Rateless Codes With Unequal Error Protection Property

Nazanin Rahnavard, *Member, IEEE*,
Badri N. Vellambi, *Student Member, IEEE*, and
Faramarz Fekri, *Senior Member, IEEE*

***Abstract*—In this correspondence, a generalization of rateless codes is proposed. The proposed codes provide unequal error protection (UEP). The asymptotic properties of these codes under the iterative decoding are investigated. Moreover, upper and lower bounds on maximum-likelihood (ML) decoding error probabilities of finite-length LT and Raptor codes for both equal and unequal error protection schemes are derived. Further, our work is verified with simulations. Simulation results indicate that the proposed codes provide desirable UEP. We also note that the UEP property does not impose a considerable drawback on the overall performance of the codes. Moreover, we discuss that the proposed codes can provide unequal recovery time (URT). This means that given a target bit error rate, different parts of information bits can be decoded after receiving different amounts of encoded bits. This implies that the information bits can be recovered in a progressive manner. This URT property may be used for sequential data recovery in video/audio streaming.**

***Index Terms*—Asymptotic analysis, finite-length analysis, iterative decoding, lossy channels, maximum-likelihood decoding, rateless codes, unequal error protection.**

I. Introduction

Recently, a new class of error-control codes called rateless (Fountain) codes has been invented. LT codes [1], Raptor codes [2], and Online codes [3] are examples of such codes. It has been shown that these codes have very simple encoding and decoding algorithms. Asymptotically good degree distributions for them were also developed [2], [3]. Rateless codes on lossy channels do not assume any knowledge of the channel. Therefore, rateless codes are very suitable candidates for applications such as transmitting data on lossy multicast channels, nonuniform channels, and time-varying channels. In some of these applications, we may not have an estimate of the channel erasure probability at all times. In some others, different users may receive data that is passed through different channels. Traditional codes cannot be optimal for such cases because of the unknown or varying characteristics of the channels. In particular, rateless codes can fit well for the Internet application in which channels are modeled as *binary erasure channels* (BEC) with unknown and time-varying erasure probabilities. Rateless erasure codes have the potential of replacing *transmission control protocol* (TCP), which is based on the *automatic repeat request* (ARQ) [4].

In all previous studies on rateless codes, equal error protection (EEP) of all data was considered. The EEP property would be sufficient for applications such as multicasting bulk data (e.g., a software file) [5]. However, in several applications, a portion of data may need more protection than the rest of data. For example, in an MPEG stream [6], I-frames need more protection than P-frames. In some other applications, a portion of data may need to be recovered prior to the other parts. An example would be video-on-demand systems, in which the stream should be reconstructed in sequence [4], [7]. Such applications raise a need for having codes with *unequal error protection* (UEP) or *unequal recovery time* (URT).

UEP codes were first studied in [8]. Since then, there has been considerable work in this area, e.g., [9], [10]. Recently, different UEP codes have been designed with LDPC codes, e.g., [11]–[13]. For the applications similar to the ones we described above, designing rateless codes with unequal error protection property (UEP-rateless codes) is of great interest. In this work, we develop, for the first time, rateless codes that can provide UEP. This implies that some portion of data would be protected more than the other parts. Theoretical and simulation results illustrate that a strong UEP can be achieved by the proposed rateless codes. These codes can also be employed in applications for which URT is desirable, i.e., the number of received symbols for recovering more important parts is less than that number for recovering less important parts. In our design and analysis, we consider both asymptotic and finite-length cases. Preliminary results were initially introduced in [14], [15].

The paper is organized as follows. In Section II, a review of LT codes is given. Section III studies design and *asymptotic* analysis of UEP-rateless codes under the *iterative decoding*. Section IV investigates design and analysis of *finite-length* UEP-rateless codes when the *maximum-likelihood* (ML) decoding is considered. Finally, we conclude the paper in Section V.

Throughout the paper, we assume the following terminologies. In a graph $G(V, E)$, where V is the set of vertices (nodes) and E is the set of edges, two vertices u and v are *adjacent* or *neighbor* if there is an edge $e = (u, v) \in E$ with ends u and v. Two edges e_1 and e_2 are *adjacent* if they share an end. A vertex v and edge e are *incident* if v is an end of e. The *degree* of a vertex v is defined as the number of edges of G incident to v. We call $G'(V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. Moreover, G' is a subgraph of G *induced by* V' if G' contains all the edges $(u, v) \in E$ with $u, v \in V'$.

II. Review of LT Codes

In this section, we briefly review LT codes introduced by Luby [1]. Suppose we want to transmit a message comprising of n input symbols. Let $\Omega_1, \ldots, \Omega_n$ be a probability distribution on $\{1, \ldots, n\}$ such that Ω_i denotes the probability that the value i is chosen. We may also denote this distribution by its generator polynomial $\Omega(x) = \sum_{i=1}^{n} \Omega_i x^i$. An encoding (output) symbol is formed as follows:

- randomly choose a degree d according to the distribution $\Omega_1, \ldots, \Omega_n$;
- choose uniformly at random d input symbols;
- perform bitwise XOR operations on the selected d input symbols to form the output symbol.

The output symbol is then transmitted. We repeat this process until a sufficient number of output symbols is obtained at the receiver. In general, the number of output symbols required to give a high probability of decoding n input symbols can be expressed as γn for a fraction $\gamma \gtrsim 1$ (γ is called the rateless overhead). The process of decoding LT codes relies on finding an output symbol such that the value of all but one of its neighbor input symbols is known. The value of the unknown input symbol is computed by simple bitwise XOR operations. This step is repeated until no more of such output symbols can be found. Obviously, the degree and the set of neighbors of an output symbol must be provided to the decoder. There are several methods to accomplish this. One method is that the encoder and decoder are synchronized and share a common random generator. The reader may refer to [1] for more details. Without loss of generality and for simplicity, throughout this paper we may assume that the symbols are binary symbols. Following

Manuscript received March 16, 2005; revised September 27, 2006. This material is based upon work supported by the National Science Foundation under Grant CCF-0430964. The material in this correspondence was presented in part at the IEEE Globecom, St. Louis, MO, November 2005, and the IEEE International Symposium on Information Theory, Seattle, WA, July 2006

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

Communicated by Ø. Ytrehus, Associate Editor for Coding Techniques.
Digital Object Identifier 10.1109/TIT.2007.892814

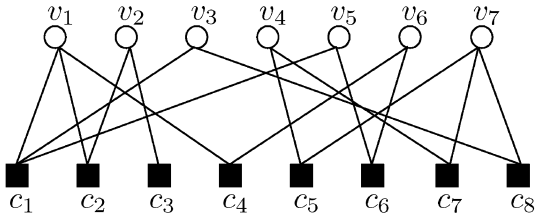

Fig. 1. An example of an LT code, where $n = 7$ and $\gamma = 8/7$. The circular and rectangular nodes correspond to input and output symbols, respectively.

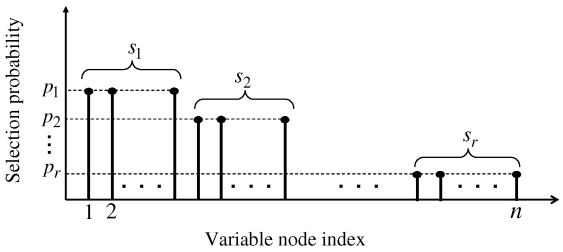

Fig. 2. Nonuniform probability distribution function for selecting a variable node (an input symbol) by an edge.

[3] and [16] we may view the input and output symbols as vertices of a bipartite graph G, where the input symbols are the variable nodes and the output symbols are the check nodes. Fig. 1 depicts a small example of an LT code, where $n = 7$ and $\gamma = 8/7$. Circular nodes correspond to the input symbols, and the rectangular nodes correspond to the output symbols. The values of the output symbols are known at the receiver, and the goal is to find the values of the input symbols. The decoding starts by copying the value of c_3 to its unique neighbor v_2. Next, since c_2 has only one unknown neighbor, it recovers the value of v_1. The next output symbol with only one unknown neighbor is c_4 and recovers v_6. The decoding continues until no output symbol with exactly one unknown neighbor exists. In this example, the decoding is successful since the values of all the input symbols are determined.

Clearly, in the mentioned encoding scheme all the input nodes have the same probability of being selected in forming each output node. Consequently, the code provides EEP for all data. In this paper, we investigate a more general case in which the neighbors of a check node are selected nonuniformly at random. As we will see later, these generalized codes have interesting properties. They are specifically useful when UEP or URT of information symbols is needed. Next, we describe the proposed codes.

III. Asymptotic Analysis of UEP-Rateless Codes

Let $\Omega(x) = \sum_{i=1}^{n} \Omega_i x^i$ be the generator polynomial corresponding to the probability distribution of the degrees of check nodes in an LT code. In our proposed scheme, the neighbors of a check node are selected nonuniformly at random. Let us partition the n variable nodes into r sets $s_1, s_2, \ldots, s_r$ of sizes $\alpha_1 n, \alpha_2 n, \ldots, \alpha_r n$ such that $\sum_{j=1}^{r} \alpha_j = 1$. Let $p_j(n)$[1] be the probability that an edge is connected to a particular variable node in s_j, for $j = 1, \ldots, r$ (see Fig. 2).

Clearly, we have $\sum_{i=1}^{r} p_i(n)\alpha_i n = 1$. The proposed ensemble at the receiver is specified by parameters $\Omega(x), n, \gamma$, and $P(x, z)$, in which $P(x, z) = \sum_{i=1}^{r}(\alpha_i x^i + p_i z^i)$. The average check-node degree is given by $\mu = \sum_{i=1}^{n} i\Omega_i = \Omega'(1)$, where $\Omega'(x)$ is the derivative of $\Omega(x)$ with respect to x. Moreover, it is straightforward to show that the degree of variable nodes in s_j has a binomial distribution, for $j = 1, 2, \ldots, r$. Specifically, the probability $\lambda_{d,j}$ that a variable node in s_j has a degree d is given by

$$\lambda_{d,j} = \binom{\mu\gamma n}{d} p_j^d (1 - p_j)^{\mu\gamma n - d}. \tag{1}$$

Asymptotically (as n goes to infinity), we can approximate distribution (1) by a Poisson distribution if the following two conditions are satisfied for $j = 1, \ldots, r$:

$C_1 : p_j(n) = o(1)$;

$C_2 : \mu\gamma n p_j = \theta_j$ is a constant.

Satisfying these conditions, $\lambda_{d,j}$ approaches to

$$\frac{e^{-\theta_j}(\theta_j)^d}{d!} \tag{2}$$

which is a Poisson distribution with mean θ_j.

