to have a fair comparison between the contribution of this paper and previous work for our simulations we chose the same rate and degree distribution as in [11]. The rate of the mother code is chosen to be $8/13$. The irregular mother code of length 3328 is generated with the optimized variable node perspective degree distribution of

$$\begin{aligned} \lambda'(x) = 0.47532x + 0.27953x^2 + 0.0348x^3 \\ +0.10889x^4 + 0.10138x^{14}. \end{aligned} \tag{3}$$

A. Extending the Parity Check Matrix

As it is said before lower-rate and more powerful codes can be constructed by extending a mother code. Fig. 1 depicts the structure of the mother code as well as the extended codes used in our scheme. As it can be seen from the figure, 12 extensions are used. The mother code parity check matrix $\mathbf{H}^{(0)}$ is a 1280×3328 matrix with degree distribution given in (3). Each extension matrix $\mathbf{H}_{ext}$ is a 256×256 square matrix that is generated by the PEG algorithm with the same degree distribution as the mother code. All identity matrices are of the same size as $\mathbf{H}_{ext}$. Therefore the parity check matrix of the l-th extended code is of size $(5+l)256\times(13+l)256, l=1,2,\ldots,12$. The identity matrices are to make correlation between the extended code bits and the previously transmitted code bits. The role of identity matrices is crucial since poorly received bits of extended code l should be related to code bits of extended code $l+1$ to be corrected. Let $\mathbf{G}^{(l)}$ be the generator matrix of the code l. For the code bits of $(l-1)$-th code to be embedded in the l-th code, the condition

$$\mathbf{G}^{(l)}=[\mathbf{G}^{(l-1)}|\mathbf{G}_l] \tag{4}$$

needs to be satisfied. The following lemma explains the special structure of the code in Fig. 1.

Lemma III.1. *Let the parity check matrices of two linear block codes on the binary field be related by*

$$\mathbf{H}^{(2)}=\begin{bmatrix}\mathbf{H}^{(1)} & \mathbf{0}\\ \mathbf{A} & \mathbf{H}_{ext}\end{bmatrix}.$$

If $\mathbf{H}_{ext}$ is a full-rank square matrix, there exists a matrix $\mathbf{G}_2$ such that the generator matrix $\mathbf{G}^{(2)}$ of the second code can be written as

$$\mathbf{G}^{(2)}=\left[\ \mathbf{G}^{(1)}|\mathbf{G}_2\ \right] \tag{5}$$

where $\mathbf{G}^{(1)}$ is the generator matrix of the first code.

Proof: $\mathbf{G}^{(2)}$ must satisfy

$$\mathbf{G}^{(2)}\mathbf{H}^{(2)^T}=\mathbf{0} \tag{6}$$

which can be written as

$$[\mathbf{G}^{(1)}|\mathbf{G}_2]\begin{bmatrix}\mathbf{H}^{(1)} & \mathbf{0}\\ \mathbf{A} & \mathbf{H}_{ext}\end{bmatrix}^T=\mathbf{0} \tag{7}$$

or

$$\left[\ \mathbf{G}^{(1)}\mathbf{H}^{(1)^T}|\mathbf{G}^{(1)}\mathbf{A}^T+\mathbf{G}_2\mathbf{H}_{ext}^T\ \right]=\mathbf{0}. \tag{8}$$

Since $\mathbf{G}^{(1)}\mathbf{H}^{(1)^T}=\mathbf{0}$ it is just enough to have

$$\mathbf{G}^{(1)}\mathbf{A}^T+\mathbf{G}_2\mathbf{H}_{ext}^T=\mathbf{0}. \tag{9}$$

If $\mathbf{H}_{ext}$ is full-rank, then $\mathbf{G}_2$ can be obtained as

$$\mathbf{G}_2=\mathbf{G}^{(1)}\mathbf{A}^T\left(\mathbf{H}_{ext}^T\right)^{-1}. \tag{10}$$

This completes the proof.

■

The above lemma explains the reason behind putting the all-zero matrix in the upper right corner of Fig. 1. In this way, going from one extension to the next satisfies the lemma conditions and the additional code bits are generated by the appropriate extending generator matrix. The extending matrix $\mathbf{H}_{ext}$ is checked for being full-rank as required by the lemma. The special matrix structure in Fig. 1 results in the simple encoder structure.

