

## Research Article

# Jointly Decoded Raptor Codes: Analysis and Design for the BIAWGN Channel

**Auguste Venkiah, Charly Poulliat, and David Declercq**

*ETIS, CNRS, ENSEA, Cergy-Pontoise University, 95014 Cergy-Pontoise Cedex, France*

Correspondence should be addressed to Auguste Venkiah, [auguste.venkiah@ensea.fr](mailto:auguste.venkiah@ensea.fr)

Received 6 August 2008; Revised 11 April 2009; Accepted 6 June 2009

Recommended by Tho Le-Ngoc

We are interested in the analysis and optimization of Raptor codes under a joint decoding framework, that is, when the precode and the fountain code exchange soft information iteratively. We develop an analytical asymptotic convergence analysis of the joint decoder, derive an optimization method for the design of efficient output degree distributions, and show that the new optimized distributions outperform the existing ones, both at long and moderate lengths. We also show that jointly decoded Raptor codes are robust to channel variation: they perform reasonably well over a wide range of channel capacities. This robustness property was already known for the erasure channel but not for the Gaussian channel. Finally, we discuss some finite length code design issues. Contrary to what is commonly believed, we show by simulations that using a relatively low rate for the precode ( $R_p \approx 0.9$ ), we can improve greatly the error floor performance of the Raptor code.

Copyright © 2009 Auguste Venkiah et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. Introduction

Fountain codes were originally introduced [1] to transmit efficiently over a binary erasure channel (BEC) with unknown erasure probability. They are of special interest for multicast or peer-to-peer applications, that is, when no feedback channel is available. Introduced by Luby [2], LT codes are the first class of efficient fountain codes: by a proper design of its so-called output distribution, an LT code produces a potentially limitless number of distinct output symbols from a set of  $K$  input symbols. The receiver can then recover the input bits from any set of  $(1 + \epsilon)K$  output bits, where  $\epsilon$  is the reception overhead. However, high performance is achieved at a decoding cost growing in  $\mathcal{O}(K \log(K))$ , which is too high to ensure linear encoding and decoding time. To overcome this complexity issue, Raptor codes have been firstly introduced by Shokrollahi in [3] for the BEC channel: it simply consists in the concatenation of an LT code with an outer code, called *precode*, which is usually a high rate error correcting code. In [4], the author independently presented the idea of precoding to obtain linear decoding time codes. More recently, Raptor codes have been studied on general binary memoryless symmetric

channels with information theoretic arguments [5]. In particular, the authors proposed an optimization procedure for designing good output degree distributions in the case of transmission on the binary input additive white Gaussian noise (BIAWGN) channel. In their optimization procedure, the LT code and the precode are decoded separately, in a tandem fashion, following the same framework as for the BEC channel. The tandem decoder can however be suboptimal, since it is possible to exchange soft information between the precode and the fountain in an iterative way.

In this paper, we assume the joint decoding of the two code components, and show that with proper design methods, we obtain Raptor codes with better performance and robustness properties than the ones proposed in literature.

In a joint decoding framework, we use the extrinsic information transfer function (EXIT function) of the precode as an additional knowledge in the system, and consider this EXIT function in the asymptotic density evolution equations of the Raptor code under Gaussian approximation. By optimizing the distribution with this new set of equations, the fountain is matched to a particular precode behavior, which leads to a substantial performance improvement. Note that our approach has the great advantage that both the

analysis and the design remain fully analytical and linear in the parameters, that is fountain distributions are easy to optimize.

Aside from the better results, we also show that optimizing Raptor codes under the joint decoding framework has also other advantages on the properties of the coded system. The first advantage relates to the robustness of the transmission to channel variations. On the BEC channel, Raptor codes are universal, as they can approach the capacity of the channel arbitrarily closely, and independently of the channel parameter [3]. This is a very special case, since the results in [5] show that Raptor codes are not universal on other channels than the BEC. Nevertheless, one can characterize the robustness of a Raptor code by considering the variation of the overhead over a wide range of channel capacities. In particular, we will show with a threshold analysis that the Raptor codes optimized under the joint decoding framework and with a smart choice of optimization parameters are more robust than the distributions proposed in [5]. An alternative solution has been proposed in [6], where the authors propose the construction of *generalized Raptor codes*, by allowing the output degree distribution to vary as the output symbols are generated. This construction has the advantage that the resulting codes can approach the capacity of a noisy symmetric channel in a rate compatible way. However, no code design technique has been proposed for generalized Raptor codes, mainly due to the fact that their structure is not as easy to optimize compared to usual Raptor codes.