Throughout the paper we assume conditions C_1 and C_2 are satisfied. For example, we can have $p_j(n) = \frac{k_j}{n}$, for some nonnegative constants k_j that satisfy $\sum_{j=1}^{r} \alpha_j k_j = 1$. Accordingly, C_2 reduces to $\mu\gamma$ has to be a constant. This condition can be easily addressed if we consider both μ and γ as constants. Assuming that μ is a constant results in constants average variable-node and check-node degrees. This is desirable since the resulting graph will be a tree as $n \to \infty$ [3], and the encoding complexity will be linear in n.

To investigate the recovery probability of an input symbol in a generalized rateless code, we use a technique called And-Or tree analysis ([17] and [3]). Next, we describe this technique and will generalize it to fit our problem. Then, we will see how And-Or tree analysis and recovery probability of input nodes in rateless coding are related.

A. And-Or Tree Analysis Technique

An And-Or tree T_l is defined as following. Let T_l be a tree of depth $2l$. The root of the tree is at depth 0, its children are at depth 1, their children at depth 2, and so forth. Each node at depth $0, 2, 4, \ldots, 2l - 2$ is called an *OR-node* (that evaluates logical OR operation on the value of its children), and each node at depth $1, 3, 5, \ldots, 2l - 1$ is called an *AND-node* (that evaluates logical AND operation on the value of its children). Suppose that each OR-node independently chooses to have i children with probability δ_i, where $\sum_i \delta_i = 1$. Similarly, each AND-node chooses to have i children with probability β_i, where $\sum_i \beta_i = 1$. Each node at depth $2l$ is assigned a value 0 or 1 independently, with y_0 being the probability that it is 0. Also OR-nodes with no children are assumed to have a value 0, whereas AND-nodes with no children are assumed to have a value 1. We are interested in finding y_l, the probability that the root node evaluates to 0, if we treat the tree as a Boolean circuit.

The following lemma from [17], which is called the And-Or tree lemma, formulates y_l. The proof is straightforward, considering that the OR-nodes at depth 2 in T_l are the roots for independent And-Or trees T_{l-1}. Therefore, y_l can be computed as a function of y_{l-1}, the probability that the root of an And-Or tree T_{l-1} evaluates to 0.

Lemma 1: The probability y_l that the root node of an And-Or tree T_l evaluates to 0 is $y_l = f(y_{l-1})$, where y_{l-1} is the probability that the root node of an And-Or tree T_{l-1} evaluates to 0, and

$$f(x) = \delta(1 - \beta(1 - x))$$
$$\delta(x) = \sum_i \delta_i x^i, \quad \text{and } \beta(x) = \sum_i \beta_i x^i. \tag{3}$$

Next, we generalize the And-Or tree construction to the case that OR-nodes are unlike each other. Specifically, suppose we have r different types of OR-nodes: Type 1, Type 2,..., Type r. Number of OR-nodes of each type is sufficiently large. Suppose the root of the generalized And-Or tree $GT_{l,j}$ is an OR-node of Type j, and the

[1]The special case $p_1 = \ldots = p_r = \frac{1}{n}$, results in the previously studied EEP-rateless codes.

depth of the tree is $2l$. We construct $GT_{l,j}$ similar to T_l except that each OR-node of Type k chooses to have i children with probability $\delta_{i,k}$, for $k = 1, \ldots, r$. Each AND-node, as before, chooses to have i children with probability β_i. However, each child of an AND-node independently will be an OR-node of Type k with probability q_k. Each node of Type k at depth $2l$, is assigned a value 0 or 1 independently, with $y_{0,k}$ being the probability that it is 0. Also, OR-nodes with no children are assumed to have a value of 0, whereas AND-nodes with no children are assumed to have a value of 1. We are interested in finding $y_{l,j}$, the probability that the root node evaluates to 0, if we treat the tree as a boolean circuit. Lemma 2 formulates $y_{l,j}$.

Lemma 2: Let $y_{l,j}$ be the probability that the root of an And-Or tree $GT_{l,j}$ evaluates to 0. Then

$$y_{l,j} = \delta_j \left(1 - \beta \left(1 - \sum_{k=1}^{r} q_k \, y_{l-1,k}\right)\right) \tag{4}$$

in which $\delta_j(x) = \sum_i \delta_{i,j} x^i$ and $\beta(x) = \sum_i \beta_i x^i$. The proof is straightforward and is similar to the proof of Lemma 1. The relation between the above analysis and the error probabilities for the generalized rateless code is given in the following section.

B. Analysis of the Generalized Rateless Codes

In this section, we examine the generalized rateless codes under iterative decoding. Let G denote the bipartite graph corresponding to the code at the receiver. Following [2] and [3], we can rephrase the belief propagation decoding algorithm for our analysis as following. At every iteration of the algorithm messages (0 or 1) are sent along the edges from check nodes to variable nodes, and then from variable nodes to check nodes. A variable node sends 0 to an adjacent check node if and only if its value is not recovered yet. Similarly, a check node sends 0 to an adjacent variable node if and only if it is not able to recover the value of the variable node. In other words, a variable node sends 1 to a neighboring check node if and only if it has received at least one message with value 1 from its other neighboring check nodes. Also, a check node sends 0 to a neighboring variable node if only if it has received at least one message with value 0 from its other neighboring variable nodes. Therefore, we see that variable nodes indeed do the logical OR operation and the check nodes do the logical AND operation. We can use the results of Lemma 2 on a subgraph G_l of G to find the probability that a variable node is not recovered after l decoding iterations (its value evaluates to zero). We choose G_l as following. Choose an edge (v, w) uniformly at random from all edges. Call the variable node v the root of G_l. Subgraph G_l is the graph induced by v and all neighbors of v within distance $2l$ after removing the edge (v, w). We can see G_l is a tree asymptotically [17]. We can map each check node to an AND-node and each variable node in s_j to an OR-node of Type j. We only need to compute the probabilities $\beta_i, \delta_{i,j}$, and q_k. We have β_i is the probability that a randomly chosen edge is connected to a check node with i children. This is the probability that the edge is connected to a check node of degree $i+1$. Therefore, we have $\beta_i = \frac{(i+1)\Omega_{i+1}}{\Omega'(1)}$ and consequently $\beta(x) = \frac{\Omega'(x)}{\Omega'(1)}$. Similarly, we have $\delta_{i,j}$ is the probability that the variable node connected to a randomly selected edge has degree $i+1$ given that the variable node belongs to s_j. It can be shown easily that $\delta_{i,j} = \frac{(i+1)\lambda_{i+1,j}}{p_j \mu \gamma n}$, in which $\lambda_{i+1,j}$ is computed from (2). After substitution, we have $\delta_j(x) = e^{n p_j \mu \gamma (x-1)}$. Additionally, we have $q_k = p_k \alpha_k n$. We summarize our results in the following lemma.

Lemma 3: Consider a generalized rateless code with parameters $\Omega(x), P(x, z), n$, and γ. Let $y_{l,j}$ be the probability that a variable node in s_j is not recovered after l decoding iterations. For $j = 1, \ldots, r$ we have

$$y_{0,j} = 1$$

and

$$y_{l,j} = \delta_j \left(1 - \beta \left(1 - \sum_{k=1}^{r} p_k \alpha_k n \, y_{l-1,k}\right)\right), \quad l \geq 1 \tag{5}$$

in which

$$\beta(x) = \Omega'(x)/\Omega'(1)$$

and

$$\delta_j(x) = e^{n p_j \mu \gamma (x-1)}$$

with $\mu = \Omega'(1)$.

Next, we prove a few lemmas that mostly represent the properties of the proposed codes.

Lemma 4: $y_{l,j}$ is a decreasing function of the number of iterations l.

Proof: We prove this by induction. We have $y_{1,j} = e^{-n p_j \gamma \Omega_1} < y_{0,j}$. Now suppose $y_{l,j} < y_{l-1,j}$ for $j = 1, \ldots, r$. We need to show $y_{l+1,j} < y_{l,j}$. This can be shown easily using the fact that $\beta(\cdot)$ and $\delta_j(\cdot)$ are both increasing functions of their argument. □

From Lemma 4, $\{y_{l,j}\}_l$ is a monotone decreasing sequence. Moreover, $\{y_{l,j}\}_l$ is a bounded sequence since we have $y_{l,j} \in [0, 1]$ for $l \geq 0$. From the monotone convergence theorem [18], we conclude that $\{y_{l,j}\}_l$ is a convergent sequence that converges to a fixed point in $[0, 1]$.

The following lemma can be proved similar to Lemma 4.

Lemma 5: $y_{l,j}$ decreases when γ increases (more check nodes are collected).

Definition 1: Define $G_{l,i,j} \triangleq \frac{y_{l,i}}{y_{l,j}}$. This parameter compares the recovery probabilities of nodes in s_i and s_j. The larger the value of $G_{l,i,j}$, the higher the recovery probability of the nodes in s_j in comparison with the nodes in s_i.

It can be shown that $G_{l,i,j} = e^{n(p_j - p_i)\mu\gamma\beta(1-\sum_{k=1}^{r} p_k \alpha_k n \, y_{l-1,k})}$. Therefore, we have:

Lemma 6: For $l \geq 1, G_{l,i,j} > 1$ if and only if $p_j > p_i$.

Lemma 7: Consider two sets s_i and s_j. Suppose that $p_j > p_i$. Then, $G_{l,i,j}$ is an increasing function of the number of iterations l and the overhead γ.

Proof: First we need to show that $G_{l+1,i,j} > G_{l,i,j}$. This can be shown easily using Lemma 4 and the fact that $\beta(\cdot)$ is an increasing function of its argument. The second part is concluded using Lemma 5. □

From Lemmas 6 and 7, we conclude the following. To increase the recovery probability of nodes in a set, we need to increase the selection probability of the nodes in that set. Moreover, if two nodes in different sets have different selection probabilities, the difference between their recovery probabilities increases by receiving more check nodes or by increasing the number of iterations in the iterative decoding algorithm.

C. A Special Case: $r = 2$

In this section, a special case of the generalized rateless codes with parameters $\Omega(x), P(x, z), n, \gamma$, and $r = 2$ is investigated.