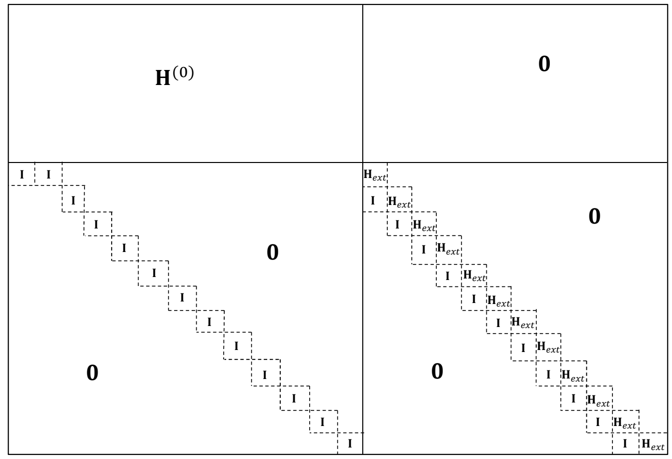

Fig. 1. Parity check matrix structure of the extended codes.

B. A Traditional Type II Hybrid ARQ Scheme

With traditional Type II hybrid ARQ, also called incremental redundancy hybrid ARQ, schemes operating on the code in Fig. 1, codeword clustering is used. The whole codeword is partitioned into clusters of S code bits, yielding a total of N/S clusters, where N is the length of the longest extended codeword. The traditional hybrid ARQ scheme uses 4 phases. In the first phase, a total of K/S clusters are sent (i.e. K code bits), where K is the number of information bits. The code bits with the highest node degrees are selected in this phase, since the high-degree nodes tend to be more problematic when punctured as they produce larger number of erasures in the initial messages.

If decoding fails (which will likely be the case except at high SNRs), the second phase starts. An additional cluster, consisting of S code bits with highest degree that have not already been sent, is transmitted. Decoding over the whole mother code is attempted, and if decoding fails, another cluster is transmitted. This process is repeated until either successful decoding occurs or the entire mother code has been transmitted.

If successful decoding is still not possible, the third phase starts. Clusters of the first extended code are transmitted, and after each cluster is received decoding is attempted on the Tanner graph of the first extended code. If, after receiving all clusters of the first extended code, decoding still fails, then clusters of the second extended code are transmitted and decoding is performed over the Tanner graph of the second extended code. This process is repeated until all the clusters of all the extended codes have been transmitted. If decoding still fails (which is only likely at low SNRs), then the fourth phase starts.

In this phase, previously transmitted clusters are retransmitted, one at a time and in the same order as previously transmitted. Decoding is done on the appropriate Tanner graph, i.e. if the retransmitted cluster belongs to the mother code it is decoded on the Tanner graph of the mother code and if not it is decoded on the Tanner graph of the corresponding extended code. The received samples of the retransmission are combined with the previously received samples using maximum ratio combining (MRC), so the initial LLR of multiple-received bit i is calculated as

$$m_i^{(0)} = \frac{4\sum_t r_i^{(t)}}{N_0} \tag{11}$$

where $\{r_i^{(t)}\}$ is the set of multiple-received samples at variable node i. This process continues until a decoding success is achieved.

C. Feedback Information

One problem with hybrid ARQ schemes with LDPC codes is the sharp throughput drop-off as the SNR decreases. Although code extension is able to widen the operating SNR range of the system, our simulation results show that it can not prevent a sharp drop-off in throughput at low SNRs. To address this issue, we propose making use of a feedback connection. The idea is to inform the transmitter of which code bits would be most beneficial to transmit next. In particular, it is desirable to retransmit those code bits that have not been received reliably. Since the magnitude of a LLR is an indicator of reliability, a LLR with a low absolute value can be considered as a *bad* LLR and corresponding code bit can be a good candidate for retransmission. To retransmit a single bit the receiver would need to feed back its index to transmitter, but this would require $\log_2(N)$ feedback bits, which is impractical because the feedback overhead would be too large. As an alternative approach, all the code bits in a cluster can be asked for retransmission. With a cluster size of S greater than 1 this will decrease the number of feedback bits to $\log_2(N/S)$ for every S transmitted bits. In practice a small value of overhead $\frac{\log_2(N/S)}{S}$ can be tolerable.

