Linear-Time Encodable Rate-Compatible Punctured LDPC Codes with Low Error Floors

Seungmoon Song, Daesung Hwang, Sunglock Seo, and Jeongseok Ha
School of Electrical and Computer Engineering
Information and Communications University (ICU)
119, Munjiro, Daejeon, 305-732, Korea
Email: smsong, gohihihigo, bassboy86 and jsha@icu.ac.kr

***Abstract*—We consider Efficiently Encodable Rate-Compatible (E^2RC) punctured Low-Density Parity-Check (LDPC) codes and show that punctured LDPC codes of the E^2RC structure have high error floors at their base code rates. Efficient type-II hybrid Automatic Repeat reQuest (ARQ) protocols are possible with punctured LDPC codes since punctured LDPC codes can support a wide range of code rates with incremental redundancy transmissions. However, such high error floors may result in excessive number of retransmission requests and end up with poor efficiency of hybrid ARQ protocols. We find that the problem of the E^2RC structure stems from dispersive right degree distribution and high maximum right degree which is an indispensable element in the E^2RC structure. In this paper, we propose a modification to E^2RC structure which eliminates the error-floor problem without compromising any of important advantages of the E^2RC structure such as linear-time encodability and good bit-error rate (BER) performance over a wide range of code rates. Our claims will be verified with BER and frame error rate (FER) simulation results.**

I. Introduction

Over time-varying channels, error correction coding systems are required to be flexible with respect to their code rates based on channel state information (CSI). That is, to maximize data rates, channel coding systems decrease their code rates until they overcome the channel impairment, often referred to as *rate adaptability* [1]. Rate adaptability can be realized with several pairs of encoders and decoders for the desired code rates. However, systems with multiple pairs of encoder and decoder are sometimes unacceptable due to their complexity. We can also realize the rate-adaptability by puncturing a low rate channel code (*mother code* or *base code*). Punctured codes need only one encoder and decoder for each mother code. It is assumed that the decoder knows the locations of punctured coded symbols. If a rate-adaptive channel code has another restriction such that the parity bits of higher rate codes in a sequence of rates are embedded in those of lower rate codes, the sequence of codes are called *rate-compatible* [1]–[3]. It is an equivalent way to define the rate-compatibility with punctured codes that punctured parity bits in a lower rate code must be punctured in higher rate codes.

Rate-compatible punctured channel codes have another advantage with type-I and II hybrid Automatic Repeat reQuest (ARQ) protocols. Here, a transmitter sends partial redundancies by puncturing redundancies in a mother code based on CSI. If the receiver fails to recover the message, the receiver requests additional redundancies which are punctured and not transmitted. In the receiver, the additional redundancies can be efficiently combined with the previously received codeword.

The coding gain of a punctured code at each code rate is usually poorer than that of the corresponding dedicated code for the code rate, i.e. optimized code for the rate. Thus, punctured bits must be chosen to minimize performance gap between dedicated and punctured codes. A selection of punctured symbols is called a *puncturing distribution* for a code rate in this paper.

Rate-compatible punctured convolutional codes were introduced in [1]. Ha *et al.* showed that puncturing can be successfully applied to LDPC codes by showing that it is possible to design capacity-approaching rate-compatible punctured LDPC (RCP-LDPC) codes at very long block lengths [2], [4]. Soon afterward puncturing methods for RCP-LDPC codes at short block lengths were presented in [5], [6]. These results served as a foundation for the work in [7]–[10], where the focus was on improving the performance of codes at high rates and designing codes with linear-time encoders.

Especially, Kim *et al.* [9] proposed a class of efficiently encodable rate-compatible (E^2RC) LDPC codes whose parity-check matrices have a favorable structure for puncturing and linear-time encoding. They showed that the E^2RC LDPC codes have superior performance to other punctured LDPC codes [5], [8]. In the E^2RC LDPC codes, the parity bits are assumed to be only degree-2 variable nodes, and the sub-matrix of the degree 2 nodes is designed in a deterministic way. In such a way, all parity bits are encoded in a recursive way which can be implemented with shift registers. In addition to the simple encoder structure, the puncturing locations and order are easily identified by investigating the structure of the sub-matrix of parity bits.