Finally, we address the issues raised by the finite length construction of Raptor codes. For practical applications, it is important that codes which perform well asymptotically also give good performance at finite length. The design of Raptor codes at finite length has already been addressed for the BEC [7, 8] and the BSC [9]. Unlike LDPC codes that can be conditioned to perform well at finite length by a careful design of the graph [10, 11], the underlying graph of a Raptor code is random by nature, and no such technique can be used. Our framework partially addresses this problem, by naturally addressing the rate repartition between the fountain code and the precode. Since the fountain is matched to the precode in our framework, the use of precodes with rates far lower than the one proposed in literature is possible without sacrificing much on the overall performance. Using this additional degree of freedom, we obtain finite length Raptor codes with considerably lower error floors, with a negligible loss in the waterfall region. This can also be seen as robustness of our constructions to varying information block lengths.

The remainder of this paper is organized as follows. In Section 2, we describe the system that we consider and give the notations used in the paper. In Section 3, we study the asymptotic performance of jointly decoded Raptor codes on the BIAWGN channel and derive an optimization method for the design of efficient output degree distributions. Then, we show with threshold computations that under the joint decoding framework, Raptor codes are robust to a channel variation. In Section 4, we consider the problem of finite length design by properly addressing the rate splitting issue,

and finally, conclusions and perspectives are drawn in Section 5.

## 2. System Description and Notations

**2.1. Definitions and Notations.** We consider in this paper only coded transmissions over the BI-AWGN channel. We call *input symbols* the set of binary information symbols to be transmitted and *output symbols* the symbols produced by an LT code from the input symbols. At the receiver side, belief propagation (BP) decoding is used to recover iteratively the input symbols from the noisy observations of the output symbols.

An LT code is described by its *output degree distribution*  $\Omega$  [2]: to generate an output symbol, a degree  $d$  is sampled from that distribution, independently from the past samples, and the output symbol is then formed as the sum of a uniformly randomly chosen subset of size  $d$  of the input symbols. Let  $\Omega_1, \Omega_2, \dots, \Omega_{d_c}$  be the distribution weights on degrees  $1, 2, \dots, d_c$ , so that  $\Omega_d$  denotes the probability of choosing the value  $d$ . Using polynomial notations, the output degree distribution can be written compactly as  $\Omega(x) = \sum_{j=1}^{d_c} \Omega_j x^j$ .  $\omega(x) = \sum_{j=1}^{d_c} \omega_j x^{j-1} = \Omega'(x)/\Omega'(1)$  is the corresponding edge degree distribution in the Tanner graph (see Figure 1). In these notations,  $d_c$  represents the maximum degree of the parity-check equations used in the generation of output symbols. Because the input symbols are chosen uniformly at random, their node degree distribution is binomial, and can be approximated by a Poisson distribution with parameter  $\alpha$  [3, 5]. Thus, the input symbol node degree distribution is defined as:  $I(x) = e^{\alpha(x-1)}$ . Then, the associated input symbol edge degree distribution  $\iota(x) = I'(x)/I'(1)$  is also equal to  $e^{\alpha(x-1)}$ . Both distributions are of mean  $\alpha$ . Technically,  $\iota(x)$  and  $I(x)$  cannot define degree distributions since they are power series and not polynomials. However, the power series can be truncated to obtain polynomials that are arbitrarily close to the exponential [5]:  $I(x) = \sum_{i=1}^{d_v} I_i x^i$  and  $\iota(x) = I'(x)/I'(1) = \sum_{i=1}^{d_v} \iota_i x^{i-1}$ . The maximum degree  $d_v$  is chosen sufficiently high. Although fountain codes are rateless, we can still define an *a posteriori* rate  $R_{LT}$  for a fountain code as follows:

$$\begin{aligned} R_{LT} &= \frac{\text{Nb input symbols}}{\text{Nb output symbols needed for successful decoding}} \\ &= \frac{\Omega'(1)}{\alpha}. \end{aligned} \quad (1)$$

For a Raptor code, the *a posteriori* rate is  $R = R_p \cdot R_{LT}$ , where  $R_p$  denotes the rate of the precode. Recall that, as a main measure of performance, Raptor codes are usually illustrated in terms of error rates versus the value of the overhead  $\epsilon$ .  $\epsilon$  is defined as  $C = R(1 + \epsilon)$  where  $C$  is the channel capacity. Finally, the Tanner graph of a Raptor code is given in Figure 1. Note that we did not represent the Tanner graph of the precode: in general, the precode can be any block code, and not necessarily a LDPC code like we considered in this paper.