The criteria for choosing the best candidate between all clusters for a retransmission should be well pondered. The worst received bit is the one with the minimum absolute LLR and thus the cluster with the code bit which has the minimum absolute LLR could be a good candidate. However, our simulation results show it is better to select the cluster with lowest sum of the absolute values of the LLRs in the cluster. Furthermore it is better to base this calculation on the final LLRs after decoding instead of the initial LLRs. This is because it is more important to retransmit code bits that can not be recovered after decoding than ones that are received unreliably. Assuming the decoder runs for a fixed maximum number of decoding iterations, $MaxItr$, the best cluster for retransmission is given by

$$Best\,Candidate = \text{argmin}_{cluster} \sum_{i \in cluster} \left| LLR_i^{(MaxItr)} \right| \tag{12}$$

where

$$LLR_i^{(MaxItr)} = m_i^{(0)} + \sum_{c' \in \mathcal{C}_i} m_{c',i}^{(MaxItr)}. \tag{13}$$

In the proposed scheme, at the beginning of the transmission the first two phases are done just as in the traditional method. However, phase 3 and 4 are modified in such a way that if the decoder fails it sends a request for transmission of the best candidate and decodes on the current extended Tanner graph. This cluster might or might not have already been transmitted. When multiple samples of the same code bit are received, MRC is used to calculate the LLRs according to (11).

IV. Simulation Results and Performance Analysis

The performance analysis of the proposed scheme is presented in this section and is compared with the previous work [11]. The throughput, which is defined as the number of bits in a message frame divided by the average number of code bits required for successful decoding, is used for the performance measure. The throughput is calculated by

$$T = \frac{K}{\sum_i N_i p_{i|(i-1)}} \tag{14}$$

where N_i is the number of transmitted code bits after the i^{th} transmission and $p_{i|(i-1)}$ is the probability of decoding success after the i^{th} transmission given decoding failures at all previous transmissions.

The maximum number of iterations of the decoder is set to be 500 and a message size $K = 2048$ is used. Simulation results for the throughput of the proposed method with $(S = 64)$ are depicted in Fig. 2 and compared with the non-feedback method as in [11]. The throughput is plotted versus E_s/N_o, where E_s is the transmitted energy per code bit. Also plotted is the information theoretical capacity of BIAWGN channel and the performance of the non-feedback method [11]. It can be seen that by using feedback as the process switches to the extension mode, the throughput drop-off issue at low SNRs is mitigated, with up to 2 dB gain over the non-feedback scheme. At intermediate SNRs, it gets as close as 0.5 dB to the Shannon limit. Fig. 3 shows the throughput of our scheme for different cluster sizes. It is observed from the

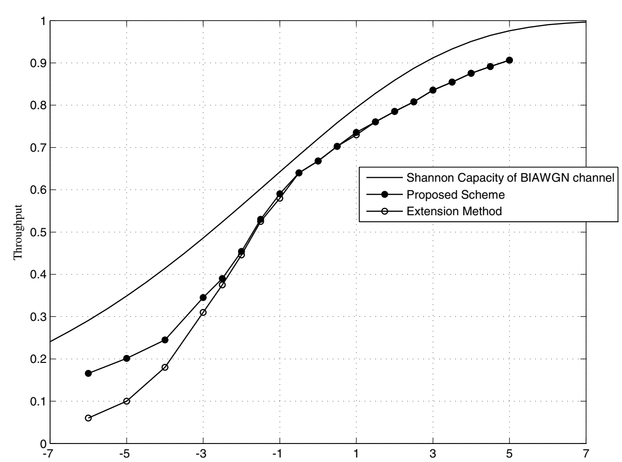

Fig. 2. Throughput of the proposed scheme.