Assume we have two levels of importance on n information bits. Assume $n_1 = \alpha n$ $(0 < \alpha < 1)$ is the number of *more important bits* (MIB), which reside in the first part of the information, and $n_2 = (1-\alpha)n$ is the number of *less important bits* (LIB). To ensure lower average BER's for MIB than LIB, the probability of selecting MIB has to be more than that of LIB by Lemma 6. We set $p_1 = \frac{k_M}{n}$ and $p_2 = \frac{k_L}{n}$ for some $0 < k_L < 1$ and $k_M = \frac{1-(1-\alpha)k_L}{\alpha}$. Let $y_{l,M}$ and $y_{l,L}$ denote the error probabilities of MIB and LIB at the lth decoding iteration, respectively. From Lemma 3, we conclude that

$$y_{l,M} = e^{-k_M \mu \gamma \beta(1-(1-\alpha)k_L y_{l-1,L} - \alpha k_M y_{l-1,M})} \tag{6}$$

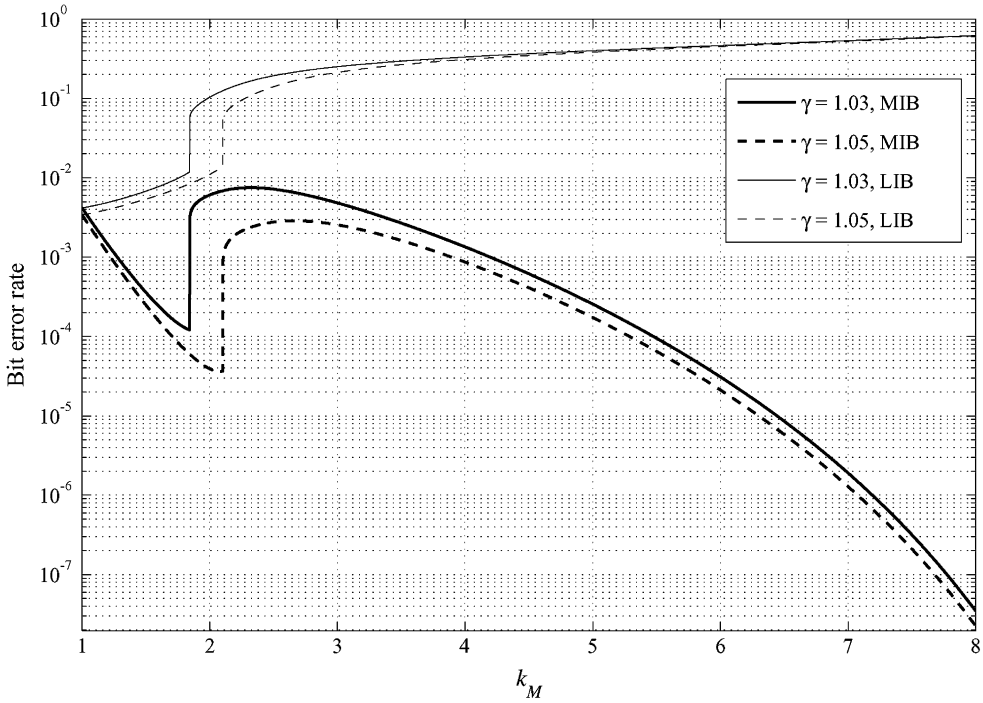

Fig. 3. Asymptotic analysis of bit error rates versus k_M for the UEP-rateless code with parameters $\Omega_1(x), n$, and $P(x,z) = 0.1x + \frac{k_M}{n}z + 0.9x^2 + \frac{k_L}{n}z^2$.

and

$$y_{l,L} = e^{-k_L \mu \gamma \beta (1-(1-\alpha)k_L y_{l-1,L} - \alpha k_M y_{l-1,M})} \tag{7}$$

with $\beta(x) = \Omega'(x)/\Omega'(1)$, $\mu = \Omega'(1)$, and $y_{0,L} = y_{0,M} = 1$.

The sequences $\{y_{l,M}\}_l$ and $\{y_{l,L}\}_l$ are convergent by Lemma 4. Let us call the corresponding fixed points as y_L and y_M, respectively. It can be shown that $\frac{\partial y_M}{\partial k_M}|_{k_M=1} = -\varphi$ and $\frac{\partial y_L}{\partial k_M}|_{k_M=1} = \varphi \frac{\alpha}{1-\alpha}$, where $\varphi = -y \ln y > 0$. Here, y is the bit error probability when uniform selection ($k_M = 1$) is done and satisfies $y = e^{-\gamma \Omega'(1-y)}$. These results express the variations of the bit error rates when k_M is slightly greater than one. We note that y_M decreases but y_L increases. However, for $0 < \alpha < \frac{1}{2}$, the decreasing slope of y_M is $\frac{1-\alpha}{\alpha}$ times greater than the increasing slope of y_L.

Example: In this example, we consider the degree distribution proposed in [2]

$$\begin{aligned}\Omega_1(x) = {} & 0.007969x + 0.493570x^2 \\ & + 0.166220x^3 + 0.072646x^4 \\ & + 0.082558x^5 + 0.056058x^8 + 0.037229x^9 \\ & + 0.055590x^{19} + 0.025023x^{64} + 0.003135x^{66}.\end{aligned}$$

Fig. 3 shows y_L and y_M versus k_M for $\alpha = 0.1$. We considered two overheads $\gamma = 1.03$ and $\gamma = 1.05$. As an example, consider the case that $\gamma = 1.05$. Uniform selection ($k_M = 1$) results in the BER of 3.4×10^{-3} for all data whereas $y_M = 5 \times 10^{-5}$ and $y_L = 9 \times 10^{-3}$ when $k_M = 1.9$. This shows that the BER of MIB has improved substantially (about two orders of magnitude) at the cost of a slight performance loss on the LIB.

Fig. 4 compares the average BER and the BER of MIB with the BER of the EEP-code for $\gamma = 1.05$. For example, for $k_M = 2$, the average performance of the UEP code is tripled. However, the performance of MIB is 87 times better than the case of EEP.

Fig. 5 depicts the BERs of MIB and LIB versus the overhead γ for $k_M = 2$. We have also included the BERs for the EEP code. Interestingly, nonuniform selection reduces BERs of both MIB and LIB for small overheads. For large overheads, the BER of MIB improves significantly while in return the performance of LIB slightly degrades.

It should be mentioned that we can also interpret the UEP as the URT. This means that given a target bit error rate, different parts of information bits can be decoded after receiving different amounts of encoded bits. In other words, the BER of MIB reaches a target BER faster (smaller overhead) than the BER of LIB (see Fig. 5).

D. Simulation Results on Iterative Decoding of A Moderate-Length UEP-Rateless Code

Here, we give simulation results for the case that the number of information bits is $n = 2000$. We considered two cases, an EEP code and a UEP code with $k_M = 2$ and $\alpha = 0.1$. We considered $\Omega_1(x)$ in both cases. Fig. 6 shows the bit error rates after performing LT decoding. We notice that the performance of MIB improves substantially in the UEP case. Even LIB has better performance than the case of EEP for small overheads. We conclude that for small overheads, UEP is provided while the overall performance of the UEP code is better than that of the EEP code. Fig. 6 also depicts a large gap between the BER's of MIB and LIB. For example, the BER of MIB is about two orders of magnitude better than that of LIB when $\gamma = 1.3$. This gap increases monotonically with the overhead.

Next, let us consider the URT problem. In URT, the BER of MIB reaches a target BER faster (smaller overhead) than the BER of LIB. For example in Fig. 6, we need to collect $1.16n = 2320$ output symbols to have $\mathrm{BER} = 10^{-3}$ for MIB. However, $1.33n = 2660$ output symbols need to be collected to achieve the same BER for LIB. This implies faster recovery for MIB than LIB.

IV. Finite-Length Analysis of UEP-Rateless Codes

In this section, finite-length analysis of LT and Raptor codes over the BEC is investigated. First, we derive upper and lower bounds on the maximum-likelihood (ML) decoding error probabilities of LT and Raptor codes when they provide EEP. We then study this for UEP-LT

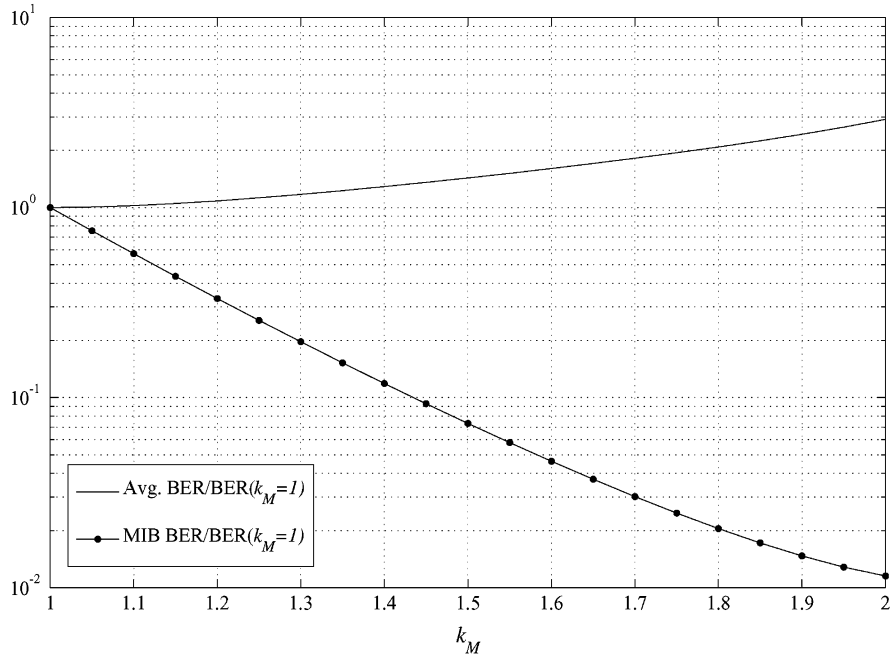

Fig. 4. The ratios of the average BER and the MIB error rate to the BER of the EEP-code versus k_M. In this case $\gamma = 1.05$.

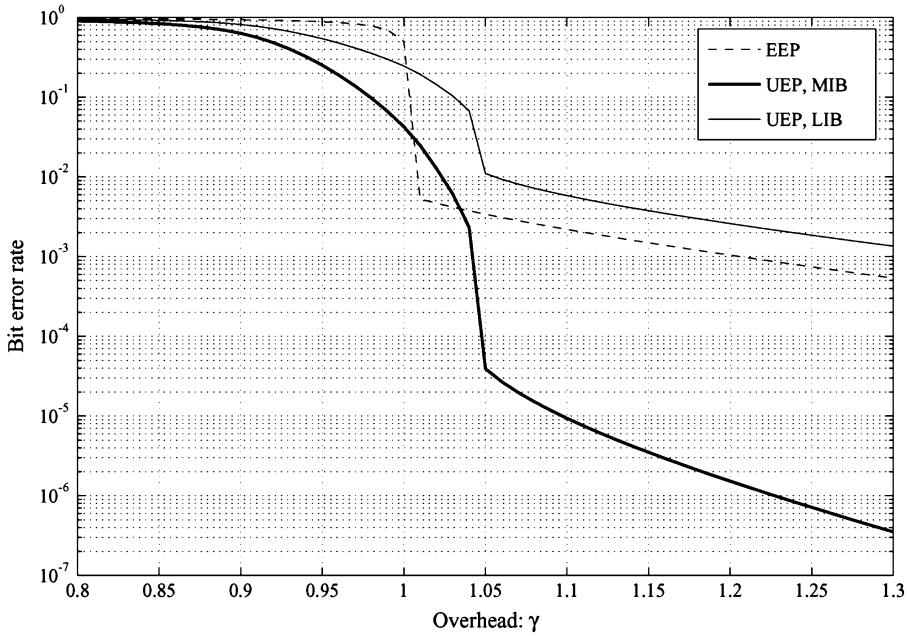

Fig. 5. Asymptotic BERs of MIB and LIB versus overhead γ for $k_M = 2$, as well as the BERs of the EEP code ($k_M = 1$).

and UEP-Raptor codes. ML decoding is computationally complex specially for long codes. However, the derivation of bounds on the ML decoding is of interest, as it provides an ultimate indication on the system performance.