Although the E^2RC LDPC codes have good performance over a wide range of code rates and the advantage of linear-time encodability, they have relatively high error floors at their base code rates. In this paper, we explain why the E^2RC LDPC codes have such high error floors and propose a modified E^2RC LDPC codes to solve the error floor problem without compromising any of important advantages of the E^2RC LDPC codes such as the linear-time encodability and good performance over a wide range of code rates.

This paper is organized as follows. In Section II, we briefly

978-1-4244-1645-5/08/$25.00

describe the puncturing algorithm in [2] which is necessary to understand the E^{2}RC LDPC codes summarized in Section III. In this section we also explain why the high error floors are unavoidable in the E^{2}RC LDPC codes. As a solution to the problem, a modified structure will be proposed in Section IV. The performance of the proposed LDPC codes is compared with that of E^{2}RC LDPC codes in Section V, and we finally conclude our work in Section VI.

II. Puncturing Algorithm for Short LDPC Codes

In this section we summarize the design rule for punctured LDPC codes at short block lengths in [6]. It is based on the level of recoverability of the punctured nodes. In general, a punctured node will be recovered with a reliable message when it has 1) more neighboring (check) nodes, and 2) each of the check nodes has reliable neighbors (variable nodes), not counting the punctured node in question. To clarify this idea further, imagine a punctured variable node. In the first iteration that node will receive nonzero messages from all neighboring check nodes whose other neighboring variable nodes are all unpunctured. This event, when a punctured node receives at least one non-zero message from its neighboring check nodes, is called recovery by analogy with the one over binary-erasure channels. The punctured node in the preceding example will be called one-step-recoverable (1-SR) since it is recovered in the first iteration. The recovered 1-SR nodes and unpunctured nodes will help recover some of the remaining punctured nodes in the second iteration, and so on. In general, the punctured nodes recovered in the k-th iteration are called k-SR nodes. An example of a 3-SR node is depicted in Fig. 1. According to the results in [6] it can be assumed that the more iterations a punctured node needs for its recovery, the less statistically reliable the recovery message is. Thus, it is preferable to puncture nodes such that they will be recovered as fast as possible, which results not only in less required iterations for decoding but also in a better performance at a given code rate. Based on this, the design rule can be defined as follows. First, the group containing 1-SR nodes, denoted by $\mathbf{G}_1$, should be maximized. Subsequently, the group with 2-SR nodes $\mathbf{G}_2$ should be maximized, and so on. The groups $\mathbf{G}_1, \mathbf{G}_2 \ldots, \mathbf{G}_k$ are used to determine the puncturing patterns for various rates in a rate-compatible fashion. Namely, the variable nodes from $\mathbf{G}_1$ are punctured first, followed by the nodes from $\mathbf{G}_2$, etc. In this manner a set of rate-compatible punctured code can easily be derived from a given mother code.

III. E^{2}RC LDPC Codes

In the E^{2}RC LDPC codes, a parity-check matrix consists of two sub-matrices which are denoted as $\mathbf{H}_p$ and $\mathbf{H}_s$ in (1).

$$\mathbf{H} = \begin{bmatrix}\mathbf{H}_p & \mathbf{H}_s\end{bmatrix}_{m\times n} \tag{1}$$

where m and n are the number of rows and columns, respectively. $\mathbf{H}_p$ is an $m \times m$ sub-matrix and whose columns correspond to parity-bits. The $m \times (n-m)$ sub-matrix $\mathbf{H}_s$ is for message bits. We consider only the case that m is a power of 2 (2^d, for $d > 0$) due to limited space. However, the choice of m does not change the structural features of E^{2}RC LDPC. The interested reader is referred to [9], [10] for details of general case. The sub-matrix $\mathbf{H}_p$ is constructed in a recursive way in (2), and the recursion ends when $m/2^{\ell^*}$ becomes 1. That is, $\ell^* = d$ since $m = 2^d$.

$$\mathbf{H}_p = \mathbf{T}^{(1)} \quad \text{and} \quad \mathbf{T}^{(\ell)} = \begin{bmatrix}\mathbf{I}_{\frac{m}{2^\ell}} & \mathbf{0}_{\frac{m}{2^\ell}} \\ \mathbf{I}_{\frac{m}{2^\ell}} & \mathbf{T}^{(\ell+1)}\end{bmatrix}, \tag{2}$$

where $\mathbf{I}_k$ and $\mathbf{0}_k$ are the $k \times k$ identity and zero matrices, respectively, and $\mathbf{I}_1$ and $\mathbf{0}_1$ are defined as 1 and 0, respectively.