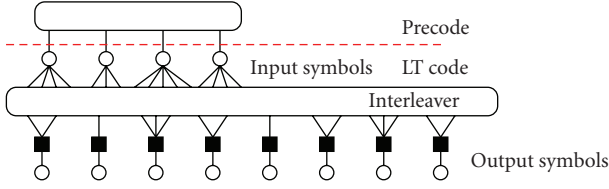


FIGURE 1: Description of a Raptor code. Tanner graph of an LT code + precode. The black squares represent parity-check nodes and the circles are variable nodes associated with input symbols or output symbols.

**2.2. Tandem and Joint Decoding of a Raptor Code.** Since a Raptor code is a serial concatenation of two component codes, two decoding schemes can be considered

- (a) In a classical setting, the tandem decoding (TD) is used: it consists in decoding the LT code first and then the precode independently, using the soft extrinsic information on the input symbols as *a priori* information for the precode.
- (b) In a joint decoding (JD) framework, both decoder components of the Raptor decoder provide extrinsic information to each other in an iterative way.

Most of the analysis and designs of Raptor codes in literature assume a tandem decoding. In this paper, we show that using an iterative joint decoder allows to obtain better coding solutions to some issues, such as robustness to channel variation and to finite length design. In the next section, we draw the density evolution analysis under Gaussian approximation, and show the advantages of considering a joint decoder.

### 3. Asymptotic Analysis and Design of Raptor Codes for Joint Decoding

**3.1. Asymptotic Analysis of Raptor Codes.** In this section, we derive the asymptotic analysis of the joint decoding of Raptor codes on the BIAWGN channel. The analysis is presented from the fountain point of view for our optimization purposes. To perform this asymptotic analysis, we adopt a monodimensional analysis of the BP decoder based on EXIT charts [12, 13]. It is based on a Gaussian approximation (GA) [14] of the density evolution (DEs) as presented in [15, 16]. In the iterative decoder, the messages are defined as log density ratios (LDRs) of the probability weights. Under GA assumption, the LDRs are considered as realizations of a Gaussian random variable with mean  $m$  and variance  $\sigma^2 = 2m$  [14]. We call *information content* (IC), the mutual information between a random variable representing a transmitted bit and another one representing an LDR message on the decoding graph. The IC associated to an LDR message is  $x = J(m)$  [12], where  $J(\cdot)$  is defined by

$$J(m) = 1 - \frac{1}{\sqrt{4\pi m}} \int_{\mathbb{R}} \log_2(1 + e^{-v}) \exp\left(-\frac{(v-m)^2}{4m}\right) dv, \quad (2)$$

Under JD framework, we assume that extrinsic information is exchanged between the precode and the fountain part from one decoding iteration of the fountain to the other. Moreover, we mainly consider the case of an LDPC precode. In this case, the Raptor code can be described by a single Tanner graph with two kinds of parity-check nodes: check nodes of the precode, referred to as *static check nodes* and parity-check nodes of the LT code, referred to as *dynamic check nodes* in the following [5]. Throughout the decoding iterations, we analytically track the evolution of the IC associated with the LDR messages that are located at the fountain side of the Tanner graph.

**3.1.1. Information Content Evolution.** We denote by  $x_u^{(l)}$  (resp.,  $x_v^{(l)}$ ) the IC associated to messages on an edge connecting a dynamic check node to an input symbol (resp., an input symbol to a dynamic check node) at the  $l$ th decoding iteration. Moreover, we denote by  $x_{\text{ext}}^{(l-1)}$  the extrinsic information passed from the LT code to the precode, at the  $l$ th decoding iteration, and  $T(\cdot) : x \mapsto T(x)$  the IC transfer function of the precode. The extrinsic information passed by the precode to the LT code is then  $T(x_{\text{ext}}^{(l)})$ . The notations are summarized in Figure 2.