A. Upper and Lower Bounds on the Maximum-Likelihood Decoding Error Probabilities of Finite-Length LT and Raptor Codes Over the BEC

We investigate the performance of finite-length LT and Raptor codes under the ML decoding. In our analysis, we consider the nonreplacement selection of the input nodes of LT codes. This means that given a check-node degree is d, a sequence of d different input nodes is selected uniformly at random from the n input nodes. Thus, a particular sequence is selected with a probability $\frac{1}{\binom{n}{d}}$.

1) ML Decoding of LT Codes Over the BEC: The ML decoding of LT codes over the BEC is the problem of recovering n information bits from $n\gamma$ received check bits. This is equivalent to solving the linear equation

$$Hx^T = b, \tag{8}$$

in which $H = [h_{ij}]$ is an $n\gamma \times n$ adjacency matrix corresponding to the graph that is formed by the input nodes and the received check nodes. Here, $h_{ij} = 1$ if the ith received check node and the jth input node are adjacent, otherwise $h_{ij} = 0$. Moreover, b is an n dimensional column vector in which b_i is the value of the ith received check node. Equation (8) has at least one solution. It has multiple solutions if and only if H is not full rank. Moreover, the ith bit does not have a unique solution if and only if H_i (the ith column of H) is in the column space spanned

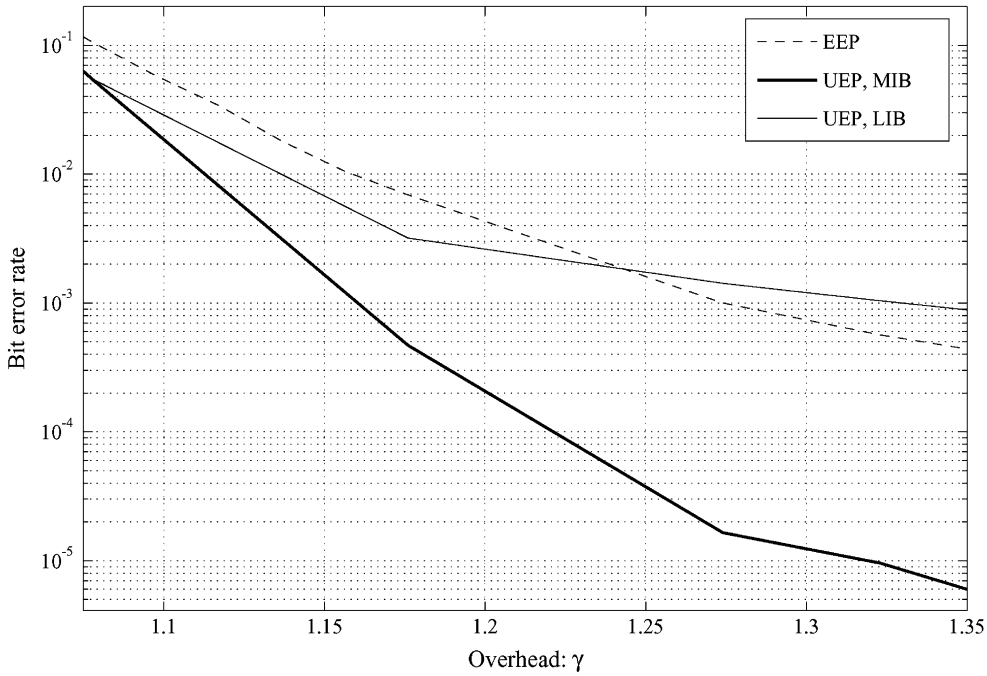

Fig. 6. Iterative decoding performance of the UEP-rateless code with parameters $\Omega_1(x)$, $n = 2000, k_M = 2$, and $\alpha = 0.1$ in comparison with the EEP-rateless code.

by $H \backslash H_i$. In the following lemma, we derive an upper bound on the ML decoding bit error probability of LT codes under the ML decoding.

Lemma 8: Given an LT code with parameters $\Omega(x)$, n, and overhead γ_L, an upper bound on the bit error probability of the LT code under the ML decoding is

$$p_b^{\mathrm{ML}} \leq \min\left\{1, \sum_{w=1}^{n} \binom{n-1}{w-1} \cdot \left(\sum_d \Omega_d \frac{\sum_{s=0,2,\ldots,2\lfloor \frac{d}{2} \rfloor} \binom{w}{s}\binom{n-w}{d-s}}{\binom{n}{d}} \right)^{n\gamma_L} \right\}. \quad (9)$$

Proof: Let p_b^{ML} be the probability that the ith bit cannot be determined by the ML decoder, for an arbitrary $i \in \{1, 2, \ldots, n\}$. We have

$$p_b^{\mathrm{ML}} = \Pr\{\exists x \in \mathrm{GF}(2)^n, x(i) = 1 : Hx^T = 0^T\} \leq \sum_{x \in \mathrm{GF}(2)^n, x(i)=1} \Pr\{Hx^T = 0^T\}.$$

Let $x \in \mathrm{GF}(2)^n, x(i) = 1$, and $I = \{i_1, i_2, \ldots, i_w\}$ be the set of indices such that $j \in I$ if and only if $x(j) = 1$. The rows of H, when viewed as random binary vectors, are generated from *independent* trials of a random variable R, such that for any vector $\nu \in \mathrm{GF}(2)^n, Pr(R = \nu) = \frac{\Omega_d}{\binom{n}{d}}$, where d is the weight of ν. Therefore

$$\Pr\{Hx^T = 0^T\} = (\Pr\{Rx^T = 0\})^{n\gamma_L}.$$

Suppose that weight$(R) = d$. Moreover, let $R(I)$ be defined as a sub-vector of R containing components of R that are specified by the elements of I, i.e., $R(I) = \{R(i_1), R(i_2), \ldots, R(i_w)\}$. We have

$$\Pr\{Rx^T = 0\} = \Pr\{R(I) \text{ contains even number of } 1's\} = \frac{\sum_{s=0,2,\ldots,2\lfloor \frac{d}{2} \rfloor} \binom{w}{s}\binom{n-w}{d-s}}{\binom{n}{d}}.$$

Since each row of H has weight d with probability Ω_d, and there are $\binom{n-1}{w-1}$ choices of x with weight w, we conclude the assertion.

A lower bound on the bit error probability of LT codes under ML decoding can be found by computing the probability that a variable node is not adjacent to any of the check nodes. This lower bound is given by [2]

$$p_b^{\mathrm{ML}} \geq \left(1 - \frac{\mu}{n}\right)^{n\gamma_L} \quad (10)$$

in which $\mu = \sum_d d\Omega_d$ is the average check-node degree.

Fig. 7 shows the upper and lower bounds on ML decoding error probabilities versus overhead γ_L for an LT code with distribution $\Omega_1(x)$ and length 500. The results imply that the bound is almost tight for $\gamma > 1.3$.

2) ML Decoding of Raptor Codes Over the BEC: Raptor codes introduced by Shokrollahi [2] are an extension of LT codes, in which an outer high-rate traditional pre-code is concatenated to an inner LT code to get practically better results than the LT code. Let $\mathcal{C}$ be a linear code of length n, rate $R = 1 - \frac{m}{n}$, and dimension $k = n - m$. A Raptor code with parameters $(k, \mathcal{C}, \Omega(x))$ is an LT code with distribution $\Omega(x)$ on n bits that are the codeword bits of the pre-code $\mathcal{C}$. If γ_L denotes the overhead of the LT code, the overhead of the Raptor code is $\gamma = \frac{\gamma_L}{R}$. In this paper, we assume the pre-code is an (n, k) LDPC code with a parity-check matrix $H' = [h'_{ij}]$ whose entries are independent and identically distributed (i.i.d) Bernoulli random variables with parameter ρ. We denote such a code by (n, k, ρ) LDPC code. The following lemmas develop upper and lower bounds on the ML decoding error probability of Raptor codes.

Lemma 9: Let $\mathcal{C}$ be an (n, k, ρ) LDPC code. Given a $(k, \mathcal{C}, \Omega(x))$ Raptor code with overhead γ, an upper bound on the ML decoding bit error probability is obtained as

$$p_b^{\mathrm{ML}} \leq \sum_{e=0}^{n} \binom{n}{e} \epsilon_U^e (1 - \epsilon_U)^{n-e} \frac{e}{n} \cdot \min\left\{1, \sum_{w=1}^{e} \binom{e-1}{w-1} (A(w, \rho))^m\right\}$$

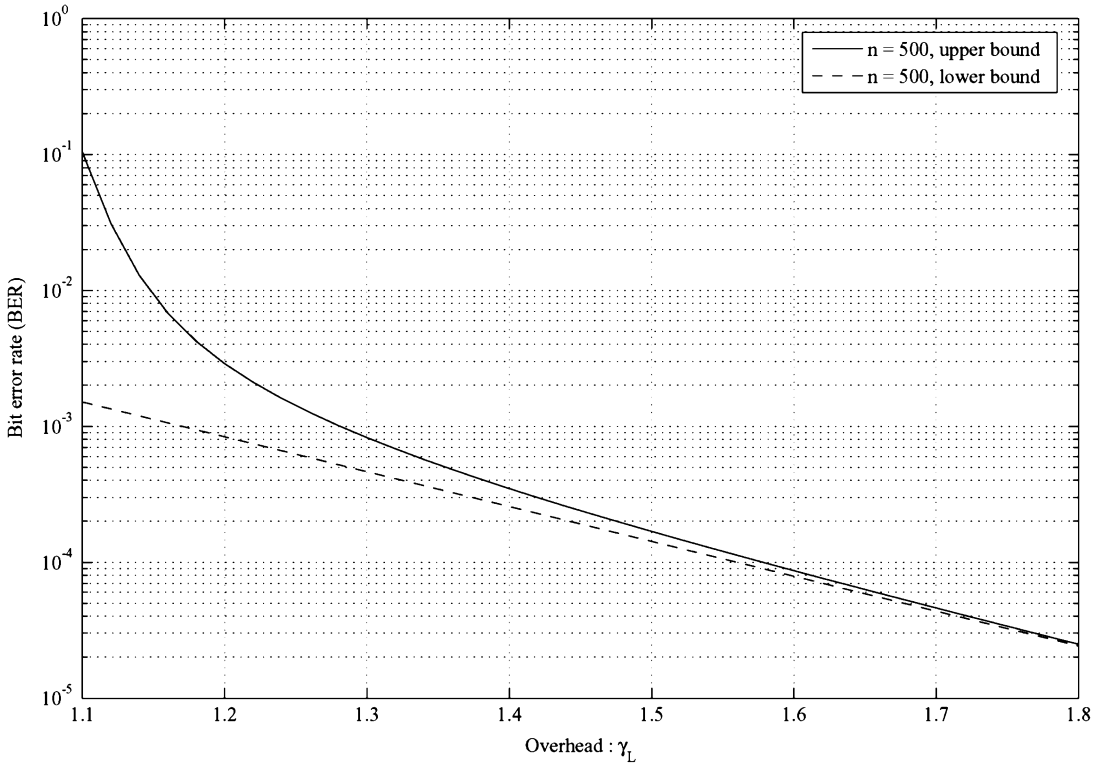

Fig. 7. Upper and lower bounds on the ML decoding bit error rates versus overhead γ_L for an LT code with distribution $\Omega_1(x)$ and length $n = 500$.