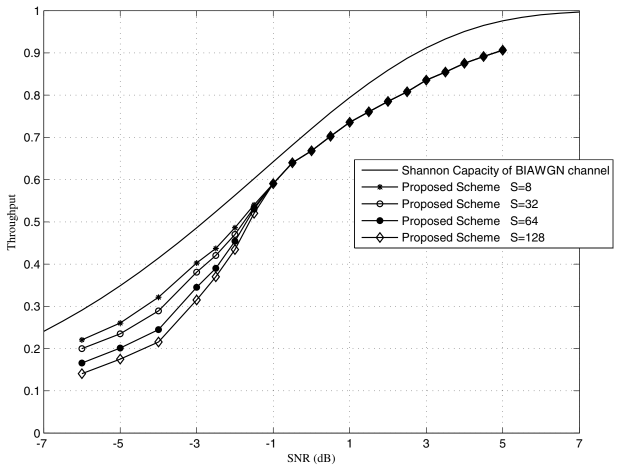

Fig. 3. Throughput of the proposed scheme for different cluster sizes.

figure that even a small amount of feedback (e.g. a large cluster size such as $S = 128$) increases the range of operation of hybrid ARQ schemes with LDPC codes and combats the throughput drop-off problem. Although a small cluster size is not practical, because of the excessive feedback overhead, its throughput is plotted to have a comparison between different cluster sizes. A notable remark about the cluster size is that it should not be too large because as the cluster size gets larger the feedback information gets less useful. To see why this is true, note that cluster candidate choice process is based on observation of a single realization of a random variable which is the summation of some other random variables. As the number of these random variables increases and with the length of the code being fixed, the mutual information between the absolute summation of LLRs for one cluster and that of some other cluster increases and feedback fails to provide useful information. It is also worth mentioning that we have used other cluster selecting criteria, such as one which chooses the cluster with minimum absolute LLR code bit, but the proposed criterion far outperforms those ones.

V. Conclusion

In this paper, a new hybrid ARQ coding scheme based on rate compatible LDPC codes is presented. The proposed scheme uses puncturing, extending and feedback information along with a powerful PEG-generated mother code to achieve high rates for a wide range of SNRs. The encoding and decoding complexity remains low as in fixed-rate LDPC codes. The idea of clustering the entire codeword to extract feedback information for the receiver, causes our scheme to outperform the previous work. The proposed method has the advantage of the non-feedback scheme at intermediate SNRs and resolves its throughput drop-off drawback at low SNRs at the same time. In our simulation it results in up to 2 dB gain at low SNRs while getting as close as 0.5 dB to the Shannon limit at intermediate SNRs.

References

[1] S. Lin and P. S. Yu, “A hybrid ARQ scheme with parity retransmission for error control of satellite channels,” *IEEE Trans. Commun.*, vol. 30, no. 7, pp. 1701-1719, Jul. 1982.

[2] J. Hagenauer, “Rate-compatible punctured convolutional codes (RCPC codes) and their applications,” *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389-400, Apr. 1988.

[3] Z. Lin and A. Svensson, “New rate-compatible repetition convolutional codes,” *IEEE Trans. Trans. Inf. Theory*, vol. 46, no. 7, pp. 2651-2659, Nov. 2000.

[4] R. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.

[5] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Trans. Inf. Theory*, vol. 47, no. 2, pp. 599-618, Feb. 2001.

[6] T. J. Richardson, A. M. Shokrollahi and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Trans. Inf. Theory*, vol. 47, no. 2, pp. 619-637, Feb. 2001.

[7] X.-Y. Hu, E. Eleftheriou and D. M. Arnold, “Regular and irregular progressive edge-growth tanner graphs,” *IEEE Trans. Trans. Inf. Theory*, vol. 51, no. 1, pp. 386-398, Jan. 2005.

[8] Y. Kou, S. Lin, and M. Fossorier, “Low-density parity-check codes based on finite geometries: A rediscovery and new results,” *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711-2736, Nov. 2001.

[9] J. Ha, J. Kim, and S. W. McLaughlin, “Rate-compatible puncturing of low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2824-2836, Nov. 2004.

[10] J. Li and K. Narayanan, “Rate-compatible low density parity check codes for capacity approaching ARQ scheme in packet data communications,” in Proc. *Int. Conf. Comm., Internet, and Info. Tech. (CIIT)*, St. Thomas, US Virgin Islands, pp. 376-132, Nov., 2002.

[11] M. R. Yazdani and A.H. Banihashemi, “Irregular rate-compatible LDPC codes for capacity approaching hybrid ARQ schemes,” in Proc. *IEEE Canadian Conf. Elec. Comp. Engg.*, Niagara Falls, Canada, pp. 303-306, May, 2004.

[12] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533-547, Sep. 1981.