Raptor Codes on Binary Memoryless Symmetric Channels

Omid Etesami and Amin Shokrollahi, *Senior Member, IEEE*

***Abstract*—In this paper, we will investigate the performance of Raptor codes on arbitrary binary input memoryless symmetric channels (BIMSCs). In doing so, we generalize some of the results that were proved before for the erasure channel. We will generalize the stability condition to the class of Raptor codes. This generalization gives a lower bound on the fraction of output nodes of degree 2 of a Raptor code if the error probability of the belief-propagation decoder converges to zero. Using information-theoretic arguments, we will show that if a sequence of output degree distributions is to achieve the capacity of the underlying channel, then the fraction of nodes of degree 2 in these degree distributions has to converge to a certain quantity depending on the channel. For the class of erasure channels this quantity is independent of the erasure probability of the channel, but for many other classes of BIMSCs, this fraction depends on the particular channel chosen. This result has implications on the "universality" of Raptor codes for classes other than the class of erasure channels, in a sense that will be made more precise in the paper. We will also investigate the performance of specific Raptor codes which are optimized using a more exact version of the Gaussian approximation technique.**

***Index Terms*—Belief-propagation, graphical codes, LT-codes, raptor codes.**

I. Introduction

ITERATIVE decoding algorithms have received much attention in the past few years. They are among the most efficient decoding algorithms to date, and perform very well even at rates extremely close to the capacity of many known symmetric channels.

One of the most prominent classes of codes for which iterative decoding methods have been quite successful is the class of low-density parity-check (LDPC) codes. These codes were invented by Gallager [1] in the early 1960s but did not receive proper attention by the information theory community until years later, when their excellent decoding qualities were rediscovered independently by the information theory and the theoretical computer science communities [2]–[7].

Gallager's LDPC codes are formed from sparse bi-regular bipartite graphs, consisting of two disjoint sets of variable and check nodes. The code is defined as the set of all binary settings of the variable nodes such that for each check node the sum (over $\mathrm{GF}(2)$) of the values of its incident variable nodes is zero. The most powerful efficient decoding algorithm for this class of codes is the iterative belief-propagation (BP) algorithm. At each iteration, this algorithm updates the probability that a given variable node is 0, given all the observations obtained in previous rounds. The complexity of the update in every iteration is proportional to the number of edges in the graph. Therefore, for a constant number of iterations, the running time of the BP algorithm is proportional to the number of edges in the graph, and hence, to the number of variable nodes if the underlying graph is sparse.

Luby *et al.* [8], [2], [3] were the first to prove that an appropriately chosen but highly irregular graph structure can yield to superior performance of the BP decoder as compared to case when regular graphs are used. Since then the concept of irregular LDPC codes has occupied center stage in the design of LDPC codes whose decoding performance is extremely close to the Shannon bounds. This performance is often calculated using the method of density evolution. This method was introduced by Luby *et al.* [9] under the name of tree analysis, and used to analyze hard-decision decoding algorithms for the binary-symmetric channel in Luby *et al.* [3]. The method was vastly generalized by Richardson and Urbanke [10] to any symmetric channel, and it was renamed to density evolution.

Classical LDPC codes do not possess a fast encoding algorithm, since the code is defined as the kernel of a sparse matrix, rather than as the image of such a matrix. There are various methods to either solve or circumvent this problem. Some of these methods circumvent the problem by considering modified codes which automatically possess fast encoders [8], [11]. Others, such as the method described in [12], stay faithful to LDPC codes and design efficient encoding algorithms that often run in linear time.

These and similar advances in the field seem to suggest that it is very difficult to substantially improve upon existing codes and their decoding algorithms. However, there are real communication scenarios in which block (or even convolutional) codes do not yield adequate results, no matter how close their performance is to the capacity of the underlying channel. For example, consider the code design problem for transmission of packets on a computer network. This transmission channel is well modeled by a binary erasure channel (BEC) [8]. However, in almost all applications, the loss rate of the channel is unknown to the sender or to the receiver. Using a block code, it is necessary to obtain a good estimate of the loss rate, and use a code with a redundancy which is as close as possible to the loss rate, and

Manuscript received March 14, 2005; revised October 31, 2005. Work on this paper was performed while O. Etesami was a summer intern at EPFL, Lausanne, Switzerland, and A. Shokrollahi was a full time employee of Digital Fountain, Inc.

O. Etesami is with the Computer Science Division, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: etesami@eecs.berkeley.edu).

A. Shokrollahi is with the School of Basic Sciences, and School of Computer Science and Communications, Swiss Federal Institute of Technology (EPFL), CH-1015 Lausanne, Switzerland (e-mail: amin.shokrollahi@epfl.ch).

Communicated by R. Koetter, Guest Editor for Networking and Information Theory.

Digital Object Identifier 10.1109/TIT.2006.872855

which has a reliable decoding algorithm. Tornado codes [8], [2] were developed for exactly this purpose. But these codes are almost useless if the loss rate is subject to frequent and transient changes, and hence cannot be reliably estimated. Here, the best option would be to design a code for the worst case. This leads to unnecessary overheads if the actual loss rate is smaller than the worst case assumption, and it leads to unreliable communication if the actual loss rate is larger than the worst case assumption.

Fountain codes are a new class of codes designed and ideally suited for reliable transmission of data over an erasure channel with unknown erasure probability. A Fountain code produces for a given set of k input symbols $(x_1, \ldots, x_k)$ a potentially limitless stream of output symbols $z_1, z_2, \ldots$. The input and output symbols can be bits, or more generally, they can be binary vectors of arbitrary length. The output symbols are produced independently and randomly, according to a given distribution on $\mathbb{F}_2^k$. A decoding algorithm for a Fountain code is an algorithm which can recover the original k input symbols from any set of m output symbols with high probability. For good fountain codes over the erasure channel, the value of m is very close to k, and the decoding time is close to linear in k.

LT-codes [13]–[15] were the first class of efficient Fountain codes. In this class, the distribution used to generate the output symbols is induced by a "degree distribution," which is a distribution on the numbers $1, \ldots, k$. For every output symbol, this distribution is sampled to obtain a degree d, and then d randomly chosen input symbols are selected and their values added to obtain the value of the output symbol.

A simple probabilistic analysis shows that for maximum-likelihoos (ML) decoding to have a vanishing error probability for an LT-code, the average degree of an output symbol has to grow at least logarithmically with the number of input symbols. This makes it very difficult to obtain a linear time encoder and decoder for an LT-code. Raptor codes [16] are an extension of LT-codes which solve this problem and yield easy linear time encoders and decoders. The main idea behind Raptor codes is to pre-code the input symbols using a block code with a linear time encoder and decoder. The output symbols are produced using the original input symbols together with the redundant symbols of the pre-code. Raptor codes solve the transmission problem over an unknown erasure channel in an almost optimal manner, as described in [16].

The success of Fountain codes for the erasure channel suggests that similar results may also be possible for other binary-symmetric channels. In this paper, we will investigate this question. As we will show, some of the properties of LT- and Raptor codes over the erasure channel can be carried over to any binary input memoryless symmetric channels (BIMSC), while some other properties cannot.

In practice, a Raptor code over a BIMSC can be used in the following way: the receiver collects output bits from the channel, and with each bit, it records the reliability of the bit. This reliability translates into an amount of information of the bit. The receiver collects bits until the sum of the informations of the individual bits is $k(1+\epsilon)$, where ϵ is an appropriate constant, called the *reception overhead*, or simply overhead. Once reception is complete, the receiver applies BP decoding (or any low-complexity flavor or it) to recover the input bits.

The main design problem for Raptor codes is to achieve a reception overhead arbitrarily close to zero, while maintaining the reliability and efficiency of the decoding algorithm. This problem has been solved for the erasure channel [16]. For general BIMSCs this problem is unsolved in full generality. In this paper, we will present some partial results in this paper.

The paper is organized as follows. In Sections II and III, we will introduce the main concepts behind Raptor codes and BP decoding. Then we will consider in Section IV degree distributions optimized for the erasure channel and study the residual bit-error rate (BER) of the decoder after a fixed number of iterations of the BP algorithm, as a function of the overhead chosen. It turns out that these degree distributions perform very well.

Then, we will show in Section V that the method of Gaussian approximation [17], [18] can be adapted to the case of Raptor codes. Using this method, and under some additional (and wrong) assumptions, we will derive a simple criterion for the output degree distribution to yield a good code. Surprisingly, even though this method is based on wrong assumptions, it gives a lower bound for the fraction of output bits of degree 2, which will turn out to be the correct lower bound necessary for the BP algorithm to achieve good performance. This will be proved in Section VI. Since the condition involves the output bits of degree 2, it is reminiscent of the stability condition for LDPC codes [10].

For a BIMSC $\mathcal{C}$, the new stability condition gives a lower bound for the fraction of output bits of degree 2 in terms of a certain parameter $\Pi(\mathcal{C})$ of the channel. This parameter involves the capacity of the channel, as well as the expected log likelihood of the channel output. For the case of the erasure channel, this parameter turns out to be equal to 1, independent of the erasure probability of the channel. This makes it possible to design "universal" codes for the class of erasure channels. Loosely speaking, universal Raptor codes for a given class of channels are Raptor codes that simultaneously approach capacity for any channel in that class when decoded by the BP algorithm. For channels other than the erasure channel, such as the binary input additive white Gaussian noise (BIAWGN) channel and the binary-symmetric channel (BSC) this quantity depends on the noise level of the particular channel, and is not a universal constant depending only on the channel class. This means that there are no universal Raptor codes for these important classes of channels.

In Section VII, we will prove that for a sequence of Raptor codes whose performance comes arbitrarily close to the capacity of the underlying channel, the fraction of output bits of degree 2 has to converge to $\Pi(\mathcal{C})/2$. On the negative side, the result suggests that on channels other than the BEC it is not possible to exhibit "universal" Raptor codes for a given class of communication channels, i.e., Raptor codes whose performance comes arbitrarily close to the capacity regardless of the noise of the channel. On the positive side, this result exhibits the limit value of the fraction of output bits of degree 2 in a capacity-achieving degree distribution for Raptor code, and shows that a weak form of the flatness condition [19] can be generalized to arbitrary BSCs, at least in the case of Raptor codes. This leaves some hope for the proof of a similar result for LDPC codes.

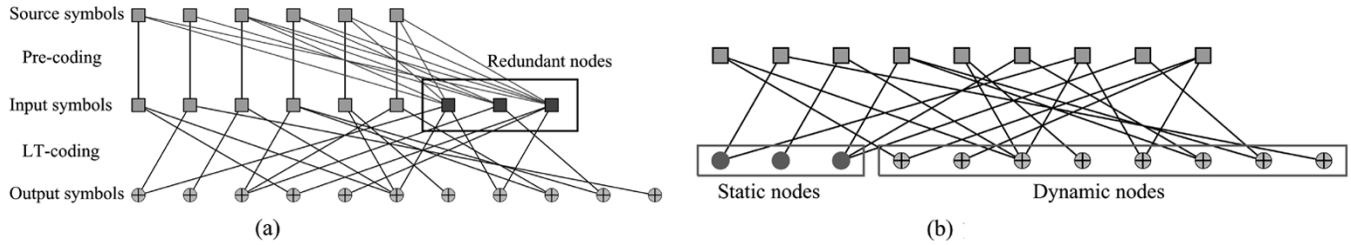


Fig. 1. (a) Raptor codes: the source symbols are appended by redundant symbols (black squares) in the case of a systematic pre-code to yield the input symbols. An appropriate LT-code is used to generate output symbols from the input symbols. (b) Decoding graph of a Raptor code. The output nodes are divided into the two categories of static and dynamic output nodes. (a) Raptor code. (b) Decoding nodes.

Since there is no hope of having universal Raptor codes for channels other than the BEC, the question arises whether one can bound the performance of Raptor codes designed for one channel when using the code for a different channel. Partial results in this direction are provided in Appendix VII. In particular, we will show that Raptor codes designed for the BEC will not perform too badly on other BSCs. More precisely, we will show that asymptotically, the overhead of universal Raptor codes for the erasure channel is at most $\log_2(\mathrm{e})$, if the codes are used on any BIMSC using the BP algorithm. This result shows that universal Raptor codes for the erasure channel simultaneously beat the cutoff rate for any BIMSC without knowing the channel beforehand.

In Section VIII, we investigate a more realistic Gaussian approximation technique, modeled after [20], and we derive some good degree distributions using random sampling and linear optimization methods.

II. Raptor Codes

Let k be a positive integer, and let $\mathcal{D}$ be a distribution on $(\mathbb{F}_2^k)^*$, the space of linear forms on $\mathbb{F}_2^k$ (also called the dual of $\mathbb{F}_2^k$). Since $\mathbb{F}_2^k$ and its dual are (noncanonically) isomorphic, $\mathcal{D}$ can be viewed as a distribution on $\mathbb{F}_2^k$ as well (after fixing standard bases). We will therefore view $\mathcal{D}$ as a distribution on $\mathbb{F}_2^k$ and its dual at the same time.

Formally, the ensemble of Fountain codes with parameter $(k, \mathcal{D})$ is an infinite vector $(X_1, X_2, X_3, \ldots)$ in which the X_i are independent random variables on $(\mathbb{F}_2^k)^*$ with distribution $\mathcal{D}$. Such an ensemble induces a probability distribution on the space of linear maps from $\mathbb{F}_2^k$ to $\mathbb{F}_2^{\mathbb{N}}$. A Fountain code with parameter $(k, \mathcal{D})$ is a mapping sampled from this distribution. The block length of a Fountain code is potentially infinite, but in applications we will solely consider truncated Fountain codes, i.e., Fountain codes with finitely many coordinates, and make frequent and implicit use of the fact that unlike block codes the length of a Fountain code is not fixed *a priori*.

The symbols produced by a Fountain code are called *output symbols*, and the k symbols from which these output symbols are calculated are called *input symbols*. The input and output symbols could be elements of $\mathbb{F}_2$, or more generally, the elements of any finite-dimensional vector space over $\mathbb{F}_2$ or any other field. In this paper, we will be primarily interested in Fountain codes over the field $\mathbb{F}_2$. For this reason, we will often use "input bits" instead of input symbols, and "output bits" instead of output symbols.

A special class of Fountain codes is furnished by LT-codes. In this class, the distribution $\mathcal{D}$ has a special form. Let $\Omega_1, \ldots, \Omega_k$ be a distribution on $\{1, \ldots, k\}$ so that Ω_i denotes the probability that the value i is chosen. Often we will denote this distribution by its generator polynomial $\Omega(x) = \sum_{i=1}^k \Omega_i x^i$. The distribution $\Omega(x)$ induces a distribution on $\mathbb{F}_2^k$ (and hence on its dual) in the following way: For any vector $v \in \mathbb{F}_2^k$, the probability of v is $\Omega_w / \binom{k}{w}$, where w is the weight of v. Abusing notation, we will denote this distribution in the following by $\Omega(x)$ again. An LT-code is a Fountain code with parameters $(k, \Omega(x))$.

Let $\mathtt{C}$ be a linear code of block length n and dimension k, and let $\Omega(x)$ be a degree distribution. A *Raptor code* with parameters $(k, \mathtt{C}, \Omega(x))$ is an LT-code with distribution $\Omega(x)$ on n symbols which are the coordinates of codewords in $\mathtt{C}$. The code $\mathtt{C}$ is called the *pre-code* of the Raptor code. The source symbols of a Raptor code are the k symbols used to construct the codeword in $\mathtt{C}$ consisting of n *input symbols*. The output symbols are the symbols generated by the LT-code from the n input symbols. The notation reflects the fact that the LT-code is applied to the encoded version of the source symbols (called input symbols), rather than the source symbols themselves. Of course, for an LT-code, the source and the input symbols are the same. A graphical presentation of a Raptor code is given in Fig. 1(a). Typically, we assume that $\mathtt{C}$ is equipped with a systematic encoding, but this is not necessary.

