# Transactions Letters

## Unequal Error Protection Using Partially Regular LDPC Codes

Nazanin Rahnavard, *Member, IEEE*, Hossein Pishro-Nik, *Member, IEEE*, and Faramarz Fekri, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a scheme to construct low-density parity-check (LDPC) codes that are suitable for unequal error protection (UEP). We derive density evolution (DE) formulas for the proposed unequal error protecting LDPC ensembles over the binary erasure channel (BEC). Using the DE formulas, we optimize the codes. For the finite-length cases, we compare our codes with some other LDPC codes, the time-sharing method, and a previous work on UEP using LDPC codes. Simulation results indicate the superiority of the proposed design methodology for UEP.**

***Index Terms*—Low-density parity-check (LDPC) codes, unequal density evolution (UDE), unequal error protection (UEP).**

### I. Introduction

MOST error-correcting codes are designed for the equal error protection (EEP) of all data. However, on several important applications, certain parts of the information may need a higher level of protection against error than other parts. For example, in optical networks, errors that occur in the header bits of a packet cause more serious damage to the subsequent process than errors within the payload. Since repeat-request protocols are not an option for error-free data delivery in optical networks (because of their multihop nature and high-speed data transmission requirement), we are left with three options. First, EEP codes with high protection for the entire packet could be used. This is not efficient, since EEP codes provide far more protection than is necessary by adding excessive redundancy. Second, two different codes could be used (time-sharing method). This approach is not prudent, since the header is very short and the performance of codes is poor for short lengths. Finally, a more interesting and challenging solution is the construction of a single code that induces a selective protection property known as unequal error protection (UEP).

Masnick and Wolf were the first to introduce UEP codes [1]. Later, other UEP codes were designed using different approaches, e.g., [2] and [3]. However, there has been little work concerning the design of low-density parity-check (LDPC) codes with the UEP property. In [4], the authors proposed UEP-LDPC codes that are based on cyclic difference families (CDFs). In [5], we investigated the notion of UEP-LDPC codes based on irregular random graphs with the degree distributions that are optimized by unequal density evolution (UDE) formulas over the binary erasure channel (BEC). In this letter, we extend the discussion.

Throughout the letter, we assume the following terminology. An LDPC ensemble can be defined to be the set of bipartite Tanner graphs with degree distribution pairs $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_i \rho_i x^{i-1}$, where $\lambda_i$ ($\rho_i$) is the fraction of edges connected to variable (check) nodes of degree $i$. The codes reported in the letter are randomly constructed following the descriptions of [6] and [7].

The letter is organized as follows. In Section II, a design method for UEP using LDPC codes is explained, and the simulation results are given. Efficient encoding is also explored for the proposed codes. Sections III and IV compare our method with time-sharing and CDF methods, respectively. Finally, we conclude the paper in Section V.

### II. UEP Using LDPC Codes

Up until now, different schemes for designing capacity achieving (CA) LDPC codes over the BEC have been devised, e.g., [8]. These schemes are based on designing codes of rate $R$ with the threshold channel-erasure probability ($\varepsilon^{th}$) as close as possible to $1 - R$. When the channel-erasure probability is less than $\varepsilon^{th}$, the average bit-error rate (BER) (the probability that a bit is not recovered after the decoding stops) goes to zero when long enough code lengths and a large enough number of decoding iterations are considered. Therefore, CA codes are superior to the UEP codes asymptotically, as they provide small enough error rates for all data. However, short-to-moderate-length codes are preferable in practice. For these lengths, UEP codes are desirable. In the proposed UEP design, we neither optimize the codes based on $\varepsilon^{th}$ nor use the average BER of all data in our analysis. Instead, we divide the codeword into different groups, and investigate the average BER for each group. The codes are optimized such that some information bits have lower BER than the other bits.

Throughout the letter, we are only concerned with the performance of information bits, thus UEP for information bits is considered. Therefore, we need to determine the positions of the information bits in the codeword. For an $(n, k)$ LDPC code that is defined by a parity-check matrix $H$, not every arbitrary

Paper approved by A. H. Banihashemi, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received August 30, 2004; revised August 7, 2005; June 5, 2006; and July 17, 2006. This work was supported in part by the National Science Foundation under Grant CCF-0430964. This paper was presented in part at the IEEE International Symposium on Information Theory, Chicago, IL, June 27–July 2, 2004.