In the last recursion, $\ell = d$, it terminates as

$$\mathbf{T}^{(d)} = \begin{bmatrix}1 & 0 \\ 1 & 1\end{bmatrix}.$$

The sub-matrix $\mathbf{H}_p$ is designed to make the columns in the upper left identity matrix $\mathbf{I}_{\frac{m}{2^\ell}}$ of $\mathbf{T}^{(\ell)}$ are all ℓ-SR nodes. Thus, $\mathbf{G}_1$ has the column indexes from 1 to $m/2$, and in general $\mathbf{G}_\ell$ has the column indexes from $\sum_{j=1}^{\ell-1} m/2^j + 1$ to $\sum_{j=1}^{\ell} m/2^j$ which implies the size of ℓ-SR node group, $|\mathbf{G}_\ell|$ to be $m/2^\ell$. The structure of the E^{2}RC LDPC codes also makes the order of puncturing simpler, which is to puncture lower indexed columns first since a node with a lower index is a lower-SR node.

For the encoding, the parity bits on $\mathbf{I}_{m/2}$ in $\mathbf{T}^{(1)}$ are determined only with the message bits. Afterward, the parity bits on $\mathbf{I}_{m/2^\ell}$ will be determined by the message bits and all previously determined parity bits on $\mathbf{I}_{m/2} \sim \mathbf{I}_{m/2^{\ell-1}}$. Due to the sparseness of the parity-check matrix, the encoding needs linear complexity. It is possible to implement the encoder with shift registers, which is well described in [10].

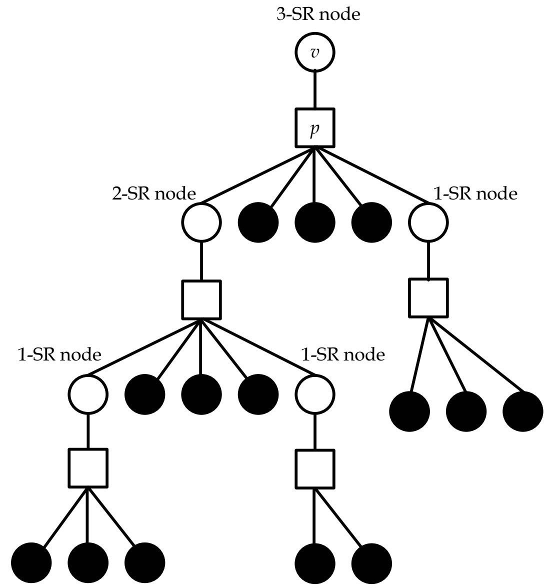

Fig. 1. A recovery tree of a 3-SR node; the squares are survived check nodes, the filled circles are unpunctured variable nodes, and the unfilled circles are punctured variable nodes.

Although the columns of $\mathbf{H}_p$ have only degree of 2 except that the degree of the last column of $\mathbf{H}_p$ is 1 the right degree distribution is dispersive, which is

$$\rho(x) = \sum_{i=1}^{d+1} \rho_i x^{i-1}, \tag{3}$$

where

$$\rho_i = \frac{\frac{2i}{m^j}}{2m-1} \quad \text{for} \quad 1 \le i \le d, \quad \rho_{d+1} = \frac{d+1}{2m-1} \quad \text{and}$$
$$d = \log_2 m.$$

When constructing the sub-matrix $\mathbf{H}_s$ of an E^{2}RC LDPC code, the authors in [9], [10] use the progressive edge growth (PEG) algorithm [11] which needs only a left degree distribution and automatically generates a right degree distribution for maximizing girth distribution of the resulting parity-check matrix. Such automatically generated right degree distributions are usually highly concentrated which empirically substantiates the well-known belief [12], [13] that good LDPC codes have right degree distributions in the concentrated form. However, when we use the PEG algorithm to design an E^{2}RC LDPC code, the right degree is lower bounded by the structure of $\mathbf{H}_p$ in (3) since the degree of each row of $\mathbf{H}$ must be larger than that of $\mathbf{H}_p$. Thus, the PEG algorithm generates a parity-check matrix with a very dispersive right degree distribution. In [9], the authors design an E^{2}RC LDPC code with a left degree distribution