The complete decoding graph of length m of a Raptor code with parameters $(k, \mathtt{C}, \Omega(x))$ is a bipartite graph with n nodes on the one side (called the input nodes or the input bits) and $m + n - k$ nodes on the other (called the output nodes or the output bits), where n is the block length of $\mathtt{C}$. The output nodes of this graph belong to two categories. One set of the output nodes corresponds to m collected output symbols, and there is an edge from such an output node to all those input nodes whose sum equals the value of the output node (before transmission over the channel). We call these output nodes *dynamic* output nodes. The notation reflects the fact that this part of the graph is not fixed, and depends on the particular output nodes collected. The second set of output nodes, called the static output nodes, corresponds to the $n - k$ parity-check equations, and such an output node is connected to all those input nodes for which the sum is equal to zero. While the graph induced by the dynamic output nodes is generally sparse, the graph induced by the static output nodes is sparse only if an LDPC code (or any of its flavors, such as an IRA code) is used as a pre-code. In this paper, we will often assume that this is the case, so that the complete

decoding graph becomes a sparse graph. An example of a decoding graph for a Raptor code is given in Fig. 1(b).

The complete decoding graph is comprised of the *dynamic* and the *static* decoding graphs. The dynamic decoding graph of a Raptor code is the subgraph of the complete decoding graph which is induced by the dynamic nodes. Similarly, the static decoding graph of the code is the subgraph of the complete decoding graph induced by the static nodes. Typically, our decoding algorithms proceed by processing the dynamic decoding graph first, and and then continue decoding by processing the static decoding graph. Since we will mostly be dealing with the decoding on the dynamic decoding graph, we will in the following refer to this graph as the "decoding graph," without further qualification.

For the analysis of Raptor codes, we need to use the degree distribution in a decoding graph of the code from the perspective of the edges rather than the nodes. We denote by ι_i the probability that a randomly chosen edge in the dynamic decoding graph of the code is connected to an input node of degree i; similarly, I_i denotes the probability that a randomly chosen input node is of degree i. We denote by $\iota(x)$ and $I(x)$ the generating functions $\sum_i \iota_i x^{i-1}$ and $\sum_i I_i x^i$, respectively. By ω_i we denote the probability that a randomly chosen edge in the dynamic decoding graph is connected to an output node of degree i. Recall that Ω_i is the probability that a randomly chosen output node is of degree i, and note that ω_i and Ω_i are independent of the number of output symbols, whereas ι_i and I_i may depend on the number of output symbols. We define $\omega(x)$ as $\sum_i \omega_i x^{i-1}$. Then we have

$$\iota(x) = \frac{I'(x)}{I'(1)}$$
$$\omega(x) = \frac{\Omega'(x)}{\Omega'(1)}$$

where $f'(x)$ denotes the formal derivative of $f(x)$ with respect to x. The following proposition shows that when the number of output nodes is large, then $I(x) \sim \mathrm{e}^{\alpha(x-1)}$ and $\iota(x) \sim \mathrm{e}^{\alpha(x-1)}$, where α is the expected average node degree of the input nodes. It can be shown using standard Chernoff bounds that the average node degree is sharply concentrated around α. We will therefore often omit the qualifier "expected" and will talk of α as the average degree of the input nodes.

Proposition 1: Let N denote the number of input and output symbols of an LT-code, respectively, α denote the average degree of an input symbol, and $I(x)$ and $\iota(x)$ be defined as above. Then we have the following.

1) We have
$$I(x) = \left(\frac{\alpha}{N}x + 1 - \frac{\alpha}{N}\right)^N$$
$$\iota(x) = \left(\frac{\alpha}{N}x + 1 - \frac{\alpha}{N}\right)^{N-1}.$$

2) Assume that $\alpha^2 < N$. Then we have for all $x \in [0,1]$
$$I(x) = \mathrm{e}^{\alpha(x-1)} + O\left(\frac{\alpha^2}{N}\right)$$
$$\iota(x) = \mathrm{e}^{\alpha(x-1)} + O\left(\frac{\alpha^2}{N}\right).$$

We will prove this proposition in Appendix I.

In all the cases considered in this paper, the degree α is a constant. In this case approximating $I(x)$ and $\iota(x)$ with $\mathrm{e}^{\alpha(x-1)}$ leads to an error term of $O(1/N)$. Since our analytical results will hold asymptotically, i.e., for very large N, this error term does not affect the approximation of $\iota(x)$ by $\mathrm{e}^{\alpha(x-1)}$.

III. The Communication Channel and the BP Algorithm

In this paper we will study BIMSCs. Three examples of such channels are furnished by the BEC with erasure probability ϵ, denoted BEC(ϵ), the BSC with error probability ϵ, denoted BSC(ϵ), and the BIAWGN channel with variance σ^2, denoted BIAWGN(σ).

We consider transmission with binary antipodal signaling. Strictly speaking, with this kind of signaling, we cannot speak of XORing bits. However, we will abuse notation slightly and denote the real product of the input values as the "XOR" of the bits.

The output of a BIMSC with binary input $\{-1,+1\}$ can be identified with a pair (S,p), where $S \in \{-1,+1\}$, and p is a real number between 0 and $1/2$. The value S is interpreted as a guess of the input value before transmission over the channel, and p can be interpreted as the probability that the guess is incorrect. The channel can be identified with the probability density function (pdf) of the error probability p. For example, BEC(ϵ) is identified with the probability distribution $(1-\epsilon)\Delta_0 + \epsilon\Delta_{1/2}$, and the channel BSC($\epsilon$) is identified with the distribution Δ_ϵ, where Δ_x denotes the Dirac delta function at x. In this notation, if f denotes the pdf of the error probability p of the channel $\mathcal{C}$, then the capacity of the channel is given as

$$\mathrm{Cap}(\mathcal{C}) = 1 - \mathrm{E}[h(p)] = 1 - \int_0^{1/2} h(p)f(p)\mathrm{d}p \qquad (1)$$

where h is the binary entropy function. A different way of presenting a channel is by means of the distribution of its log-likelihood ratio (LLR). The LLR of a received symbol Y is defined as

$$\ln \frac{\Pr[X=0 \mid Y]}{\Pr[X=1 \mid Y]}$$

where X is the bit sent over the channel. The LLR of a BIMSC is often presented under the assumption that the all-zero codeword was sent over the channel. In this representation, the channel BEC(ϵ) is given by $\epsilon\Delta_0 + (1-\epsilon)\Delta_\infty$, while the channel BSC($\epsilon$) is given by the distribution $\epsilon\Delta_{\ln\left(\frac{\epsilon}{1-\epsilon}\right)} + (1-\epsilon)\Delta_{\ln\left(\frac{1-\epsilon}{\epsilon}\right)}$. The channel capacity can be given via the pdf $g(x)$ of the LLR as

$$\mathrm{Cap}(\mathcal{C}) = 1 - \int_{-\infty}^{\infty} \log_2\left(1+\mathrm{e}^{-x}\right) g(x)\mathrm{d}x. \qquad (2)$$

For example, we have

$$\mathrm{Cap}(\mathrm{BEC}(\epsilon)) = 1-\epsilon$$
$$\mathrm{Cap}(\mathrm{BSC}(\epsilon)) = 1-h(\epsilon)$$

and

$$\mathrm{Cap}(\mathrm{BIAWGN}(\sigma)) = 1 - \frac{1}{2\sqrt{\pi m}} \int_{-\infty}^{\infty} \log_2\left(1+\mathrm{e}^{-x}\right) \cdot \mathrm{e}^{-\frac{(x-m)^2}{4m}}\mathrm{d}x$$
$$m = \frac{2}{\sigma^2}.$$

All these (and a lot more) facts can be found in the upcoming book by Richardson and Urbanke [21].

The remainder of this section will give a description of the BP algorithm that is used in the decoding process of Raptor codes over BIMSCs. The algorithm proceeds in rounds. In every round, messages are passed from input bits to output bits, and then from output bits back to input bits along the edges of a decoding graph for the Raptor code. The message sent from the input bit i to the output bit o in the ℓth round of the algorithm is denoted by $\mathtt{m}_{i,o}^{(\ell)}$, and similarly the message sent from an output bit o to an input bit i is denoted by $\mathtt{m}_{o,i}^{(\ell)}$. These messages are elements in $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. We will perform additions in this set according to the following rules: $a + \infty = \infty$ for all $a \neq -\infty$, and $a - \infty = -\infty$ for all $a \neq \infty$. The values of $\infty - \infty$ and $-\infty + \infty$ are undefined. Moreover, $\tanh(\infty/2) := 1$ and $\tanh(-\infty/2) := -1$.

In the following, for every output bit o, we denote by Z_o the corresponding LLR. In round 0 of the BP algorithm the input bits send to all their adjacent output bits the value 0. Thereafter, the following update rules are used to obtain the messages passed at each round $\ell \geq 0$:

$$\tanh\left(\frac{\mathtt{m}_{o,i}^{(\ell)}}{2}\right) := \tanh\left(\frac{Z_o}{2}\right) \cdot \prod_{i' \neq i} \tanh\left(\frac{\mathtt{m}_{i',o}^{(\ell)}}{2}\right) \quad (3)$$

$$\mathtt{m}_{i,o}^{(\ell+1)} = \sum_{o' \neq o} \mathtt{m}_{o',i}^{(\ell)} \quad (4)$$

where the product is over all input bits adjacent to o other than i, and the sum is over all output bits adjacent to i other than o.

After running the BP algorithm for ℓ rounds, the LLR of each input bit i can be calculated as the sum $\sum_o \mathtt{m}_{o,i}^{(\ell)}$, where the sum is over all the output bits o adjacent to i. We then gather these LLRs, and run a decoding algorithm for the pre-code on the static decoding graph of the Raptor code, where in this phase we set the prior LLRs of the input bits to be equal to the calculated LLRs according to the preceding formula.

The neighborhood of depth ℓ of an input (output) bit i (o) consists of all the input (output) bits which are connected to i (o) by a path of length at most 2ℓ, together with all their adjacent output (input) bits. Under the assumption that this neighborhood is a tree, the BP algorithm correctly calculates the LLR of an input or output bit, given the observations of all the output bits in this tree. It is well known that for any fixed ℓ, the number of output or input bits for which the neighborhood of depth ℓ is not a tree is $o(k)$, where k is the number of input bits, and hence, for all but at most $o(k)$ output bits the BP algorithm correctly calculates the LLR. We will call the assumption that the neighborhood of depth ℓ of an input (or an output) bit is a tree, the *tree assumption*.

The messages passed during the BP algorithm are random variables. Under the tree assumption the messages passed at round ℓ from input bits to output bits have the same density function. Similarly, the messages passed at round ℓ from output bits to input bits have the same density function. In the following, we let X_ℓ denote a representative random variable which has the same density as the messages passed at round ℓ from input to output bits, and similarly, we let Y_ℓ denote a representative random variable with the same density as the messages passed from output bits to input bits at round ℓ. Furthermore, we let Z denote the random variable describing the channel LLR. Then we have the following simple result.

Proposition 2: Let X_ℓ, Y_ℓ, and Z be defined as above, let $\Omega(x)$ and $\omega(x)$ denote the output node and edge degree distributions of a Raptor code, and let α denote the average degree of the input bits in the dynamic decoding graph of the code. Further, let N denote the number of output symbols collected by the decoder. Then, under the tree assumption, we have

$$\mathrm{E}[X_{\ell+1}] = \alpha\left(1 - \frac{1}{N}\right)\mathrm{E}[Y_\ell] \quad (5)$$

and

$$\mathrm{E}\left[\tanh\left(Y_\ell/2\right)\right] = \mathrm{E}\left[\tanh\left(Z/2\right)\right]\omega\left(\mathrm{E}\left[\tanh\left(X_\ell/2\right)\right]\right). \quad (6)$$

Proof: As before, let $\iota(x) = \sum_{d \geq 1} \iota_d x^{d-1}$ denote the edge degree distribution of the input bits in the dynamic decoding graph. If we are observing the message on a random edge in the dynamic decoding graph, then the probability that the edge is connected to an input bit of degree d is ι_d, in which case the mean of this message will be $(d-1)\mathrm{E}[Y_{\ell-1}]$. Multiplying this value with the probability ι_d that the edge is connected to an input bit of degree d, we see that

$$\mathrm{E}[X_{\ell+1}] = \sum_d \iota_d (d-1)\mathrm{E}[Y_\ell] = \iota'(1)\mathrm{E}[Y_\ell].$$

The edge degree distribution of the input nodes in this graph is given by $\iota(x) = (\beta x/k + (1-\beta/k))^{N-1}$, where β is the average degree of the output symbols in the dynamic decoding graph, and k is the number of input symbols, see Proposition 1. Hence, $\iota'(1) = (N-1)\beta/k$. Note that $\alpha k = \beta N$ since this is the number of edges in the dynamic decoding graph (counted from the point of view of the input and the output symbols, respectively. This proves (5).

The proof of (6) follows the same line. The tree assumption and (4) immediately imply that

$$\mathrm{E}\left[\tanh\left(m_{o,i}^{\ell}\right) \mid \deg(o) = d\right] = z\mathrm{E}\left[\tanh(X_\ell/2)\right]^{d-1}$$

where z is $\mathrm{E}[\tanh(Z/2)]$. Therefore,

$$\begin{aligned}\mathrm{E}\left[\tanh\left(Y_\ell/2\right)\right] &= z\sum_d \omega_d \mathrm{E}\left[\tanh\left(X_\ell/2\right)\right]^{d-1} \\ &= z\omega\left(\mathrm{E}\left[\tanh\left(X_\ell/2\right)\right]\right).\end{aligned}$$

This completes the proof. □

We denote by $\mathrm{E}(\mathcal{C})$ the expectation $\mathrm{E}[\tanh(Z/2)]$. If $g(x)$ denotes the pdf of the LLR of the channel, we have

$$\mathrm{E}(\mathcal{C}) = \int_{-\infty}^{\infty} \tanh\left(\frac{x}{2}\right) g(x)\mathrm{d}x. \quad (7)$$

For example, as remarked above for BEC(ϵ), the distribution of the LLR is equal to $\varepsilon\Delta_0 + (1-\varepsilon)\Delta_\infty$, so the distribution of $\tanh(Z/2)$ is given as $\varepsilon\Delta_0 + (1-\varepsilon)\Delta_1$ and hence,

$$\mathrm{E}(\mathrm{BEC}(\epsilon)) = 1 - \epsilon. \quad (8)$$

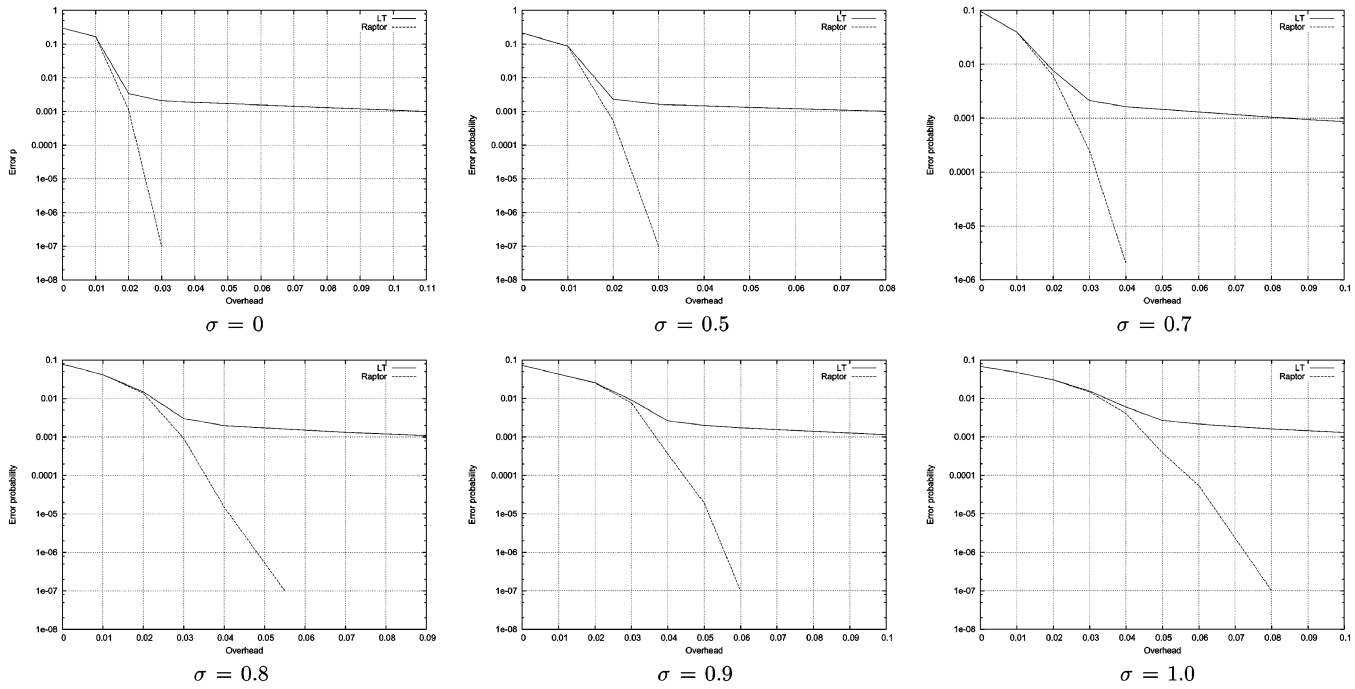


Fig. 2. Overhead versus error probability for LT- and Raptor code of length 65536 with output distribution given in (14) and with a right-Poisson, left-regular pre-code of left degree 4 and rate 0.98. The different graphs correspond to different values of the standard deviation σ of the channel BIAWGN (σ).

