TABLE V
NUMBERS OF COSETS OF WEIGHT 6

(s,t)	$(1,1)$	$(1,2)$	$(1,3)$	$(2,1)$	$(2,2)$
#	31303	242098	537786	42941	380328
(s,t)	$(2,3)$	$(3,1)$	$(3,2)$	$(3,3)$	$(4,1)$
#	857647	27554	282210	680455	10994
(s,t)	$(4,2)$	$(4,3)$	$(5,1)$	$(5,2)$	$(5,3)$
#	133032	360939	2047	28566	97865
(s,t)	$(6,1)$	$(6,2)$	$(6,3)$	$(7,2)$	$(7,3)$
#	161	4048	17549	230	2001
(s,t)	$(8,3)$				
#	276				

TABLE VI
NUMBERS OF COSETS OF WEIGHT 7

(s,t)	$(1,1)$	$(2,1)$	$(3,1)$	$(3,2)$	$(4,1)$
#	276	2001	17549	345	97865
(s,t)	$(4,2)$	$(5,1)$	$(5,2)$	$(6,1)$	$(6,2)$
#	1725	360939	18630	680455	36570
(s,t)	$(7,1)$	$(7,2)$	$(8,1)$	$(8,2)$	$(9,1)$
#	857647	74865	537786	31050	217097

TABLE VII
NUMBERS OF COSETS OF WEIGHT 8

(s,t)	$(1,1)$	$(2,1)$	$(3,1)$	$(3,2)$
#	12512	45218	217097	15180

weightdistribution is uniquely determined. In the tables, # lists the numbers of cosets for these weights. For cosets of weights $4, 5, 6, 7, 8$, the weight distributions are written using parameters s, t, and the numbers of cosets with a given weight distribution are listed in Tables III–VII, respectively. We remark that the unique coset of weight 11 is the shadow of C_{46} (see, e.g., [4] and [9] for the definition of the shadow).

REFERENCES

[1] E. F. Assmus Jr. and V. Pless, "On the covering radius of extremal self-dual codes," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 3, pp. 359–363, May 1983.

[2] D. J. Bergstrand, "New uniqueness proofs for the (5, 8, 24), (5, 6, 12) and related Steiner systems," *J. Combin. Theory Ser. A*, vol. 33, pp. 247–272, 1982.

[3] P. Delsarte, "Four fundamental parameters of a code and their combinatorial significance," *Inf. Control*, vol. 23, pp. 407–438, 1973.

[4] M. Harada, A. Munemasa, and V. D. Tonchev, "A characterization of designs related to an extremal doubly-even self-dual code of length 48," *Ann. Comb.*, to be published.

[5] M. Harada and M. Ozeki, "Extremal self-dual codes with the smallest covering radius," *Discr. Math.*, vol. 215, pp. 271–281, 2000.

[6] S. K. Houghten, C. W. H. Lam, L. H. Thiel, and J. A. Parker, "The extended quadratic residue code is the only $(48, 24, 12)$ self-dual doubly-even code," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 53–59, Jan. 2003.

[7] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.

[8] C. L. Mallows and N. J. A. Sloane, "An upper bound for self-dual codes," *Inf. Control*, vol. 22, pp. 188–200, 1973.

[9] E. M. Rains, "Shadow bounds for self-dual codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 1, pp. 134–139, Jan. 1998.

Nonuniform Error Correction Using Low-Density Parity-Check Codes

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

***Abstract*—This correspondence introduces a framework to design and analyze low-density parity-check (LDPC) codes over nonuniform channels. We study LDPC codes for channels with nonuniform noise distributions, rate-adaptive coding, and unequal error protection. First, we propose a technique to design LDPC codes for volume holographic memory (VHM) systems for which the noise distribution is nonuniform. We show that the proposed coding scheme has an easy design procedure and results in efficient codes for holographic memories. An important property of the proposed technique is the design of the codes that have a low error floor and low variable node degrees, while maintaining performance close to the Shannon limit. We then show that punctured LDPC codes can be studied as a special case of our design methodology for nonuniform channels. Finally, we propose a method to generate LDPC codes that can provide unequal error protection in addition to having a good overall performance. Moreover, the highly protected bits can be decoded without requiring the entire word to be decoded.**

***Index Terms*—Bipartite graphs, error floor, iterative decoding, low-density parity-check (LDPC) codes, nonuniform channels, punctured codes, unequal error protection.**

I. INTRODUCTION

In this correspondence, we study three closely related applications of low-density parity-check (LDPC) codes: coding for nonuniform channels, rate-compatible coding using punctured codes, and unequal error protection. In the first application, we concentrate on the design and analysis of LDPC codes over nonuniform channels. Specifically, we focus on volume holographic memory (VHM) systems that can be modeled as a set of parallel channels as in Fig. 1. In [1], [2] we have already shown that using proper LDPC codes instead of conventional coding schemes can result in more than a 50% increase in the storage capacity of these systems. In the second application, we investigate punctured LDPC codes and show that they can be considered as a special case of our model for nonuniform channels. Finally, we study unequal error protection using LDPC codes.

First, we investigate the design of LDPC codes over a set of parallel subchannels. Consider Fig. 1, where we transmit bits over several binary-input output-symmetric channels. For simplicity, we may assume that the channels are independent. One trivial approach is to design a separate error-correcting code for each of the channels. Here, we are interested in designing only one LDPC code as shown in Fig. 1. Suppose we use a code of length n. We transmit any codeword over the set of channels such that $n^{(j)}$ bits in any codeword are transmitted over the jth channel. Let $p_j = \frac{n^{(j)}}{n}$. Assume $0 < p_j$ for $j = 1, \dots, k_r$. Let z_j be the random variable that is equal to the log-likelihood ratio (LLR) of a received bit from the jth channel. Then, if the bits that are transmitted over the different channels are chosen randomly from the

Manuscript received May 9, 2003; revised January 22, 2005. This work was supported in part by the National Science Foundation under Grant CCF-0430964 and the Air Force Office of Scientific Research under Grant F49620-02-1-0053 (Kent Miller).

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0250 USA (e-mail: hossein@ece.gatech.edu; nazanin@ece.gatech.edu; fekri@ece.gatech.edu).

Communicated by R. Koetter, Associate Editor for Coding Theory.
Digital Object Identifier 10.1109/TIT.2005.850230

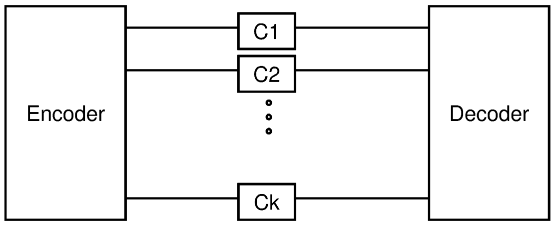

Fig. 1. Several parallel channels.

n bits in a codeword, the set of parallel channels can be modeled as a single channel having the LLR that has the distribution

$$P_Z(z) = \sum_{j=1}^{k_r} p_j P_{Z_j}(z). \tag{1}$$

Therefore, good degree distributions for LDPC codes can be found by the methods described in [3] and [4] for the corresponding $P_Z(z)$.

The first goal of this correspondence is to show that for certain practical problems, we can employ an improved method that provides some advantages over the above method. We consider VHM systems and show that they can be modeled as a set of parallel channels as in Fig. 1. Then, we introduce the ensemble of graphs that are used over parallel channels. We present the asymptotic analysis of the performance of the corresponding codes. We then discuss the design methodology for practical systems and we present some results for VHM systems. Relevant work regarding the application of LDPC codes for parallel channels can be found in [5].

Second, we consider the construction of rate-compatible LDPC codes via puncturing, one of the most common methods used to construct rate-compatible codes. In this method, in order to change the rate of a code to a higher rate, we puncture (delete) a subset of the codeword bits. Puncturing has been studied for convolutional and turbo codes [6]–[8]. The near Shannon limit performance of LDPC codes [4], [9], [10] motivates us to construct rate-adaptive LDPC codes. Previous work on finding puncturing patterns for LDPC codes is given in [11] where it is shown that punctured LDPC codes exhibit desirable properties. First, the performance of a good LDPC code is maintained for a wide range of rates (as defined in Section III, we define the performance as the ratio of the code rate to the channel capacity for small enough bit-error rates (BERs)). Second, there is no theoretical limitation on the number of rates or the values of rates we can generate. In Section III, we present some results on punctured codes and show that a randomly punctured LDPC code usually has a good performance. We show that a punctured code can be modeled as a code that is used over two parallel channels as Fig. 1. In this model, punctured bits are transmitted over the second channel that has a zero capacity. Thus, our proposed density evolution formulas for the parallel channels can be used to find optimum puncturing patterns for the LDPC codes.

Third, we consider a closely related problem of unequal error protection (UEP). Some previous works on UEP codes can be found in [12]–[15]. In Section IV, we will be concerned with a possibly uniform channel; however, we would like to impose intentional nonuniform BERs for different sets of bits. In other words, we would like to protect some bits more than others. In particular, we are interested in unequal error correction for data frames. A transfer frame consists of a header, a body, and a trailer. We usually want a smaller error probability for the header information, which contains important routing information such as the destination address and the frame number. It is also desirable to be able to read the header data without decoding the whole frame. This prevents all intermediary routers from having to decode the entire frame.

Specifically, suppose we send data in the forms of frames of length n over a network. These n bits include the redundant bits due to the error-correcting code. Moreover, suppose a very small fraction of the data in a frame (the header bits), consisting of $\xi(n)$ bits, is very important to us. Let us call them important bits. We need a coding scheme with the following properties. First, the important bits must have a considerably smaller error rate than the rest of the bits in the codeword. Second, for a given code rate, the average BER of the code must be acceptable. In other words, we want to minimize the price that we may have to pay for the unequal error protection. Thus, we would like the UEP code to have overall performance close to the best ordinary codes for the same rate and block length. Third, we want to be able to decode the header data without decoding the whole frame. Our goal in Section IV is to show that we can satisfy the above requirements with LDPC codes. The good performance of LDPC codes makes them good candidates for the problem described above.