When accounting for the transfer function of the precode, the IC update rules in the Tanner graph can be written as follows (see other references for the detailed explanation of such system of equations [5, 12, 15])

- (i) Input symbol message update:

$$x_v^{(l)} = \sum_{i=1}^{d_v} \iota_i J\left((i-1)J^{-1}\left(x_u^{(l-1)}\right) + J^{-1}\left(T\left(x_{\text{ext}}^{(l-1)}\right)\right)\right). \quad (3)$$

- (ii) Dynamic check node message update is

$$x_u^{(l)} = 1 - \sum_{j=1}^{d_c} \omega_j J\left((j-1)J^{-1}\left(1 - x_v^{(l)}\right) + f_0\right) \quad (4)$$

$$\text{with } f_0 \triangleq J^{-1}(1 - J(2/\sigma^2)).$$

- (iii) Precode extrinsic information update is

$$x_{\text{ext}}^{(l)} = \sum_i \iota_i J\left(iJ^{-1}\left(x_u^{(l)}\right)\right). \quad (5)$$

Replacing (3) in (4) gives (7), the monodimensional recursive equation:  $x_u^{(l)} = F(x_u^{(l-1)}, \sigma^2, T(\cdot))$  that describes the evolution through one joint decoding iteration of the IC of the LDRs at the output of the dynamic check nodes (fountain part):

$$\begin{aligned} x_u^{(l)} &= F\left(x_u^{(l-1)}, \sigma^2, T(\cdot)\right) \\ &= 1 - \sum_{j=1}^{d_c} \omega_j J\left((j-1)J^{-1}\left(1 - \sum_{i=1}^{d_v} \iota_i J\left((i-1)J^{-1}\left(x_u^{(l-1)}\right) + J^{-1}\left(T\left(x_{\text{ext}}^{(l-1)}\right)\right)\right) + f_0\right)\right) \end{aligned} \quad (6)$$

$$+ J^{-1}\left(T\left(x_{\text{ext}}^{(l-1)}\right)\right) \quad (7)$$

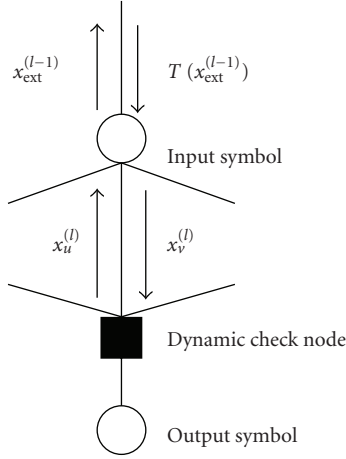


FIGURE 2: Notations used for the asymptotic analysis of a Raptor code with IC evolution.

Note that for a given distribution  $\iota(x)$ , this expression is linear with respect to the coefficients of  $\omega(x)$ , which is the distribution that we intend to optimize. Let us also point out that (7) is general since it reduces to the classical tandem decoding case by setting the extrinsic transfer function to  $x \mapsto T(x) = 0$  for all  $x \in [0; 1]$ , thus assuming that no information is exchanged between the precode and the fountain.

**3.1.2. On the Precode IC Transfer Function.** If the precode is an error correcting code that has a soft-input soft-output decoding algorithm, then its transfer function  $x \mapsto T(x)$  can be estimated with Monte Carlo simulations. When the precode is an LDPC code, an analytical expression of the transfer function can be given. Let  $\lambda(x)$  (resp.,  $\Lambda(x)$ ) denote the variable edge (resp., node) degree distribution and  $\rho(x)$  the check edge degree distribution, then the IC transfer function [17] is given by:

$$T(x) = \sum_{i=2}^{d_v} \Lambda_i J \left( i J^{-1} \left( 1 - \sum_{j=2}^{d_c} \rho_j J((j-1)J^{-1}(1-x)) \right) \right). \quad (8)$$

Note that even if we used—for simplification—the same notation  $d_v$  for the maximum connexion degree for the precode (8) and the fountain (7), these two degrees could take different values. Using (8) as stated implicitly implies that: (a) one inner iteration is performed and (b) the messages in the precode Tanner graph are reinitialized each time the fountain passes its soft information to the precode. This pessimistic assumption is crucial to lead to a linear optimization problem with respect to the optimization parameter. However, it has been found sufficient for the design of good output degree distributions. Note that, in practice, we will keep during the decoding the computed values of the extrinsic messages everywhere in the Tanner graph, without any re-initialization.