Similarly, for BSC(ε) the distribution of the LLR is given by $\epsilon\Delta_{\ln(\frac{\epsilon}{1-\epsilon})}+(1-\epsilon)\Delta_{\ln(\frac{1-\epsilon}{\epsilon})}$, so the distribution of $\tanh(Z/2)$ is given by $(1-\varepsilon)\Delta_{1-2\varepsilon}+\varepsilon\Delta_{2\varepsilon-1}$, and so

$$\mathrm{E}(\mathrm{BSC}(\epsilon))=(1-2\epsilon)^2. \tag{9}$$

If $\mathcal{C}$ is BIAWGN(σ), then

$$\mathrm{E}(\mathrm{BIAWGN}(\sigma))=\frac{1}{2\sqrt{\pi m}}\int_{-\infty}^{\infty}\tanh\left(\frac{x}{2}\right)\mathrm{e}^{-\frac{(x-m)^2}{4m}}\mathrm{d}x \tag{10}$$

where $m=2/\sigma^2$. For future reference, we will mention the following well-known estimates: as $\sigma\to\infty$, $\mathrm{Cap}(\mathrm{BIAWGN}(\sigma))$ behaves as for $\mathrm{E}(\mathrm{BIAWGN}(\sigma))$ as

$$\frac{1}{\ln(2)}\left(\frac{m}{4}-\frac{m^2}{16}+\frac{m^3}{48}+O(m^4)\right) \tag{11}$$

while $\mathrm{E}(\mathrm{BIAWGN}(\sigma))$ behaves as

$$\frac{1}{2}m-\frac{1}{4}m^2+\frac{5}{24}m^3+O(m^4). \tag{12}$$

These estimates can be found, for example, in [21]. However, for the reader's convenience, we will give a proof in Appendix II.

One of the main parameters of the channel that we will interested in is denoted by $\Pi(\mathcal{C})$ and defined as

$$\Pi(\mathcal{C}):=\frac{\mathrm{Cap}(\mathcal{C})}{\mathrm{E}(\mathcal{C})}. \tag{13}$$

The following equalities can be easily verified:

$$\Pi(\mathrm{BEC}(\epsilon))=1$$
$$\Pi(\mathrm{BSC}(\epsilon))=\frac{1-h(\epsilon)}{(1-2\epsilon)^2}.$$

For $\Pi(\mathrm{BIAWGN}(\sigma))$, we have the following equivalent formula:

$$\frac{1-\frac{1}{2\sqrt{\pi m}}\int_{-\infty}^{\infty}\log_2\left(1+\mathrm{e}^{-x}\right)\mathrm{e}^{-\frac{(x-m)^2}{4m}}\mathrm{d}x}{\frac{1}{2\sqrt{\pi m}}\int_{-\infty}^{\infty}\tanh\left(\frac{x}{2}\right)\mathrm{e}^{-\frac{(x-m)^2}{4m}}\mathrm{d}x}$$

where $m=2/\sigma^2$.

A random variable X on $\overline{\mathbb{R}}$ is called *symmetric* if its pdf f satisfies $f(-x)=\mathrm{e}^{-x}f(x)$ [10]. It is easy to show that the random variable describing the LLR of a symmetric channel is binary symmetric. Moreover, as was shown in [10], the random variables describing the messages passed during any round of the BP algorithm are symmetric (irrespective of the tree assumption).

Finally, we give a formal definition of the *reception overhead* of a decoder: For each received bit, let p_i be the probability that the bit was zero before transmission, and let $E=\sum_{i=1}^{m}(1-h(p_i))$, where m is the number of collected output bits to which the decoding algorithm is to be applied. We say that the decoding algorithm has a reception overhead of ε if $E=k(1+\varepsilon)$. In other words, the algorithm works with a number of output bits that is only ε away from the optimal number.

IV. Simulations of Good Distributions for the BEC

In this section, we will report on simulations we performed for degree distributions that were optimized for the BEC, as reported in [16]. Our results are analogous to those of Palanki and Yedidia [22].

Our experiments used the output distribution

$$\begin{aligned}\Omega(x)=&0.008x+0.494x^2+0.166x^3+0.073x^4\\&+0.083x^5+0.056x^8+0.037x^9+0.056x^{19}\\&+0.025x^{65}+0.003x^{66}.\end{aligned} \tag{14}$$

We chose a Raptor code with parameters $(65536,\mathtt{C},\Omega(x))$, where $\mathtt{C}$ is a right-Poisson, left-regular LDPC code of rate 0.98, as described in [16]. The communication channel is BIAWGN(σ), and the simulations were done for various values of the standard deviation σ. The results of these simulations are summarized in Fig. 2.

We performed our experiments in the following way: each time we ran enough experiments to see 200 bit errors, or 2000 decoding runs, whichever was first, and we also ran at least 200 decoding runs. Then we calculated the average fraction of bit errors at the end of the decoding process. During the decoding process, we ran the BP algorithm for at most 300 rounds.

Since the degree distribution was optimized for the BEC, the smallest overhead to ensure a good error probability is expected to occur for the case $\sigma = 0$; this is also what the simulations suggest. Moreover, as σ is increased, the corresponding overhead needs to be increased as well.

The graphs in Fig. 2 clearly show the advantage of Raptor codes over LT-codes. It is clear that for a small average degree the LT-codes exhibit a bad error floor behavior. This is due to the fact that not all the input bits will be covered by the output bits, as shown in [15] and [16].

The experiments seem to suggest that, although degree distributions optimized for the erasure channel do not perform bad on the AWGN, there is room for improvement. This will be the topic of Section V.

V. Gaussian Approximation

In [17], the authors have presented a simple method called *Gaussian approximation* which approximates message densities as a Gaussian (for regular LDPCs) or a mixture of Gaussians (for irregular LDPCs). As will be discussed below, using such an approximation it is possible to collapse the density of the messages passed at each round of the BP algorithm to a recursion for the mean of a Gaussian, and hence to a single-variable recursion.

We are going to apply similar Gaussian approximation techniques to Raptor codes. We will assume in this section that the edge degree distribution of the dynamic decoding graph of the Raptor code from the point of view of the output nodes and the input nodes is given by $\omega(x) = \sum_d \omega_d x^{d-1}$ and $\iota(x) = \sum_d \iota_d x^{d-1}$, respectively. Note that we can approximate $\iota(x)$ by $\mathrm{e}^{\alpha(x-1)}$, where α is the average degree of an input node (see Proposition 1).

A Gaussian distribution is completely specified by two quantities, its mean μ and its variance σ^2. It is possible to express a symmetric Gaussian by one parameter only (either its mean or its variance), since in this case $\sigma^2 = 2\mu$. Note that if X is a symmetric Gaussian with mean μ (and variance 2μ), then

$$\mathrm{E}\left[\tanh\left(\frac{X}{2}\right)\right] = \frac{1}{2\sqrt{\pi\mu}} \int_{-\infty}^{\infty} \tanh\left(\frac{u}{2}\right) \mathrm{e}^{-\frac{(u-\mu)^2}{4\mu}} \mathrm{d}u.$$

As in [17], we define $\varphi(x)$ for $x \in [0, \infty)$ as

$$\varphi(x) = 1 - \frac{1}{2\sqrt{\pi x}} \int_{-\infty}^{\infty} \tanh\left(\frac{u}{2}\right) \mathrm{e}^{-\frac{(u-x)^2}{4x}} \mathrm{d}u$$

for $x > 0$. φ has limit 1 at $x = 0$, and so we define $\varphi(0) = 1$. For example, we have

$$1 - \varphi\left(\frac{2}{\sigma^2}\right) = \mathrm{E}(\mathrm{BIAWGN}(\sigma)).$$

It can be verified that $\varphi(x)$ is continuous, monotonically decreasing, and convex in the interval $[0, \infty)$. As a result, the inverse function $\varphi^{-1}(x)$ exists on this interval, and is also continuous, monotonically decreasing, and convex. Moreover, (12) shows that $1 - \varphi(x) = \frac{1}{2}x + O(x^2)$, hence,

$$\varphi'(0) = -\frac{1}{2}. \tag{15}$$

We will make use of the nonvanishing of the derivative of φ at 0 later in this section.

Now we assume that the individual message sent from an input or an output node is Gaussian. The mean of a message sent from an input node of degree d at iteration $(\ell + 1)$ is given by

$$\mathrm{E}\left[\mathtt{m}_{i,o}^{(\ell+1)} \mid \deg(i) = d\right] = (d-1) \cdot \mathrm{E}\left[\mathtt{m}_{o,i}^{(\ell)}\right]$$

where $\mathrm{E}[\mathtt{m}_{o,i}^{(\ell)}]$ is the mean of $\mathtt{m}_{o,i}$ at the ℓth iteration. Therefore, considering the Gaussian mixture density of the message sent from an input node to an output node, we will have

$$\mathrm{E}\left[\tanh\left(\frac{\mathtt{m}_{i,o}^{(\ell+1)}}{2}\right)\right] = 1 - \sum_d \iota_d \varphi\left((d-1)\mathrm{E}\left[\mathtt{m}_{o,i}^{(\ell)} \mid \deg(i) = d\right]\right).$$

Next, considering the update rule for the output nodes, we can compute the mean of the Gaussian message sent from an output node with degree b. To save space, we denote by z the expectation $\mathrm{E}[\tanh(\frac{Z_o}{2})]$. Then

$$\mathrm{E}\left[\mathtt{m}_{o,i}^{(\ell+1)} \mid \deg(o) = b\right] = \varphi^{-1}\left(1 - z\left[1 - \sum_d \iota_d \varphi\left((d-1)\mathrm{E}\left[\mathtt{m}_{o,i}^{(\ell)}\right]\right)\right]^{b-1}\right).$$

We just need to keep track of the mean of the messages sent to input nodes, which we can do by

$$\mathrm{E}\left[\mathtt{m}_{o,i}^{(\ell+1)}\right] = \sum_b \omega_b \varphi^{-1}\left(1 - z\left(1 - \sum_d \iota_d \varphi\left((d-1)\mathrm{E}\left[\left(\mathtt{m}_{o,i}^{(\ell)}\right)\right]\right)\right)^{b-1}\right).$$

This finally gives us the update rule for $\mathrm{E}[\mathtt{m}_{o,i}^{(\ell)}]$.

For successful decoding under the Gaussian assumption, we need to guarantee that

$$y < \sum_b \omega_b \varphi^{-1}\left(1 - z\left(1 - \sum_d \iota_d \varphi\left((d-1)y\right)\right)^{b-1}\right).$$

This inequality cannot hold for all values of y. In fact, the monotonicity of φ^{-1} shows that the inequality cannot be valid for $y \geq \varphi^{-1}(1-z)$. However, the inequality needs to be valid around 0. So, the derivative of the left-hand side is majorized by the derivative of the right-hand side at zero. This shows that

$$\sum_b \omega_b (\varphi^{-1})'(1) z(b-1) \cdot \sum_b \sum_d \iota_d (d-1) \varphi'(0) \left(1 - \sum_d \iota_d \varphi(0)\right)^{b-2}$$

is larger than 1, where $(\varphi^{-1})'(1)$ is the derivative of the inverse function of φ at 1, i.e., the reciprocal of the derivative of φ at

0. Since $\varphi'(0) \neq 0$ by (15), we have $(\varphi^{-1})'(1)\varphi'(0) = 1$. Moreover, in the preceding sum, the contribution of the terms for $b \neq 2$ is zero, since $\sum_d \iota_d \varphi(0) = \sum_d \iota_d = 1$. This shows that

$$\omega_2 \geq \frac{1}{\alpha z}.$$

Note that $\omega_2 = 2\Omega_2/\beta$, where β is the average degree of the output nodes. Therefore, we obtain

$$\Omega_2 \geq \frac{\beta}{\alpha}\frac{1}{2z}.$$

Since the quantity β/α is the "code rate," the maximum value for β/α is the capacity of the channel. Hence, for a capacity-achieving degree distribution, this would imply

$$\Omega_2 \geq \frac{\mathrm{Cap}(\mathrm{BIAWGN}(\sigma))}{2\mathrm{E}(\mathrm{BIAWGN}(\sigma))} = \frac{\Pi(\mathcal{C})}{2}.$$

This inequality is an analogue of the stability condition in [10]. Although its derivation is based on the incorrect Gaussian assumption, it turns out that it can be proved rigorously. We will give a rigorous proof of this inequality in Section VI.

We have used the formulas above and linear programming to design degree distributions for the Gaussian channels. Unfortunately, the designs obtained this way perform rather poorly in practice. One possible reason for this is that the assumption that the messages passed from output bits to input bits are Gaussian is a very unrealistic one. Later, in Section VIII, we will introduce a different, more realistic version of this technique which will yield more practical codes.

VI. A Bound on the Fraction of Output Symbols of Degree 2 of an LT-Code

The stability condition for LDPC codes derives an upper bound on the fraction of message nodes of degree 2, such that if the fraction is smaller than this bound, then the BP algorithm converges to the correct solution if it is started at a neighborhood of the correct solution. Hence, the correct solution is a stable fixed point of the density evolution. Therefore, the stability condition for LDPC codes spells a condition for successful termination of the algorithm, given that it is close to termination. In this section, we will prove an equivalent but different condition for LT-codes. Unlike LDPC codes, our bound is a lower bound (rather than an upper bound). Moreover, this bound gives a condition for the successful "start" of the algorithm, rather than the end of it.