$$\lambda(x) = 0.30780x + 0.27287x^2 + 0.41933x^6 \tag{4}$$

which shows good threshold performance at its designed code rate, 0.5 with the right degree distribution in the concentrated form:

$$\rho(x) = 0.4x^5 + 0.6x^6.$$

However, in the case of E^{2}RC LDPC code, the right degree distribution becomes very dispersive due to the structure of $\mathbf{H}_p$. When the code length n is 1024 and the number of parities m is 512, the right degree distribution in [9] is

$$\begin{aligned}\rho(x) =& 0.41147x^5 + 0.54626x^6 + 0.01892x^7 + 0.01064x^8 + \\ & 0.00592x^9 + 0.00325x^{10} + 0.00354x^{11}.\end{aligned}$$

Moreover, the maximum right degree grows with increasing number of rows as shown in (3), which means the right degree distribution will be more dispersive with larger block lengths or lower rates. The dispersive right degree distributions in the E^{2}RC structure result in high error floors in BER evaluations at their base code rates. If the error floors are due to decoding failures, it is less serious than decoder errors since the decoder can request retransmissions. However, we find that decoder errors significantly contribute to the error floors of the E^{2}RC LDPC codes, which makes hybrid AQR protocols useless due to the undetected errors.

In Section IV, we propose a modified E^{2}RC structure which has concentrated right degree distributions without losing any of important advantages of the E^{2}RC structure.

IV. Proposed LDPC Codes

To address the dispersive right degree distributions in the E^{2}RC structure, we propose a modified E^{2}RC structure which includes the E^{2}RC structure as a special case. In the proposed structure, the sub-matrix $\mathbf{H}_p$ is constructed as shown in (5).

$$\mathbf{H}_p = \begin{bmatrix} \mathbf{E}_{b,1}^{(1)} & \mathbf{0}_b & \cdots & \cdots & \mathbf{0}_b \\ \mathbf{C}_b & \mathbf{E}_{b,2}^{(1)} & \ddots & \cdots & \mathbf{0}_b \\ \mathbf{0}_b & \mathbf{C}_b & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0}_b \\ \mathbf{0}_b & \cdots & \mathbf{0}_b & \mathbf{C}_b & \mathbf{E}_{b,m_b}^{(1)} \end{bmatrix}, \tag{5}$$

$$\mathbf{E}_{b,j}^{(\ell)} = \begin{bmatrix} \mathbf{I}_{\frac{b}{2^\ell}} & \mathbf{0}_{\frac{b}{2^\ell}} \\ \mathbf{I}_{\frac{b}{2^\ell}} & \mathbf{E}_{b,j}^{(\ell+1)} \end{bmatrix}, \ \mathbf{E}_{b,j}^{(\delta)} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},$$

$$\text{for} \quad 1 \le j \le m_b, \ \text{and}$$

$$\mathbf{C}_b = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \ddots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix},$$

where $m = 2^d$, $b = 2^\delta$, $0 < \delta < d$, $m_b = m/b = 2^{d-\delta}$, $\mathbf{E}_{b,j}^{(1)}$, $\mathbf{C}_b$ are $b \times b$ rectangular matrices, and $\mathbf{E}_{b,j}^{(1)}$ is recursively constructed with the termination $\mathbf{E}_{b,j}^{(\delta)}$. If $\delta = d$, $\mathbf{E}_{m,1}^{(1)}$ will be $\mathbf{H}_p$, and the proposed structure becomes the same as the E^{2}RC structure. In each $\mathbf{E}_{b,j}^{(\ell)}$, the columns of the upper left identity matrix $\mathbf{I}_{b/2^\ell}$ are all ℓ-SR nodes.

The right degree distribution of the sub-matrix $\mathbf{H}_p$ in (5) from the node perspective is derived as in (6).

$$\begin{aligned}\tilde{\rho}_i &= \frac{1}{2_i} + \left(-\frac{m_b}{m}\right)^i \quad \text{for} \quad i = 1, 2, \\ \tilde{\rho}_i &= \frac{1}{2^i} \quad \text{for} \quad 3 \le i \le \delta, \quad \text{and} \quad \tilde{\rho}_{\delta+1} = \frac{m_b}{m} = 2^{-\delta}.\end{aligned} \tag{6}$$

The right degree distribution from the edge perspective, ρ_i has a relation with $\tilde{\rho}_i$ as

$$\rho_i = \frac{mi}{2m-1}\tilde{\rho}_i.$$

In the E2RC structure, the maximum right degree is directly proportional to the number of rows m but in the proposed structure, we can control the maximum right degree by choosing δ. Thus, if we have a small value of δ, the PEG algorithm will generate the sub-matrix $\mathbf{H}_s$ to make a concentrated right degree distribution of $\mathbf{H}$.