In the rest of the section, we derive the conditions on the distribution first monomials such that the density evolution

equations under Gaussian approximation converge to a stable fixed point. The same study with similar results has been conducted in [5], but for a different set of equations since the authors used the evolution of the mean of the Gaussian density, instead of the information content.

**3.1.3. Fixed Point Characterization.** In an IC evolution analysis, the convergence is guaranteed by the condition  $F(x, \sigma^2, T(\cdot)) > x$ . Unfortunately, there are no trivial solutions for the fixed point of (7). Replacing  $x_u^{(l-1)}$  by 1 (its maximal value) and using the fact that  $T(1) = 1$  in (7), we can however obtain the following upper bound:

$$\lim_{x \rightarrow 1} F(x, \sigma^2, T(\cdot)) = J\left(\frac{2}{\sigma^2}\right) \triangleq x_0. \quad (9)$$

Thus, the IC of the LT part of a Raptor code is upper bounded through the decoding iterations by  $x_0$ , which is equal to the capacity of a BIAWGN channel with noise variance  $\sigma^2$ .

**3.1.4. Starting Condition.** If the following condition is not met, then the decoding of a Raptor code to a zero error fixed point is not possible.

**Proposition 1** (Starting condition). *The decoding process can begin if and only if  $F(0, \sigma^2, T(\cdot)) > 0$  and the following holds:*

$$F(0, \sigma^2, T(\cdot)) > \varepsilon \iff \omega_1 > \frac{\varepsilon}{J(2/\sigma^2)}. \quad (10)$$

*Proof.* The decoding process can begin if and only if  $x_u^{(1)} > \varepsilon$ , for some arbitrarily small  $\varepsilon > 0$ . At the first iteration,  $x_u^{(0)} = 0$ , and (7) gives  $x_u^{(1)} = F(0, \sigma^2, T(\cdot)) = \omega_1 J(2/\sigma^2)$ .  $\square$

Roughly speaking, one must have  $\omega_1 > 0$  for the decoding process to begin. Thus, the parameter  $\varepsilon$  appears to be a design parameter to ensure that  $\omega_1 \neq 0$ . In practice, the value of  $\varepsilon$  can be chosen arbitrarily small.

**3.1.5. Lower Bound on  $\omega_2$  (Flatness Condition).** In [5], an important bound on  $\Omega_2$ , the proportion of output symbols of degree 2, has been derived for sequences of capacity achieving distributions, which is a counterpart of the stability condition [16] for LDPC codes. Following steps of [5], we derive a similar bound for the proportion  $\omega_2$  of a capacity achieving distribution  $\omega(x)$ , specifically for the IC evolution system of equations.

**Proposition 2.** *When considering IC evolution, the necessary condition for a distribution  $\omega(x)$  to be capacity achieving is:*

$$F'(0, \sigma^2, T(\cdot)) > 1 \iff \omega_2 > \frac{1}{\alpha e^{-f_0/4}}. \quad (11)$$

*Proof.* see the appendix  $\square$

This lower bound on the output nodes of degree 2 for a capacity achieving output degree distribution ensures that  $x = 0$  is *not* an attractive fixed point of the decoder (i.e., the decoder successfully starts). We point out that the IC



evolution method leads to a slightly different result than the one obtained with mean evolution [5]. However, the same phenomenon has been observed for the derivation of the stability condition of LDPC codes.

**3.2. Design of Output Degree Distributions.** In this section, we explicit the optimization problem for the design of good output degree distributions, and give some complementary results that we use for the choice of the design parameters. We assume that the channel parameter  $\sigma^2$  is known, that is to say that the output degree distribution is optimized for a given channel parameter.

**3.2.1. Optimization Problem Statement.** For a given value  $\alpha$ , the optimization of an output distribution consists in maximizing the rate of the corresponding LT code: this is achieved when maximizing  $\Omega'(1) = \sum_i \Omega_i i$ , which is equivalent to minimizing  $\sum_i \omega_i / i$ . Thus, the optimization problem can be stated as follows:

$$\omega_{\text{opt}}(x) = \arg \min_{\omega(x)} \sum_j \frac{\omega_j}{j} \quad (12)$$

subject to the following constraints  $[C_i]$  (according to the previous section).