N. Rahnavard and F. Fekri are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: nazanin@ece.gatech.edu; fekri@ece.gatech.edu).

H. Pishro-Nik was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA. He is now with the Department of Electrical and Computer Engineering, University of Massachusetts–Amherst, Amherst, MA 01003 USA (e-mail: pishro@ecs.umass.edu).

Digital Object Identifier 10.1109/TCOMM.2007.892436

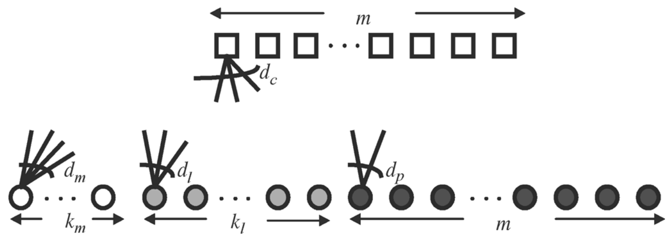

Fig. 1. Tanner graph of the proposed ensemble for the UEP property.

collection of $k$ bits in the codeword can correspond to the information bits. The following should be satisfied by $H$.

*Fact 1:* Let an $(n-k)\times n$ matrix $H=[h_1,h_2,\ldots,h_n]$ be the parity-check matrix corresponding to an $(n,k)$ linear code. To have $[i_1,i_2,\ldots,i_k]$ as the positions of the information bits in the codeword, matrix $H_P=H\setminus H_I$ must be full rank, where $H_I=[h_{i_1},h_{i_2},\ldots,h_{i_k}]$.

*Proof:* Let us define $X=(x_1,x_2,\ldots,x_n)$, $X_I=(x_{i_1},\ldots,x_{i_k})$, and $X_P=X\setminus X_I$. Then, $X$ is a valid codeword if and only if $H_IX_I^T+H_PX_P^T=0^T$. Using this equation, we can find parity bits (PB) $X_P$ as a function of information bits $X_I$ if and only if $H_P$ is full rank. ■

In this letter, the information bits are divided into two groups with two levels of importance. One group consists of the more important bits (MIB) that need higher protection. The other group contains the less important bits (LIB). Next, we describe how to design UEP-LDPC codes. For simplicity, we use asymptotic tools for designing the UEP codes. However, the results are used in constructing finite-length codes.

### A. Problem Statement

Suppose we want to transmit $k$ information bits with two levels of importance over an erasure channel with erasure probability $\varepsilon$. For this, we want to design an $(n,k)$ UEP code having rate $R$. Let $k_m=\alpha k$ (where $0<\alpha<1$) be the number of MIB and $k_l=(1-\alpha)k$ be the number of LIB. Let $m=n-k$ be the number of PB.

Next, a method for the UEP is proposed. In this method, we consider the conventional bipartite Tanner graph with $n$ variable nodes and $m$ check nodes. For the simplicity of design, we assume having partially regular ensembles. By partially regular, we mean that all the MIB, LIB, and PB have the same degrees $d_m$, $d_l$, and $d_p$, respectively. Further, all check nodes have the same degree $d_c$. Fig. 1 shows the Tanner graph of this ensemble. Let $H=[H_m|H_l|H_p]$ denote the corresponding parity-check matrix of this graph, where $H_m$, $H_l$, and $H_p$ are submatrices that correspond to the MIB, LIB, and PB, respectively. By *Fact 1*, we conclude that the assumption of separating information bits and PB as specified above is valid if and only if $H_p$ is full rank. Next, we derive DE formulas for the proposed partially regular ensemble.