Throughout the correspondence we assume the following terminology. By a graph we mean a simple graph, i.e., a graph with no loops (edges joining a vertex to itself) and no multiple edges (several edges joining the same two vertices). Let A be a subset of the vertices in the graph g. Then $N(A) = N^1(A)$ shows the set of neighbors of A in g. More generally, for $j \in \mathbb{N}$, $N^j(A)$ is the set of vertices in g from which there is path of length j to a vertex in A. Let D be a subgraph of g such that its vertex set is A. We say D is induced by A if D contains all edges of g that join two vertices in A. For a square matrix M, $r(M)$ denotes the spectral radius of M. In other words,

$$r(M) = \max\{|\eta| : \eta \text{ is an eigenvalue of } M\}.$$

Similar to [4], for a random variable X with distribution F_X we define

$$P_e(F_X) = \Pr\{X < 0\} + \frac{1}{2}\Pr\{X = 0\}.$$

II. Nonuniform Error Correction

A. VHM Systems

Some practical applications may benefit from the use of nonuniform error protection. For example, in holographic data storage, information is recorded and retrieved in the form of two-dimensional data pages (i.e., two-dimensional patterns of bits). The bits in a page are subject to different sources of noise and interference (such as inter-page interference (IPI), limited diffraction, aberration, misalignment error, and nonuniform erasure [16]). The noise distribution at any point in the page is obtained by the superposition of these noise sources. We assume that the noise is Gaussian and the signal-to-noise ratio (SNR) decreases as we move from the center to the corner of the page [16]. Typically, the raw BER might vary by two or three orders of magnitude over a page. The common approach to solve the nonuniform error protection problem is to use an interleaver followed by a Reed–Solomon (RS) code [16]. It has been shown through simulations that LDPC codes optimized for nonuniform channels, result in an increase in the storage capacity of a typical holographic data storage by more than 50% compared to the approach using an interleaver and an RS code [1], [2]. In this section, we discuss the design methodology for the LDPC codes that are used in the VHM systems. However, note that this design procedure is also applicable to other systems such as rate-compatible codes, orthogonal frequency-division multiplexing (OFDM) systems and multilevel coding.

Consider a VHM page of $N \times N$ pixels. Each pixel is subject to noise with a probability density that is dependent on the pixel location in the page. Generally, pixels at the corner of a data page have higher probability of error than those at the center of the page. We divide this page into k_r regions in which pixels are subject to almost the same noise power. Let the regions be $R_1, R_2, \ldots, R_{k_r}$. All the bits in a page are written or read simultaneously. Thus, this page can be modeled as k_r parallel binary input channels as in Fig. 1.

B. Ensemble $g(\Lambda, \rho)$

There are several ways to define the ensembles of LDPC codes suitable for nonuniform channels. We introduced such an ensemble in [1] but in this correspondence we use a slightly simpler ensemble. Again, suppose we use a code of length n, and we transmit each codeword over the set of channels such that $n^{(j)}$ bits from every codeword are transmitted through the jth channel. Let $(x_1, x_2, \ldots, x_n)$ be a codeword. Let also $W^{(j)}$ be the set of bits in the codeword that are transmitted over the jth channel (type j bits). Thus, we have $|W^{(j)}| = n^{(j)}$, where $|\cdot|$ denotes the cardinality of the set. For example, in the VHM system, $W^{(j)}$ is the set of bits in the jth region (i.e., $W^{(j)} = \{x_i : x_i \in R_j\}$). Now we define the ensemble $g(\Lambda, \rho)$ of bipartite graphs for nonuniform error protection. Let E be the set of edges in the graph and let $E^{(j)}$ be the set of edges that are incident with a variable node of type j. Also, let $E_i^{(j)}$ be the set of the edges that are adjacent to the variable nodes of type j and degree i. We define

$$\lambda^{(j)}(x) = \sum \lambda_i^{(j)} x^{i-1} \tag{2}$$

where

$$\lambda_i^{(j)} = \frac{|E_i^{(j)}|}{|E^{(j)}|}. \tag{3}$$

Let $\Lambda = \{\lambda^{(j)}(x) : j = 1, \ldots, k_r\}$. Let also $\rho(x) = \sum \rho_i x^{i-1}$, where ρ_i is the fraction of edges connected to a check node of degree i [4]. We define the ensemble $g(\Lambda, \rho)$ as the ensemble of bipartite graphs with the degree distributions given by Λ and ρ. In other words, in the ensemble $g(\Lambda, \rho)$, variable nodes corresponding to bits of different types may have different degree distributions. In fact, we propose to design codes with the prior knowledge of which bits are transmitted over each channel. Our aim in this correspondence is to show that this method has some advantages in certain applications.

C. Asymptotic Analysis

Similar to [4], we can find the density evolution formulas for the ensemble $g(\Lambda, \rho)$. Let us define

$$q^{(j)} = \frac{|E^{(j)}|}{|E|}. \tag{4}$$

Let $m_{vc}^{(l),(j)}$ denote the message that is sent from a variable node v of type j (i.e., $v \in W^{(j)}$) to its incident check node c at the lth iteration of the message passing algorithm. Let also $m_{cv}^{(l)}$ denote the message that the check node c sends to its incident variable node. Let $P_l^{(j)}$ and Q_l denote the densities of random variables $m_{vc}^{(l),(j)}$ and $m_{cv}^{(l)}$, respectively. Let also P_l' be the density of the message that is sent on a randomly chosen edge (from the variable node to the check node) at the lth iteration. Then, it can be shown that the formulas for the density evolution can be written as

$$P_l^{(j)} = P_0^{(j)} \otimes \lambda^{(j)}(Q_l) \tag{5}$$
$$P_l' = \sum q^{(j)} P_l^{(j)} \tag{6}$$
$$Q_l = \Gamma^{-1}\left(\rho\left(\Gamma\left(P_{l-1}'\right)\right)\right) \tag{7}$$

where $\otimes$ denotes convolution and Γ is defined in [4] in the following way. If Z is a random variable with the distribution F_Z, then $\Gamma(F_Z)$ is defined as [4]

$$\Gamma(F_Z)(s, x) = I_{(s=0)}\Gamma_0(F_Z)(x) + I_{(s=1)}\Gamma_1(F_Z)(x) \tag{8}$$

where I is the indicator function and

$$\begin{aligned} \Gamma_0(F_Z)(x) &= 1 - F_Z^-\left(-\ln\tanh\left(\frac{x}{2}\right)\right) \\ \Gamma_1(F_Z)(x) &= F_Z\left(\ln\tanh\left(\frac{x}{2}\right)\right). \end{aligned} \tag{9}$$

Note that F_Z in (8) will be the corresponding distribution for P_{l-1}'.

Let $c^{(j)}$ be the capacity of the jth binary channel in Fig. 1 and suppose that we use a randomly chosen LDPC code from the ensemble $g(\Lambda, \rho)$. Using (5)–(7) we can prove the following lemma.

Lemma 1: Suppose $c^{(j)} < 1$ for $j = 1, \ldots, k_r$ and $k_r < \infty$. Then, for any $i, j \in \{1, 2, \ldots, k_r\}$ we have $\lim_{l\to\infty} P_e(P_l^{(j)}) = 0$ if and only if $\lim_{l\to\infty} P_e(P_l^{(i)}) = 0$.

Proof: (Sketch) Using (5) and the assumption $c^{(j)} < 1$, we conclude that to have $\lim_{l\to\infty} P_e(P_l^{(j)}) = 0$, the density Q_l should converge to a Delta function at infinity. This means that for any $i \in \{1, 2, \ldots, k_r\}$, we must have

$$\lim_{l\to\infty} P_e(P_l^{(i)}) = 0. \qquad \square$$

Using Lemma 1 and (5)–(7) we can optimize the degree distribution of the code for the given channels. It seems that the design of good codes from the ensemble $g(\Lambda, \rho)$ is more difficult than the design of the ordinary irregular LDPC codes because of the larger number of parameters involved in the optimization. However, we will show that finding a good degree distribution for a set of parallel channels is simpler than the optimization of ordinary LDPC codes. The reason is that we can use simpler ensembles such as semiregular ensembles (which will be defined later). In fact, the simplicity of design is one advantage of using the ensemble $g(\Lambda, \rho)$.

Most of the results for ordinary LDPC codes such as the concentration theorem, the cycle-free convergence, the stability condition of [3] and [4], and the Gaussian approximation of [17] can also be generalized for the ensemble $g(\Lambda, \rho)$. The Gaussian approximation formulas for ensemble $g(\Lambda, \rho)$ are given in the Appendix. Here we give the stability condition for this ensemble. Other generalizations are straightforward. For simplicity, we derive the stability condition when all the channels in Fig. 1 are binary erasure channels (BECs) with different erasure probabilities. Let ϵ_j be the erasure probability of the jth channel. Note that for this case, the system of parallel channels is equivalent to a BEC with the erasure probability

$$\epsilon = \sum_{j=1}^{k_r} p_j \epsilon_j \tag{10}$$

where $p_j = \frac{n^{(j)}}{n}$. However, as we mentioned before, it is better to work with the set of parallel channels instead of the derived single channel. Let $x_l^{(j)}$ be the fraction of erasure messages emitted from the variable nodes of type j in the lth iteration. Then the density evolution formulas are

$$\begin{aligned} x_l^{(j)} &= x_0^{(j)} \lambda^{(j)}(1 - \rho(1 - y_{l-1})) \\ y_l &= \sum_{j=1}^{k_r} q^{(j)} x_l^{(j)} \\ x_0^{(j)} &= \epsilon_j, \qquad \text{for } j = 1, 2, \ldots, k_r. \end{aligned} \tag{11}$$

Let $\underline{X}_l$ be

$$\underline{X}_l = \begin{pmatrix} x_l^{(1)} \\ x_l^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ x_l^{(k_r)} \end{pmatrix}. \tag{12}$$

Then, the stability condition for the ensemble $g(\Lambda, \rho)$ can be stated as follows.

Theorem 1: Let $\epsilon_j < 1$ for $j = 1, \dots, k_r$ and M be a $k_r \times k_r$ matrix whose element in the jth row and the ith column is

$$\alpha_{ji} = \lambda'^{(j)}(0)\rho'(1)q^{(i)}x_0^{(j)}.$$

Then, we have the following conditions.

- If $r(M) > 1$, then there exists a strictly positive constant $\zeta = \zeta(\Lambda, \rho, \underline{X}_0)$ such that for all $l \in \mathbb{N}$ and for $j = 1, \dots, k_r$, we have $x_l^{(j)} > \zeta$.
- If $r(M) < 1$, then there exists a strictly positive constant $\zeta = \zeta(\Lambda, \rho, \underline{X}_0)$ such that if $x_l^{(j)} \leq \zeta$ for some $l \in \mathbb{N}$ and for $j = 1, \dots, k_r$, then $\lim_{l \to \infty} x_l^{(j)} = 0$ for $j = 1, \dots, k_r$.

Proof: By expanding the density evolution formula into the Taylor series at zero and neglecting high order terms, we get

$$\underline{X}_l = M\underline{X}_{l-1}. \tag{13}$$

If $\|\underline{X}_l\|$ is sufficiently small, then we have $\lim_{l \to \infty} \underline{X}_l = 0$ if and only if $\lim_{k \to \infty} M^k = 0$. This is equivalent to $r(M) < 1$. The rest of the proof is similar to the proof of the stability condition in [4]. □

We finally give an upper bound for the rate of the codes from the ensemble $g(\Lambda, \rho)$ with the maximum likelihood (ML) decoding. This bound is valid for the iterative decoding as well. It is similar to the bound given in [18]. Let φ_i be the fraction of check nodes of degree i. Let us define $\Phi(x) = \sum_i \varphi_i x^i$. By a simple observation, we can find the following upper bound on the capacity of the LDPC codes over the BEC. The proof is similar to the one presented in [18] on the bound for uniform channels.