where

$$A(w,\rho) := \frac{1+(1-2\rho)^w}{2}. \tag{11}$$

Here, $m = n - k$. Also, ϵ_U is the upper bound on the ML decoding bit error rate of the LT code with parameters $\Omega(x), n$, and overhead $\gamma_L = \frac{k}{n}\gamma$ that was found by Lemma 8.

Proof: Let us assume that H' is the parity-check matrix corresponding to the pre-code $\mathcal{C}$. Moreover, let H'_e be an $m \times e$ matrix composed of the columns of H' that correspond to the variable nodes that have not been recovered after the LT-decoding process. Note that elements of H'_e are independently one with probability ρ. We want to obtain the probability that the ith bit cannot be determined either by the LT decoder or by the pre-code decoder, for an arbitrary $i \in \{1, 2, \ldots, n\}$. Let the j th column in H'_e correspond to the ith input bit of the LT code. We have

$$\begin{aligned}&\Pr\{\text{The pre-code fails to determine the } i\text{th bit}\}\\ &\quad= \Pr\{\exists x \in \mathrm{GF}(2)^e, x(j) = 1 : H'_e x^T = 0^T\}\\ &\quad\le \sum_{x\in \mathrm{GF}(2)^e,\ x(j)=1} \Pr\{H'_e x^T = 0^T\}.\end{aligned}$$

Let $x \in \mathrm{GF}(2)^e, x(j) = 1$, and weight of x is w. Let R_l denote the lth row of H'_e. We have

$$\Pr\{H'_e x^T = 0^T\} = \prod_{l=1}^{m} \Pr\{R_l x^T = 0\}.$$

Note that the events $R_l x^T = 0$ for $l = 1, \ldots, m$ are equiprobable and independent. Let $A(w,\rho)$ be the probability that even number of 1's occurs in a stream of independent 0's and 1's of length w when probability of 1 is ρ. We have

$$\Pr\{R_l x^T = 0\} = A(w,\rho) \tag{12}$$

$$= \frac{1+(1-2\rho)^w}{2}. \tag{13}$$

Assuming that ϵ is the bit error probability of the LT code, we conclude

$$p_b^{\mathrm{ML}} \le \sum_{e=0}^{n} \binom{n}{e} \epsilon^e (1-\epsilon)^{n-e} \frac{e}{n} \cdot \min\left\{1, \sum_{w=1}^{e} \binom{e-1}{w-1} A^m(w,\rho)\right\}. \tag{14}$$

However, instead of the exact value of ϵ, we have bounds on it. Assuming ϵ_U is an upper bound on ϵ and using Lemma 1 in Appendix A we can easily conclude the assertion. □

Lemma 10: Let $\mathcal{C}$ be an (n, k, ρ) LDPC code. Given a $(k, \mathcal{C}, \Omega(x))$ Raptor code with overhead γ, a lower bound on the ML decoding bit error probability is given by

$$\begin{aligned}p_b^{\mathrm{ML}} \ge \max\Bigg\{&0, \sum_{e=0}^{n} \binom{n}{e} \epsilon_L^e (1-\epsilon_L)^{n-e} \frac{e}{n}\\ &\times \min\left\{1, \sum_{w=1}^{e} \binom{e-1}{w-1} A^m(w,\rho)\right\}\\ &- \frac{1}{2}\sum_{e=0}^{n} \binom{n}{e} \epsilon_U^e (1-\epsilon_U)^{n-e} \frac{e}{n}\\ &\times \min\Bigg\{1, \sum_{w_0=1}^{e-1} \sum_{w_1=0}^{e-w_0} \sum_{w_2=0}^{e-w_0-w_1} \mathbf{1}(w_1+w_2)\\ &\cdot \binom{e-1}{w_0-1}\binom{e-w_0}{w_1}\binom{e-w_0-w_1}{w_2}\\ &\times D^m(w_0, w_1, w_2, \rho, \rho, \rho)\Bigg\}\Bigg\}\end{aligned}$$

where

$$\begin{aligned}D(w_0, w_1, w_2, \rho, \rho, \rho) := &A(w_0,\rho)A(w_1,\rho)A(w_2,\rho)\\ &+ \bar{A}(w_0,\rho)\bar{A}(w_1,\rho)\bar{A}(w_2,\rho)\end{aligned}$$

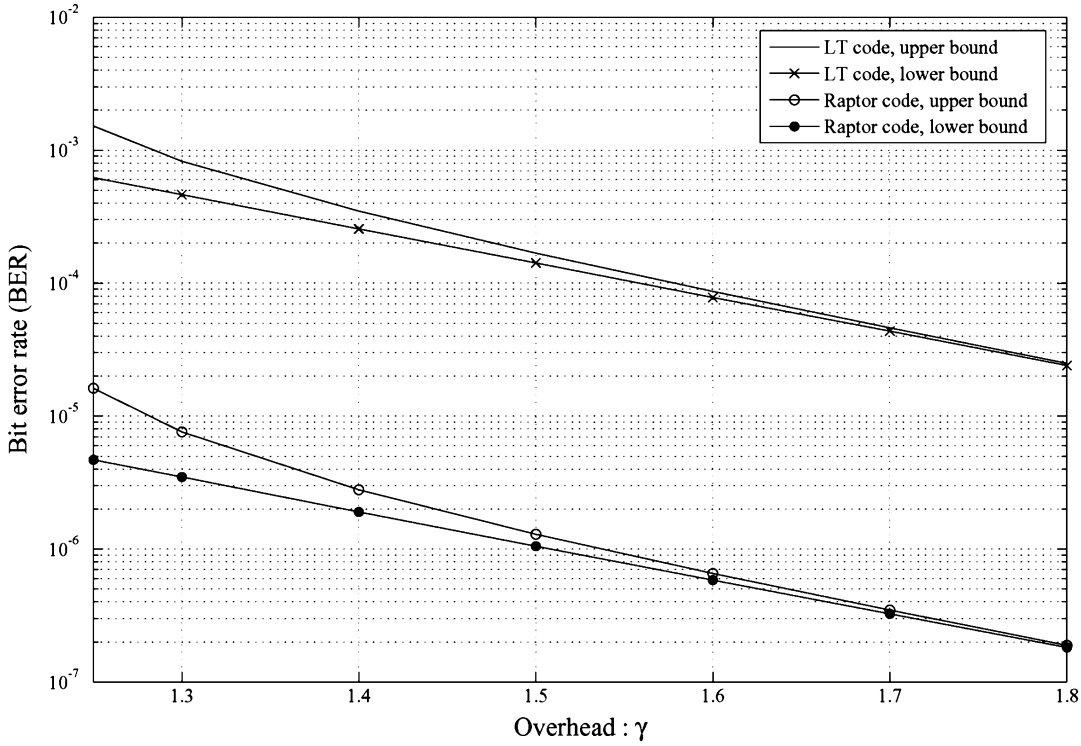

Fig. 8. Upper and lower bounds on the ML decoding bit error probabilities of LT and Raptor codes versus overhead γ for transmitting 500 information bits over an erasure channel.

$$\mathbf{1}(x) := \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{otherwise} \end{cases}$$

$\overline{A(\cdot)} = 1 - A(\cdot)$, and $m = n - k$. Also, $A(\cdot)$ is defined as (11). Moreover, $\epsilon_L(\epsilon_U)$ is the lower bound (upper bound) on the ML decoding bit error rate of the LT code with parameters $\Omega(x)$, n, and overhead $\gamma_L = \frac{k}{n}\gamma$ found by (10) and (9).

Proof: Consider H'_e as was defined before. Let the jth column in H'_e correspond to the ith bit. We have

$$\begin{aligned}
&\Pr\{\text{The precode fails to determine the } i\text{th bit}\} \\
&\quad = \Pr\{\exists x \in \mathrm{GF}(2)^e, x(j)=1 : H'_e x^T = 0^T\} \\
&\quad \geq \sum_{x \in \mathrm{GF}(2)^e, x(j)=1} \Pr\{H'_e x^T = 0^T\} \\
&\qquad - \frac{1}{2} \sum_{x,y \in \mathrm{GF}(2)^e,\ x(j)=1,\ y(j)=1,\ x \neq y} \Pr\{H'_e x^T = 0^T, H'_e y^T = 0^T\}
\end{aligned}$$

in which the inequality results in from the Bonferroni inequality [19]. The first term can be calculated using Lemma 9. Let $x, y \in \mathrm{GF}(2)^e$ such that $x(j) = 1, y(j) = 1$, and $x \neq y$. We define three binary vectors z_0, z_1, and $z_2 \in \mathrm{GF}(2)^e$ such that for $t = 1, \ldots, e$, $z_0(t) = 1$ if and only if $x(t) = 1$ and $y(t) = 1$, $z_1(t) = 1$ if and only if $x(t) = 1$ and $y(t) = 0$, and $z_2(t) = 1$ if and only if $x(t) = 0$ and $y(t) = 1$. Let w_0, w_1, and w_2 be the weights of vectors z_0, z_1, and z_2, respectively. We have

$$\Pr\{H'_e x^T = 0^T, H'_e y^T = 0^T\} \tag{15}$$

$$= \prod_{l=1}^{m} \Pr\{R_l z_0^T = R_l z_1^T = R_l z_2^T\}, \tag{16}$$

$$= \left(A(w_0,\rho)A(w_1,\rho)A(w_2,\rho) + \bar{A}(w_0,\rho)\bar{A}(w_1,\rho)\bar{A}(w_2,\rho)\right)^m \tag{17}$$

in which R_l denotes the lth row of H'_e, (16) is resulted from the independency of the elements of H'_e, and (17) is obtained easily by the definition of $A(.)$ as (11) and $\overline{A(\cdot)} := 1 - A(\cdot)$. Summing over all possible values for e, w_0, w_1, and w_2 and noting that w_1 and w_2 cannot be zero simultaneously (since $x \neq y$), we conclude the assertion. □

Fig. 8 depicts the upper and lower bounds on the ML decoding bit error probabilities versus overhead γ for the fixed degree distribution $\Omega_1(x)$. We considered an LT code with $n = 500$ and a Raptor code with $k = 500$ and a precode $\mathcal{C}$ as an $(510, 500, 0.4)$ LDPC code with $R \approx 0.98$. Note that in each case we assumed the decoder starts the decoding after receiving 500γ check bits. As we can see, the bounds are tight for small error rates. Moreover, as we expected and it was shown in [2], Raptor codes can achieve lower error rates than LT codes.