### B. UEP DE

Let us consider the standard iterative decoding algorithm for the BEC. To achieve UEP with a significant gap among the different protection levels, we modify the DE formulas introduced in [9]. In our formulation, three parameters $m_i$, $l_i$, and $p_i$ are introduced. These parameters denote the expected fractions of the erasure messages at the $i$th iteration that are passed from the variable nodes that correspond to the MIB, LIB, and PB, respectively. Furthermore, let $q_i$ denote the probability that an erasure message is passed from the check nodes to the variable nodes at the $i$th iteration. Then, the UDE formulas are given as

$$m_0=l_0=p_0=\varepsilon \tag{1}$$
$$m_{i+1}=m_0q_i^{d_m-1},\ \ l_{i+1}=l_0q_i^{d_l-1},\ \ p_{i+1}=p_0q_i^{d_p-1} \tag{2}$$
$$q_i=1-\left(1-\lambda_{d_m}m_i-\lambda_{d_l}l_i-\lambda_{d_p}p_i\right)^{d_c-1} \tag{3}$$

where $\lambda_{d_m},\lambda_{d_l}$, and $\lambda_{d_p}$ are the fractions of the edges that are connected to the MIB, LIB, and PB, respectively. These parameters can be obtained by $\lambda_{d_m}=\alpha Rd_m/E,\lambda_{d_l}=(1-\alpha)Rd_l/E$, and $\lambda_{d_p}=(1-R)d_p/E$, in which $E=\alpha Rd_m+(1-\alpha)Rd_l+(1-R)d_p$.

The following lemma points out the UEP property of the proposed ensemble.

*Lemma 1:* Let $\varepsilon$ be the erasure probability of a BEC and $\beta_{i,\varepsilon}\triangleq(l_{i+1,\varepsilon}/m_{i+1,\varepsilon})$ be the UEP gain at the $i$th decoding iteration. Then, $\beta_{i,\varepsilon}$ increases when the erasure probability of the channel, $\varepsilon$, decreases. Moreover, $\beta_{i,\varepsilon}$ is an increasing function of the number of iterations $i$.

*Proof:* Using (2), we have $\beta_{i,\varepsilon}=(1/q_{i,\varepsilon})^{d_m-d_l}$. Since $d_m>d_l$, we need to show that $q_{i,\varepsilon}$ is an increasing function of $\varepsilon$. This can be proven by induction. Assume $\varepsilon_2>\varepsilon_1$. From (3), we have $q_{0,\varepsilon}=1-(1-\varepsilon)^{d_c-1}$. This implies that $q_{0,\varepsilon_2}>q_{0,\varepsilon_1}$. Now we assume that $q_{i-1,\varepsilon_2}>q_{i-1,\varepsilon_1}$. From (2), we have $m_{i,\varepsilon_2}>m_{i,\varepsilon_1}$, $l_{i,\varepsilon_2}>l_{i,\varepsilon_1}$, and $p_{i,\varepsilon_2}>p_{i,\varepsilon_1}$. Using (3), we conclude that $q_{i,\varepsilon_2}>q_{i,\varepsilon_1}$. This proves the first part of the lemma.

To prove the second part, we must show that $q_{i,\varepsilon}$ is a decreasing function of $i$. This can be done by induction on $i$. First, note that $q_{0,\varepsilon}<1$. From (3), we have $q_{1,\varepsilon}<q_{0,\varepsilon}$. Now assume that $q_{i,\varepsilon}<q_{i-1,\varepsilon}<\ldots<q_{1,\varepsilon}<q_{0,\varepsilon}<1$. Let

$$f(q_{i,\varepsilon})\triangleq\varepsilon\lambda_{d_m}q_{i,\varepsilon}^{d_m-1}+\varepsilon\lambda_{d_l}q_{i,\varepsilon}^{d_l-1}+\varepsilon\lambda_{d_p}q_{i,\varepsilon}^{d_p-1}. \tag{4}$$

We have $f(q_{i,\varepsilon})<1$ and

$$\begin{aligned}q_{i+1,\varepsilon}-q_{i,\varepsilon}&=(1-f(q_{i-1,\varepsilon}))^{d_c-1}-(1-f(q_{i,\varepsilon}))^{d_c-1}\\&=(f(q_{i,\varepsilon})-f(q_{i-1,\varepsilon}))\times K\end{aligned}$$

in which $K>0$. Using (4), the value of $f(q_{i,\varepsilon})-f(q_{i-1,\varepsilon})$ can be seen to be negative since $q_{i,\varepsilon}<q_{i-1,\varepsilon}$. Therefore, we have $q_{i+1,\varepsilon}<q_{i,\varepsilon}$. This completes the proof. ■