The order of puncturing was the column index in the ascending order in the E^{2}RC LDPC codes. However, in the proposed structure, the column index p_j of the j-th punctured parity bit is computed as

$$p_j = r \times b + q + 1, \ \text{for} \ 1 \le j \le m - m_b, \tag{7}$$

where $r = (j-1) \mod m_b$, and $q = \lfloor (j-1)/m_b \rfloor$. Although the equation of the puncturing order seems to be involved, it simply takes one lower indexed ℓ-SR node from $\mathbf{E}_{b,((j \mod m_b)+1)}$ and take the next ℓ-SR node from $\mathbf{E}_{b,(((j+1) \mod m_b)+1)}$ and so on.

We have to note that in (7), the maximum number of punctured symbols is limited up to $m - m_b$ since we do not punctured the last column in $\mathbf{E}_{b,j}$ for all j, which limits the highest code rate to

$$r_{\text{highest}} = \frac{k}{n-m+m_b} = \frac{k}{k+m_b} = \frac{1}{1+\frac{2^{d-\delta}}{k}}. \quad (8)$$

We show that concentrated right degree distribution is possible with a choice of a small δ but due to the limit of the highest code rate in (8), δ cannot be arbitrary small. However, a small loss of the highest rate ($r_{\text{highest}} = 0.9412$ when $k = 512$, $m = 512$, and $\delta = 4$) provides a highly concentrated right degree distribution.

Since all the parity bits in (5) are determined in a recursive way, encoding will be done with a linear complexity as we did in the E^{2}RC structure.

V. Simulation Results

We verify our claim by evaluating and comparing performances of punctured LDPC codes with the E^{2}RC and the proposed structures. For the performance evaluations, we design LDPC codes at length 2048 and rate 0.5 with the degree distribution in (4). For the linear-time encoding, both the E2RC and the proposed structure need one degree 1 parity bit on the last column of $\mathbf{H}_p$, which slightly changes the left degree distribution to

$$\lambda(x) = 0.00015 + 0.30235x + 0.27126x^2 + 0.42624x^6. \quad (9)$$

For the left degree distribution we have a dispersive right degree distribution of the E^{2}RC LDPC code,

$$\rho(x) = 0.390784x^5 + 0.576872x^6 + 0.017723x^7 + 0.006646x^8 + 0.002954x^9 + 0.003249x^{10} + 0.001772x^{11}$$

from the PEG algorithm with the "best effort sense" to maximize local girths. However, the right degree distribution of the proposed LDPC code looks highly concentrated

$$0.355339x^5 + 0.639935x^6 + 0.004726x^7,$$

where we set δ in (5) to 4.

The BER[1] performances of the punctured LDPC codes are evaluated at rates 0.5, 0.6, 0.7, 0.8, and 0.9 and depicted in Fig. 2 where the proposed punctured LDPC codes have the same performances as those of the E^{2}RC LDPC codes. However, the proposed LDPC code shows a much lower error floor at it base code rate 0.5, which is due to the concentrated right degree distribution. Such performance improvement at the base rate becomes more distinctive in the comparisons of the frame error rate (FER) in Fig. 3. Here, FER means the rate of the number of decoding failures and errors to the total number of transmitted codewords.