B. Upper and Lower Bounds on the Maximum-Likelihood Decoding Error Probabilities of Finite-Length UEP-LT and UEP-Raptor Codes Over the BEC

In this section, we consider the problem of finite-length UEP-rateless codes. Suppose we want to transmit n bits with two different levels of importance over a BEC. Assume $n_1 = \alpha n$ $(0 < \alpha < 1)$ is the number of MIB and $n_2 = (1 - \alpha)n$ is the number of LIB. A UEP-LT code is constructed similar to a traditional LT code except that the check nodes select their adjacent variable nodes nonuniformly at random. This means that a check node with degree d, selects $d_1 = \min([\alpha d k_M], n_1)$ ($[x]$ means the nearest integer to x) variable nodes from MIB (for some $k_M > 1$) and $d_2 = d - d_1$ variable nodes from LIB as shown in Fig. 9. Note that here the nonreplacement selection is considered. This means that any sequence of $d_1(d_2)$ different variable nodes in MIB (LIB) is selected uniformly with probability $\frac{1}{\binom{n_1}{d_1}}\left(\frac{1}{\binom{n_2}{d_2}}\right)$. By cascading a UEP-LT code and a traditional precode

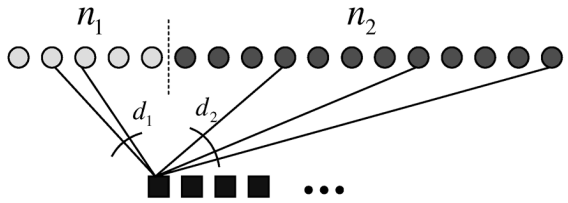

Fig. 9. Nonuniform selection of variable nodes (input symbols) in UEP-LT codes.

$\mathcal{C}$, we can form a UEP-Raptor code.[2] This implies that the codeword bits of $\mathcal{C}$ are the input bits of the UEP-LT code. Let $\mathcal{C}$ be a linear code of length n, rate $R = 1 - \frac{m}{n}$ and dimension $k = n - m$. Let also H' be the parity-check matrix that corresponds to $\mathcal{C}$. Here the number of information bits is k. We may design the precode $\mathcal{C}$ such that all the first n_1 bits of the codeword bits correspond to the more important information bits. This is possible if and only if the submatrix of H' containing the last n_2 columns has full rank. In this case, the ratio of the number of more important information bits to the total number of information bits is $\alpha_R = \frac{\alpha}{R}$. As before, let us assume the pre-code $\mathcal{C}$ is an (n, k, ρ) LDPC code. Next, we derive upper and lower bounds on the ML decoding error probabilities of the UEP-LT and UEP-Raptor codes.

1) ML Decoding of UEP-LT Codes: In this section, we examine the performance of UEP-LT codes under the ML decoding. In the following lemma, upper bounds on the ML decoding error probabilities of the proposed ensemble are derived.

Lemma 11: Consider a UEP-LT code with parameters $\Omega(x), n, \alpha, k_M$, and overhead γ_L. The upper bounds on the bit error probabilities of MIB and LIB under the ML decoding are given in (18) and (19) at the bottom of the page, respectively. Here, $w_2 = w - w_1, n_1 = \alpha n, n_2 = (1-\alpha)n, d_1 = \min([\alpha d k_M], n_1)$, and $d_2 = d - d_1$.

Proof: Let $H = [h_{cv}]$ be the adjacency matrix corresponding to the graph that is formed by the input nodes and the received check nodes. This means that $h_{cv} = 1$ if and only if the cth received check node is adjacent to vth variable node. Let $p_{b,i}^{\mathrm{ML}}$ be the bit error probability of the ith bit under ML decoding. For an arbitrary $i \in \{1, 2, \ldots, n\}$ we have

$$p_{b,i}^{\mathrm{ML}} = \Pr\{\exists x \in \mathrm{GF}(2)^n, x(i) = 1 : Hx^T = 0^T\} \le \sum_{x \in \mathrm{GF}(2)^n, x(i)=1} \Pr\{Hx^T = 0^T\}. \quad (20)$$

Let $x \in \mathrm{GF}(2)^n, x(i) = 1$, and $I = \{i_1, i_2, \ldots, i_{w_1}\}$ be the set of indices such that $j \in I$ if and only if $x(j) = 1$ and $j \in \{1, \ldots, n_1\}$. Similarly, $J = \{j_1, j_2, \ldots, j_{w_2}\}$ is the set of indices such that $j \in J$ if and only if $x(j) = 1$ and $j \in \{n_1 + 1, \ldots, n\}$. As in the proof of Lemma 8

$$\Pr\{Hx^T = 0^T\} = (\Pr\{Rx^T = 0\})^{n\gamma_L},$$

where R is any row of H. Suppose that weight$(R) = d$. We have (21) at the bottom of the page. For $i \in$ MIB, there are $\binom{n_1-1}{w_1-1}\binom{n_2}{w_2}$ possible different x's, and for $i \in$ LIB, this value is $\binom{n_1}{w_1}\binom{n_2-1}{w_2-1}$. This completes the proof. □

Lower bounds on the bit error probabilities of MIB and LIB under the ML decoding are given by

$$p_{b,\mathrm{MIB}}^{\mathrm{ML}} \ge \left(1 - \sum_d \Omega_d \frac{d_1}{n_1}\right)^{n\gamma_L} \quad (22)$$

and

$$p_{b,\mathrm{LIB}}^{\mathrm{ML}} \ge \left(1 - \sum_d \Omega_d \frac{d_2}{n_2}\right)^{n\gamma_L} \quad (23)$$

respectively. These are the probabilities that a node in MIB or LIB is not a neighbor of any of the check nodes.

Fig. 10 shows the upper bound (UB) and lower bound (LB) on the ML decoding BER's of MIB and LIB versus overhead γ_L for a UEP-LT code with parameters $n = 500, \Omega_1(x), k_M = 2$, and $\alpha = 0.1$. We also included the bounds on the ML decoding performance of an EEP-LT code with $n = 500$ and $\Omega_1(x)$. As an example, for $\gamma = 1.8$ where the bounds are tight, we note that BER of LIB is increased less than one order of magnitude in comparison with the EEP code. However, BER of MIB is decreased by about four orders of magnitude.

2) ML Decoding of UEP-Raptor Codes: Let us consider the case that we cascade a UEP-LT code by a pre-code $\mathcal{C}$ to form a UEP-Raptor code. Similar to Lemma 9, we can show the following.

Lemma 12: Let $\mathcal{C}$ be an (n, k, ρ) LDPC code. Consider a UEP-Raptor code that has a UEP-LT code with parameters $\Omega(x), n, \gamma_L, \alpha$,

$$p_{b,\mathrm{MIB}}^{\mathrm{ML}} \le \min\left\{1, \sum_{w=1}^{n}\sum_{w_1=1}^{w}\binom{n_1-1}{w_1-1}\binom{n_2}{w_2}\cdot\left(\sum_d \Omega_d \frac{\sum_{t=0}^{1}\left(\prod_{r=1}^{2}\left(\sum_{s=t,2+t,\ldots,2\lfloor\frac{d_r}{2}\rfloor-t}\binom{w_r}{s}\binom{n_r-w_r}{d_r-s}\right)\right)}{\binom{n_1}{d_1}\binom{n_2}{d_2}}\right)^{n\gamma_L}\right\} \quad (18)$$

and

$$p_{b,\mathrm{LIB}}^{\mathrm{ML}} \le \min\left\{1, \sum_{w=1}^{n}\sum_{w_1=0}^{w-1}\binom{n_1}{w_1}\binom{n_2-1}{w_2-1}\cdot\left(\sum_d \Omega_d \frac{\sum_{t=0}^{1}\left(\prod_{r=1}^{2}\left(\sum_{s=t,2+t,\ldots,2\lfloor\frac{d_r}{2}\rfloor-t}\binom{w_r}{s}\binom{n_r-w_r}{d_r-s}\right)\right)}{\binom{n_1}{d_1}\binom{n_2}{d_2}}\right)^{n\gamma_L}\right\} \quad (19)$$

$$\begin{aligned}\Pr\{Rx^T = 0\} &= \Pr\{R(I) \text{ contains even number of 1's}\}\cdot\Pr\{R(J) \text{ contains even number of 1's}\} \\ &\quad + \Pr\{R(I) \text{ contains odd number of 1's}\}\cdot\Pr\{R(J) \text{ contains odd number of 1's}\} \\ &= \frac{\sum_{t=0}^{1}\left(\prod_{r=1}^{2}\left(\sum_{s=t,2+t,\ldots,2\lfloor\frac{d_r}{2}\rfloor-t}\binom{w_r}{s}\binom{n_r-w_r}{d_r-s}\right)\right)}{\binom{n_1}{d_1}\binom{n_2}{d_2}}.\end{aligned} \quad (21)$$

[2] An alternative way to form a UEP-Raptor code is by cascading a traditional LT code and a UEP pre-code. Although we do not consider this case in this correspondence, the analysis will be similar.