Using the UDE formulas, the asymptotic behavior of a code for a given degree distribution can be estimated.[1] Moreover, we can optimize the degrees such that we have low error rates for MIB while keeping the overall performance comparable with other codes. For a given $R$ and $\alpha$, we need to find optimal values for $d_m,d_l,d_p$, and $d_c$. However, we have one equality constraint that is imposed by the edges as

$$\alpha Rd_m+(1-\alpha)Rd_l=(1-R)(d_c-d_p). \tag{5}$$

[1]An alternative way to obtain the performance of a code over the BEC is by determining the stopping sets' characteristics. Such an approach is more complicated, especially for the UEP case. However, the results of two approaches will be consistent asymptotically.

TABLE I
DEGREE DISTRIBUTIONS, $m_{25}, l_{25}$, AND $p_{25}$ OF SOME OPTIMIZED UEP-LDPC CODES OF RATE 1/2 AND $\alpha = 0.1$ FOUND BY THE PROPOSED METHOD

| Code | $d_m$ | $d_l$ | $d_p$ | $d_c$ | $m_{25}$ | $l_{25}$ | $p_{25}$ |
|---|---|---|---|---|---|---|---|
| 1 | 23 | 3 | 2 | 7 | 2.18e-6 | 1.48e-1 | 2.58e-1 |
| 2 | 24 | 4 | 2 | 8 | 2.31e-12 | 1.52e-2 | 1.45e-1 |

Therefore, we have three independent variables to optimize. We considered $d_c$ as the dependent variable. By assuming a maximum value for the degrees ($d_{\max}$) and considering $d_m > d_l > d_p$, we can easily search through all the possible values for the degrees and select the ones that result in very low error rates for MIB. The cost function is considered as $m_I$ (for some large integer $I$).

*Example:* Assume we want to design a UEP code with $\alpha = 0.1$ and rate 1/2. By setting $d_{\max} = 25$, $\varepsilon = 0.45$,[2] and $I = 25$, we minimized the cost function $m_I$. Table I shows the degrees, $m_I$, $l_I$, and $p_I$ for two optimized codes.

As it is shown in the table, asymptotically, the performance gaps between the BERs of MIB and the rest of the codeword bits are several orders of magnitude for 25 decoding iterations. Increasing the number of iterations results in even larger gaps.

To measure the performance of the proposed codes for the finite-length case, we found the BER versus $\varepsilon$ for Code 1 ($\varepsilon^{th} = 0.455$) when the length of the code is $n = 4000$ [Fig. 2(a)]. Two other codes were chosen for comparison with our code: the regular (3,6) ($\varepsilon^{th} = 0.429$) and a BEC-optimized irregular code, referred to as Code A, found from [10] by setting the maximum allowable degree to 25. The degree distribution of Code A[3] is given by $\lambda(x) = 0.249765x + 0.247164x^2 + 0.148003x^5 + 0.0033269x^6 + 0.351741x^{19}$ and $\rho(x) = x^7$ with $\varepsilon^{th} = 0.489$. To have a fair comparison, we showed the performance of $k_m = 200$ highest-degree nodes (as MIB) and rest of the nodes (LIB and PB) separately for Code A. As we can see in Fig. 2(a), there is a large gap between the BERs of the MIB and LIB in the proposed code.[4] This gap is at least two orders of magnitude, and it increases when the channel erasure probability decreases as in *Lemma 1* for the asymptotic case. Moreover, the performance of the MIB in the proposed code is always better than the performance of the MIB in the two other codes. In addition, the error floor in the LIB and PB are lower in the proposed code in comparison with Code A. We also note that the performance of the proposed code is far better than the performance of the regular (3,6) for $\varepsilon > 0.3921$. For smaller $\varepsilon' s$, the performance of the regular (3,6) beats the performance of LIB in the proposed method. This is because of the well-known result that the regular (3,6) does not show an error floor, unlike the irregular codes.

[2]If we optimize a code for a large value of $\varepsilon$, asymptotically, the code will have a good performance for large $\varepsilon' s$. On the other hand, if we optimize a code for a small value of $\varepsilon$, asymptotically, the code will have a good performance in the error-floor region.