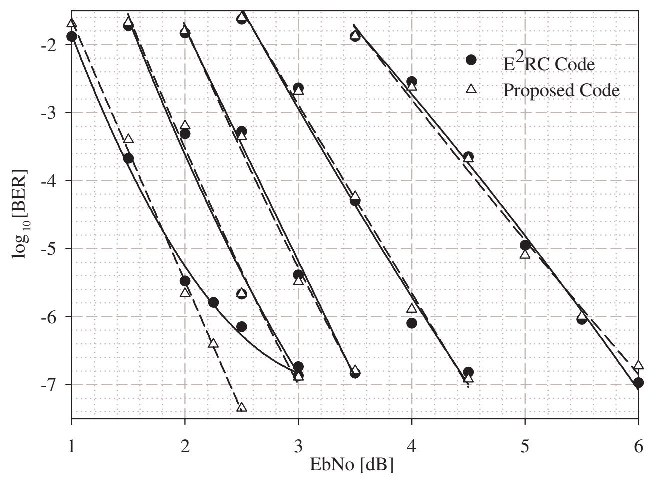

Fig. 2. Comparison between BERs of the punctured LDPC codes at block length 2048 with the E^{2}RC (circles) and the proposed (triangles) structures; the left degree distribution of the base code is in (9) and the code rates are 0.5, 0.6, 0.7, 0.8 and 0.9 from the left to the right.

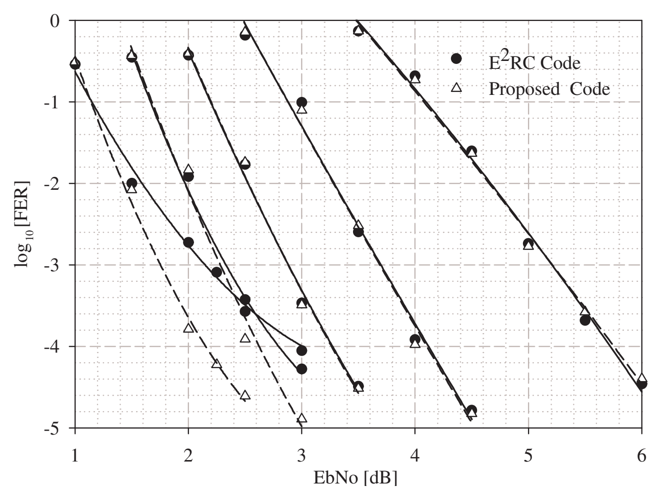

Fig. 3. Comparison between FERs of the punctured LDPC codes at block length 2048 with the E^{2}RC (circles) and the proposed (triangles) structures; the left degree distribution of the base code is (9) and the code rates are 0.5, 0.6, 0.7, 0.8 and 0.9 from the left to the right.

We also verify our claims with different degree distribution pair from [12]:

$$\lambda(x) = 0.30780x + 0.27287x^2 + 0.41933x^6 \text{ and}$$
$$\rho(x) = 0.4x^5 + 0.6x^6,$$

which is for rate 0.5 LDPC codes.

Again, we design LDPC codes at length 2048 and introduce one degree 1 parity bit to make them linear-time encodable, which slightly changes the left degree distribution to

$$\lambda(x) = 0.00015 + 0.30765x + 0.27287x^2 + 0.41933x^6. \quad (10)$$

[1]For each BER, we watched more than 50 unsuccessful decodings.

For the left degree distribution we have two right degree distributions,

$$\rho(x) = 0.441899x^5 + 0.521797x^6 + 0.017854x^7 + 0.012052x^8 + 0.002976x^9 + 0.001637x^{10} + 0.001785x^{11}, \text{ and}$$

$$\rho(x) = 0.400833x^5 + 0.596786x^6 + 0.002381x^7.$$

from the PEG algorithm for the E^2RC and proposed structures, respectively. By changing the random seed in the PEG algorithm, we design 10 different parity-check matrices for each structure and compare their performances at the base code rate, 0.5. The comparisons are depicted in Fig. 4 where the proposed structure consistently outperforms the E^2RC structure, which confirms our claims.

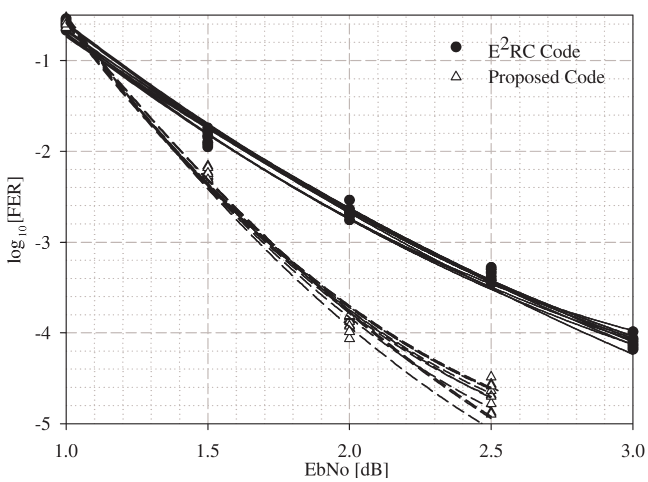

Fig. 4. Comparison between FERs of the 10 LDPC codes at block length 2048 with the E^2RC (circles) and the proposed (triangles) structures; the left degree distribution of the base code is in (10).