We use the following notation throughout.

- The decoding graph of the LT-code with input parameters $(k, \Omega(x))$ has N output bits, where $\Omega(x) = \sum_{i\geq 1} \Omega_i x^i$ is the output node degree distribution.
- $\beta = \Omega'(1)$ is the average degree of the output symbols and $\omega(x) = \sum_{i\geq 1} \omega_i x^{i-1}$ is the edge degree distribution of the output symbols.
- α is the average degree of an input node, and $\iota(x) = \sum_{i\geq 1} \iota_i x^{i-1}$ is the edge degree distribution of the input nodes. By Proposition 1, this distribution is approximately equal to $\mathrm{e}^{\alpha(x-1)}$.
- $R := k/N = \beta/\alpha$ is the nominal rate of the code (the assertion $k/N = \beta/\alpha$ can be proved by counting the number of edges in the decoding graph in two different ways).
- Y_ℓ is the log likelihood sent through a random edge from an output node to an input node at round ℓ of the BP algorithm, $\ell \geq 0$.
- μ_ℓ is the expectation of $\tanh(Y_\ell/2)$.
- X_ℓ is the log likelihood sent through a random edge from an input node to an output node at round ℓ of the BP algorithm, $\ell \geq 0$.
- ξ_ℓ is the expectation of $\tanh(X_\ell/2)$.

The proof of the following two results are provided in Appendices III and IV.

Proposition 3: For a random variable T on $\overline{R}$ let $\tau(T)$ denote $\mathrm{E}[\tanh(\frac{T}{2})]$. Suppose that X and Y are symmetric random variables on $\overline{\mathbb{R}}$. Then $X + Y$ is also symmetric, and we have

$$\tau(X+Y) \geq \tau(X)\tau(Y) - 2\tau(X)\tau(Y)$$

and

$$\tau(X+Y) \leq \tau(X) + \tau(Y) - \tau(X)\tau(Y).$$

Lemma 4: Let $\ell \geq 0$.

a) $\mu_\ell = \mathrm{E}(\mathcal{C})\omega(\xi_\ell) = \frac{\mathrm{E}(\mathcal{C})}{\beta}\Omega'(\xi_\ell)$.
b) $\frac{1}{2}(1 - \iota(1 - 2\mu_\ell)) \leq \xi_{\ell+1} \leq 1 - \iota(1 - \mu_\ell)$.

The stability condition gives an upper bound on the fraction of variable nodes of degree 2 for an LDPC code to reduce the error probability below any constant. There is an analogue of this property for LT-codes. In this case, the stability condition is a *lower* bound on the fraction of output nodes of degree 2. The following theorem explains this condition, and a partial converse thereof.

Theorem 5:

1) For all $\epsilon > 0$ there exist positive γ, δ depending only on ϵ, $\gamma < 1$, such that if $\Omega_1 \leq \delta$ and $\Omega_2 \leq (1-\epsilon)\frac{R}{2\mathrm{E}(\mathcal{C})}$, then for all $\ell \geq 0$ we have $\xi_\ell \leq \gamma$.
2) For all $\epsilon > 0$ there exists a positive $\delta > 0$ and $\ell \geq 0$ depending only on ϵ such that if $\Omega_1 > 0$ and $\Omega_2 \geq (1+\epsilon)\frac{R}{2\mathrm{E}(\mathcal{C})}$, then $\xi_\ell \geq \delta - O(\alpha/N)$.

Proof:

1) If $\xi_\ell \leq 0.75$, then a small calculation reveals that $i\xi_\ell^{i-1} \leq 3\xi_\ell^2$ for $i \geq 3$, and hence,

$$\begin{aligned}\mu_\ell &= \frac{\mathrm{E}(\mathcal{C})}{\beta}\sum_{i\geq 1} i\Omega_i \xi_\ell^{i-1} \quad \text{(by Lemma 4(a))} \\ &\leq \frac{\mathrm{E}(\mathcal{C})}{\beta}\left(\Omega_1 + 2\Omega_2\xi_\ell + 3\xi_\ell^2\sum_{i\geq 3}\Omega_i\right) \\ &\leq \frac{\mathrm{E}(\mathcal{C})}{\alpha R}\Omega_1 + \frac{1-\epsilon}{\alpha}\xi_\ell + \frac{3\mathrm{E}(\mathcal{C})}{\alpha R}\xi_\ell^2.\end{aligned}$$

The input edge degree distribution of the code is $\iota(x) = (\beta x/k + (1 - \beta/k))^{N-1}$ by Proposition 1. By Lemma 4(b), we have

$$\begin{aligned}\xi_{\ell+1} &\leq 1 - \iota(1-\mu_\ell) \\ &= 1 - (1 - \beta\mu_\ell/k)^{N-1} \leq (N-1)\beta\mu_\ell/k.\end{aligned}$$

(Here we use the inequality $(1-x)^m \leq 1-mx$ valid for all $m \geq 0$ and all $x \in [0,1]$.) Since $\beta/k = \alpha/N$, we see that $\xi_{\ell+1} < \alpha\mu_\ell$.

2) By Lemma 4(b) and the fact the above formula for $\iota(x)$ we have

$$\xi_{\ell+1} \geq \frac{1}{2}\left(1-\iota(1-2\mu_\ell)\right) = \frac{1}{2}\left(1-\left(1-\frac{2\beta\mu_\ell}{k}\right)^{N-1}\right).$$

Using the well-known inequality $(1-x)^m \leq \mathrm{e}^{-mx}$, we obtain

$$\begin{aligned}\xi_{\ell+1} &\geq \frac{1}{2}\left(1-\left(1-\frac{2\beta\mu_\ell}{k}\right)^{N-1}\right)\\ &\geq \frac{1}{2}\left(1-\mathrm{e}^{-\frac{2\beta\mu_\ell(N-1)}{k}}\right)\\ &= \frac{1}{2}\left(1-\mathrm{e}^{-2\alpha\mu_\ell}\mathrm{e}^{-\frac{2\alpha\mu_\ell}{N}}\right)\\ &= \frac{1}{2}\left(1-\mathrm{e}^{-2\alpha\mu_\ell}\left(1+O\left(\frac{\alpha}{N}\right)\right)\right)\\ &= \frac{1}{2}\left(1-\mathrm{e}^{-2\alpha\mu_\ell}\right)-O\left(\frac{\alpha}{N}\right).\end{aligned}$$

Trivially, by Lemma 4(a), we have $\mu_\ell \geq \omega_2\xi_\ell \mathrm{E}(\mathcal{C})$, hence, by our assumption, we have $\mu_\ell \geq (1+\epsilon)\xi_\ell/\alpha$. This shows that

$$\xi_{\ell+1} \geq \frac{1}{2}\left(1-\mathrm{e}^{-2(1+\epsilon)\xi_\ell}\right)-O\left(\frac{\alpha}{N}\right).$$

Let $\delta > 0$ be such that $(1-\mathrm{e}^{-2(1+\epsilon)\delta})/2 > \delta$. (Such a δ exists as can be seen from the Taylor expansion of $1-\mathrm{e}^{-2(1+\epsilon)x}$ around $x=0$.) Suppose that $\xi_\ell \leq \delta - O(\alpha/N)$. Then

$$\begin{aligned}\xi_{\ell+1} &\geq \frac{1}{2}\left(1-\mathrm{e}^{-2(1+\epsilon)\delta}\mathrm{e}^{2(1+\epsilon)O(\alpha/N)}\right)-O\left(\frac{\alpha}{N}\right)\\ &= \frac{1}{2}\left(1-\mathrm{e}^{-2(1+\epsilon)\delta}\left(1+O\left(\frac{\alpha}{N}\right)\right)\right)-O\left(\frac{\alpha}{N}\right)\\ &= \frac{1}{2}\left(1-\mathrm{e}^{-2(1+\epsilon)\delta}\right)-O\left(\frac{\alpha}{N}\right)\\ &> \delta - O\left(\frac{\alpha}{N}\right).\end{aligned}$$

So, we see that $\xi_\ell \leq \delta - O(\alpha/N)$ cannot hold for all ℓ, and we are done. □

The results of the previous theorem can be translated into assertions on the error probability of the BP decoder. For that, we need the following lemma, the proof of which can be found in Appendix V.

Lemma 6: Suppose that X is a symmetric random variable on $\overline{\mathbb{R}}$, and let t be a positive real number.

a) If $\mathrm{E}[\tanh(X/2)] \leq t$, then for all $a > 0$ we have

$$\Pr[X \leq 0] \geq \frac{\mathrm{e}^{-a}}{2}\left(1-t/\tanh(a/2)\right).$$

b) If $\mathrm{E}[\tanh(X/2)] \geq t$, then $\Pr[X \leq 0] \leq \frac{1-t}{2}$.

From this lemma and the previous theorem we can immediately deduce the following

Corollary 7: For all $\epsilon > 0$, there exist $\pi, \delta > 0$ depending only on ϵ, such that if $\Omega_1 \leq \delta$ and $\Omega_2 \leq (1-\epsilon)\frac{R}{2\mathrm{E}(\mathcal{C})}$, then the error probability of the BP decoder is at least π.

Proof: Theorem 5 part 1) shows that $\mathrm{E}[\tanh(X_\ell/2)] \leq \gamma$ for some $0 < \gamma < 1$. Let $a > 0$, and set

$$\pi = \frac{\mathrm{e}^{-a}}{4}\left(1-\gamma/\tanh(a/2)\right).$$

Lemma 6 part a) shows that $\Pr[X_\ell \leq 0] \geq 2\pi$. Since

$$\Pr[X_\ell < 0] + \frac{1}{2}\Pr[X_\ell = 0]$$

is the error probability of the decoder at round ℓ, and since

$$\Pr[X_\ell < 0] + \frac{1}{2}\Pr[X_\ell = 0] \geq \frac{1}{2}\Pr[X_\ell \leq 0] \geq \pi$$

the result follows. □

The preceding corollary is analogous to the classical stability condition for LDPC codes [10]: if the fraction of nodes of degree 2 in the decoding graph of the LT-code is too small, then the BP decoder will not be successful. Note that this type of result is opposite to the case of LDPC codes: in that case, decoding is not successful when the fraction of nodes of degree 2 is larger than dictated by the stability condition, while in our case decoding is not successful.

In Section VII, we will connect a lower bound on the expectation of the $\tanh$ with an upper bound on some mutual information. This will enable us to show that the value $\frac{\mathrm{Cap}(\mathcal{C})}{2\mathrm{E}(\mathcal{C})}$ is critical for the fraction of output bits of degree 2.

VII. Fraction of Output Symbols of Degrees One and Two

In this section, we will derive exact formulas for the fraction of output symbols of degrees one and two for Raptor codes that are to achieve capacity. For these codes, the residual error probability of the BP decoder applied to the LT-code has to be very small. This residual error is then decoded using the pre-coder. For this reason, our investigation will be solely concerned with the LT part of the decoding process, and would like to study under what circumstances the residual error probability of the BP decoder applied to the LT-code is small.

Let d be a fixed integer, and let $x_1, \ldots, x_k$ be the input bits of an LT-code with distribution x^d. We denote by x the vector $(x_1, \ldots, x_k)$. Further, let $y_1, \ldots, y_n$ be output bits of this LT-code, which are supposed to have been received after transmission through a BIMSC $\mathcal{C}$. Analogously, we denote by y the vector $(y_1, \ldots, y_n)$. Assume that the neighborhood of depth ℓ of T of the output bits is *not* a tree. Further, let X_ℓ and Y_ℓ be prototype random variables denoting the messages of the BP algorithm passed at round ℓ from the input bits to the output bits, and from the output bits to the input bits, respectively.

In this section, we will use the following result whose proof is given in Appendix VI.

Theorem 8: Assume that there is some $t > 0$ such that $\mathrm{E}[\tanh(X_\ell/2)] > t$. Then there exists a constant $\epsilon > 0$ depending only on t, on the degree d, and on the channel $\mathcal{C}$, such that

$$I(x;y) \leq n\left(\mathrm{Cap}(\mathcal{C}) - \epsilon\right) + T\mathrm{Cap}(C).$$

Theorem 8 is, in fact, a weak generalization of the "Flatness Condition" [19]. Indeed, the theorem shows that if a sequence of Raptor codes is to achieve capacity, then asymptotically the outgoing error probability needs to be equal to the incoming error probability for BP applied to the LT-code.

A sequence of Raptor codes with parameters $(k_m, C_m, \Omega^{(m)}(x))$, $m \geq 1$, is called *capacity-achieving* for a channel $\mathcal{C}$ if the following conditions hold: a) k_m goes to ∞ as m grows, and b) the error probability of the BP algorithm applied to $k_m/\mathrm{Cap}(\mathcal{C}) + o(k_m)$ output symbols of the mth code in the sequence approaches zero as m approaches infinity.

The following lemma states an obvious fact: for a Raptor code that operates at a rate very close to the capacity of the channel, the mutual information between the input bits and the output bits has to be the maximum possible value up to terms of the form $o(n)$.

Lemma 9: Suppose that $(k_m, C_m, \Omega^{(m)}(x))$, $m \geq 1$, is a capacity-achieving sequence of Raptor codes for the BIMSC $\mathcal{C}$. Then, for any set of output symbols $z = (z_1, \ldots, z_n)$ of the mth code in the sequence, we have

$$I(x; z) \geq \min\{k_m - o(k_m), n\mathrm{Cap}(\mathcal{C}) - o(n)\}$$

where $x = (x_1, \ldots, x_{k_m})$ are the input bits of the Raptor code.

Proof: Suppose that there is some $\epsilon > 0$ such that for infinitely many m there exist output bits $y_1, \ldots, y_n$ such that

$$I(x; y_1, \ldots, y_n) \leq n(\mathrm{Cap}(\mathcal{C}) - \epsilon).$$

Clearly, n has to be smaller than $k_m/\mathrm{Cap}(\mathcal{C}) + o(k_m)$ since the sequence of Raptor codes is capacity-achieving. Add to $y_1, \ldots, y_n$ an additional number $t := k_m/\mathrm{Cap}(\mathcal{C}) + o(k_m) - n$ of further output bits $y_{n+1}, \ldots, y_{n+t}$. Then $I(x; y_1, \ldots, y_{n+t})$ is at least $k_m - o(k_m)$ since the sequence of Raptor codes is capacity-achieving. On the other hand

$$\begin{aligned} I(x; y_1, \ldots, y_{n+t}) &\leq I(x; y_1, \ldots, y_n) \\ &\quad + I(x; y_{n+1}, \ldots, y_{n+t}) \\ &\leq n(\mathrm{Cap}(\mathcal{C}) - \epsilon) + t\mathrm{Cap}(\mathcal{C}) \\ &= n(\mathrm{Cap}(\mathcal{C}) - \varepsilon) \\ &\quad + k_m - n\mathrm{Cap}(\mathcal{C}) + o(k_m) \\ &= k_m - n\epsilon + o(k_m) \end{aligned}$$

which is a contradiction. □

Theorem 10: Suppose that $\big(k_m, \mathtt{C}_m, \Omega^{(m)}(x)\big)$, $m \geq 1$, is a capacity-achieving sequence of Raptor codes for the channel $\mathcal{C}$ and suppose that $\mathrm{E}(\mathcal{C}) \neq 0$. Then we have

$$\forall\, m\colon \quad \Omega_1^{(m)} > 0 \text{ and } \lim_{m\to\infty} \Omega_1^{(m)} = 0. \tag{16}$$

Proof: First, we prove that $\Omega_1^{(m)}$ has to be larger than zero for the BP algorithm to start. Assume that $\Omega_1^{(m)} = 0$. Then all the messages going from output nodes to input nodes in the decoding graph in the first round are zero (in the LLR domain), and hence, the messages will be zero throughout the decoding process.