Theorem 2: Consider k_r parallel binary erasure subchannels as in Fig. 1 with erasure probabilities $\epsilon_1, \epsilon_2, \dots, \epsilon_{k_r}$. Then for an arbitrarily small error probability we must have

$$1 - R \geq \frac{\epsilon}{1 - \Phi(1 - \epsilon')} \tag{14}$$

where $\epsilon' = \sum_{j=1}^{k_r} q^{(j)}\epsilon_j$ and $q^{(j)}$ and ϵ are given by (4) and (10).

D. Advantages of the Ensemble $g(\Lambda, \rho)$

Here we briefly explain the advantages of using the ensemble $g(\Lambda, \rho)$. These advantages are further explained and verified using simulations in Section V. Note that in our model, we know which subsets of bits are transmitted through each channel. The important fact about the ensemble $g(\Lambda, \rho)$ is that we use this information in the code design. Note that in ordinary ensembles of LDPC codes, we do not use this information in the code design, instead we use the average density of the LLRs of channels for each bit. This extra information results in several advantages of the ensemble $g(\Lambda, \rho)$ over the ordinary ensembles. The first advantage is that we can use lower values for variable nodes in the degree distribution. In other words, we can obtain sparser codes using the ensemble $g(\Lambda, \rho)$ having the same performance of ordinary LDPC codes. This results in faster decoding and more efficient implementation.

In ordinary LDPC codes ensembles, in order to approach the channel capacity we need to have a high number of degree-two variable nodes in the graph [19]. Thus capacity-approaching LDPC codes usually suffer from the error floor problem. However, in the ensemble $g(\Lambda, \rho)$, since we use more information in the code design, we can have codes with a low number or even no degree-two variable nodes that still have thresholds close to the Shannon limit. This is particularly very important in data storage systems such as holographic memories because a very low error probability is required.

Another advantage is simpler design. It is worth noting that in ordinary LDPC codes, regular ensembles usually do not have thresholds close to the Shannon limit. Thus, in ordinary LDPC codes in order to approach channel capacity we need to use highly irregular codes. However, in the ensemble $g(\Lambda, \rho)$ part of the required irregularity is achieved by channel nonuniformity. In fact, we will show in Section V that we can approach the channel capacity by using semiregular codes (codes in which bits that are transmitted through the same channel correspond to variable nodes with the same degrees). This will simplify the degree optimization significantly.

Finally, for short codes, we can get better performance by using the ensemble $g(\Lambda, \rho)$ because more information is available in the code design. Note that ensemble $g(\Lambda, \rho)$ is a generalization of the ordinary ensembles of LDPC codes. In fact, by choosing all $\lambda^{(j)}(x)$ equivalent we obtain an ordinary ensemble of LDPC codes. Thus, in all circumstances, the performance of the codes obtained from the ensemble $g(\Lambda, \rho)$ is at least as good as the codes obtained from ordinary ensembles.

III. Rate-Compatible LDPC Codes

In this section, we are concerned with punctured codes over binary-input output-symmetric memoryless (BIOSM) channels. We restrict ourselves to normalized channels [10]. A normalized channel is defined as the channel obtained by concatenation of a BIOSM channel with log-likelihood mappings. The normalization of a channel is a lossless process because the set of log likelihoods is a sufficient statistic for decoding. Thus, we say two channels C_1 and C_2 are equivalent if their normalized channels are the same. We represent the capacities of the channels by c_1 and c_2, respectively. We first prove the following lemma that is useful for modeling of punctured codes.

Lemma 2: A normalized BIOSM channel has zero capacity if and only if the received LLR is equal to zero with probability one.

Proof: Let X and Y be the random variables representing the input and output of a BIOSM channel, respectively. We define the random variable U in the following way. We let the input to the channel be $X = 1$. If y is the output of the channel, then

$$U = \log \frac{\mathrm{pr}\{X = 1 | Y = y\}}{\mathrm{pr}\{X = -1 | Y = y\}}, \text{ when the channel input is } X = 1. \tag{15}$$

Then the capacity of the normalized channel is given by [10]

$$c = 1 - E[\log_2(1 + e^{-U}) | X = 1]. \tag{16}$$

Thus, if $U = 0$ with probability one, then $c = 0$. Moreover, if $c = 0$, we have $E[\log_2(1 + e^{-U})] = 1$. If we assume $p(u)$ is the probability density function of U, we have

$$\begin{aligned} 1 &= \int_{-\infty}^{+\infty} \log_2(1 + e^{-u})p(u)du \log_2(1 + e^{-u})p(u)du \\ &= \mathrm{pr}\{U = 0\} \\ &\quad + \int_{(0,+\infty)} [\log_2(1 + e^{-u}) - \log_2(1 + e^{u})e^{-u}]p(u)du \end{aligned} \tag{17}$$

where we used $p(u) = e^u p(-u)$ [4]. Since we have $\mathrm{pr}\{U = 0\} \leq 1$ and $[\log_2(1 + e^{-u}) - \log_2(1 + e^{u})e^{-u}] < 0$ for $u \in (0, \infty)$, we conclude $\mathrm{pr}\{U = 0\} = 1$. □

We would like to design rate-adaptive LDPC codes that use the same encoder and decoder for all rates. Let $\mathfrak{R} = \{r_1, r_2, \dots, r_s\}$ be the set of different rates that are needed. Let r_p be the rate of the parent

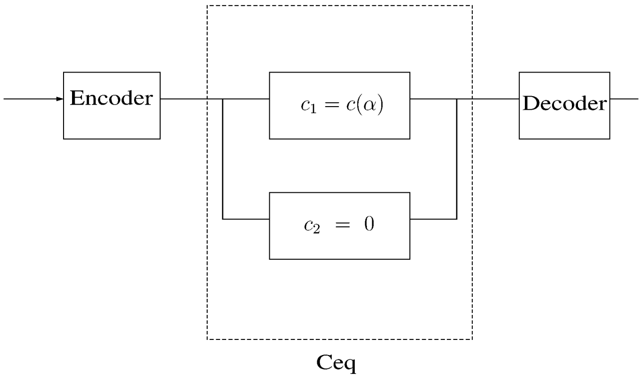

Fig. 2. A model that describes puncturing over a binary channel.

code (i.e., the lowest rate in $\mathfrak{R}$). We consider the following scheme. We design an optimized LDPC code of rate $r_p = k/n$ where k and n are the lengths of information blocks and the codewords, respectively. To generate a code with a new rate, we find an optimum puncturing of a subset of bits in the codeword and send the punctured codeword to the receiver. It is assumed that the decoder knows the positions of the punctured bits in the codeword. At the beginning of the iterative decoding, we need to compute LLRs in the decoder. The LLRs for the punctured bits are set to zero.

Let us define the performance of a rate-compatible code over a channel as $\frac{r}{c}$ where c is the channel capacity and r is the maximum rate of the code for which the error probability is less than a required value. When we consider asymptotic behavior of codes, r is the maximum rate for which an arbitrarily small error probability is achievable. Now consider a time-varying binary-input output-symmetric channel [4] which can be described by its transmission conditional probability $P(y|x, \alpha)$ where $\alpha \in [\alpha_{\min}, \alpha_{\max}]$ is time variant. For example, for binary-input additive white Gaussian noise (BIAWGN) channels, α can be the variance of the noise. Let $c(\alpha)$ be the capacity of this channel. We may assume that $c(\alpha)$ is a decreasing function of α. We design an optimal LDPC code for the rate r_p that is used when $\alpha = \alpha_{\max}$. Now, suppose the channel quality improves. In other words, the value of the parameter α is reduced to a value less than $\alpha_{\max}$. By puncturing, we increase the code rate from r_p to $r(\alpha)$ such that the error probability still becomes less than the required value. If

$$\frac{r_p}{c(\alpha_{\max})} > \frac{r(\alpha)}{c(\alpha)}$$

then we would have a performance loss due to puncturing. Our goal is to minimize the performance loss by finding a good puncturing pattern.

To investigate the performance of punctured LDPC codes, we consider the model depicted in Fig. 2. In this model, it is assumed that the unpunctured bits are transmitted through the channel and the punctured bits are transmitted through a virtual channel with a zero capacity. In fact, by Lemma 2, a normalized BIOSM channel with zero capacity is equivalent to a BEC with erasure probability one. Let p be the fraction of punctured bits and define $r_{eq}(\alpha)$ to be the code rate of the overall channel in Fig. 2. In other words, $r_{eq}(\alpha)$ is the code rate if we consider both punctured and unpunctured bits. With this definition, it is clear that $r_{eq}(\alpha) = r_p$. Note that $C_{eq} = C_{eq}(\alpha)$ is the channel that consists of two subchannels C_1 and C_2 with capacities $c(\alpha)$ and zero, respectively. Therefore, a fraction p of bits are transmitted through C_2 and the rest of the bits are transmitted through C_1. Let $c_{eq}(\alpha)$ be the capacity of $C_{eq}(\alpha)$. In Fig. 2, we have

$$c_{eq}(\alpha) = (1-p)c(\alpha), \qquad r(\alpha) = \frac{r_p}{(1-p)}. \tag{18}$$

Therefore, we have a performance loss due to puncturing, if and only if $c_{eq}(\alpha) > c(\alpha_{\max})$. Let z denote the LLR of the received bits and $\varpi(z;\alpha)$ be the density of z when the all-zero codeword is sent. Then by the following theorem, we identify the channels for which the code performance does not change due to random puncturing. For example, as a special case of the following theorem, we conclude that for a BEC in which α is chosen to be the erasure probability, a random puncturing results in no performance loss. In fact, the performance of the randomly punctured code is the same for all rates. It is important to note that for other types of channels, we usually have some performance degradation because of puncturing. Therefore, we need to optimize the puncturing pattern for these types of channels.

Theorem 3: Let $\varpi(z;\alpha) = \theta(\alpha)\delta(z) + (1-\theta(\alpha))f(z)$ specify a normalized channel in which θ is an increasing function of α such that for all $\alpha \in [\alpha_{\min}, \alpha_{\max}]$, we have

$$0 \le \theta(\alpha) \le \theta(\alpha_{\max}) \le 1 - r_p \quad \text{and} \quad \int_{-\infty}^{+\infty} f(z)dz = 1.$$

Then, the average performance of any binary block code does not change by random puncturing if we choose the puncturing fraction $p(\alpha)$ properly. Moreover, the class of channels defined by $\varpi(z;\alpha)$ is the only class of normalized BIOSM channels having this property.