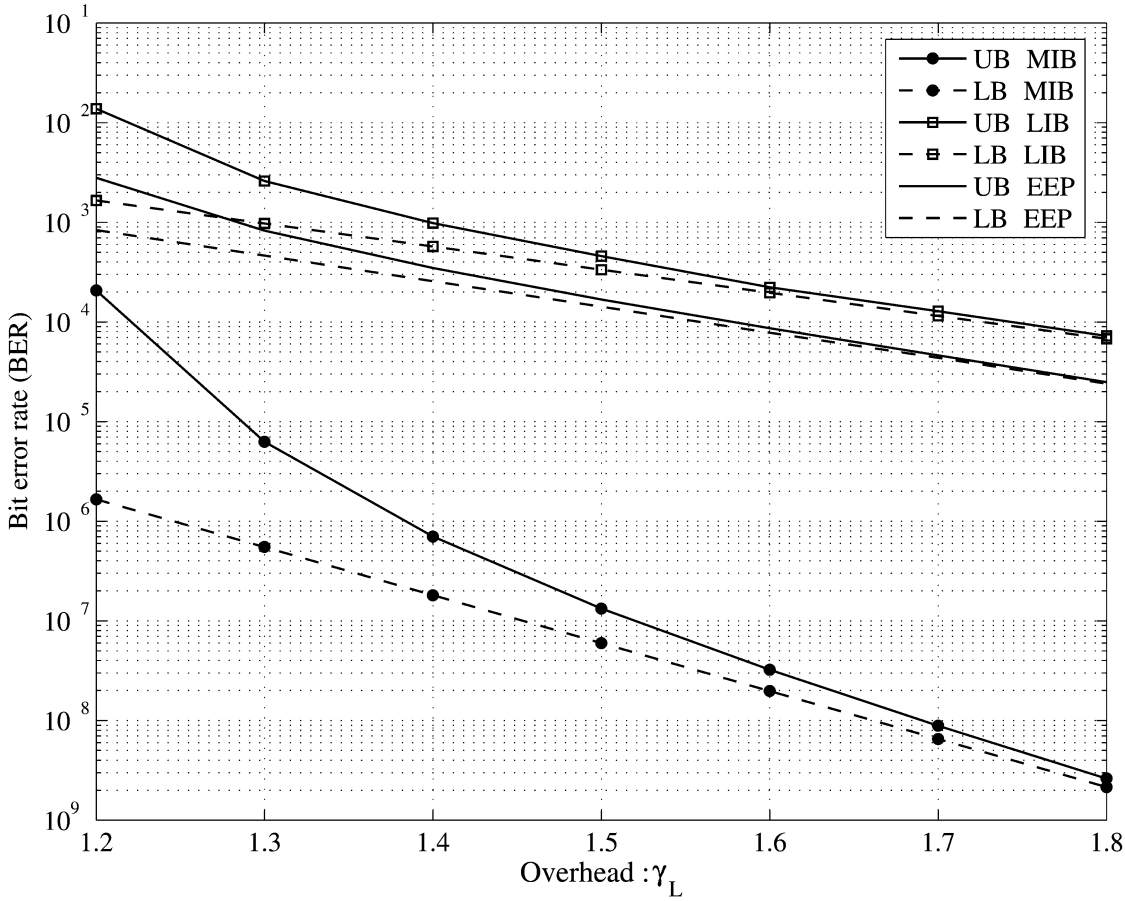

Fig. 10. Upper and lower bounds on the ML decoding BER's of MIB and LIB versus overhead γ_L for a UEP-LT code with parameters $n = 500, \Omega_1(x), k_M = 2$, and $\alpha = 0.1$. The bounds on the decoding performance of the EEP-LT code are also depicted.

and k_M together with the precode $\mathcal{C}$. Upper bounds on the bit error probabilities of MIB and LIB under the ML decoding are given by

$$p_{b,\mathrm{MIB}}^{\mathrm{ML}} \leq \sum_{e=1}^{n} \sum_{e_1=\max(1,e-n_2)}^{\min(n_1,e)} \binom{n_1-1}{e_1-1}\binom{n_2}{e-e_1} \epsilon_{U1}^{e_1}(1-\epsilon_{U1})^{n_1-e_1} \cdot \epsilon_{U2}^{e-e_1}(1-\epsilon_{U2})^{n_2-e+e_1} \min\left\{1, \sum_{w=1}^{e}\binom{e-1}{w-1}A^m(w,\rho)\right\}$$

and

$$p_{b,\mathrm{LIB}}^{\mathrm{ML}} \leq \sum_{e=1}^{n} \sum_{e_1=\max(0,e-n_2)}^{\min(n_1,e-1)} \binom{n_1}{e_1}\binom{n_2-1}{e-e_1-1} \epsilon_{U1}^{e_1}(1-\epsilon_{U1})^{n_1-e_1}\epsilon_{U2}^{e-e_1}(1-\epsilon_{U2})^{n_2-e+e_1} \times \min\left\{1, \sum_{w=1}^{e}\binom{e-1}{w-1}A^m(w,\rho)\right\}$$

respectively. Here, ϵ_{U1} and ϵ_{U2} are the upper bounds on the ML decoding BER's of MIB and LIB in the UEP-LT code, respectively, $m = n-k$, and $A(.)$ is defined as in Lemma 9. Likewise, similar to Lemma 10, we can show the following.

Lemma 13: Let $\mathcal{C}$ be an (n,k,ρ) LDPC code. Consider a UEP-Raptor code that has a UEP-LT code with parameters $\Omega(x), n, \gamma_L, \alpha$, and k_M together with the pre-code $\mathcal{C}$. Lower bounds on the bit error probabilities of MIB and LIB under the ML decoding are given by

$$\begin{aligned} p_{b,\mathrm{MIB}}^{\mathrm{ML}} \geq \max\Bigg\{0, &\sum_{e=1}^{n} \sum_{e_1=\max(1,e-n_2)}^{\min(n_1,e)} \binom{n_1-1}{e_1-1} \\ &\times \binom{n_2}{e-e_1}\epsilon_{L1}^{e_1}(1-\epsilon_{L1})^{n_1-e_1} \cdot \epsilon_{L2}^{e-e_1}(1-\epsilon_{L2})^{n_2-e+e_1} \\ &\times \min\left\{1, \sum_{w=1}^{e}\binom{e-1}{w-1}A^m(w,\rho)\right\} \\ &- \frac{1}{2}\sum_{e=1}^{n} \sum_{e_1=\max(1,e-n_2)}^{\min(n_1,e)} \binom{n_1-1}{e_1-1} \\ &\times \binom{n_2}{e-e_1}\epsilon_{U1}^{e_1}(1-\epsilon_{U1})^{n_1-e_1} \cdot \epsilon_{U2}^{e-e_1}(1-\epsilon_{U2})^{n_2-e+e_1} \\ &\times \min\Bigg\{1, \sum_{w_0=1}^{e-1}\sum_{w_1=0}^{e-w_0}\sum_{w_2=0}^{e-w_0-w_1} \mathbf{1}(w_1+w_2) \\ &\cdot \binom{e-1}{w_0-1}\binom{e-w_0}{w_1}\binom{e-w_0-w_1}{w_2} \\ &\times D^m(w_0,w_1,w_2,\rho,\rho,\rho)\Bigg\}\Bigg\} \end{aligned}$$

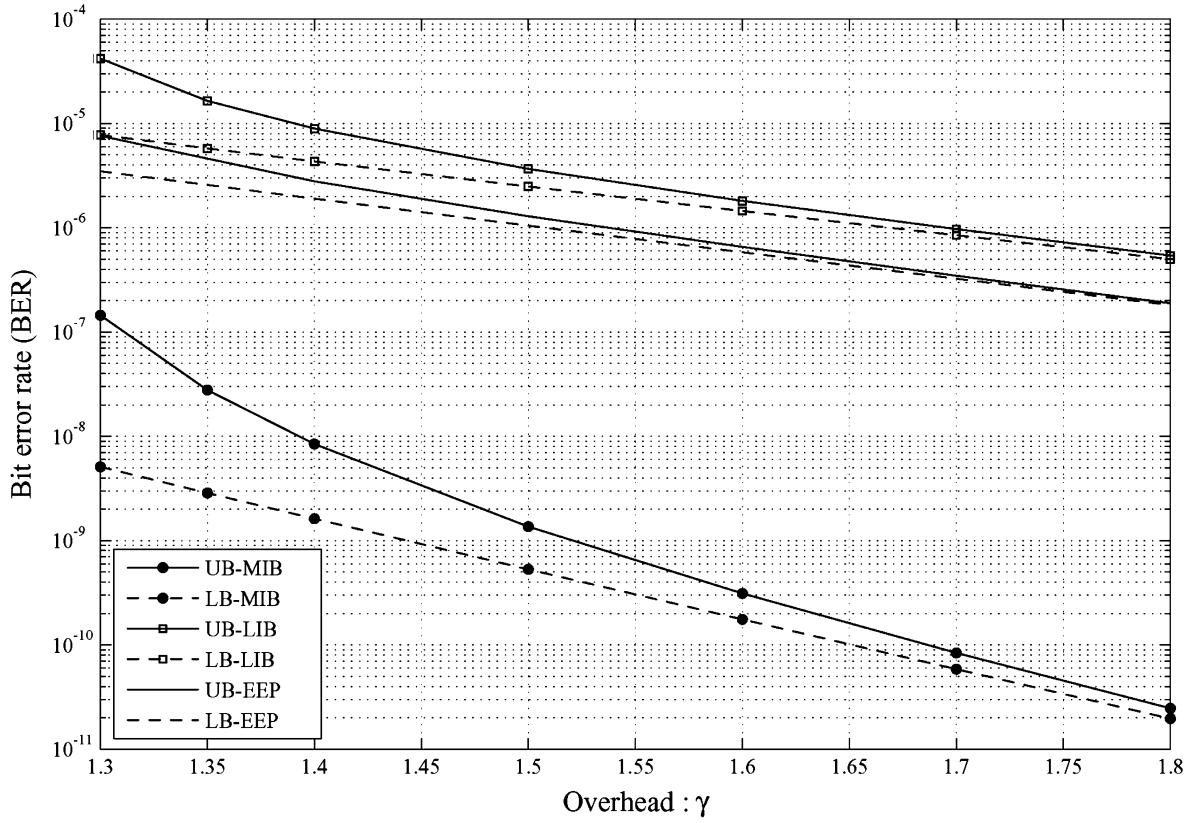

Fig. 11. Upper and lower bounds on the ML decoding BER's of MIB and LIB for the UEP-Raptor code with parameters $k = 500, \Omega_1(x), k_M = 2, \alpha = 0.1$, and a (5105000.4) LDPC code as the precode. The bounds on the decoding performance of the EEP-Raptor code are also depicted.

and

$$p_{b,\mathrm{LIB}}^{\mathrm{ML}} \geq \max \Bigg\{ 0, \sum_{e=1}^{n} \sum_{e_1=\max(0,e-n_2)}^{\min(n_1,e-1)} \binom{n_1}{e_1} \times \binom{n_2-1}{e-e_1-1} \epsilon_{L1}^{e_1} (1-\epsilon_{L1})^{n_1-e_1} \cdot \epsilon_{L2}^{e-e_1} (1-\epsilon_{L2})^{n_2-e+e_1} \times \min \Bigg\{ 1, \sum_{w=1}^{e} \binom{e-1}{w-1} A^m(w,\rho) \Bigg\} - \frac{1}{2} \sum_{e=1}^{n} \sum_{e_1=\max(0,e-n_2)}^{\min(n_1,e)} \binom{n_1}{e_1} \times \binom{n_2-1}{e-e_1-1} \epsilon_{U1}^{e_1} (1-\epsilon_{U1})^{n_1-e_1} \cdot \epsilon_{U2}^{e-e_1} (1-\epsilon_{U2})^{n_2-e+e_1} \times \min \Bigg\{ 1, \sum_{w_0=1}^{e-1} \sum_{w_1=0}^{e-w_0} \sum_{w_2=0}^{e-w_0-w_1} \mathbf{1}(w_1+w_2) \cdot \binom{e-1}{w_0-1} \binom{e-w_0}{w_1} \binom{e-w_0-w_1}{w_2} \times D^m(w_0,w_1,w_2,\rho,\rho,\rho) \Bigg\} \Bigg\}$$

respectively. Here, $\epsilon_{L1}(\epsilon_{U1})$ and $\epsilon_{L2}(\epsilon_{U2})$ are the lower bounds (upper bounds) on the ML decoding BER's of MIB and LIB in the UEP-LT code, respectively, and $m = n - k$. Moreover, $A(\cdot)$, $\overline{A(\cdot)}$, and $D(\cdot)$ are defined as in Lemmas 9 and 10.