Using Theorem 8 we now give a proof that for a capacity-achieving sequence, we need to have $\lim_m \Omega_1^{(m)} = 0$. An alternative proof is given in Appendix VIII.

Suppose that $\Omega_1^{(m)} > \epsilon$ for infinitely many m. Consider the graph formed between the input symbols and the output symbols of degree one. Since all the output symbols are of degree one, this graph is a forest, i.e., a disjoint union of trees. We therefore refer to it as the "degree-one forest" in the following. If n denotes the number of output symbols in the original Raptor code, then the number of output symbols of degree one is sharply concentrated around its expectation $\epsilon n = \delta k_m$, where $\delta = \epsilon n/k_m$.

Let $I(x)$ denote the input symbol degree of the degree-one forest. In other words, if I_i denotes the probability that a randomly chosen input symbol in the degree-one forest is of degree i, then $I(x) = \sum_{i\geq 0} I_i x^i$. An argument similar to Proposition 1 shows that $I(x) = \mathrm{e}^{\delta(x-1)} + O(\delta^2/\epsilon n)$.

We will now estimate the expectation of the messages passed from input symbols to output symbols in the first round of the BP algorithm on the degree-one forest. In the first round, output symbols of degree one send the channel LLR to their adjacent input symbols. The expectation of these messages is $\mathrm{E}(\mathcal{C})$. Thereafter, input symbols that are connected to d output symbols in the degree-one forest send a message whose expectation is $(d-1)\mathrm{E}(\mathcal{C})$. It follows that the expectation of the of the messages passed from input to output symbols in the first round of the BP algorithm is $\mathrm{E}(\mathcal{C}) \sum_{d\geq 2} I_d(d-1)$. Since

$$\sum_{d\geq 2} I_d x^d = \mathrm{e}^{\delta(x-1)} - \mathrm{e}^{-\delta}(1 + \delta x) + O(\delta^2/n)$$

we see that this expectation equals $\mathrm{E}(\mathcal{C})\delta(1 - \mathrm{e}^{-\delta}) + O(\delta^2/n)$. Hence, for large enough n, this expectation is strictly positive, since $\mathrm{E}(\mathcal{C}) \neq 0$. It follows from Theorem 8 that there exists a positive η such that the mutual information between the input symbols of the Raptor code and the output symbols of degree one is at most $\epsilon n(\mathrm{Cap}(\mathcal{C}) - \eta)$, which contradicts the fact that the sequence is capacity-achieving. □

Theorem 11: Suppose that $\big(k_m, \mathtt{C}_m, \Omega^{(m)}(x)\big)$, $m \geq 1$, is a capacity-achieving sequence of Raptor codes for the BIMSC $\mathcal{C}$ and suppose that $\mathrm{Cap}(\mathcal{C}) \neq 0$. Then we have

$$\lim_{m\to\infty} \Omega_2^{(m)} = \frac{\Pi(\mathcal{C})}{2}. \tag{17}$$

Proof: Suppose that $\Omega_2^{(m)} > (1+\epsilon)\Pi(\mathcal{C})/2$ for infinitely many m. Since $\Omega_1^{(m)} > 0$ by Theorem 10, Theorem 5 part 2) implies that the expectation $\mathrm{E}[\tanh(X_\ell/2)]$ of the messages passed at the ℓth round of BP is larger than δ, for some $\delta > 0$, if m is large enough. Let $y_1, \ldots, y_n$ denote the output bits of degree 2, and set $y = (y_1, \ldots, y_n)$. By Theorem 8, there exists a constant $\eta > 0$ depending on δ and the channel, such that $I(x; y) \leq n(\mathrm{Cap}(\mathcal{C}) - \eta) + T\mathrm{Cap}(\mathcal{C})$, where T is the number of output symbols if degree 2 whose neighborhood of depth ℓ is not a tree. (The theorem also assumes that the constant depends on the degree of the output symbols; but since this degree is fixed to two in our application, we can disregard this dependency.) A standard argument shows that $T = o(n)$. (See, for example, [2] or [23].) Hence, $I(x; y) = n(\mathrm{Cap}(\mathcal{C}) - \eta) + o(n)$, which contradicts Lemma 9.

On the other hand, suppose that $\Omega_2^{(m)} < \Pi(\mathcal{C})\frac{1-\varepsilon}{2}$ for infinitely many m. Let δ and γ be the constants given in Theorem 5 part 1). Since $\Omega_1^{(m)} \to 0$ as $m \to \infty$ by Theorem 10, there exists some m_0 such that $\Omega_1^{(m)} \leq \delta$ for $m \geq m_0$. Then

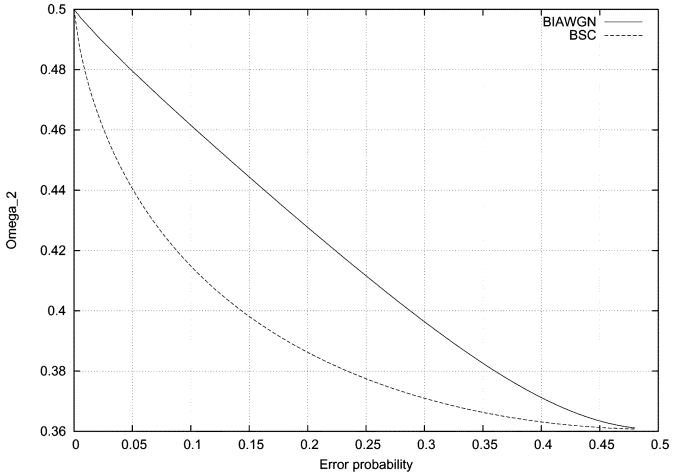


Fig. 3. $\Omega_2(\mathcal{C})$ as a function of the error probability of the channel for the BIAWGN and the BSC.

Theorem 5 part 1) implies that there is some $\gamma < 1$ such that $\mathrm{E}[\tanh(X_\ell/2)] < \gamma$ for all ℓ. This shows that the error probability of BP cannot converge to zero. □

In the following we define

$$\Omega_2(\mathcal{C}) := \frac{\Pi(\mathcal{C})}{2}.$$

By virtue of the previous result, this quantity is the asymptotic fraction of output symbols of degree 2 for a Raptor code achieving the capacity of the BIMSC $\mathcal{C}$. We have the following results.

Proposition 12: Suppose that the BIMSC $\mathcal{C}$ is either BEC(ϵ), or BSC(ϵ), or BIAWGN(σ), where $\epsilon, \sigma > 0$. Then we have

1) $0 \leq \Omega_2(\mathcal{C}) \leq 1/2$;
2) $\Omega_2(\mathcal{C}) = 1/2$ if and only if $\mathcal{C} = \mathrm{BEC}(\epsilon)$;
3) if $\mathcal{C}$ is BSC(ϵ) or $\mathcal{C}$ is BIAWGN(σ), then $\Omega_2(\mathcal{C}) \geq 1/\ln(16)$;
4) $\lim_{\epsilon \to 1/2} \Omega_2(\mathrm{BSC}(\epsilon)) = \lim_{\sigma \to \infty} \Omega_2(\mathrm{BIAWGN}(\sigma)) = 1/\ln(16)$.

Proof: Results 1) and 2) are proved by simple examination for which we refer to Fig. 3. It is also not hard to see that the function $\Omega_2(\mathrm{BSC}(\epsilon))$ is a monotonically decreasing function of ϵ, and that $\Omega_2(\mathrm{BIAWGN}(\sigma))$ is a monotonically decreasing function of σ. We therefore concentrate on proving 4) which would also prove 3). We first show that $\lim_{\epsilon \to 1/2} \Omega_2(\mathrm{BSC}(\epsilon)) = 1/\ln(16)$. To see this, note that by l'Hospital's rule, this limit equals

$$\frac{-h''(1/2)}{16} = \frac{\frac{4}{\ln(2)}}{16} = \frac{1}{4\ln(2)} = \frac{1}{\ln(16)}.$$

To prove that $\lim_{\sigma \to \infty} \Omega_2(\mathrm{BIAWGN}(\sigma)) = 1/\ln(16)$, we use the estimates (11) and (12) from Section III, which gives us

$$\begin{aligned}\Omega_2(\mathrm{BIAWGN}(\sigma)) &= \frac{1}{4\ln(2)} \left(\frac{m}{2 \cdot \frac{m}{2}} + O(m) \right) \\ &= \frac{1}{4\ln(2)} (1 + O(m)).\end{aligned}$$

The result follows, since m approaches 0 as σ approaches infinity. □

The preceding result seems to suggest that $\Omega_2(\mathcal{C})$ converges to $1/\ln(16)$ when $\mathcal{C}$ is a BIMSC whose error probability converges to $1/2$. In fact, it has been proved independently by Pakzad [24] and Sasson [25] that this is true for a large class of BSCs. In Appendix IX, we will reproduce Pakzad's proof.

We finish this section with an important remark on the nonexistence of universal Raptor codes for important classes of channels such as the BIAWGN and the BSC. Let $\boldsymbol{C}$ be a class of BIMSCs. We call a sequence of Raptor codes with parameters $(k_m, C_m, \Omega^{(m)}(x))$, $m \geq 1$ *universal* for the class $\boldsymbol{C}$, if the sequence is capacity-achieving simultaneously for all $\mathcal{C} \in \boldsymbol{C}$. For example, [16] exhibits a sequence of universal Raptor codes for the class of erasure channels. The results of this section imply the following.

Corollary 13: Let $\boldsymbol{C}$ be a class of BIMSCs. If there exists a universal sequence of Raptor codes for $\boldsymbol{C}$, then there exists π such that for all $\mathcal{C} \in \boldsymbol{C}$ we have $\Pi(\mathcal{C}) = \pi$. In particular, there are no universal Raptor codes for the classes of BIAWGN and BSC channels.

Proof: For any given $\mathcal{C} \in \boldsymbol{C}$ we need to have

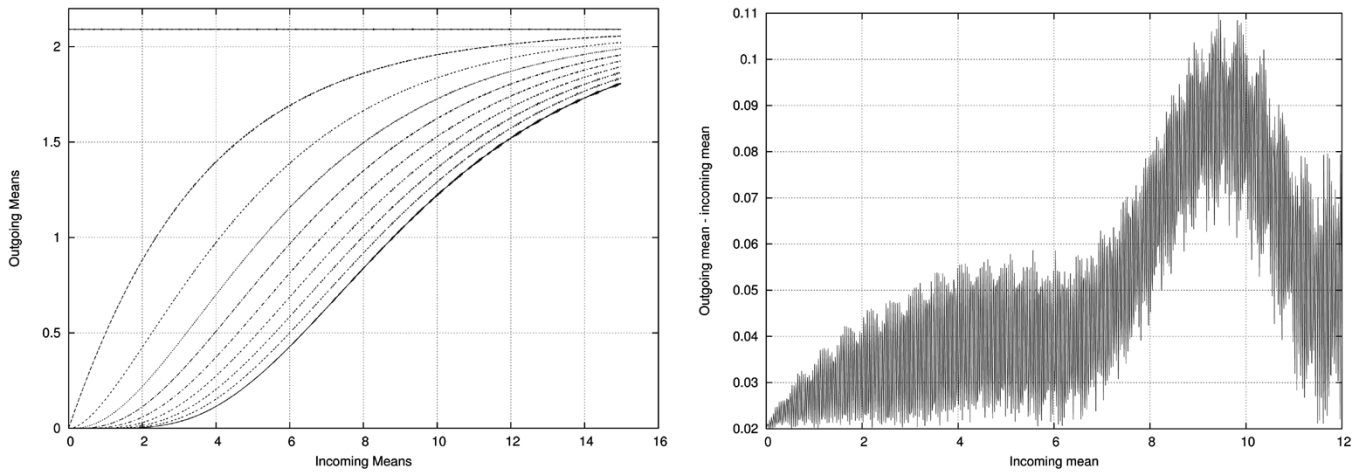
$$\lim_{m \to \infty} \Omega_2^{(m)} = \Pi(\mathcal{C}).$$

This shows that $\Pi(\mathcal{C})$ has to be constant on $\boldsymbol{C}$. For the classes of BIAWGN and BSC channels the value $\Pi(\mathcal{C})$ depends on the particular noise parameter of the channel, so universal Raptor codes cannot exist for these channel classes. □

VIII. A More Refined Gaussian Approximation

In this section, we will assume that the communication channel is a BIAWGN with variance σ^2. Requiring all the messages passed at every iteration of the BP algorithm to be Gaussian is very strong, and often wrong. Simulations and computations suggest that the messages passed from output bits of small degree are very far from being Gaussian random variables. On the other hand, the messages passed from input bits at every round are close to Gaussian. The rationale for this is simple: these messages are obtained as a sum of independent random variables of finite mean and variance from the same distribution. If the number of these additions is large, then, by the central limit theorem, the resulting density function is close to a Gaussian.

This is of course a heuristic assumption, but it seems that it is much closer to the truth than the "all-Gaussian"-assumption of Section V. In this section, we will assume that at any given round, the messages from input bits are symmetric Gaussian random variables with the same distribution, and we will derive a recursion for their means. Under the assumption that the codeword sent over the channel is the all-zero codeword (which we can do by the symmetry of the channel), we want the mean to increase from iteration to iteration. This condition implies linear inequalities for the unknown coefficients of the output degree distribution, and leads to a linear programming problem which can be easily solved using any of the standard algorithms for this task. Our solution is an adaptation of a method of Ardakani and Kschischang [20] to the case of Raptor codes.



(a) (b)

Fig. 4. (a) Mean of the messages going out of input nodes versus their difference to the mean of the messages going out of output nodes for the above degree distribution. (b) Outgoing mean of messages passed from input bits to check bits as a function of the means of the incoming messages under the assumption that the incoming messages are symmetric Gaussians. The different graphs correspond to different degrees of the output symbol, starting from one and going to 10. The standard deviation of the underlying AWGN channel is 0.9786.

Let μ denote the mean of the symmetric Gaussian distribution representing the messages passed from input bits to output bits at some round ℓ. The message passed from an output bit of degree d to an input bit in the same round has an expectation equal to

$$2\mathrm{E}\left[\operatorname{atanh}\left(\tanh\left(\frac{Z}{2}\right)\prod_{i=1}^{d-1}\tanh\left(\frac{X_i}{2}\right)\right)\right] =: f_d(\mu)$$

where $X_1,\ldots,X_{d-1}$ are independent symmetric Gaussian random variables of mean μ, and Z is the LLR of the channel. If α denotes the average input bit degree, then the mean of the messages passed from input bits to output bits at round $\ell+1$ of the algorithm equals

$$\alpha\sum_d \omega_d f_d(\mu)$$

where $\omega(x)=\sum_d \omega_d x^{d-1}$ is the output edge degree distribution of a decoding graph for the Raptor code.

The objective is to keep the average degree of the output bits as close as possible to $\alpha\mathrm{Cap}(\mathcal{C})$, while respecting the condition

$$\mu < \alpha\sum_d \omega_d f_d(\mu) \tag{18}$$

for all $\mu>0$. In other words, we want $\mathrm{Cap}(\mathcal{C})\alpha\sum_d \frac{\omega_d}{d}$ to be as close to 1 as possible, while ensuring the inequality (18).

If only a finite number of the ω_d's are nonzero, then it is impossible to satisfy (18) for all $\mu\geq 0$. But it is also not necessary to do so. If we assume that (18) holds only for a range of μ, say $\mu\in(0,\mu_0)$, then this would mean that at the time the BP decoder stops on the LT-code, the reliability of the output bits is large enough so that a pre-code of high rate would be sufficient to finish off the decoding process.