Proof: First we prove the following lemma,

Lemma 3: The performance of an arbitrary block code with an arbitrary decoder does not change by random puncturing in the scheme of Fig. 2 if and only if there exists a puncturing fraction function $p(\alpha)$ such that for all $\alpha \in [\alpha_{\min}, \alpha_{\max}]$, we have $c_{eq}(\alpha) \equiv c(\alpha_{\max})$.

Proof: Suppose the error probability of all decoders in Fig. 2 stays the same for any random puncturing. Then, the probability density function of the input of the decoders must remain unchanged by puncturing. This implies that $C_{eq} \equiv C(\alpha_{\max})$. Moreover, suppose there exists a puncturing fraction function $p(\alpha)$ such that for all $\alpha \in [\alpha_{\min}, \alpha_{\max}]$ we have $C_{eq}(\alpha) \equiv C(\alpha_{\max})$. Then if we perform random puncturing according to $p(\alpha)$, we have

$$\frac{r(\alpha)}{c(\alpha)} = \frac{\frac{r_p}{1-p}}{\frac{c_{eq}(\alpha)}{1-p}} = \frac{r_p}{c(\alpha_{\max})}. \tag{19}$$

Therefore, the performance of the code stays the same. □

Proof of Theorem 3: Suppose the assumptions of the theorem hold for $\varpi(z;\alpha) = \theta(\alpha)\delta(z) + (1-\theta(\alpha))f(z)$. We choose

$$p(\alpha) = \frac{\theta(\alpha_{\max}) - \theta(\alpha)}{1-\theta(\alpha)}. \tag{20}$$

It is clear that we have $0 \le p(\alpha) \le \theta(\alpha_{\max}) \le 1 - r_p$. The LLR corresponding to $C_{eq}(\alpha)$ is equal to

$$\begin{aligned}
&(1-p(\alpha))\varpi(z;\alpha) + p(\alpha)\delta(z)\left(1 - \frac{\theta(\alpha_{\max}) - \theta(\alpha)}{1-\theta(\alpha)}\right) \\
&\quad \times \left(\theta(\alpha)\delta(z) + (1-\theta(\alpha))f(z)\right) \\
&\quad = \theta(\alpha_{\max})\delta(z) + (1-\theta(\alpha_{\max}))f(z) \\
&\quad = \varpi(z;\alpha_{\max}).
\end{aligned} \tag{21}$$

This is the same as the LLR for $C(\alpha_{\max})$. Therefore, we have $C_{eq}(\alpha) \equiv C(\alpha_{\max})$. By Lemma 3, we conclude that the performance of the codes on this channel does not change by random puncturing.

Now suppose we have a time-varying channel that is defined by $\varpi(z;\alpha)$ such that the performance of codes stays the same by random puncturing. By Lemma 3, we have $C_{eq}(\alpha) \equiv C(\alpha_{\max})$. Therefore, we conclude that $\varpi(z;\alpha)(1-p(\alpha))+p(\alpha)\delta(z) = \varpi(z;\alpha_{\max})$. This results in

$$\varpi(z;\alpha) = \frac{\varpi(z;\alpha_{\max})}{1-p(\alpha)} - \frac{p(\alpha)\delta(z)}{1-p(\alpha)}. \tag{22}$$

Let p_m be the maximum possible fraction of punctured bits. It is clear that $p_m \le 1 - r_p$. Let us define $\theta(\alpha_{\max}) = p_m$ and

$$\theta(\alpha) = \frac{\theta(\alpha_{\max}) - p(\alpha)}{1-p(\alpha)},$$
$$f(z) = \frac{\varpi(z;\alpha_{\max}) - \theta(\alpha_{\max})\delta(z)}{1-\theta(\alpha_{\max})}. \tag{23}$$

Since $p(\alpha)$ is decreasing in α and $0 \le p(\alpha) \le \theta(\alpha_{\max}) \le 1 - r_p$, we conclude that θ is an increasing function of α and $0 \le \theta(\alpha) \le \theta(\alpha_{\max}) \le 1 - r_p$. Moreover, we have

$$\varpi(z;\alpha) = \theta(\alpha)\delta(z) + (1-\theta(\alpha))f(z) \quad \text{and} \quad \int_{-\infty}^{+\infty} f(z)dz = 1.$$

□

It is shown in [10] that if we optimize an LDPC code for a symmetric channel, the code usually has good performance on other types of symmetric channels for which the code is not optimized. This property of LDPC codes can be used to explain the good performance of punctured LDPC codes by examining Fig. 2 as follows. The figure implies that the puncturing process can be considered as a change in the channel instead of the change in the code rate. Therefore, although the LDPC code is optimized for the channel with the parameter $\alpha = \alpha_{\max}$ (or $p = 0$) we expect that it also performs well for other values of α for which $p > 0$ (note that C_{eq} is a symmetric channel). However, we can optimize the puncturing pattern to further improve the performance.

Considering Fig. 2, we can find the density evolution formulas for a punctured LDPC code over a BIOSM channel using the density evolution formulas for the ensemble $g(\Lambda,\rho)$. Then, using these formulas, we obtain good puncturing distributions for LDPC codes. If the channel is subject to Gaussian noise we can also apply the Gaussian approximation method. We now show that by applying the Gaussian approximation formulas of the Appendix we get the same result as [11].

Let $m_u(l)$ denote the mean of the messages from check nodes to variable nodes in the lth iteration. Let also $m_0 = \frac{2}{\sigma^2}$ where σ is the variance of C_1 in Fig. 2. We define $\psi_i^{(1)}$ to be the fraction of unpunctured variable nodes of degree i among all the unpunctured variable nodes in the graph. Define $\psi_i^{(2)}$ for the punctured variable nodes similarly. If the puncturing fraction is p, we have

$$\psi_i = (1-p)\psi_i^{(1)} + p\psi_i^{(2)} \tag{24}$$
$$\sum_i \psi_i = \sum_i \psi_i^{(1)} = \sum_i \psi_i^{(2)} = 1. \tag{25}$$

Our goal is to find $\{\psi_i^{(2)}\}_{i>1}$ such that the performance of the code is optimized. We have

$$\lambda_i^{(j)} = \frac{i\psi_i^{(j)}}{\sum_k k\psi_k^{(j)}}, \qquad j = 1,2. \tag{26}$$

Let $p_{pe} = q^{(2)} = \frac{|E^{(2)}|}{|E|}$. Using the Appendix we define

$$h_i^{(1)}(s,r) = \phi\left(s + (i-1)\sum_j \rho_j \phi^{-1}\right.$$
$$\left.\times\left(1-(1-r)^{(j-1)}\right)\right) \tag{27}$$

$$h_i^{(2)}(s,r) = \phi\left((i-1)\sum_j \rho_j \phi^{-1}\right.$$
$$\left.\times\left(1-(1-r)^{(j-1)}\right)\right) \tag{28}$$

$$h(s,r) = (1-p_{pe})\sum_i \lambda_i^{(1)} h_i^{(1)}(s,r)$$
$$+ p_{pe}\sum_i \lambda_i^{(2)} h_i^{(2)}(s,r). \tag{29}$$

Then we have

$$r_l = h(s, r_{l-1}) \tag{30}$$

where $s = m_0$ and $r_0 = (1-p_{pe})\phi(s) + p_{pe}$. As stated in the Appendix, $r_l(s) \longrightarrow 0$ if and only if $r > h(s,r)$ for all $r \in (0,1)$. We now set up a linear program to optimize the puncturing pattern. Let us define $\mu_i = (1-p_{pe})\lambda_i^{(1)}$ and $\beta_i = p_{pe}\lambda_i^{(2)}$. Here we maximize p for the given α. Thus, we have the following optimization problem:

$$\max_{\mu_i,\beta_i} p = \frac{|E|}{n}\sum_i \frac{\beta_i}{i} \tag{31}$$

with the constraints

$$h(s,r) = \sum_i \mu_i h_i^{(1)}(s,r) + \sum_i \beta_i h_i^{(2)}(s,r) < r,$$
$$0 < r < 1 \tag{32}$$
$$\mu_i + \beta_i = \lambda_i. \tag{33}$$

After finding the optimum values of β_i and μ_i, we can find p_i (the fraction of the variable nodes of degree i that should be punctured) by the following equations:

$$P_{pe} = \sum_i \beta_i,$$
$$\lambda_i^{(2)} = \frac{\beta_i}{p_{pe}}$$
$$\psi_i^{(2)} = \frac{\frac{\lambda_i^{(2)}}{i}}{\sum_k \frac{\lambda_k^{(2)}}{k}}$$
$$p_i = \frac{p\psi_i^{(2)}}{\psi_i}. \tag{34}$$

We note that this is the same as the result in [11].

IV. Unequal Error Protection Using LDPC Codes

We now consider a problem closely related to code design for the nonuniform channels. We will be concerned with uniform channels; however, we would like to impose intentional nonuniformity at the

BERs of different sets of bits. In other words, we would like to protect some bits more than others. In particular, we are interested in unequal error correction for data frames. A transfer frame consists of a header and a body. The header length is usually small compared to the body. In fact, it has usually logarithmic length with respect to the frame length [20]. Thus, if $\xi(n)$ is the header length, it is reasonable to assume $\lim_{n\to\infty} \frac{\xi(n)}{n} = 0$. We usually want a very small error probability for the header information.

Suppose we want to transmit a block of k bits over a BIOSM channel. We also want to use at most $n - k$ redundant bits. Let $\xi(n)$ be the number of important bits that require higher protection. One approach is to use two different block codes, one for the important bits and the other for the rest of the bits. However, it is more interesting to design only one block code that provides unequal error protection. More importantly, using two different LDPC codes is not efficient for the following reason. Since $\xi(n)$ is usually a very small number, we have to use a short-length code for the important bits. However, as we know LDPC codes do not perform well for short lengths. Therefore, to get a good BER we must use a very-low-rate code which is inefficient. Thus, we need to use a different type of code for the important bits. On the other hand, it is not clear that using only one LDPC code and imposing the unequal error protection on the code would result in an efficient coding scheme. In fact, our aim in this section is to study this.

A. Perfect Protection

Suppose we transmit binary bits over a binary channel with capacity $c(\alpha)$ where α is the parameter of the channel. We want to use a block code of rate R that performs unequal error protection. Let $P_E(C,\alpha)$ be the average error rate of the code C when the channel parameter is equal to α. Let also $P_E^{\xi}(C,\alpha)$ be the average error rate of the important bits. Let $\mathcal{C}_R$ be the class of codes of type $\mathcal{C}$ and rate R. For example, LDPC$_R$ is the class of LDPC codes of rate R. We define $\mathcal{C}_R^{\epsilon,\delta}$ as the class of unequal error protection codes of type $\mathcal{C}$ and rate R that satisfy the following property. For any $C \in \mathcal{C}_R^{\epsilon,\delta}$ if $c(\alpha) > \delta$, then we have $P_E^{\xi}(C,\alpha) < \epsilon$.