[3]We need to make a subtle change to the distribution of finite-length codes. For example, we used $\lambda(x) = 0.249625x + 0.2475x^2 + 0.148125x^5 + 0.0035x^6 + 0.35125x^{19}$ and $\lambda(x) = 0.249x + 0.2475x^2 + 0.15x^5 + 0.0035x^6 + 0.35x^{19}$, for $n = 4000$ and $n = 1000$, respectively. In both cases, $\varepsilon^{th} = 0.489$.

[4]The BER for MIB is found by averaging over the fraction of the bits in MIB that has not been recovered when decoding stops. Similarly, BERs of LIB and PB can be obtained.

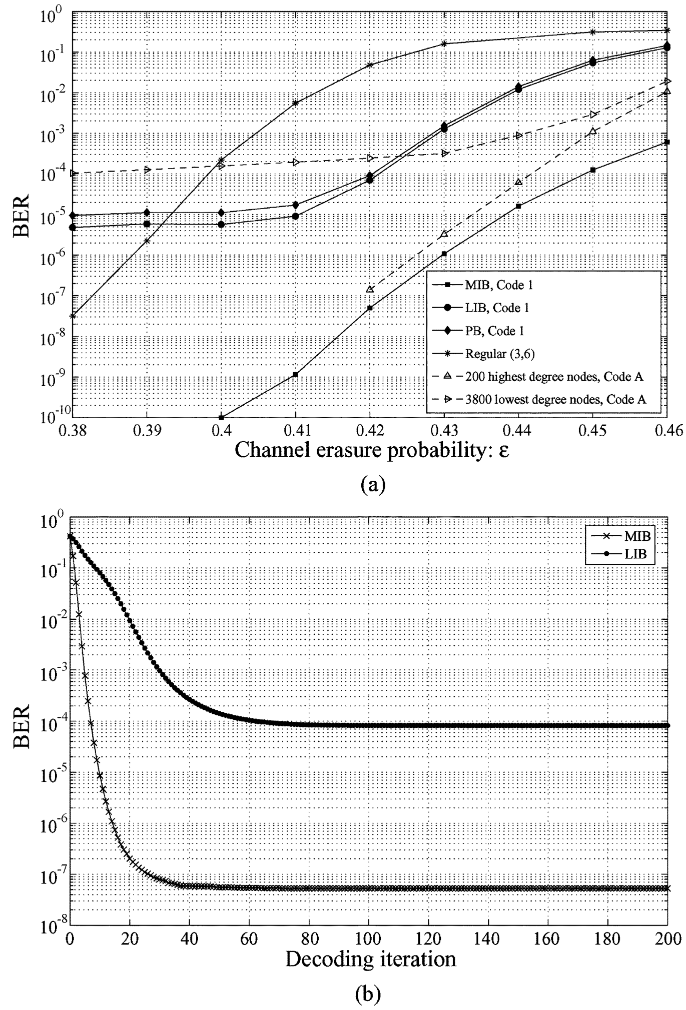


Fig. 2. (a) Comparison of the BERs of Code 1 with Code A and the regular (3,6). All codes are of length 4000 and rate 1/2. (b) Recovery convergence rate of MIB and LIB Code 1 at $\varepsilon = 0.42$.

However, we note that the performance of MIB in the proposed code is superior to the performance of the regular (3,6).

It is worth noting that not only will MIB be retrieved with much less error than LIB, but also MIB converges in fewer decoding iterations than LIB. This can be seen in Fig. 2(b) for Code 1 at $\varepsilon = 0.42$. This is useful when fast decoding for MIB is needed.

We also illustrated the performance of Code 1 when $n = 1000$ (Fig. 3). Again, we compared the proposed code with the regular (3,6) and Code A of lengths 1000. As we can see, the proposed code is superior to the regular (3,6) in the shown range. Moreover, although the performances of LIB and PB are close in Code 1 and Code A, the performance of MIB is far better than the performance of the 50 highest-degree nodes in Code A.