In practice, it is possible to transform (18) to a linear programming problem in the following way. We fix σ^2, α, μ_0, and integers N and D. Further, we choose N equidistant points $\mu_{N-1}<\mu_{N-2}<\cdots<\mu_1<\mu_0$ in the interval $(0,\mu_0]$, and minimize

$$\mathrm{Cap}(\mathcal{C})\alpha\sum_{d=1}^{D}\frac{\omega_d}{d}$$

subject to the three constraints

1) $\forall\, i=0,\ldots,N-1:\quad \alpha\sum_{d=1}^{D}\omega_d f_d(\mu_i)>\mu_i$
2) $\sum_{d=1}^{D}\omega_d = 1$
3) $\forall\, d=1,\ldots,D:\quad \omega_d\geq 0.$

This linear program can be solved by standard means, e.g., the simplex algorithm.

It remains to show how to calculate $f_d(\mu_i)$ for a given d. This can be done either by means of calculating the distribution of

$$2\operatorname{atanh}\left(\tanh(Z/2)\prod_{i=1}^{d-1}\tanh(X_i/2)\right)$$

or by simply sampling from this distribution many times and calculating an empirical mean. The latter is very fast, and can be implemented very easily. Fig. 4 (a) shows the graphs of $f_d(\mu)$ for $0\leq\mu\leq 16$, $\sigma=0.9786$, and $d=1,2,\ldots,10$. In this example, we used 100 000 samples per value, and discretized the interval of the means using a step size of 0.01.

These graphs were obtained by progressive sampling. As can be seen, the accuracy of this method decreases with increasing mean (and hence variance, since the densities are assumed to be symmetric). Nevertheless, these approximations provide a fast and robust means of designing good output degree distributions. We include one example of our optimization technique for the value $\sigma=0.977$. The corresponding output degree distribution is given by

$$\begin{aligned}\Omega(x) =& 0.006x + 0.492x^2 + 0.0339x^3 + 0.2403x^4\\ &+ 0.006x^5 + 0.095x^8 + 0.049x^{14} + 0.018x^{30}\\ &+ 0.0356x^{33} + 0.033x^{200}.\end{aligned}$$

The average output degree equals 11.843. Fig. 4 (b) shows the graph of the difference between the outgoing means minus the incoming means at an input node, versus the incoming mean at the input node. The rather "blurry" picture is an artifact of the fact that we used random sampling rather than density evolution

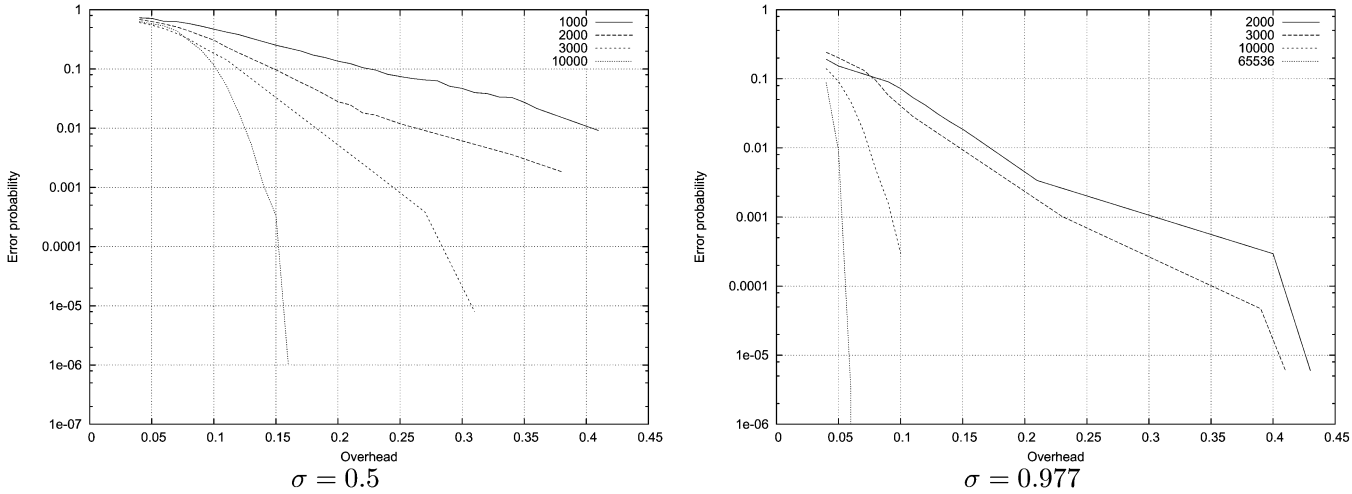


Fig. 5. Overhead versus the error probability for the above degree distribution for various input lengths, and $\sigma = 0.5$ and $\sigma = 0.977$.

to calculate the outgoing means. Nevertheless, this method has the advantage of taking into account some of the variance in the decoding process, and hence the degree distributions produced from these data should perform well in practice.

Fig. 5 gives a plot of the error probability of the decoder versus the overhead of the Raptor code. In this example, we did not use any pre-code, so that the code used is actually an LT-code. As can be seen, for a fixed residual BER, the overhead decreases with the length of the code.

Appendix I
Proof of Proposition 1

Fix an input symbol. For every generated output symbol, the probability that this input symbol is a neighbor of the output symbol is β/k, where k is the number of input symbols, and β is the average degree of the output symbols. Set $p := \beta/k$. Note that $p = \alpha/N$ as well. The probability that an input symbol is of degree ℓ is

$$\binom{N}{\ell} p^{\ell}(1-p)^{N-\ell}.$$

It turns out that $I(x) = (px + (1-p))^N$ and $\iota(x) = I'(x)/I'(1) = (px + (1-p))^{N-1}$. This proves part 1).

For part 2), we will concentrate on proving the assertion of Proposition 1 for $I(x)$; the assertion for $\iota(x)$ is done completely analogously.

We have

$$I(x) = (1-p)^N \left(\frac{px}{1-p} + 1\right)^N.$$

Noting that $\beta/k = \alpha/N$, we obtain

$$\begin{aligned}(1-p)^N &= \exp\left(-N\left(\frac{\beta}{k} + O\left(\frac{\beta^2}{k}\right)\right)\right)\\ &= \exp\left(-\alpha + O\left(\frac{\alpha^2}{N}\right)\right)\\ &= \exp(-\alpha)\left(1 + O\left(\frac{\alpha^2}{N}\right)\right)\end{aligned}$$

where $\exp(t) = \mathrm{e}^t$, and the last equality follows from the Taylor expansion of $\exp(t)$ around zero. Similarly, we have

$$\begin{aligned}\left(\frac{px}{1-p} + 1\right)^N &= \exp\left(N \cdot \left(\frac{px}{1-p} + O\left(\frac{\alpha^2 x^2}{N}\right)\right)\right)\\ &= \exp\left(\alpha x\left(1 + O\left(\frac{\beta}{k}\right)\right) + O\left(\frac{\alpha^2 x^2}{N}\right)\right)\\ &= \exp\left(\alpha x + O\left(\frac{\alpha^2 x}{N}\right)\right)\\ &= \exp(\alpha x)\left(1 + O\left(\frac{\alpha^2 x}{N}\right)\right).\end{aligned}$$

The one before last equality follows from $\alpha\beta/k = \alpha^2/N$ and $x^2 \le x$ for $x \in [0,1]$. Altogether we obtain

$$I(x) = \mathrm{e}^{\alpha(x-1)}\left(1 + O\left(\frac{\alpha^2}{N}\right)\right) = \mathrm{e}^{\alpha(x-1)} + O\left(\frac{\alpha^2}{N}\right)$$

since $\mathrm{e}^{\alpha(x-1)} \le 1$ for $x \in [0,1]$ (note that $\alpha > 0$). This concludes the proof.

Appendix II
Estimates for the Capacity of the BIAWGN Channel

In this appendix, we will prove (11) and (12). Let $m = \frac{2}{\sigma^2}$. Both proofs are based on the following observation: Let $f(x)$ be a kernel function which has a Taylor expansion at zero, say $f(x) = \sum_{k=0}^{\infty} f_k x^k$. Then

$$\frac{1}{2\sqrt{\pi m}} \int_{-\infty}^{\infty} \mathrm{e}^{-(x-m)^2/4m} f(x)\mathrm{d}x = \sum_{k=0}^{\infty} f_k \mathrm{E}[X^k]$$

where X is a symmetric Gaussian random variable with mean m. The higher moments for the Gaussian distribution can be calculated by means of the following integral:

$$\mathrm{E}[X^k] = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} t^n \mathrm{e}^{-x^2}\mathrm{d}x$$

where $t = 2x\sqrt{m} + m$. The first moments can be calculated as

$$\begin{aligned}
\mathrm{E}[X^0] &= 1 \\
\mathrm{E}[X^1] &= m \\
\mathrm{E}[X^2] &= m^2 + 2m \\
\mathrm{E}[X^3] &= m^3 + 6m^2 \\
\mathrm{E}[X^4] &= m^4 + 12m^3 + 12m^2 \\
\mathrm{E}[X^5] &= m^5 + 20m^4 + 60m^3 \\
\mathrm{E}[X^6] &= m^6 + 30m^5 + 180m^4 + 120m^3 \\
\mathrm{E}[X^7] &= m^7 + 42m^6 + 420m^5 + 840m^4 \\
&\vdots \quad \vdots
\end{aligned}$$

To calculate a series expansion of the capacity of the BIAWGN around $m = 0$, we use the series expansion of $f(x) = \ln(1 + \mathrm{e}^{-x})$

$$\begin{aligned}
&\log_2(1 + \mathrm{e}^{-x}) \\
&\quad = 1 - \frac{1}{\ln(2)} \left(\frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{192}x^4 - \frac{1}{2880}x^6 + O(x^8) \right).
\end{aligned}$$

This gives us

$$\mathrm{Cap}(\mathrm{BIAWGN}(\sigma)) = \frac{1}{\ln(2)} \left(\frac{m}{4} - \frac{m^2}{16} + \frac{m^3}{48} + O(m^4) \right)$$

which proves assertion (11).

To prove (12), we calculate a series expansion for $\tanh(x/2) = 1 - \frac{2}{\mathrm{e}^x + 1}$

$$\tanh\left(\frac{x}{2}\right) = \frac{1}{2}x - \frac{1}{24}x^3 + \frac{1}{240}x^5 + O(x^7).$$

This yields

$$\mathrm{E}(\mathrm{BIAWGN}(\sigma)) = \frac{1}{2}m - \frac{1}{4}m^2 + \frac{5}{24}m^3 + O(m^4),$$

which proves (12).

Appendix III
Proof of Proposition 3

Let $f_t(x)$ be the pdf given by

$$f_t(x) = \frac{\mathrm{e}^t}{\mathrm{e}^t + 1}\Delta_t(x) + \frac{1}{\mathrm{e}^t + 1}\Delta_{-t}(x)$$

where Δ_z is the Dirac delta function with peak at z. The corresponding random variable will be denoted by Z_t. By $T(t)$ we denote the quantity $\mathrm{E}[\tanh(Z_t/2)] = \tau(Z_t)$.

Lemma 14: For $x, y \in \overline{\mathbb{R}}$ we have

$$\begin{aligned}
\tau(Z_x + Z_y) &\geq T(x) + T(y) - 2T(x)T(y) \\
\tau(Z_x + Z_y) &\leq T(x) + T(y) - T(x)T(y).
\end{aligned}$$

Proof: It is easily checked that

$$T(x) = (\mathrm{e}^x - 1)^2/(\mathrm{e}^x + 1)^2.$$

Let f_t denote the pdf of Z_t. The pdf of $Z_x + Z_y$ is given by

$$\frac{1 + \mathrm{e}^{x+y}}{(1 + \mathrm{e}^x)(1 + \mathrm{e}^y)} f_{x+y} + \frac{\mathrm{e}^x + \mathrm{e}^y}{(1 + \mathrm{e}^x)(1 + \mathrm{e}^y)} f_{x-y}.$$

Therefore,

$$\begin{aligned}
\tau(Z_x + Z_y) = \frac{1 + \mathrm{e}^{x+y}}{(1 + \mathrm{e}^x)(1 + \mathrm{e}^y)} T(x + y) \\
+ \frac{\mathrm{e}^x + \mathrm{e}^y}{(1 + \mathrm{e}^x)(1 + \mathrm{e}^y)} T(x - y).
\end{aligned}$$

This shows that

$$\tau(Z_x + Z_y) = \frac{T(x) + T(y) - 2T(x)T(y)}{1 - T(x)T(y)}.$$

Since $0 \leq T(x), T(y) < 1$, we have

$$T(x) + T(y) - 2T(x)T(y) \leq \frac{T(x) + T(y) - 2T(x)T(y)}{1 - T(x)T(y)}$$

and

$$\frac{T(x) + T(y) - 2T(x)T(y)}{1 - T(x)T(y)} \leq T(x) + T(y) - T(x)T(y)$$

which proves the assertion. □

Proof: (Of Proposition 3) It is easily proved that if X and Y are symmetric, then so is $X + Y$.

Let f and g denote the pdfs of X and Y, respectively. Then we have

$$\begin{aligned}
&\tau(X + Y) \\
&= \int_{y=0}^{\infty} \int_{x=0}^{\infty} \tau(Z_x + Z_y)(1 + \mathrm{e}^{-x})(1 + \mathrm{e}^{-y}) f(x) g(y) \mathrm{d}x \mathrm{d}y.
\end{aligned}$$

Using the upper bound of the previous lemma we obtain the following upper bound for $\mathrm{E}[\tanh((X + Y)/2)]$:

$$\begin{aligned}
&\int_{y=0}^{\infty} \int_{x=0}^{\infty} (T(x) + T(y) - T(x)T(y)) \\
&\quad \cdot (1 + \mathrm{e}^{-x})(1 + \mathrm{e}^{-y}) f(x) g(y) \mathrm{d}x \mathrm{d}y \\
&\quad = \int_{x=0}^{\infty} T(x) f(x)(1 + \mathrm{e}^{-x}) \mathrm{d}x \\
&\qquad + \int_{x=0}^{\infty} T(y) g(y)(1 + \mathrm{e}^{-y}) \mathrm{d}y \\
&\qquad - \int_{x=0}^{\infty} T(x)T(y) g(x) g(y)(1 + \mathrm{e}^{-x})(1 + \mathrm{e}^{-y}) \mathrm{d}y \\
&\quad = \tau(X) + \tau(Y) - \tau(X)\tau(Y).
\end{aligned}$$

The lower bound in the assertion of the proposition follows in a similar manner from the lower bound of the previous lemma. □

APPENDIX IV
PROOF OF LEMMA 4

For $i \geq 1$ let $Y_\ell^{(j)}$, $j = 1, \ldots, i$ be i independent copies of Y_ℓ, and set

$$a_\ell^{(i)} := \mathrm{E}\left[\tanh\left(\frac{1}{2}\sum_{j=1}^{i} Y_\ell^{(i)}\right)\right].$$

.

a) This equation follows directly from the density evolution equations.

b) By Proposition 3 we have for all $i \geq 1$ we have

$$a_\ell^{(i)} \geq a_\ell^{(i-1)}(1-2\mu_\ell) + \mu.$$

This shows that

$$a_\ell^{(i)} \geq \mu_\ell \sum_{j=0}^{i-1}(1-2\mu_\ell)^j = \frac{1}{2}\left(1-(1-2\mu_\ell)^i\right).$$

Since $\xi_{\ell+1} = \sum_i \iota_i a_\ell^{(i-1)}$, we conclude that

$$\xi_{\ell+1} \geq \frac{1}{2}(1-\iota(1-2\mu_\ell))$$

since $\sum_i \iota_i x^{i-1} = \iota(x)$. The other inequality is proved similarly: Proposition 3 implies that

$$a_\ell^{(i)} \leq \mu_\ell \sum_{j=0}^{i-1}(1-\mu_\ell)^j = 1-(1-\mu_\ell)^i$$

and we proceed the same way as with the lower bound.