Definition 1: We say that an unequal error protection scheme $\mathcal{C}_R^{\epsilon,\delta}$ perfectly protects the important bits if for any positive numbers ϵ and δ and any code C in $\mathcal{C}_R$, there exists a code C' in $\mathcal{C}_R^{\epsilon,\delta}$ such that $P_E(C',\alpha) \leq P_E(C,\alpha)$.

Intuitively, perfect protection implies an unequal error protection without paying any price. In other words, even if the channel capacity $c(\alpha)$ becomes arbitrary close to zero, we are able to get arbitrarily small error probability for the important bits without loosing anything with respect to other bits. It can be seen that for asymptotically good codes, perfect protection is possible only when $\lim_{n\to\infty} \frac{\xi(n)}{n} = 0$; otherwise, we violate the fundamental theorem of Shannon capacity. This assumption is reasonable for applications such as data frames where $\xi(n) \ll n$.

B. An Unequal Error Protection Scheme

Now we propose a scheme for unequal error protection using LDPC codes. Conventional LDPC codes provide almost equal error protection. Although high-degree variable nodes have lower error probabilities in irregular LDPC codes, the difference between the error rates of the variable nodes of different degrees is not considerable (usually, less than one order of magnitude). Moreover, this difference reduces when the channel becomes worse. Note that it is usually difficult, if not impossible, to find good unequal error protection LDPC codes by searching for different degree distributions. This is because we have to choose the degree distribution to be extremely irregular (i.e., we have to choose very high degrees for the important bits), which is usually harmful if we cannot have a large enough code length.

Let A be the set of important variable nodes and $|A| = \xi(n)$. We propose a scheme based on the degree distributions of the vertices in the sets $N^0(A) = A, N(A), N^2(A), \ldots, N^h(A)$, where h is a constant. Note that $N^j(A)$ consists of variable nodes if j is even. Otherwise, $N^j(A)$ consists of check nodes. As explained in [21], from the point of view of variable nodes, it is best to have high degrees. On the contrary, for check nodes, it is best to have low degrees. In fact, our scheme is based on the above fact. Let $g_n(\lambda,\rho)$ be the ensemble of irregular graphs introduced in [4] and [22]. That is the ensemble of bipartite graphs having degree distribution (λ,ρ) and length n. We define $g_n(\lambda,\rho,h,d_v,d_c)$ as the ensemble of bipartite graphs for which the degree of each vertex in $N^j(A)$ for $j = 0,1,\ldots,h$ is equal to d_v if j is even. Otherwise, it is equal to d_c. The degree distribution of the vertices in the rest of the graph is determined by λ and ρ similar to the ensemble $g_n(\lambda,\rho)$. Note that for simplicity we assume that the degrees of all the variable nodes in the sets $N^0(A) = A, N(A), N^2(A), \ldots, N^h(A)$ are the same and the degrees of all the check nodes in these sets are the same. We could have also assigned an irregular degree distribution to the vertices in $N^j(A)$. In general, a graph from the ensemble $g_n(\lambda,\rho,h,d_v,d_c)$ is similar to a graph from the ensemble $g_n(\lambda,\rho)$ having the extra condition that the vertices of A and their neighborhood of depth h must have certain degree distributions.

Theorem 4: If $\lim_{n\to\infty} \frac{\xi(n)}{n} = 0$, then the ensemble of the codes defined by $g_n(\lambda,\rho,h,d_v,d_c)$ satisfies the perfect protection property. In other words, for any positive numbers ϵ and δ and any code C in $g_n(\lambda,\rho)$ of rate R, there exists a code C' in $g_{n'}(\lambda,\rho,h,d_v,d_c)$ having the same rate as C such that $P_E(C',\alpha) \leq P_E(C,\alpha)$. Furthermore, if $c(\alpha) > \delta$, we have $P_E^{\xi}(C',\alpha) < \epsilon$.

Proof: For simplicity, we prove the theorem for the BEC. The extension to other channels is immediate. Let α be the erasure probability. Thus, $c(\alpha) = 1 - \alpha$. Suppose we are given positive numbers ϵ and δ and a degree distribution pair (λ,ρ) of rate R. We show that for sufficiently large n', the ensemble $g_{n'}(\lambda,\rho,h,d_v,d_c)$ satisfies the requirements of the theorem. Define

$$B = \bigcup_{j=0}^{h} N^j(A). \tag{35}$$

Since $|B| \leq \max(d_v,d_c)^h \xi(n)$ we have $\lim_{n\to\infty} \frac{|B|}{n} = 0$. Let x_l be the average fraction of erasure messages which are passed in the lth iteration of the iterative decoding on a graph g in $g_n(\lambda,\rho)$. Let also y_l be the average fraction of erasure messages which are passed in the lth iteration of the iterative decoding on a graph g' in $\mathcal{C}_{n'}(\lambda,\rho)$. As $n' \to \infty$, if we pick a vertex from g' at random, with high probability, its neighborhood of depth d does not contain any vertices from the set B for any constant d. Thus, by arguments similar to those in [3], we conclude that $y_l \to x_l$ as $n, n' \to \infty$. Therefore, for sufficiently large n', we have $P_E(C',\alpha) \leq P_E(C,\alpha)$.

Now let I be the subgraph of g' that is induced by the vertices in B, as shown in Fig. 3. Let t_l be the probability that the value of a variable node in A, the set of important bits, is unknown after the lth iteration. Note that with high probability the graph I is cycle free. Thus, using the similar arguments as in [3] and [22], we conclude that for $l \leq h$, t_l satisfies $t_l = z_0[1-(1-z_{l-1})^{d_c-1}]^{d_v}$ where

$$\begin{aligned} z_0 &= \alpha \\ z_l &= z_0[1-(1-z_{l-1})^{d_c-1}]^{d_v-1}. \end{aligned} \tag{36}$$

Using (36), we see that for any $0 \leq \alpha < 1$ we can always choose the parameters h, d_v, and d_c such that $t_h < \epsilon$. Hence, we have $P_E^{\xi}(C',\alpha) < \epsilon$. □

Theorem 4 does not guarantee that the proposed scheme is efficient for short-length codes. However, it gives some ideas how to design

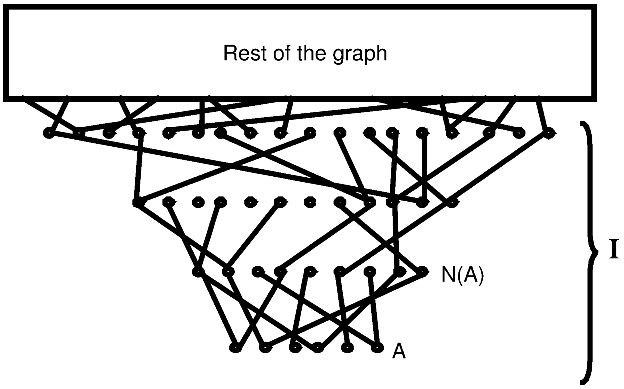

Fig. 3. Illustration of the subgraph I.

short-length codes. Our simulations suggest that the scheme also results in good performance for short-length codes.

C. Decoding of Highly Protected Bits

As we mentioned previously, it is desirable in some applications that we can decode the important bits without having to decode the entire received word. This is particularly interesting in network applications where the header has to be protected more than the rest of the bits and extracted in routers. Here, we show that this is possible using the proposed unequal error-protection LDPC (UELDPC) codes. The key point is that by the proof of Theorem 4, we conclude that only h iterations are sufficient to obtain a small enough error probability for the important bits. Note that we only need the messages sent to the important bits at the hth iteration (note that h is an even number). Thus, for the decoding we only need the subgraph I in Fig. 3.

To decode the important bits we perform the following procedure. First, the variable nodes in $N^h(A)$ send messages to the check nodes in $N^{h-1}(A)$. These messages are simply the LLRs of the variable nodes based on the observation of the channel. Then, the check nodes in $N^{h-1}(A)$ send messages to the variable nodes in $N^{h-2}(A)$. These messages are computed based on the messages from $N^{h-1}(A)$. We continue until the messages to $N^0(A) = A$ are computed. Thus, we need to compute $|E(I)|$ messages for decoding the important bits ($|E(I)|$ is the number of the edges of the graph I). Note that the number of messages that must be computed for decoding the entire block is equal to $|E| \times 2l$, where l is the total number of iterations in the message-passing algorithm and $|E|$ is the total number of edges in the Tanner graph of the code. Let T_ξ and T be the amount of time required for decoding the important bits and the whole block, respectively. Then we have

$$\frac{T_\xi}{T} = \frac{|E(I)|}{|E| \times 2l} \to 0, \qquad \text{as } n \to \infty. \tag{37}$$

In fact, $T_\xi = \Theta(\xi(n)) = o(n)$ but $T = \Theta(n)$. Therefore, we conclude that the important bits can be decoded in a much shorter time than the time required for decoding the entire block.

V. Practical Code Design and Simulation Results

A. Practical Code Design for Nonuniform Channels

Let us consider the problem of designing efficient LDPC codes from the ensemble $g(\Lambda, \rho)$ for the VHM systems. It is known that long LDPC codes can have performance close to the Shannon limit. The use of long LPDC codes is possible in the VHM systems because the whole memory page is read or stored simultaneously. Since BERs of less than 10^{-12} are desirable for the VHM systems, we require that the code do not present an error floor at least for BERs higher than 10^{-12}.

A stopping set S is defined in [23] as a subset of variable nodes such that all neighbors of S are connected to S at least twice. It is shown in [24] and [25] that if $\lambda_2\rho'(1) < 1$ (λ_2 is the fraction of the edges connected to the variable nodes of degree two), then the minimum distance and the size of the minimum stopping set in the expurgated ensemble increase linearly with respect to the code length. Here, a constant fraction of the codes in the ensemble with low minimum stopping set size are removed in the expurgation. On the other hand, if $\lambda_2\rho'(1) > 1$, these quantities are sublinear with high probability. Until now, all the discovered capacity-achieving sequences of LDPC codes over the BEC, satisfy $\lambda_2\rho'(1) > 1$ [26]. Therefore, for achieving the capacity, we should have a small minimum distance [24]. This implies that capacity-achieving codes have the error floor effect. In fact, capacity-approaching codes of practical lengths usually have an error floor at a BER of 10^{-7} or higher. On the other hand, if the minimum distance is linear, the error-floor effect is reduced substantially. Although we do not have a rigorous proof for this, simulations show the superiority of these codes in terms of the error-floor effect over the codes with sublinear minimum distance. Thus, in our designs we always use the expurgated ensembles with linear minimum distance and linear minimum stopping set size. Now, we discuss the code design for the nonuniform error correction. For simplicity, we first consider the BEC. The VHM systems in which we have BIAWGN channels will be discussed afterwards.