### C. Efficient Encoding

As we can see in Table I, the degree-distribution optimization has resulted in $d_p = 2$. We also observed the same result for most of the other UEP code designs. In fact, we exploit this property of the parity nodes to simplify the encoding of the proposed codes as follows. Since $d_p = 2$, all columns of $H_p$ have weight two. However, given that $H_p$ is $m \times m$ and full rank by *Fact 1*,

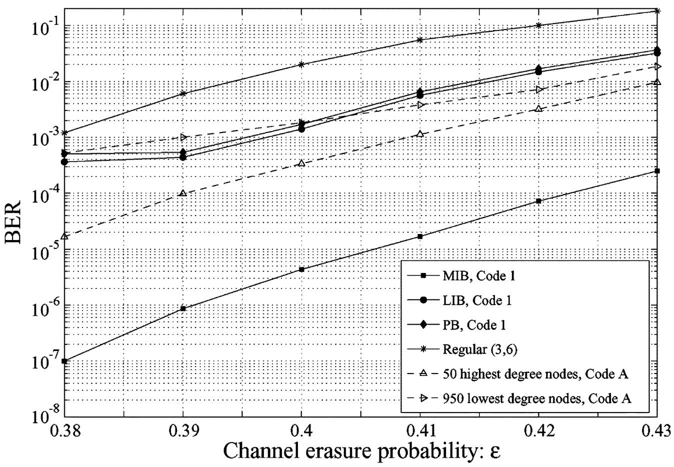


Fig. 3. Comparison of the BERs of Code 1 with Code A and the regular (3,6). All codes are of length 1000 and rate 1/2.

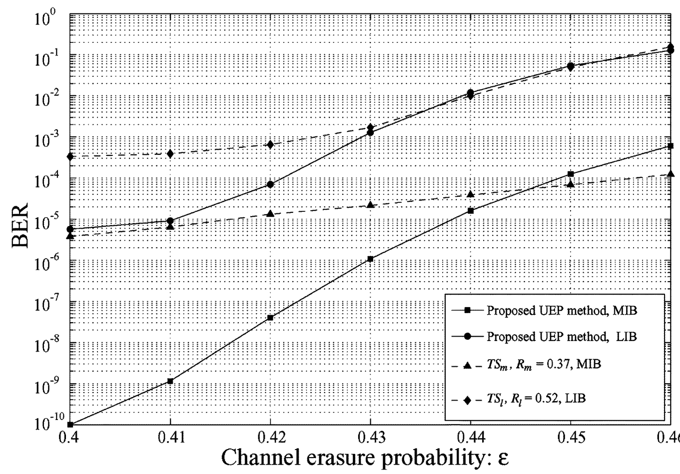


Fig. 5. Comparison of the proposed UEP method with the time-sharing method.

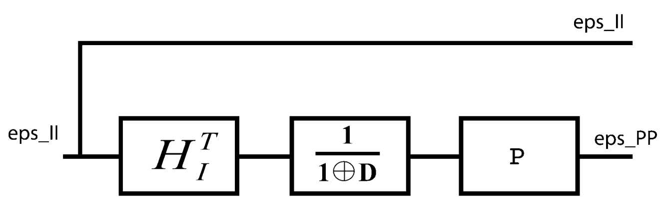


Fig. 4. Efficient encoding for the proposed UEP codes having $d_p = 2$.

no more than $m-1$ columns of weight two are allowed. To overcome this problem, we use the method proposed in [11]. One of the weight-two columns is replaced with a weight-one column. This does not have an important effect on the performance of the code, but ensures that $H_p$ is full rank. It can be shown that $H_p$ is either a dual-diagonal matrix $Q$ or a column permutation of $Q$ [11]. In other words, $H_p = Q\Pi$, where $\Pi$ is a random permutation matrix. A systematic generator matrix for the parity-check matrix $H = [H_m|H_l|Q\Pi]$ is given by $G = [I|H_I^T Q^{-T}\Pi]$, in which $H_I = [H_m|H_l]$. The matrix $Q^{-T}$ corresponds to a differential encoder whose transfer function is $1/(1 \bigoplus D)$ [11]. The encoder for these codes is depicted in Fig. 4. Thus, these codes are a generalized form of the repeat-accumulate (RA) codes for which $\Pi$ is equal to the identity matrix.