APPENDIX V
PROOF OF LEMMA 6

Let $f(x)$ be the pdf of X. Since X is symmetric we have

$$\mathrm{E}[\tanh(X/2)] = \int_0^\infty \tanh(x/2) f(x)(1-\mathrm{e}^{-x})\mathrm{d}x.$$

The proofs of both assertions are accomplished by appropriate decompositions of the integral and by using the fact that the function $\tanh(x/2)$ is increasing.

a) We have

$$\begin{aligned}
t &\geq \int_0^\infty \tanh\left(\frac{x}{2}\right) f(x)(1-\mathrm{e}^{-x})\mathrm{d}x \\
&\geq \tanh\left(\frac{a}{2}\right)\int_a^\infty f(x)(1-\mathrm{e}^{-x})\mathrm{d}x \\
&= \tanh\left(\frac{a}{2}\right)\cdot\left(1-\int_0^a f(x)(1-\mathrm{e}^{-x})\mathrm{d}x\right) \\
&\geq \tanh\left(\frac{a}{2}\right)\cdot\left(1-\int_0^a f(x)(1+\mathrm{e}^{-x})\mathrm{d}x\right) \\
&\geq \tanh\left(\frac{a}{2}\right)\cdot\left(1-2\int_0^a f(x)\mathrm{d}x\right) \\
&= \tanh\left(\frac{a}{2}\right)\cdot\left(1-2\int_{-a}^0 f(x)\mathrm{e}^{-x}\mathrm{d}x\right) \\
&\geq \tanh\left(\frac{a}{2}\right)\cdot\left(1-2\mathrm{e}^{a}\int_{-a}^0 f(x)\mathrm{d}x\right) \\
&\geq \tanh\left(\frac{a}{2}\right)\cdot\left(1-2\mathrm{e}^{a}\int_{-\infty}^0 f(x)\mathrm{d}x\right) \\
&= \tanh\left(\frac{a}{2}\right)\cdot(1-2\mathrm{e}^{a}\mathrm{Pr}[X\leq 0]).
\end{aligned}$$

The assertion is obtained by a simple manipulation of the last inequality.

b) The proof is similar to that of the last part

$$\begin{aligned}
t &\leq \int_0^\infty \tanh\left(\frac{x}{2}\right) f(x)(1-\mathrm{e}^{-x})\mathrm{d}x \\
&\leq \int_0^\infty f(x)(1-\mathrm{e}^{-x})\mathrm{d}x.
\end{aligned}$$

Using the symmetry of f, this shows that

$$\Pr[X\leq 0] = \int_{-\infty}^0 f(x)\mathrm{d}x \leq \int_0^\infty f(x)\mathrm{d}x - t.$$

Adding $\Pr[X\leq 0]$ to both sides of the inequality gives

$$2\Pr[X\leq 0] \leq 1-t$$

which proves the assertion.

APPENDIX VI
PROOF OF THEOREM 8

For the proof of this theorem we will proceed in several steps.

Lemma 15: Assumptions being as in the previous theorem, suppose that the neighborhood of depth ℓ of y_i is a tree, and let $\hat{y}_i$ denote the set of output bits that appear in this tree. Then, there exists a constant $\tau > 0$ depending on t, the degree of y_i, and the channel $\mathcal{C}$, such that

$$I(x; y_i \mid \hat{y}_i) \leq \mathrm{Cap}(\mathcal{C}) - \tau.$$

Proof: Let d denote that degree of the output bit y_i, and let Y denote the LLR of y_i given $\hat{y}_i$. Since the neighborhood of y_i is a tree, we have

$$\mathrm{E}[\tanh(Y/2)] = \mathrm{E}(\mathcal{C})\mathrm{E}[\tanh(X_\ell/2)]^{d-1} > \mathrm{E}(\mathcal{C})t^{d-1} =: \eta > 0.$$

If z_i denotes the value of the received bit y_i prior to the transmission, then Lemma 6 part b) implies that

$$\Pr[z_i = -1 \mid \hat{y}_i] \leq \frac{1-\eta}{2} < \frac{1}{2}$$

so that $h(y_i \mid \hat{y}_i) < 1-\tau$ for some $\tau > 0$. (Note that z_i is a binary random variable, hence, the upper bound on the conditional probability gives an upper bound on the conditional entropy.) Note that τ depends on η, which itself depends on d, t, and $\mathrm{E}(\mathcal{C})$. By definition, we have $I(x; y_i \mid \hat{y}_i) = h(y_i \mid \hat{y}_i) - h(y_i \mid x, \hat{y}_i)$, which translates to

$$\begin{aligned}
I(x; y_i \mid \hat{y}_i) &= \mathrm{Cap}(\mathcal{C}) - (1 - h(y_i \mid \hat{y}_i)) \\
&\leq \mathrm{Cap}(\mathcal{C}) - \tau.
\end{aligned}$$

This completes the proof. □

The next lemma shows that the previous condition is valid even if we condition on fewer output bits in the neighborhood of y_i.

Lemma 16: Assumptions being as in the previous lemma, for any τ' with $0 < \tau' \leq \tau$ there exists $p > 0$ depending on τ' such that if $T_{\ell,p}$ denotes the neighborhood of depth ℓ of y_i in which the output bits are removed with probability p, then

$$I(x; y_i \mid T_{\ell,p}) \leq \mathrm{Cap}(\mathcal{C}) - \tau'.$$

Proof: Let $\mathcal{C}(p)$ denote the channel obtained by concatenating the channel $\mathcal{C}$ with a BEC of probability p. Let $X_\ell(p)$ be a prototype random variable describing the messages passed from input bits to output bits at round ℓ of the BP algorithm applied to a decoding graph of the LT-code on this channel. Then it is clear that $\mathrm{E}[\tanh(X_\ell(p)/2)]$ is a continuous function of p. Moreover, when $p = 1$, this expectation is zero, whereas for $p = 0$, this expectation is at least t, by assumption. Therefore, for any t' with $0 \leq t' \leq t$ there exists a p such that $\mathrm{E}[\tanh(X_\ell(p)/2)] \geq t'$, and t' is a continuous function of p. As far as the BP algorithm is concerned, the neighborhood of depth ℓ of y_i equals $T_{\ell,p}$, since the erased output bits do not contribute to the BP algorithm. Therefore, by Lemma 6 part b), we have $\Pr[y_i = -1 \mid T_{\ell,p}] \leq (1-t')/2$. From this, we deduce that $h(y_i \mid T_{\ell,p}) \leq 1 - \tau'$ for some τ' which is a continuous function of p. As a result,

$$I(x; y_i \mid T_{\ell,p}) \leq \mathrm{Cap}(C) - \tau'.$$

The result follows from the Mean Value Theorem by using the continuity of τ' as a function of p. □

We need one more result, the proof of which is well known [26].

Lemma 17: Suppose that X, Y, and Z are random variables such that Y and Z are independent given X. Then

$$I(X; Y|Z) = I(X; Y) - I(Y; Z) \leq I(X; Y).$$

Note that the assumption that Y and Z are independent given X is crucial as the following example shows: suppose that X and Y are independent binary random variables, and set $Z = X \oplus Y$ be their XOR. Then $I(X; Y) = 0$, but $I(X; Y|Z) = 1$. Now we are ready to prove the main theorem.

Proof: (Of Theorem 8) Let τ be as in the statement of Lemma 15; choose some τ' with $0 < \tau' < \tau$. Then, by Lemma 16 there exists some $p > 0$ such that

$$I(x; y_i \mid T_{\ell,p}) < \mathrm{Cap}(\mathcal{C}) - \tau', \qquad \text{for all } y_i$$

for which the neighborhood of depth ℓ is a tree. Let B be the number of y_i for which this neighborhood is not a tree, and let $m = n(1-p)$. We then have

$$\begin{aligned} I(x; z) &= \sum_{i=1}^{n} I(x; z_i | z_1, \ldots, z_{i-1}) \\ &= \sum_{i=1}^{m} I(x; z_i | z_1, \ldots, z_{i-1}) \\ &\quad + \sum_{i>m, z_i \in B} I(x; z_i | z_1, \ldots, z_{i-1}) \\ &\quad + \sum_{i>m, z_i \notin B} I(x; z_i | z_1, \ldots, z_{i-1}). \end{aligned}$$

Using Lemma 17, the first summand can be estimated from above by $\sum_{i=1}^{m} I(x; z_i) = m\mathrm{Cap}(\mathcal{C})$. Similarly, the second summand is at most $T\mathrm{Cap}(\mathcal{C})$. By Lemmas 17 and 16 we have

$$I(x; y_i \mid y_1, \ldots, y_{i-1}) \leq I(x; y_i \mid T_{\ell,p}) \leq \mathrm{Cap}(\mathcal{C}) - \tau'.$$

Therefore, we have altogether

$$\begin{aligned} I(x; z) &\leq n(1-p)\mathrm{Cap}(C) + np(\mathrm{Cap}(C) - \tau') + T\mathrm{Cap}(\mathcal{C}) \\ &= n(\mathrm{Cap}(\mathcal{C}) - p\tau') + T\mathrm{Cap}(\mathcal{C}). \end{aligned}$$

Note that τ' and p depends on τ, which in turn depends on t, the degree of y_i, and the channel $\mathcal{C}$. Setting $\epsilon = p\tau'$, this proves the assertion. □

APPENDIX VII
RELATING BIMSCS WITH THE BEC

In this section, we state an upper bound and a lower bound on the decoding threshold of the BP algorithm over an arbitrary BIMSC in terms of its decoding threshold over the erasure channel.

The upper and the lower bound both only depend on the density evolution analysis of the BP algorithm. Therefore, the bounds hold for BP over all classes of graphical codes where density evolution is valid, e.g., LDPC codes, irregular repeat–accumulate (IRA) codes, bounded-degree LT-codes, and Raptor codes.

A. Lower Bound

Assume that $\mathcal{C}$ is an arbitrary BIMSC. The Bhattacharya parameter of $\mathcal{C}$, written $\gamma(\mathcal{C})$, is defined as $\gamma(\mathcal{C}) = \mathrm{E}[\exp(-Z/2)]$ where Z is the LLR of the bit obtained from the channel under the assumption that the all-zero codeword is transmitted. For example, we have

$$\gamma(\mathrm{BEC}(p)) = p \quad \text{and} \quad \gamma(\mathrm{BSC}(p)) = 2\sqrt{p(1-p)}.$$

The following theorem [27, Ch. 4] gives a lower bound on the performance of BP decoding over binary symmetric channels.

Theorem 18: Let $\gamma(\mathcal{C})$ be the Bhattacharya parameter of an arbitrary channel $\mathcal{C}$. If BP can decode an ensemble of codes on $\mathrm{BEC}(\gamma(\mathcal{C}))$, then BP can decode the same ensemble of codes with the same length on channel $\mathcal{C}$.

Corollary 19: If the overhead of a Raptor code on an erasure channel is ϵ, then its overhead over any BIMSC $\mathcal{C}$ is at most $(1+\epsilon)\mathrm{Cap}(\mathcal{C})/(1-\gamma(\mathcal{C})) - 1$.

Proof: Assume that we can decode k bits from $k(1+\epsilon)$ correct output bits. Then we can also obtain k bits from $k(1+\epsilon)/(1-\gamma(\mathcal{C}))$ bits received from $\mathrm{BEC}(\gamma(\mathcal{C}))$. By Theorem 18, to obtain k bits, it is enough to get $k(1+\epsilon)/(1-\gamma(\mathcal{C}))$ bits from $\mathcal{C}$. Thus, the overhead of the code over $\mathcal{C}$ is $\leq (1+\epsilon)\mathrm{Cap}(\mathcal{C})/(1-\gamma(\mathcal{C})) - 1$ □

For a sequence of Raptor codes that achieves the capacity of BEC, Corollary 19 shows that these codes simultaneously beat the so-called "cutoff" rate on all BIMSCs [27]. The rate was considered to be a limit for "practical communication" before the advent of graphical codes and iterative decoding. The interesting point about this result is that this performance is achieved while the encoder is totally oblivious of the underlying channel $\mathcal{C}$!

Corollary 20: For arbitrary small $\epsilon > 0$, the reception overhead of Raptor codes optimized for the BEC is at most $\log_2(\mathrm{e}) - 1 + \epsilon = 0.442\ldots$ on any BIMSC.

Proof: Let $\mathcal{C}$ be a BIMSC. Let $f(p)$ be the density function of the probability that a symbol received from the channel is incorrect with probability p ($0 \leq p \leq 1/2$). We have

$$\frac{\mathrm{Cap}(\mathcal{C})}{1-\gamma(\mathcal{C})} = \frac{\int_p (1-h(p))f(p)\mathrm{d}p}{\int_p (1-2\sqrt{p(1-p)})f(p)\mathrm{d}p} \leq \sup_{0\leq p<1/2} \frac{1-h(p)}{1-2\sqrt{p(1-p)}}.$$

By l'Hospital's rule

$$\lim_{p\to 1/2}(1-h(p))/(1-2\sqrt{p(1-p)}) = \log_2(\mathrm{e}).$$

The result follows from Corollary 19. □

B. Upper Bound

Recall the parameter $\mathrm{E}(\mathcal{C})$ defined for a BIMSC $\mathcal{C}$ in (7).

The following theorem establishes an upper bound on the performance of BP over BIMSCs in terms of its performance on the BEC.

Theorem 21: For any BIMSC $\mathcal{C}$, if BP can decode an ensemble of codes on channel $\mathcal{C}$, then it can also decode the same ensemble of codes with the same length on $\mathrm{BEC}(1-\mathrm{E}(\mathcal{C}))$.

Proof: We will use the notation at the beginning of Section VI and track μ_ℓ and ξ_ℓ for the BP algorithm over a BIMSC $\mathcal{C}$. By Lemma 4 part b) we have

$$1-\xi_{\ell+1} \geq \iota(1-\mu_\ell). \tag{19}$$

Let $\mathcal{C}_0 := \mathrm{BEC}(1-\mathrm{E}(\mathcal{C}))$. Clearly, $\mathrm{E}(\mathcal{C}_0) = \mathrm{E}(\mathcal{C})$.

When BP is run over the BEC, ξ_ℓ (resp., μ_ℓ) is the probability that the message passed through a random edge from an input node to an output node (resp., from an output node to an input node) is not an erasure at round ℓ. Thus, inequality (19) is an equality in the case of the BEC $\mathcal{C}_0$. Hence, by induction on ℓ, the values of ξ_ℓ and μ_ℓ for BP on $\mathcal{C}$ are no more than the corresponding values for BP on $\mathcal{C}_0$.

In particular, if ξ_ℓ converges to 1 as $\ell \to \infty$ for BP on $\mathcal{C}$, then it also converges to 1 for BP on $\mathcal{C}_0$. Finally, we note that ξ_ℓ converges to 1 if and only if the decoding error probability converges to 0. □

We now apply Theorem 21 to Raptor codes.

Theorem 22: If a sequence of Raptor codes achieves the capacity of a BIMSC, then the overhead of the sequence on the BEC is exactly equal to $\mathrm{E}(\mathcal{C})/\mathrm{Cap}(\mathcal{C}) - 1 = 1/\Pi(\mathcal{C}) - 1$.