1) BEC: Consider the case that all the channels in Fig. 1 are BECs having different erasure probabilities. Let ϵ_j be the erasure probability of the jth channel. Here, we compare the performance of ordinary irregular LDPC codes and the codes from the ensemble $g(\Lambda, \rho)$ selecting each column of H with probability erased bits.

As an example, consider the case that the number of channels $k_r = 4$ and

$$\epsilon_1 = .1\kappa, \quad \epsilon_2 = .25\kappa, \quad \epsilon_3 = .5\kappa, \quad \epsilon_4 = .95\kappa \tag{38}$$

where κ is a constant. Suppose we use half-rate LDPC codes of length 10^4 in which 2500 bits are transmitted over each channel. Note that the whole system can be modeled as a BEC with the erasure probability $\epsilon = .45\kappa$. As a first approach, we consider the performance of the optimized half-rate LDPC codes for the erasure channel in [27]. We also design codes using the ensemble $g(\Lambda, \rho)$ as follows. In our design to alleviate the error floor problem we require that $\lambda_2\rho'(1) < 1$. For design simplicity, we choose the degree distribution to be semiregular. By a semiregular degree distribution we mean a degree distribution in which the variable nodes of the same type (variable nodes corresponding to bits that are transmitted through the same subchannel) have the same degree. We also require that the degree distribution of the check nodes be concentrated at two consecutive values. It is observed that this limitation does not result in considerable performance loss. However, it makes the optimization simpler [10] and [4]. We denote the ensemble of semiregular codes by $g(D, \rho)$ where $D = \{d_j : j = 1, \ldots, k_r\}$ and d_j is the degree of the variable nodes of type j. Thus, for the above example, a semiregular degree distribution consists of at most four distinct degrees for the variable nodes. It may sound that the semiregularity is too restrictive and the performance of the resulting codes would be much worse than the fully optimized codes. However, this is not the case. For the length $n = 10^4$, the best half-rate ordinary irregular code that we found in [27] has the following degree distribution:

$$\begin{aligned} &\lambda_1(x) = 0.2498x + 0.2472x^2 + 0.1480x^5 + 0.0033x^6 + 0.3517x^{19} \\ &\rho_1(x) = x^7. \end{aligned} \tag{39}$$

Let Code A be a randomly chosen code of length 10^4 from the ensemble defined by (λ_1, ρ_1). Note that the maximum variable-node degree is 20 for this code. We now design a semiregular code from the ensemble

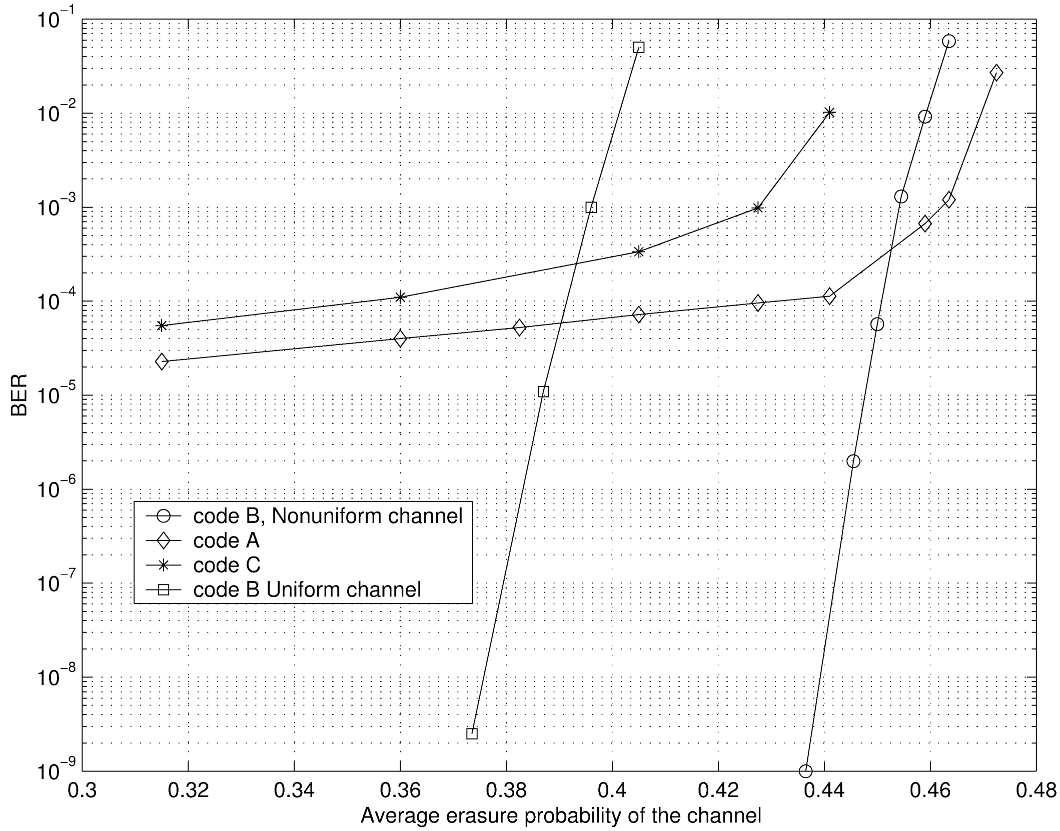

Fig. 4. Performance of different half-rate LDPC codes over the BEC.

$g(D, \rho)$. To simplify the design we restrict the maximum variable-node degree to be 7. Therefore, only a few choices are left. We can easily find the best possible code with the given constraints using density evolution. For example, we found the following degree distribution:

$$d_1 = 4, \quad d_2 = 7, \quad d_3 = 3, \quad d_4 = 2, \quad d_c = 8. \tag{40}$$

Let Code B be a randomly chosen code of length 10^4 from the ensemble that is defined by the above degree distribution. Since the maximum variable-node degree in Code B is 7, we also generated the best code (with respect to threshold) given in [27] with the maximum variable-node degree 7. Let Code C be a randomly chosen code of length 10^4 from this ensemble. Fig. 4 shows the performance of these codes. First, we note that both Codes A and C have an error floor higher than 10^{-5} while Code B does not have any error floor at least for BERs higher than 10^{-9}. Furthermore, for almost all practical purposes, Code B is the best among these codes. It is worth noting that the maximum variable-node degree of Code A is much higher than that of Code B. We also conclude that Code B has lower BERs than Code C for all values of ϵ, the average channel erasure probability. The performance of Code B over one single BEC is also shown in the figure. We observe that the performance of the code over the nonuniform channel (four parallel subchannels) is much better than its performance over the equivalent single channel. This verifies that we have utilized of the nonuniformity of the channel in the code design. Additionally, if it is desired, by slightly relaxing the constraints on the ensemble $g(D, \rho)$, we can get closer performance to the capacity.

Asymptotically, the probability of a small (logarithmic size) stopping set in the expurgated ensemble that we defined in above (Code B) goes to zero. Since we are concerned with the error floor, we need to be careful about variable nodes of degree two. In fact, any cycle whose variable nodes have degree two constructs a stopping set. Thus, if one of these cycles exists in our code, we just regenerate the code. Since the probability of having these small stopping sets is bounded away from one, it is very likely that we get a code with no small stopping set by a few trials. It is worth noting that these cycles can be found using simple graph algorithms [28]. Therefore, we avoid degree-two variable nodes in our design at Section V-A2. We will show that we can still find very good codes from the ensemble $g(D, \rho)$.

2) BIAWGN Channel: We now consider the VHM systems. As we mentioned before, each page of the VHM system can be considered as a set of parallel channels having different noise powers. As an example, we use the VHM system in [2]. In this system, we divide each page into four regions $(k_r = 4)$. The noise is assumed to be Gaussian. Therefore, the system can be modeled as a set of four BIAWGN channels. The relative SNRs of different regions are

$$\begin{aligned} \mathrm{SNR}_2 - \mathrm{SNR}_1 &= 1.61 \text{ dB} \\ \mathrm{SNR}_3 - \mathrm{SNR}_1 &= 2.80 \text{ dB} \\ \mathrm{SNR}_4 - \mathrm{SNR}_1 &= 3.74 \text{ dB}. \end{aligned} \tag{41}$$

We want to design a code of rate 0.85 from the ensemble $g(D, \rho)$. (In VHM systems codes with rates between 0.7 and 0.9 are typically used.) Since it is very important to prevent the error floor (because a BER of at least 10^{-12} is needed), we avoid degree-two variable nodes in the graph. We found the following degree distribution:

$$d_1 = 3, \quad d_2 = 4, \quad d_3 = 7, \quad d_4 = 10, \quad d_c = 40. \tag{42}$$

Fig. 5 shows the performance of this code for the block lengths of 10000 and 100000. As shown in the figure, at the BER of 10^{-9}, the distances from the capacity are only 0.65 and 1.04 dB for the lengths 10^5 and 10^4, respectively. Moreover, the codes do not present any error floor at a BER of 10^{-9}. Another interesting property that is verified

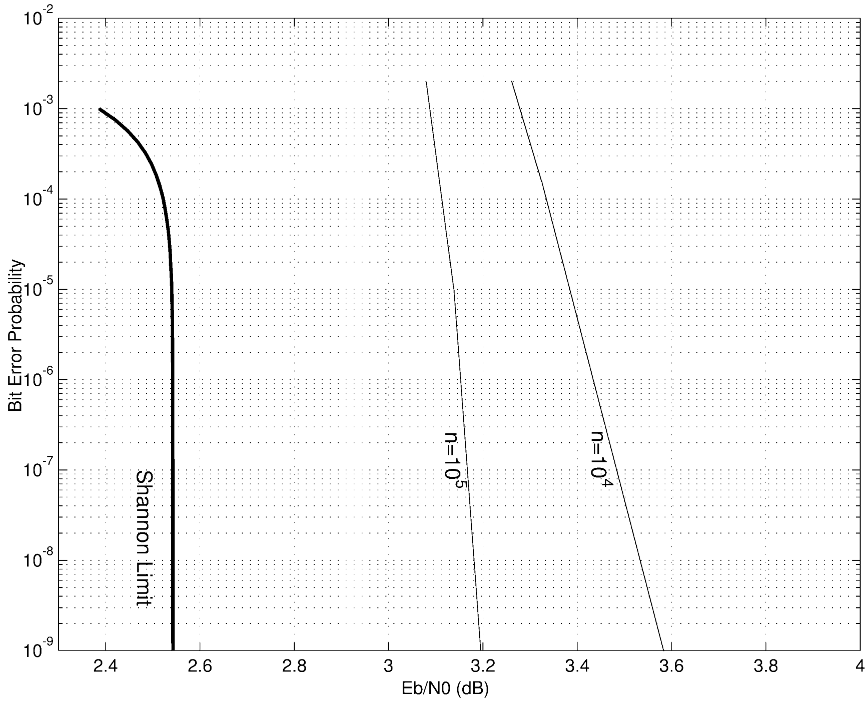

Fig. 5. Performance of the irregular LDPC code of rate 0.85 over four parallel BIAWGN channels.

by our experiments is that it is almost always possible to find very good codes from the ensemble $g(D, \rho)$ having small maximum variable-node degree when the number of channels is greater than three. This implies that we can avoid the high complexity degree distribution optimization. When the number of channels is two or three, a small relaxation on the restrictions is needed. For example, one possibility is to allow two distinct degrees for the variable nodes of each type.