- [C1] *Proportion constraint.*  $\sum_i \omega_i = 1$ . Since  $\omega(x)$  is a probability distribution, its coefficients must sum up to 1.
- [C2] *Convergence constraint.*  $F(x, \sigma^2, T(\cdot)) > x$  for all  $x \in [0; x_0 - \delta]$  for some  $\delta > 0$ . To ensure the convergence of the iterative process, we must have  $F(x, \sigma^2, T(\cdot)) > x$ . However, this inequality cannot hold for each and every value of  $x$ : the analysis in Section 3.1.3 shows that the fixed point of  $F(x, \sigma^2, T(\cdot))$  is smaller than  $x_0 = J(2/\sigma^2)$ . Therefore, we must fix a margin  $\delta > 0$  away from  $x_0$ , and then by discretizing  $[0; x_0 - \delta]$  and requiring inequality to hold on the discretization points, we obtain a set of inequalities that need to be satisfied. The influence of the parameter  $\delta$  is discussed in Section 3.2.3.
- [C3] *Starting condition.*  $\omega_1 > \varepsilon/J(2/\sigma^2)$  for some  $\varepsilon > 0$ .
- [C4] *Flatness condition.*  $\omega_2 > 1/\alpha e^{-f_0/4}$ .

For a given value of  $\alpha$ , and a given channel parameter  $\sigma^2$ , the cost function and the constraints are linear with respect to the unknown coefficients  $\omega_i$ . Therefore, the optimization of an output degree distribution can be written as a linear optimization problem that can be efficiently solved with linear programming.

**3.2.2. Parameter  $\alpha$ .** The average degree of input symbols  $\alpha$  is the main design parameter of the optimization problem. For increasing values of the design parameter  $\alpha$ , we optimized output degree distributions as explained in the previous section. As illustrated on Figure 3, there is a value for  $\alpha$  that maximizes the corresponding rate of the LT code. In this example, the distributions are optimized for a BIAWGN

channel of capacity  $C = 0.5$ , with a regular (3,60) precode of rate  $R_p = 0.95$ . Remarking that we have as performance limit  $R_{\text{LT}} R_p < C$ , we get a lower bound on  $R_{\text{LT}}^{-1}$ . In our case, this is given by

$$R_{\text{LT}}^{-1} > \frac{R_p}{C} = \frac{0.95}{0.5} = 1.9. \quad (13)$$

*Remark 1.* The preceding example leads to the following general remark. As  $R_{\text{LT}} R_p < C$ , we get an upper bound on the maximum achievable rate of the fountain:  $R_{\text{LT}} < C/R_p$ . Note that it is always greater than  $C$ . So, an effective optimization of the fountain should give a rate as close as possible to this limit, as observed in our example. Note that effectively, the “best” fountain obtained through optimization has an effective rate  $R_{\text{LT}} > C$ .

We now show that there is a minimum value  $\alpha_{\min}$  under which it is not possible to design *zero error* output degree distributions. Let us first assume that the fountain part of the Tanner graph has converged to its fixed point  $x_u^{(\infty)} < x_0 < 1$ . The extrinsic information content transmitted to the precode is upper bounded by

$$x_{\text{ext}} \leq J\left(\alpha J^{-1}\left(x_u^{(\infty)}\right)\right). \quad (14)$$

With the re-initialization assumption of the precode Tanner graph (see Section 3.1), we can assume that the precode is an LDPC code with asymptotic decoding threshold  $x_p$ . This means that if the precode is initialized with an information content—coming from the fountain—greater than  $x_p$ , then the information content of the precode alone will converge to 1, and the Raptor code has a threshold behavior. It follows that the minimum value of  $\alpha$  is given by the condition  $x_{\text{ext}} > x_p$ , which gives

$$\alpha \geq \frac{\sigma^2 J^{-1}(x_p)}{2} \triangleq \alpha_{\min}. \quad (15)$$

Note that although this condition looks like what we implied a tandem decoder, the value of  $x_u^{(\infty)}$  is effectively obtained with the joint decoder equations (7).