## III. Comparison With the Time-Sharing Method

One approach to UEP is the time-sharing method. In this method, several codes of different rates are used for different parts of the data. This method increases the complexity of the system. Additionally, since the MIB is usually very short, the code length would be short. We expect that this causes performance degradation. The following simulation confirms that the time-sharing technique does not perform as well as the proposed method. Suppose we want two levels of protection for a message whose $\alpha$ fraction is MIB. In the first method, a UEP code of rate $R$ is considered. Alternatively, we can design two codes $TS_m$ and $TS_l$ with rates $R_m$ and $R_l$ for MIB and LIB, respectively. By fixing the total number of the PB in both methods, we get $(\alpha/R_m) + ((1-\alpha)/R_l) = 1/R$. For a given $R$ and $\alpha$, we can have different pairs of $R_m$ and $R_l$, where choosing the best pair can be done by trial and error. Fig. 5 compares the performance of UEP Code 1 of length 4000, $R = 0.5$, and $\alpha = 0.1$ with the time-sharing method having $R_l = 0.52$ and $R_m = 0.37$. The codes that are used in the time-sharing method are the best codes that we find from [10] for the given rates. They have the following degree distributions:

$$
\begin{aligned}
\lambda_{TS_l}(x) &= 0.299488x + 0.144426x^2 + 0.0159922x^3 \\
&\quad + 0.165696x^4 + 0.175288x^9 \\
&\quad + 0.134321x^{24} + 0.0647887x^{25} \\
\rho_{TS_l}(x) &= x^7 \\
\lambda_{TS_m}(x) &= 0.249839x + 0.128926x^2 + 0.113474x^4 \\
&\quad + 0.0514865x^5 + 0.0767592x^{10} \\
&\quad + 0.0636197x^{11} + 0.142641x^{27} + 0.174203x^{99} \\
\rho_{TS_m}(x) &= 0.7x^6 + 0.3x^7
\end{aligned}
$$

with $\varepsilon_l^{th} = 0.474$ and $\varepsilon_m^{th} = 0.628$, respectively. To have better performance for MIB and LIB in the time-sharing method, we assume that MIB and LIB correspond to the higher degree variable nodes in $TS_m$ and $TS_l$, respectively. Fig. 5 indicates that the proposed UEP scheme outperforms the time-sharing scheme for $\varepsilon < 0.446$. For example, at $\varepsilon = 0.42$, the MIB (LIB) in Code 1 has more than two orders of magnitude (about one order of magnitude) less BER in comparison with the case where $TS_m(TS_l)$ is used. Further, the superiority of the proposed method versus time-sharing increases when the channel erasure probability decreases.

## IV. Comparison With the Previous UEP-LDPC Codes

In [4], authors proposed UEP-LDPC codes constructed based on the orbits of CDFs. We note that the codes have very high protection for some codeword bits. This approach is desirable in applications such as holographic memory systems, where the noise has a nonuniform pattern. Therefore, different protection levels for *codeword bits* are provided to achieve uniform BER after the decoding. In applications where UEP for *information bits* is needed, this approach may not be efficient. Specifically, it can be shown that the most highly protected codeword bits in [4]

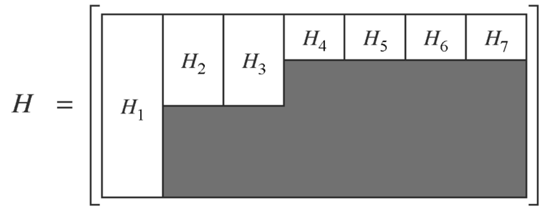


Fig. 6. Structure of the parity-check matrix constructed using the CDF method.

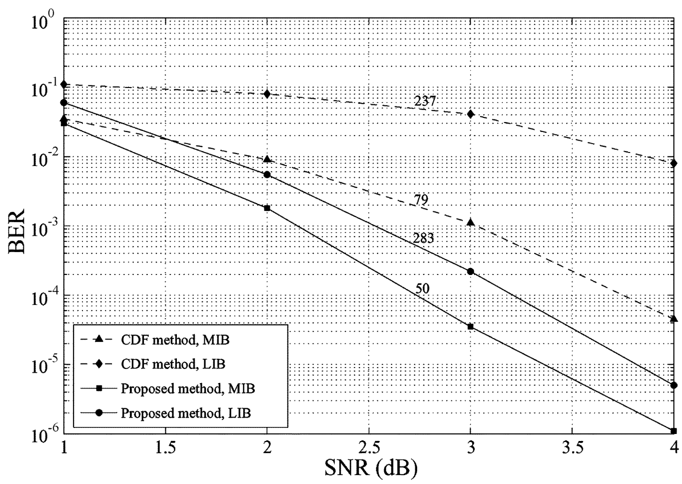


Fig. 7. Comparison of the codes designed by the CDF method and the proposed method.