B. Unequal Error Protection

In this subsection, we describe experiments to measure the performance of some codes from the ensemble $g_{n'}(\lambda, \rho, h, d_v, d_c)$. We showed in previous sections that the asymptotic performance of the UELDPC codes is good. Thus, here we concentrate on finite-length UELDPC codes and show that these codes have good performance even for short block lengths. To design UELDPC codes, we first need to choose h. Note that taking $h = 0$ results in an ordinary LDPC code from the ensemble $g_n(\lambda, \rho)$ in which we assign the high-degree variable nodes to the important bits. As we mentioned before, this is not an efficient approach. Our experiment shows that usually $h = 2$ results in good codes. For short-length codes (with lengths between 1000 and 5000), it is not suitable to choose h larger. We first consider the BEC. We designed half-rate UELDPC codes of lengths $n = 2000$ and $n = 4000$. The value of $\xi(n)$ was chosen 50 and 100 for $n = 2000$ and $n = 4000$, respectively. Let c' be either one of these codes and $g_{c'}$ be its corresponding graph. Let A be the set of important variable nodes. We chose the degree of the important bits as $d_v = 12$ and the degree of the vertices in $N(A)$ as $d_c = 5$. The degree of the vertices in $N^2(A)$ was one of the values $8, 3$, and 2. The degrees of all the other check nodes was 8. The degrees of the rest of the variable nodes were either two or three. To construct UELDPC codes we use a method similar to one described in [3]. We assign sockets to the vertices and construct the graph using a random permutation. The only difference with [3] is that the permutation that we use is a restricted random permutation to make sure that the vertices in $N^0(A) = A, N(A), N^2(A), \dots, N^h(A)$ take the desired degrees.

Fig. 6 shows the performance of the code c' when the length of the code is $n = 2000$. It also shows the performance of the regular $(3, 6)$ code of length 2000 which is the best regular LDPC code. We did not consider irregular codes because their performance is only slightly better than the regular codes for short lengths. Moreover, there is no efficient method to find good short-length irregular codes. In Fig. 6, by bad bits we mean the bits other than the important bits in the code c'. The figure shows both important bits and the bad bits have smaller BERs than the BER of the $(3, 6)$ regular code. In particular, the important bits have a much lower BER. Fig. 7 shows the same results when the lengths of the codes are $n = 4000$. We observe that the results are similar to the case of $n = 2000$.

For the BIAWGN channel, let us consider the code c' and compare its performance with the $(3, 6)$ regular code. Although we designed this code for the BEC, it is useful to evaluate its performance over the BIAWGN channel. Fig. 8 shows the performance of the code c' and the regular code for the length $n = 2000$. We notice that both bad bits and important bits of the code c' have better BERs than the regular code. Additionally, the BER of the important bits is considerably lower than the BER of the regular code. Since the $(3, 6)$regular code is considered to be a good code for length $n = 2000$, we conclude that the code c' is also a good UELDPC code for the BIAWGN channel.

Finally, we would like to emphasize that the assumption $\xi(n) \ll n$ is crucial for the above proposed scheme. For finite-length cases, if the ratio of the most important bits is comparable with the code length, it is still possible to design UELDPC codes by choosing higher degrees for these bits. However, it is an open question whether this code would be efficient compared with two separate LDPC codes.

VI. Other Applications

The proposed framework to design LDPC codes for the nonuniform error correction has several other applications. Multicarrier OFDM and multilevel coding are among these applications. The OFDM systems consist of several parallel channels in which some bits experience higher SNRs than others. Thus, we need to perform nonuniform error protection. In this case, we are not usually concerned about the error

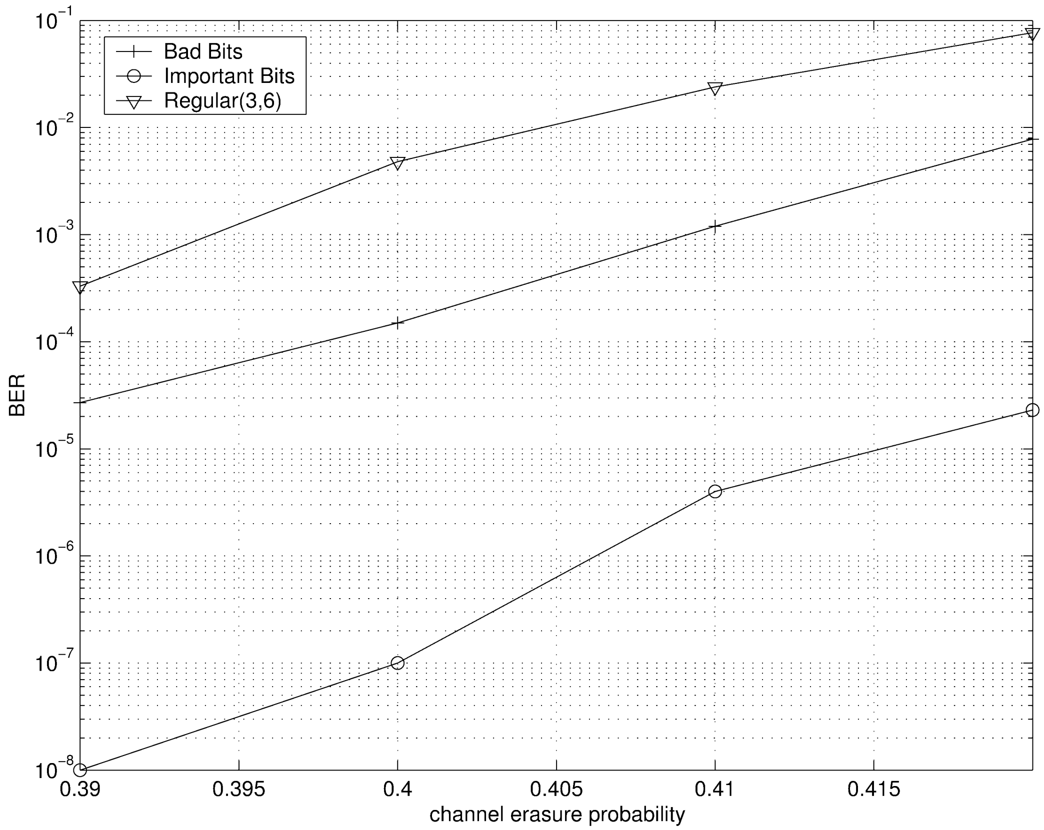

Fig. 6. Comparison between the performance of an UELDPC code of length 2000 and the regular $(3,6)$ code of the same length over the BEC.

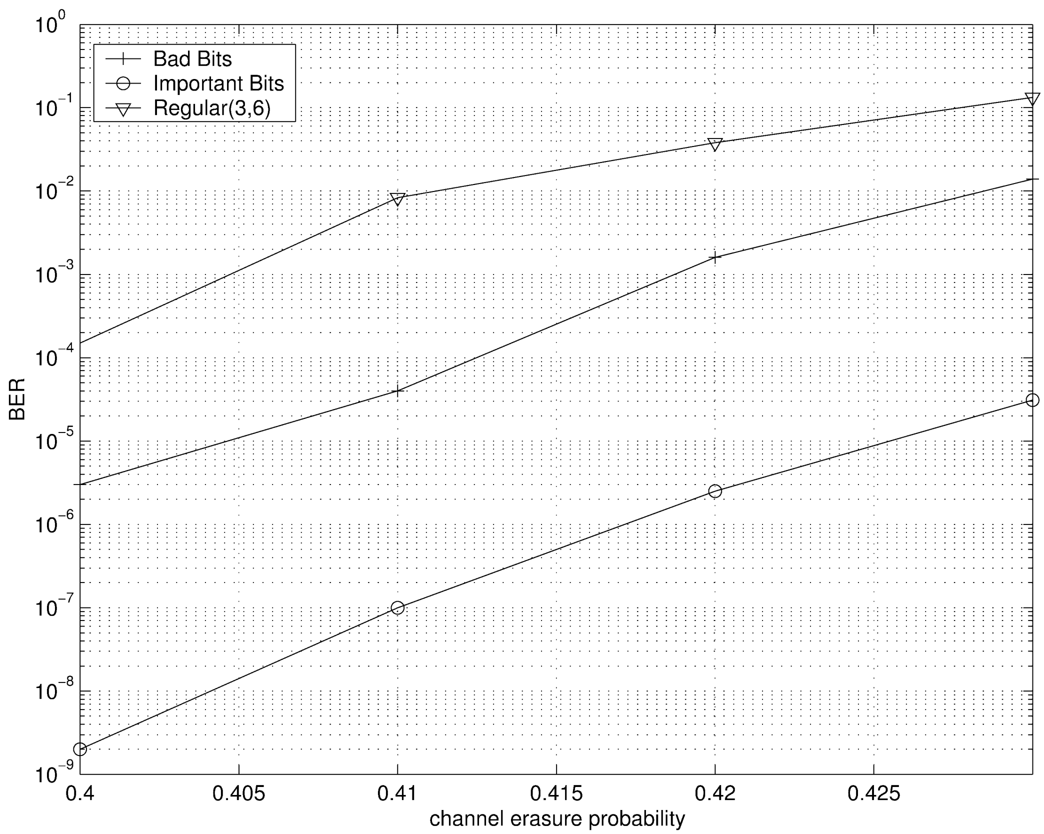

Fig. 7. Comparison between the performance of an UELDPC code of length 4000 and the regular $(3,6)$ code of the same length over the BEC.