Proof: Consider a sequence of Raptor codes that achieves the capacity of the BIMSC $\mathcal{C}$. When the code is used over channel $\mathcal{C}$, $k/\mathrm{Cap}(\mathcal{C})(1+o(1))$ output bits are sufficient to decode k input bits. Using Theorem 21, $k/\mathrm{Cap}(\mathcal{C})(1+o(1))$ output bits are sufficient to decode k inputs bits over $\mathrm{BEC}(1-\mathrm{E}(\mathcal{C}))$. Thus, $k\mathrm{E}(\mathcal{C})/\mathrm{Cap}(\mathcal{C})(1+o(1))$ received output bits are sufficient on the BEC.

We now only need to show that the overhead of the code over the BEC, say ϵ, is $\geq 1/\Pi(\mathcal{C}) - 1$ in the limit. Using Theorem 10, the asymptotic fraction of input bits of degree 2 is $\Omega_2 = \Pi(\mathcal{C})/2$. Using Theorem 5, in the limit we have $\Omega_2 \geq 1/(2(1+\epsilon))$. This implies that $1+\epsilon \geq 1/\Pi(\mathcal{C})$. □

Using part 2) of Theorem 5, it is possible to construct a sequence of degree distributions for Raptor codes such that $\Omega_2 = \frac{1}{2}\Pi(\mathcal{C})/(1+\epsilon)$ in the limit, where ϵ is the overhead of the sequence of codes in the limit. Using part 1) of Theorem 5, the overhead of that sequence of codes over the BEC is $\geq (1+\epsilon)/\Pi(\mathcal{C})$. This shows that in Theorem 21, the constant $1-\mathrm{E}(\mathcal{C})$ (the erasure probability) cannot be improved.

Appendix VIII
Alternative Proof of Theorem 10

In this appendix, we will give an alternative proof of Theorem 10; we wish to thank Bixio Rimoldi and Emre Telatar for their help with this proof.

The intuition of the proof is the following: if a noisy version of a bit x_i with $\Pr[x_i = 0] = 1/2$ is observed more than once, then the mutual information between x_i and its observations is less than the number of observations times the capacity of the channel. A standard argument can then be used to deduce that the mutual information between the input symbols and the output symbols of degree one is too small if there is a constant fraction of output symbols of degree one.

To put the intuition on firm ground, we again look at the degree-one forest formed by the input symbols and the output symbols of degree one. Let us denote the input symbols by $x = (x_1, \ldots, x_k)$ and the output symbols by $y = (y_1, \ldots, y_s)$. Suppose that input symbol x_i is connected to output symbols $y_{i,1}, \ldots, y_{i,r_i}$. Then

$$I(x;y) = \sum_{i=1}^{k} I(x_i; y_{i,1}, \ldots, y_{i,r_i})$$

since the other y_j's are independent of x_i. Let $Y_i := (y_{i,1}, \ldots, y_{i,r_i})$. Then

$$\begin{aligned} I(x_i;Y_i) &= h(Y_i) - h(Y_i \mid x_i) \\ &\leq \sum_{j=1}^{r_i} h(y_{i,j}) - h(Y_i \mid x_i) \\ &= \sum_{j=1}^{r_i} (h(y_{i,j}) - h(y_{i,j} \mid x_i)) \\ &= \sum_{i=1}^{r_j} I(x_i; y_{i,j}) \\ &= r_i \mathrm{Cap}(\mathcal{C}). \end{aligned}$$

The last equality follows from the fact that x_i is a binary random variable with $\Pr[x_i = 0] = 1/2$. Since the $y_{i,j}$ for different j are not independent, the inequality in the second step above is sharp if $r_i > 1$. As a result, if $r_i > 1$, then there exists $\epsilon_i > 0$ such that

$$I(x_i;Y_i) \leq r_i \mathrm{Cap}(\mathcal{C}) - \epsilon_i.$$

Hence, we have

$$I(x;y) \leq s\mathrm{Cap}(\mathcal{C}) - \sum_{i, r_i>1} \epsilon_i \leq s\mathrm{Cap}(\mathcal{C}) - t\epsilon_2$$

where t is the number of i such that $r_i = 2$.

Now we proceed as in the original proof of Theorem 10. The fraction of input symbols connected to two output symbols of

degree one is a constant, if the fraction of the output symbols of degree one is a constant (measured with respect to the number of input symbols). Therefore, $t \geq \mu s$ for some constant μ, and we have $I(x;y) < s(\mathrm{Cap}(C) - \mu)$, which shows that the sequence cannot be capacity-achieving.

Appendix IX
The Limiting Behavior of $\Omega_2(\mathcal{C})$

In this appendix, due to Payam Pakzad, we will prove that Proposition 12 part 4) holds in a much more general setting.

For a symmetric density $f(x)$, we define the *error probability* associated with $f(x)$ as

$$e(f) := \int_{-\infty}^{\infty} \frac{f(x)}{1+\mathrm{e}^{|x|}}\mathrm{d}x = \int_{0}^{\infty} f(x)\mathrm{e}^{-x}\mathrm{d}x \qquad (20)$$

where the last equality follows from the symmetry of f. It is easy to see that $0 \leq e(f) \leq \frac{1}{2}$. We also define

$$\Omega_2(f) := \frac{\int_{-\infty}^{\infty} \log_2\left(\frac{2}{1+\mathrm{e}^{-x}}\right) f(x)\mathrm{d}x}{2\int_{-\infty}^{\infty} \tanh\left(\frac{x}{2}\right) f(x)\mathrm{d}x}. \qquad (21)$$

Note that if f is the density of the LLR of the channel $\mathcal{C}$, then $\Omega_2(f) = \Omega_2(\mathcal{C})$.

Theorem 23: Let $\{f_i(x)\}$ be a family of symmetric probability densities such that each f_i comprises of countably many point masses and a piecewise smooth function with countably many discontinuity points, and such that $\lim_{i\to\infty} e(f_i) = \frac{1}{2}$. Then

$$\lim_{i\to\infty} \Omega_2(f_i) = \frac{1}{\ln(16)} = 0.36067\cdots.$$

Proof: First we will show that for large enough i, $f_i(x)$ must have all its mass in close vicinity of $x = 0$. More precisely, if we define $\delta_i(\epsilon) := 1 - \int_{-\epsilon}^{\epsilon} f_i(x)\,\mathrm{d}x$, then we claim that

$$\forall\, \epsilon > 0\text{: } \lim_{i\to\infty} \delta_i(\epsilon) = 0.$$

To see this, first note that for all $x \notin (-\epsilon, \epsilon)$, we have

$$\frac{1}{1+\mathrm{e}^{|x|}} \leq \frac{1}{2} - \epsilon'$$

where $\epsilon' \approx \frac{\epsilon}{4}$ is a positive real. Now suppose there is an $\alpha > 0$ such that $\delta_i(\epsilon) \geq \alpha$ for infinitely many i's. Then for infinitely many i's we have

$$\begin{aligned} e(f_i) &= \int_{-\infty}^{\infty} \frac{f_i(x)}{1+\mathrm{e}^{|x|}}\mathrm{d}x \\ &= \int_{-\epsilon}^{\epsilon} \frac{f_i(x)}{1+\mathrm{e}^{|x|}}\mathrm{d}x + \int_{-\infty}^{-\epsilon} \frac{f_i(x)}{1+\mathrm{e}^{|x|}}\mathrm{d}x + \int_{\epsilon}^{\infty} \frac{f_i(x)}{1+\mathrm{e}^{|x|}}\mathrm{d}x \\ &\leq (1-\delta_i(\epsilon))\cdot\frac{1}{2} + \delta_i(\epsilon)\cdot\left(\frac{1}{2}-\epsilon'\right) \\ &= \frac{1}{2} - \frac{\delta_i(\epsilon)\epsilon'}{2} \\ &\leq \frac{1}{2} - \frac{\alpha\epsilon'}{2} < \frac{1}{2} \end{aligned}$$

which is a contradiction, since $\lim_{i\to\infty} e(f_i) = \frac{1}{2}$.

We deduce that there is a sequence $\{\epsilon_i\}$ such that $\lim_{i\to\infty}\epsilon_i = 0$ and $\delta_i(\epsilon_i)$ also goes to zero, say $\delta_i(\epsilon_i) = O\left(\epsilon_i^4\right)$ (the reason for this particular choice of ϵ_i^4 will be apparent below).

Now suppose that each density function $f_i(x)$ can be decomposed as the sum of a smooth function and a pair of delta functions in $(-\epsilon_i, \epsilon_i)$, i.e.,

$$f_i(x) = g_i(x) + a_i\delta_{x_i}(x) + b_i\delta_{-x_i}(x)$$

where $g_i(x) = g_i^0 + g_i^1 x + g_i^2 x^2 + \cdots$ for $x \in (-\epsilon_i, \epsilon_i)$, and $0 \leq x_i \leq \epsilon_i$. Then, from the symmetry of $f_i(x)$, we must have

$$g_i^0 = 2g_i^1 \quad \text{and} \quad b_i = a_i\mathrm{e}^{-x_i}.$$

We will now write the small-order expansions of the numerator and the denominator of (21). The numerator of the expression has the expansion

$$\begin{aligned} &\int_{-\infty}^{\infty} \log_2\left(\frac{2}{1+\mathrm{e}^{-x}}\right) f_i(x)\mathrm{d}x \\ &\quad= \int_{-\epsilon_i}^{\epsilon_i} \log_2\left(\frac{2}{1+\mathrm{e}^{-x}}\right) \\ &\qquad\cdot\Big(\left(2g_i^1 + g_i^1 x + \cdots\right) + a_i\Big(\delta_{x_i}(x) + \mathrm{e}^{-x_i}\delta_{-x_i}(x)\Big)\Big)\mathrm{d}x \\ &\qquad + O\left(\delta_i(\epsilon_i)\right) \\ &\quad= \frac{g_i^1\epsilon_i^3}{6\ln(2)} + O\left(\epsilon_i^4\right) + \frac{a_i x_i^2}{4\ln(2)} + O\left(x_i^3\right) + O(\delta_i(\epsilon_i)) \end{aligned}$$

whereas the denominator has the expansion

$$\begin{aligned} &\int_{-\infty}^{\infty} \tanh\left(\frac{x}{2}\right) f_i(x)\mathrm{d}x \\ &\quad= \int_{-\epsilon_i}^{\epsilon_i} \tanh\left(\frac{x}{2}\right) \\ &\qquad\cdot\Big(\left(2g_i^1 + g_i^1 x + \cdots\right) + a_i\left(\delta_{x_i}(x) + \mathrm{e}^{-x_i}\delta_{-x_i}(x)\right)\Big)\mathrm{d}x \\ &\qquad + O(\delta_i(\epsilon_i)) \\ &\quad= \frac{g_i^1\epsilon_i^3}{3} + O\left(\epsilon_i^4\right) + \frac{a_i x_i^2}{2} + O\left(x_i^3\right) + O(\delta_i(\epsilon_i)). \end{aligned}$$

Therefore, using the fact that $\delta_i(\epsilon_i) = O\left(\epsilon_i^4\right)$, we can write the desired limit as

$$\begin{aligned} &\lim_{i\to\infty} \Omega_2(f_i) \\ &\quad= \lim_{i\to\infty} \frac{\frac{1}{6\ln(2)} g_i^1\epsilon_i^3 + O\left(\epsilon_i^4\right) + \frac{1}{4\ln(2)} a_i x_i^2 + O\left(x_i^3\right)}{2\left(\frac{1}{3}g_i^1\epsilon_i^3 + O\left(\epsilon_i^4\right) + \frac{1}{2}a_i x_i^2 + O\left(x_i^3\right)\right)} \\ &\quad= \lim_{i\to\infty} \frac{\frac{1}{6\ln(2)} g_i^1\epsilon_i^3 + \frac{1}{4\ln(2)} a_i x_i^2}{2\left(\frac{1}{3}g_i^1\epsilon_i^3 + \frac{1}{2}a_i x_i^2\right)} \\ &\quad= \frac{1}{4\ln(2)}. \end{aligned}$$

It is easy to see that the proof can be extended if each $f_i(x)$ comprises of countable point-masses and a piecewise smooth function with countable discontinuity points. □

Acknowledgment

The authors wish to thank Mehdi Molkaraie for discussions that led to the information-theoretic proofs of Section VII, Rüdiger Urbanke for interesting discussions and his help with some of the references in the paper as well as with the proof of Proposition 12 part 4), and Lorenz Minder for pointing out a deficiency in the random number generator used to compute the first version of the results in Section VIII. Many thanks go

to Michael Luby for careful proofreading of an earlier version of the paper and for many important suggestions for improving its readability. The authors would also like to thank Payam Pakzad for carefully reading an earlier version of the paper and providing numerous suggestions and corrections, and Igal Sasson for feedbacks on an earlier version. Thanks go also to Amir Dana for spotting some errors in an earlier version of the manuscript. Last but not least, we would like to express our gratitude to two anonymous referees for spending the time to go through a prior submitted version of this paper and providing us with very detailed comments. These comments have led to a major improvement of the style and the presentation of the results.

REFERENCES

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

[2] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[3] ——, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

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

[5] D. MacKay and R. Neal, "Good codes based on very sparse matrices," in *Proc. 5th IMA Conf., Cryptography and Coding (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1995, vol. 1025, pp. 100–111.

[6] M. Sipser and D. Spielman, "Expander codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.

[7] D. Spielman, "Linear-time encodable and decodable error-correcting codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1723–1731, Nov. 1996.

[8] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. 29th Annu. ACM Symp. Theory of Computing*, El Paso, TX, May 1997, pp. 150–159.

[9] M. Luby, M. Mitzenmacher, and A. Shokrollahi, "Analysis of random processes via and-or tree evaluation," in *Proc. 9th Annu. ACM-SIAM Symp. Discrete Algorithms*, San Francisco, CA, Jan. 1998, pp. 364–373.

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

[11] D. Divsalar, H. Jin, and R. McEliece, "Coding theorems for 'turbo-like' codes," in *Proc. 1998 Allerton Conf. Communication, Control and Computing*, Monticello, IL, Oct. 1998, pp. 201–210.

[12] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.

[13] M. Luby, "Information Additive Code Generator and Decoder for Communication Systems," U.S. Patent,307 487, Oct. 23, 2001.

[14] ——, "Information Additive Code Generator and Decoder for Communication Systems," U.S. Patent,373 406, Apr. 16, 2002.

[15] ——, "LT-codes," in *Proc. 43rd Annu. IEEE Symp. Foundations of Computer Science*, Vancouver, BC, Canada, Nov. 2002, pp. 271–280.

[16] A. Shokrollahi, "Raptor codes," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, Jun./Jul. 2004, p. 36.

[17] S.-Y. Chung, T. Richardson, and R. 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. El Gamal and A. Hammons, "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 671–686, Feb. 2001.

[19] A. Shokrollahi, *Capacity-Achieving Sequences*. Minneapolis, MN: Inst. Mathematics and its Applications (IMA), 2000, vol. 123, IMA Volumes in Mathematics and its Applications, pp. 153–166.

[20] M. Ardakani and F. Kschischang, "A More Accurate One-Dimensional Analysis and Design of LDPC Codes," preprint, 2003.

[21] T. Richardson and R. Urbanke, "Modern Coding Theory," preprint, 2004.

[22] R. Palanki and J. Yedidia, "Rateless codes on noisy channels," in *Proc. IEEE In. Symp. Information Theory*, Chicago, IL, Jun./Jul. 2004, p. 37.

[23] T. Richardson and R. 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.

[24] P. Pakzad, private communication, 2004.

[25] I. Sason, private communication, 2004.

[26] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[27] A. Khandekar, "Graph-based codes and iterative decoding," Ph.D. dissertation, Calif. Ins. Technol., Pasadena, 2002.