are not the information bits. This is because of the parity-check matrix structure that is used. As an example, a code of length $n = 553$ and $R \cong .57$ would have an $H$ matrix of the form depicted in Fig. 6, in which $H_1$ is a 316×79 submatrix, $H_2$ and $H_3$ are 158×79, and $H_4, H_5, H_6$, and $H_7$ are 79×79 submatrices. Note that all elements in the gray part of $H$ are zeros. Moreover, the bits corresponding to $H_1$ are the most protected bits. We claim that the codeword bits corresponding to $H_1$ are the PB. Otherwise, we must have rank $(H \setminus H_1) = n(1-R) = 237$, which is impossible. Therefore, the most highly protected bits are the PB. By a similar argument, it is shown that $H_2$ and $H_3$ cannot together correspond to the information bits. Therefore, a possible choice for information bits can correspond to $H_2$, $H_4$, and $H_5$. For comparison, we also give a code based on our proposed method having length $n = 555, R = 0.6, d_m = 23, d_l = 3, d_p = 2, d_c = 11$, and $\alpha \cong 0.15$. It should be mentioned that in this example, the channel is AWGN. We used the same code that we designed using DE formulas over the BEC. Fig. 7 shows the BER versus SNR for the information bits. Note that BER of the PB was not shown in this figure. The number over each graph represents the number of information bits in each part. It is concluded that although our proposed code has a slightly higher rate, it has much better performance than the code in [4].

## V. Conclusion

This letter investigated the design of high-performance UEP-LDPC codes over the BEC. A method based on partially regular random graphs was proposed. We derived UDE formulas over the BEC to optimize the codes for the UEP property. Using the DE formulas, we found codes with good performance that also have a significant UEP property. Simulation results show that we achieve much higher protection for the MIB in comparison with the LIB. Our results were compared with a BEC-optimized irregular code and the regular (3,6) code for lengths 4000 and 1000. We also compared our method with the time-sharing method and the previous UEP-LDPC method. The results suggest the superiority of the proposed method versus the aforementioned methods for the UEP.

## References


[1] B. Masnick and J. Wolf, “On linear unequal error protection codes,” *IEEE Trans. Inf. Theory*, vol. IT-3, no. 10, pp. 600–607, Oct. 1967.

[2] C. C. Kilgus and W. C. Gore, “Cyclic codes with unequal error protection,” *IEEE Trans. Inf. Theory*, vol. IT-17, no. 3, pp. 214–215, Mar. 1971.

[3] I. Boyarinov and G. Katsman, “Linear unequal error protection codes,” *IEEE Trans. Inf. Theory*, vol. IT-27, no. 3, pp. 168–175, Mar. 1981.

[4] B. Vasic, A. Cvetkovic, S. Sankaranarayanan, and M. Marcellin, “Adaptive error protection low-density parity-check codes for joint source-channel coding schemes,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jun.–Jul. 2003, p. 267.

[5] N. Rahnavard and F. Fekri, “Unequal error protection using low-density parity-check codes,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jun.–Jul. 2004, p. 449.

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

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

[8] P. Oswald and A. Shokrollahi, “Capacity-achieving sequences for the erasure channel,” *IEEE Trans. Inf. Theory*, vol. 48, no. 12, pp. 3017–3028, Dec. 2002.

[9] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, “Practical loss-resilient codes,” in *Proc. 29th Annu. ACM Symp. Theory Comput.*, 1997, pp. 150–159.

[10] Ecole Polytech. Federale de Lausanne (EPFL). Lausanne, Switzerland [Online]. Available: http://ltchwww.epfl.ch/research/ldpcopt/

[11] M. Yang, Y. Li, and W. Ryan, “Design of efficiently encodable moderate-length high-rate irregular LDPC codes,” *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 564–571, Apr. 2004.