**3.2.3. Parameter  $\delta$ .** Following the same trend as in the previous section, and recalling that  $x_u^{(\infty)} = x_0 - \delta$  for a converging output distribution, we can also discuss how to fix the value of  $\delta$  in the optimization procedure. Again, one must have  $x_{\text{ext}} > x_p$ , and for some value  $\alpha \geq \alpha_{\min}$ , it follows that

$$\delta \leq x_0 - J\left(\frac{\sigma^2 J^{-1}(x_p)}{2}\right). \quad (16)$$

We recall that  $\delta$  represents a margin away from  $x_0$ : the choice  $\delta = 0$  leads to an overly stringent optimization problem. Moreover, the larger  $\delta$ , the higher the asymptotic rate, because the optimization problem becomes less constrained when  $\delta$  becomes larger. However, inequality (16) shows that  $\delta$  cannot be chosen arbitrarily. In practice, a good choice for  $\delta$  is therefore a value as close as possible to the right hand of (16).

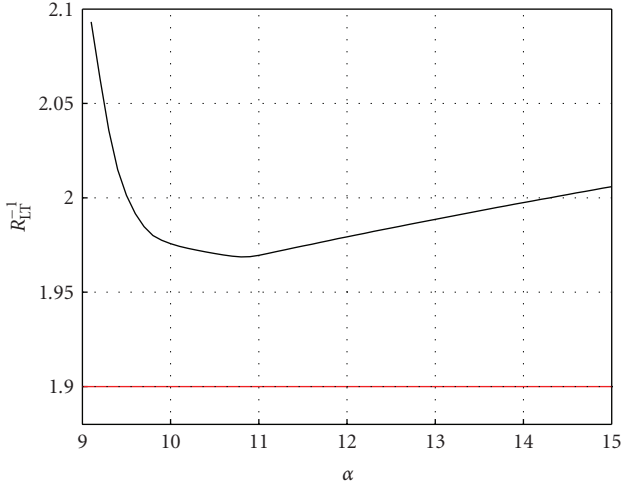


FIGURE 3: Asymptotic rate of an LT code:  $R_{LT}^{-1}$  versus  $\alpha$ . For increasing values of  $\alpha$ , we optimize a distribution to match a (3,60) regular LDPC precode of rate  $R_p = 0.95$  on a BIAWGN channel of capacity  $C = 0.5$  ( $\sigma = 0.9786$ ), and compute the a posteriori rate  $R_{LT} = \Omega'(1)/\alpha$ . It appears that is an optimal value for  $\alpha$  that minimizes  $R_{LT}^{-1}$ , that is, that minimizes the asymptotic overhead.

**3.2.4. Simulation Results.** The simulation results are illustrated in terms of BER versus overhead  $\epsilon$ . We used a regular (3,60) LDPC precode of length  $N = 65000$ , generated randomly. We compare the distribution  $\Omega_E(x)$  proposed in [5, page 2044], with both and decoders, to the following distribution that we optimized for with our method:

$$\begin{aligned} \Omega_B(x) = & 0.00428x + 0.49924x^2 + 0.01242x^3 + 0.34367x^4 \\ & + 0.04604x^{10} + 0.06181x^{11} + 0.02163x^{22} \\ & + 0.01091x^{23}. \end{aligned} \quad (17)$$

Simulation results are reported on Figure 4.

For the state-of-the-art distribution  $\Omega_E(x)$  there is very little difference between and decoders. This can be explained by the fact that the distribution has not been optimized to take into account the information provided by the precode. Compared to the distribution  $\Omega_E(x)$ , our distribution  $\Omega_B(x)$  appears to operate closer to the channel capacity: the overhead is more that 10% in the first case and less than 5% for our distribution. This result shows that one can design better output degree distributions by proper optimization with a joint decoding framework.

**3.3. Threshold of a Raptor Code.** In this section, we discuss the threshold behavior of Raptor codes under joint decoding with the IC evolution model, and compute numerically the thresholds for the two distributions  $\Omega_E(x)$  and  $\Omega_B(x)$ .

#### 3.3.1. Threshold Behavior of a Raptor Code.

**Definition 1 (Threshold).** The *a posteriori* rate is the rate below which the decoding is successful. The threshold  $\epsilon^*$  of