Since type-II hybrid ARQ protocol can ask for retransmissions only if it detects errors, error control codes in hybrid ARQ protocols must have good error detecting capability. In other words, the decoding error rate (DER), i.e. the number of decoding errors/the number of transmitted codeword, must be minimized. In Table I, we summarize DERs of the E^2RC and the proposed LDPC codes whose left degree distribution is from (4). In this comparison, the proposed LDPC code shows 10 times lower DERs than those of the E^2RC codes. Thus, the LDPC codes with the proposed structure provide not only better performance in forward error controls but also in the hybrid ARQ protocols.

VI. Conclusions

Although the E^2RC punctured LDPC codes provide good performances over a wide range of code rates, they have high error floors at their base code rates. We show that this is due to the dispersive right degree distributions which come from the structure of the parity-check sub-matrix in the E^2RC LDPC codes. We propose a modified E^2RC structure with which we can have both good performance over the range of code rates and lower error floors. We also investigate the decoder error rate (DER) and show that the proposed LDPC codes have 10 times lower DERs in our experiment.

TABLE I
Decoding error rates of the E^2RC and the proposed LDPC codes at their base code rate, 0.5 with the left degree distribution in (4).

Decoding Error Rates					
E_b/N_0	1.00	1.50	2.00	2.25	2.50
E^2RC	1.15E-2	2.81E-3	7.16E-4	3.74E-4	2.25E-4
Proposed	0	1.67E-4	3.26E-5	2.39E-5	9.41E-6

The work was supported by in part by Samsung Electronics, Korea.

References

[1] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, pp. 389–400, Apr. 1988.

[2] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. IT-50, pp. 2824– 2836, Nov. 2004.

[3] J. Li and K. Narayanan, "Rate-compatible low density parity check codes for capacity-approaching ARQ scheme in packet data communicaitons," in *Int. Conf. on Comm., Internet and Info. Tech. (CIIT)*, 2002.

[4] J. Ha and S. W. McLaughlin, "Optimal puncturing of irregular low-density parity-check codes," in *IEEE Int. Conf. Commun.*, Anchorage, Alaska, May 2003, pp. 3110–3114.

[5] J. Ha, J. Kim, and S. W. McLaughlin, "Puncturing for finite length low-density parity-check codes," in *Proc. Int. Symp. Information Theory*, Chicago, USA, 2004, p. 151.

[6] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, "Rate-compatible punctured low-density parity-check codes with short block lengths," *IEEE Trans. Inform. Theory*, vol. IT-52, pp. 728 – 738, Feb. 2006.

[7] E. Choi, S.-B. Suh, and J. Kim, "Rate-compatible puncturing for low-density parity-check codes with dual-diagonal parity structure," in *16th Int. Symp. on Personal Indoor and Mobile Radio Commun. (PIMRC)*, Berlin, Germany, 2005.

[8] G. Yue, X. Wang, and M. Madihian, "Design of rate-compatible irregular repeat accumulate codes," *IEEE Trans. Commun.*, vol. 55, no. 6, pp. 1153 – 1163, June 2007.

[9] J. Kim, A. Ramamoorthy, and S. W. McLaughlin, "Design of efficiently-encodable rate-compatible irregular LDPC codes," in *IEEE Int. Conf. Commun.*, Istanbul, Turkey, June 2006, pp. 1131–1136.

[10] ——, "The design ofefficiently-encodable rate-compatible ldpc codes," *IEEE Trans. Commun.*, submitted.

[11] X. Hu, E. Eleftheriou, and D. M. Arnold, "Progressive edge-growth Tanner graphs," in *Proc. IEEE GLOBECOM*, San Antonio, Texas, Nov. 2001, pp. 995–1001.

[12] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. IT-47, pp. 657–670, Feb. 2001.

[13] T. J. Richardson, A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. IT-47, pp. 619–637, Feb. 2001.