Fig. 11 shows the upper and lower bounds on the ML decoding BER's of MIB and LIB versus overhead γ for a UEP-Raptor code with parameters $k = 500, \Omega_1(x), k_M = 2, \alpha = 0.1$, and a pre-code $\mathcal{C}$ as an $(510, 500, 0.4)$ LDPC code with $R \approx 0.98$. We also included the bounds on the ML decoding performance of an EEP-Raptor code with $k = 500, \Omega_1(x)$, and the same precode. As an example, for $\gamma = 1.8$ where the bounds are tight, the BER of LIB is increased less than one order of magnitude but the BER of MIB is decreased by about four orders of magnitude. This shows a large gap between the BERs of MIB and LIB and very low error rates for the MIB.

V. Conclusion

In this paper, we proposed a modification in the structure of rateless codes to provide unequal error protection (UEP) and unequal recovery time (URT) properties. We analyzed the performance of the proposed structure asymptotically. It was shown that UEP-rateless codes can provide very low error rates for more important bits with only a subtle loss on the performance of less important bits. Next, we focused on finite-length rateless codes and derived upper and lower bounds on the maximum-likelihood decoding bit error rates of EEP- and UEP-rateless codes. The results show that the bounds are tight for small error rates. Moreover, the bit error rates of more important bits are significantly improved with respect to the bit error rates of less important bits for finite-length cases. We also discussed that the UEP problem can be viewed as the URT problem for a fixed bit error rate.

Appendix A

Lemma 1: Let us define $g(\epsilon) = \sum_{e=0}^{n} \binom{n}{e} \epsilon^e (1-\epsilon)^{n-e} f(e)$. Then, $g(\epsilon)$ is a nondecreasing function of ϵ if $f(e)$ is a nondecreasing function of e.

Proof: Let us define $h_{\epsilon_1}(e) = \binom{n}{e}\epsilon_1^e(1-\epsilon_1)^{n-e}$ and $h_{\epsilon_2}(e) = \binom{n}{e}\epsilon_2^e(1-\epsilon_2)^{n-e}$. We need to show that if $\epsilon_2 > \epsilon_1$ then

$$\sum_{e=0}^{n} h_{\epsilon_2}(e)f(e) \geq \sum_{e=0}^{n} h_{\epsilon_1}(e)f(e).$$

Since $\epsilon_2 > \epsilon_1$ and due to the nature of functions $h_{\epsilon_1}(e)$ and $h_{\epsilon_2}(e)$ it can be easily shown that there exists an integer e_0 such that $h_{\epsilon_2}(e) > h_{\epsilon_1}(e)$ if and only if $e \geq e_0$. Therefore

$$\begin{aligned}
&\sum_{e=0}^{n}(h_{\epsilon_2}(e) - h_{\epsilon_1}(e))f(e) \\
&\quad = \sum_{e:h_{\epsilon_2}(e)>h_{\epsilon_1}(e)} (h_{\epsilon_2}(e) - h_{\epsilon_1}(e))\, f(e) \\
&\qquad - \sum_{e:h_{\epsilon_2}(e)\leq h_{\epsilon_1}(e)} (h_{\epsilon_1}(e) - h_{\epsilon_2}(e))\, f(e) \\
&\quad \geq f(e_0) \sum_{e=e_0}^{n} (h_{\epsilon_2}(e) - h_{\epsilon_1}(e)) - f(e_0 - 1) \\
&\qquad \times \sum_{e=0}^{e_0-1} (h_{\epsilon_1}(e) - h_{\epsilon_2}(e)) \\
&\quad = (f(e_0) - f(e_0 - 1)) \sum_{e=e_0}^{n} (h_{\epsilon_2}(e) - h_{\epsilon_1}(e)) \geq 0 \qquad (24)
\end{aligned}$$

where in (24), we use the fact that

$$\sum_{e=0}^{n} h_{\epsilon_1}(e) = \sum_{e=0}^{n} h_{\epsilon_2}(e) = 1$$

Therefore

$$\sum_{e=e_0}^{n} (h_{\epsilon_2}(e) - h_{\epsilon_1}(e)) = \sum_{e=0}^{e_0-1} (h_{\epsilon_1}(e) - h_{\epsilon_2}(e)).$$ □

REFERENCES

[1] M. Luby, "LT codes," in *Proc. 43rd Ann. IEEE Symp. Found. Comp. Sci.*, 2002.

[2] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, pp. 2551–2567, Jun. 2006.

[3] P. Maymounkov, Online codes NYU Tech. Rep. TR2003-883, 2002.

[4] M. Mitzenmacher, "Digital fountains: A survey and look forward," in *Proc. Inf. Theory Workshop*, Oct. 2004, pp. 271–276.

[5] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM*, Vancouver, BC, Canada, Aug. 1998, pp. 56–67.

[6] T. Sikora, "MPEG digital video coding standards," *IEEE Signal Process. Mag.*, vol. 14, pp. 82–100, Sep. 1997.

[7] L. Xu, "Resource-efficient delivery of on-demand streaming data using UEP codes," *IEEE Trans. Commun.*, vol. 51, pp. 63–71, Jan. 2003.

[8] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Trans. Inf. Theory*, vol. IT-3, pp. 600–607, Oct. 1967.

[9] L. A. Bassalygo, V. A. Zinov'ev, V. V. Zyablov, and G. S. P. M. S. Pinsker, "Bounds for codes with unequal protection of two sets of messages," *Problemy Peredachi Informatsii*, vol. 15, pp. 40–49, Jul.–Sep. 1979.

[10] C. C. Kilgus and W. C. Gore, "Cyclic codes with unequal error protection," *IEEE Trans. Inf. Theory*, vol. 17, pp. 214–215, Mar. 1971.

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

[12] C. Poulliat, D. Declercq, and I. Fijalkow, "Optimization of LDPC codes for UEP property," in *IEEE Int. Symp. Inf. Theory*, Jun.–Jul. 2004, p. 450.

[13] H. Pishro-Nik, N. Rahnavard, and F. Fekri, "Nonuniform error correction using low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 51, pp. 2702–2714, Jul. 2005.

[14] N. Rahnavard and F. Fekri, "Finite-length unequal error protection rateless codes: Design and analysis," in *Proc. IEEE GLOBECOM*, St. Louis, MO, Nov.–Dec. 2005.

[15] N. Rahnavard and F. Fekri, "Generalization of rateless codes for unequal error protection and recovery time: Asymptotic analysis," in *Proc. IEEE Int. Symp. Inf. Theory*, Seattle, WA, Jul. 2006, pp. 523–527.

[16] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. 29th Ann. ACM Symp. Theory Computing (STOC)*, 1997, pp. 150–159.

[17] M. Luby, Mitzenmacher, and A. Shokrallahi, "Analysis of random processes via and-or tree evaluation," in *Proc. 9th Ann. ACM-SIAM Symp. Discrete Algorithms*, 1998, pp. 364–373.

[18] R. G. Bartle, *The Elements of Real Analysis*, 2nd ed. New York: Wiley, 1976.

[19] L. Comtet, Advanced Combinatorics, 1974.

On Z_{2^k}-Dual Binary Codes

Denis S. Krotov

***Abstract*—A new generalization of the Gray map is introduced. The new generalization $\Phi : Z_{2^k}^n \to Z_2^{2^{k-1}n}$ is connected with the known generalized Gray map φ in the following way: if we take two dual linear Z_{2^k}-codes and construct binary codes from them using the generalizations φ and Φ of the Gray map, then the weight enumerators of the binary codes obtained will satisfy the MacWilliams identity. The classes of Z_{2^k}-linear Hadamard codes and co-Z_{2^k}-linear extended 1-perfect codes are described, where co-Z_{2^k}-linearity means that the code can be obtained from a linear Z_{2^k}-code with the help of the new generalized Gray map.**

***Index Terms*—Gray map, Hadamard codes, MacWilliams identity, perfect codes, Z_{2^k}-linearity.**

I. INTRODUCTION

As discovered in [1], [2], [20] certain nonlinear binary codes can be represented as linear codes over Z_4. The variant of this representation founded in [3] uses the mapping $\phi : 0 \to 00, 1 \to 01, 2 \to 11, 3 \to 10$, which is called the *Gray map*, to construct binary so-called Z_4-linear codes from linear quaternary codes. The main property of ϕ from this point of view is that it is an isometry between Z_4 with the Lee metric and Z_2^2 with the Hamming metric. In [4] (and in [5] in more general form) the Gray map is generalized to construct Z_{2^k}-linear codes. The generalized Gray map (say φ; see Section II-A for recalling basic facts on the generalized Gray map) is an isometric imbedding of Z_{2^k} with the metric specified by the homogeneous weight [6] into $Z_2^{2^{k-1}}$ with the Hamming metric.

In this correspondence, we introduce another generalization Φ of the Gray map (Section II-B). This generalization turns out to be dual to the previous in the following sense. If $\mathcal{C}$ and $\mathcal{C}^\perp$ are dual linear Z_{2^k}-codes, then the binary Z_{2^k}-linear code $\varphi(\mathcal{C})$ and the *co-Z_{2^k}-linear* code $\Phi(\mathcal{C}^\perp)$ are formally dual. The formal duality is that the weight enumerators of these two codes satisfy the MacWilliams identity

Manuscript received July 2, 2006; revised December 8, 2006. The material in this correspondence was presented in part at the 4th International Workshop on Optimal Codes and Related Topics OC 2005, Pamporovo, Bulgaria, June 2005.

The author is with the Sobolev Institute of Mathematics, Novosibirsk, 630090 Russia (e-mail: krotov@math.nsc.ru).

Communicated by T. Etzion, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2007.892787