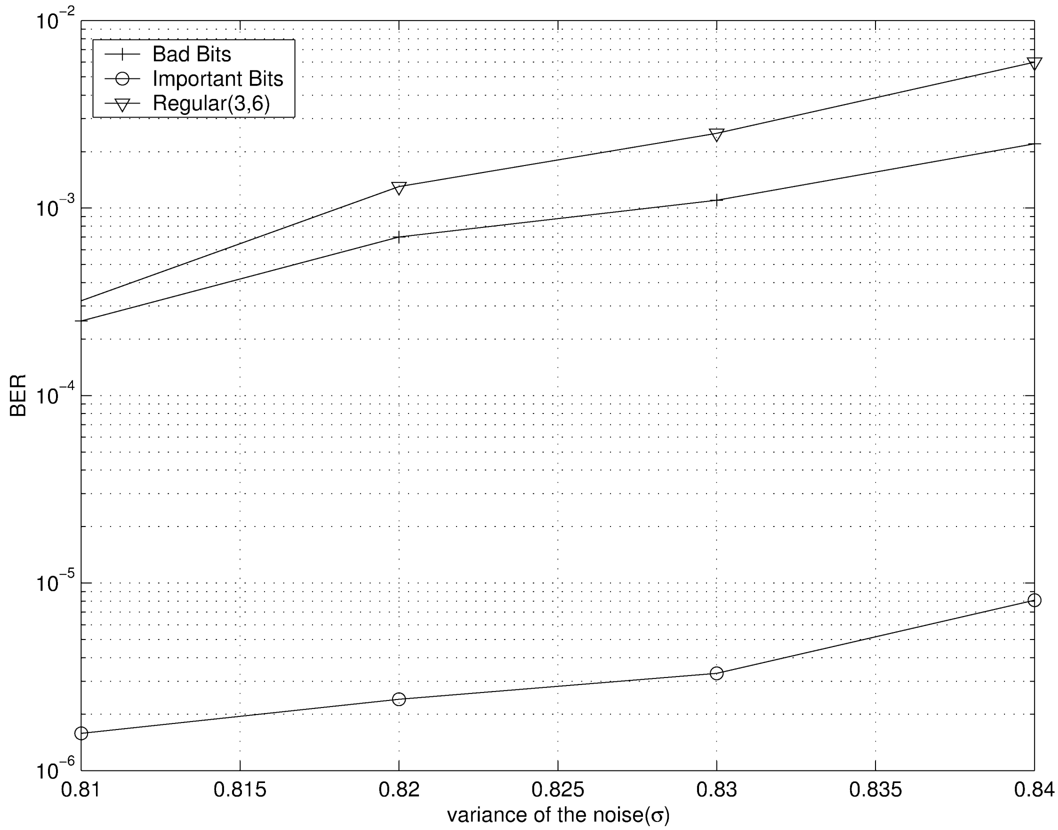

Fig. 8. Comparison between the performance of an UELDPC code of length 2000 and the regular $(3, 6)$ code of the same length over the BIAWGN channel.

floor problem. Instead we have other restrictions. Specifically, we may not be able to use long codes. It can be shown by simulations that for finite-length cases, the codes from the ensemble $g(\Lambda, \rho)$ may perform better than the conventional irregular codes over the parallel channels. A similar situation exists in multilevel coding.

VII. Conclusion

In this correspondence, we proposed a framework to design good LDPC codes over a set of parallel channels. This method is useful for many applications such as the volume holographic memories, OFDM, and rate-adaptive coding systems. We showed that the proposed method has several advantages over the conventional method. First, the design procedure is very simple since we do not need to perform the high-complexity degree optimization algorithms that are necessary for conventional LDPC codes. Second, using the proposed method, we can find codes that have near Shannon limit performance and have lower error floor. Third, for the applications that the code length cannot be large, the proposed codes can have better performance than the ordinary LDPC codes. The proposed framework can also be used to design LDPC codes in other applications such as OFDM systems and multilevel coding.

We also showed that the analysis and optimization of rate-compatible LDPC codes can be done as a special case of our analysis for parallel channels. The developed LDPC code employs single encoder and decoder for all combination of rates that are desired. As opposed to traditional rate-adaptive convolutional codes, we can generate any combination of rates very easily. We plan to expand on this research and find methods to design good rate-compatible LDPC codes.

Finally, we investigated unequal error protection using LDPC codes. In particular, we showed that good UELDPC codes exist for certain applications in which a small fraction of bits are highly protected. We proposed a technique to design these codes. We showed that these codes are asymptotically as good as any equal protecting LDPC codes. For short-length codes, simulations demonstrate that these codes outperform regular LDPC codes. Additionally, with the proposed scheme we can decode the important bits without having to decode the entire block. In ongoing research, we are exploring this issue further.

Appendix
Gaussian Approximation

If the subchannels in Fig. 1 are BIAWGN, it is possible to use a Gaussian approximation similar to [17]. This method is useful for designing codes for VHM systems and for finding optimal puncturing distributions over the Gaussian channels. Here, we give the Gaussian approximation formulas for the $g(\Lambda, \rho)$. We use the function ϕ which is defined in [17]. Let $m_u^{(l)}$ denote the mean of messages from the check nodes to variable nodes in the lth iteration. Let also $m_0^{(j)} = \frac{2}{\sigma_j^2}$ where σ_j is the variance of the noise in channel C_j in Fig. 1. Then we have

$$m_u^{(l)} = \sum_d \rho_d \phi^{-1}\left(1 - \left[1 - \sum_{j,i} q^{(j)} \lambda_i^{(j)} \phi(m_0^{(j)} + (i-1)m_u^{(l-1)})\right]^{(d-1)}\right) \quad (43)$$

where $q^{(j)} = \frac{|E^{(j)}|}{|E|}$. Similar to [17] we define

$$f_d(\underline{s}, t) = \phi^{-1}\left(1 - \left[1 - \sum_{j,i} q^{(j)} \lambda_i^{(j)} \phi(s^{(j)} + (i-1)t)\right]^{(d-1)}\right) \quad (44)$$

$$f(\underline{s}, t) = \sum_d \rho_d f_d(\underline{s}, t) \tag{45}$$

for $0 \leq t < \infty$. We can rewrite (43) as

$$t_l = f(\underline{s}, t_{l-1}) \tag{46}$$

where

$$s = (s^{(1)}, s^{(2)}, \ldots, s^{(k_r)}) = (m_0^{(1)}, m_0^{(2)}, \ldots, m_0^{(k_r)})$$

and $t_l = m_u^{(l)}$, and $t_0 = 0$. Similar to [17], one can show that $t_l(s)$ converges to infinity if and only if $t < f(\underline{s}, t)$ for all $t \in \mathbb{R}^+$. An equivalent formulation can be made by the following change of the variable:

$$r_l = \sum_{j,i} q^{(j)} \lambda_i^{(j)} \phi(s^{(j)} + (i-1)t_l). \tag{47}$$

We also define

$$h_i^{(j)}(s, r) = \phi\left(s + (i-1)\sum_d \rho_d \phi^{-1}([1-(1-r)^{(d-1)}])\right) \tag{48}$$

$$h(\underline{s}, r) = \sum_{j,i} \lambda_i^{(j)} q^{(j)} h_i^{(j)}(s^{(j)}, r). \tag{49}$$

Then, we have

$$r_l = h(\underline{s}, r_{l-1}) \tag{50}$$

where

$$s = (s^{(1)}, s^{(2)}, \ldots, s^{(k_r)}) = (m_0^{(1)}, m_0^{(2)}, \ldots, m_0^{(k_r)})$$

and $r_0 = \sum_j q^{(j)} \phi(s^{(j)})$. Again $r_l(s) \longrightarrow 0$ if and only if $r > h(s, r)$ for all $r \in (0, r_0)$. It is easy to show that $r_l(s) \longrightarrow 0$ if and only if $r > h(s, r)$ for all $r \in (0, 1)$. This fact is useful when we use linear programming for the optimization of the degree distribution or puncturing pattern.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers who helped to improve the content of the work. They are also thankful to Ralf Koetter, the Associate Editor, for handling the review.

REFERENCES

[1] H. Pishro-Nik, N. Rahnavard, and F. Fekri, "Nonuniform error correction using low-density parity check codes," in *Proc. 40th Annu. Allerton Conf.*, Monticello, IL, Oct. 2002.

[2] H. Pishro-Nik, N. Rahnavard, J. Ha, F. Fekri, and A. Adibi, "Low-density parity-check codes for volume holographic memory systems," *Appl. Opt.*, vol. 42, pp. 861–870, 2003.

[3] 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.

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

[5] K. Nakamura, Y. Kabashima, D. Saad, and R. M. Zaragoza, "Statistical mechanics of broadcast channels using low-density parity-check codes," in *Proc. 2003 IEEE Int. Symp. Information Theory*, Yokohama, Japan, Jun./Jul. 2003, p. 294.

[6] J. Hagenauer, "Rate-compatible punctured convolutional codes (rcpc codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.

[7] ——, "Rate-compatible punctured turbo (rcpt) codes in a hybrid fec/arq system," in *Proc. Communications Theory Mini-Conf. of IEEE GLOBECOM '97*, Phoenix, AZ, Nov. 1997, pp. 55–59.

[8] P. Jung and J. Plechinger, "Performance of rate-compatible punctured turbob codes for mobile radio applications," *IEE Electron. Lett.*, vol. 33, pp. 389–400, 1997.

[9] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[10] S. Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, MIT, Cambridge, MA, 2000.

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

[12] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 4, pp. 600–607, Oct. 1967.

[13] D. Mandelbaum, "Unequal-error-protection codes dervied from difference sets," *IEEE Trans. Inf. Theory*, vol. IT-18, pp. 686–687, Sep. 1972.

[14] C. C. Kilgus and W. C. Gore, "A class of cyclic unequal error protection codes," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 5, pp. 687–690, Sep. 1972.

[15] I. Boyarinov and G. Katsman, "Linear unequal error protection code," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 1, pp. 168–175, Jan. 1981.

[16] W. Chou and M. A. Neifeld, "Interleaving and error correction in volume holographic memory systems," *Appl. Opt.*, vol. 37, pp. 6951–6968, 1998.

[17] 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. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[18] H. Pishro-Nik and F. Fekri, "On decoding of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 50, no. 3, pp. 439–454, Mar. 2004.

[19] M. A. Shokrollahi, *Codes, Systems, and Graphical Models*. New York: Springer-Verlag, 2001, ch. Capacity-achieving sequences, pp. 153–166.

[20] J. Justesen, "Coding for frames," in *Proc. 40th Annual Allerton Conference*, Monticello, IL, Oct. 2002.

[21] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[22] ——, "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[23] C. Di, D. Proietti, İ. E. Telatar, T. Richardson, and R. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.

[24] C. Di, T. Richardson, and R. Urbanke, "Weight distributions: How deviant can you be?," in *Proc. 2001 IEEE Int. Symp. Information Theory*, Washington, DC, 2001, p. 50.

[25] T. Richardson, A. Shokrollahi, and R. Urbanke, "Finite-length analysis of low-density parity-check ensembles for the binary erasure channel," in *Proc. 2002 IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, 2002, p. 1.

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

[27] R. Urbanke. Report. [Online]. Available: http://lthcwww.epfl.ch/research/ldpcopt/

[28] T. H. Cormen, C. E. Leiserson, and C. S. R. L. Rivest, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.