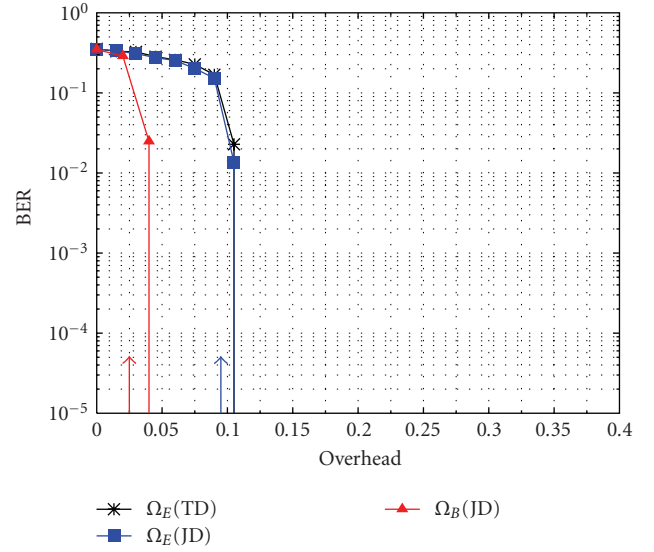


FIGURE 4: BER versus overhead for a Raptor code defined with a regular (3,60) LDPC precode of size  $N = 65000$ . We compare  $\Omega_B(x)$ , a distribution that we optimized for joint decoding, to  $\Omega_E(x)$  proposed in [5] under tandem decoding (squares) and under joint decoding (stars). The thresholds of the corresponding distributions are also reported on the figure (c.f. Section 3.3). Simulations are run on a BIAWGN channel of capacity  $C = 0.5$  ( $\sigma = 0.9786$ ) with 600 decoding iterations.

a Raptor code is the asymptotic overhead corresponding to expectation of its a posteriori rate.

We only consider the case such that the precode is a block error correcting code with a threshold behavior (an LDPC code e.g.). For tandem decoding, it is clear that the Raptor code has a threshold behavior: when LT code converges to its fixed point, it is sufficient that this fixed point is such that the extrinsic information passed to the precode is higher than the precode's threshold.

In the case of joint decoding, we adopt the same strategy, except that during the convergence of the extrinsic information passed from the fountain to the precode to its limiting value  $x_u^{(\infty)}$ , we assume belief propagation decoding on the whole Raptor code Tanner graph. The scheduling that we propose has then two steps: during the first step, the Raptor code is decoded under joint decoding, and the LT part of the Tanner graph converges to its fixed point. The convergence is guaranteed by (7) under Gaussian approximation. During the second step, the precode is decoded alone, and the extrinsic information passed from the LT code is used as *a priori* information for the precode. Since the precode is assumed to have a threshold, the joint decoding of a Raptor code with the proposed scheduling exhibits a threshold behavior.

**3.3.2. Robustness against Channel Parameter Mismatch.** To compute the threshold of a Raptor code, we use a numerical method that is an instance of Density Evolution (DEs), by Monte Carlo simulations. This method gives as good

TABLE 1

	$(d_v, d_c)$	Code rate (R)	Number 4-cycles	Number 6-cycles
$K = 4096$	(3,80)	0.9625	643	683392
	(3,60)	0.95	0	259567
	(3,40)	0.925	0	47157
	(3,30)	0.9	0	9728
	$(d_v, d_c)$	Code rate (R)	Number 4-cycles	Number 6-cycles
$K = 2048$	(3,80)	0.9625	3536	693044
	(3,60)	0.95	860	289392
	(3,40)	0.925	0	70493
	(3,30)	0.9	0	19982
	$(d_v, d_c)$	Code rate (R)	Number 4-cycles	Number 6-cycles
$K = 1024$	(3,80)	0.9625	5356	716492
	(3,60)	0.95	2328	295760
	(3,40)	0.925	90	83869
	(3,30)	0.9	0	31394

estimations for the decoding thresholds as the histogram approach. We used the estimation of thresholds of DE to show the robustness of the designed output distribution to channel parameter mismatch.

The results in [5] show that Raptor codes are not universal on other channels than the BEC: they cannot adapt to themselves to an unknown channel noise *and* approach the capacity of the channel arbitrarily closely. However, it turns out that the distributions are quite robust to channel variation, when a joint decoder is used. In order to show this robustness, we have computed for different channel capacities the thresholds of the distribution  $\Omega_E(x)$  [5] and  $\Omega_B(x)$  (our distribution). The results are reported on Figure 5 and it can be seen that both distributions have almost constant thresholds for all considered capacities, which shows that even though not universal, Raptor codes on the BIAWGN channel with joint decoding are very robust. Moreover, one can see that our optimization procedure produces an output degree distribution with thresholds outperforming the one of [5] for all capacities. For example, at  $C = 0.4$ , the threshold is only 2% away from the capacity of the channel.

#### 4. Finite Length Design

In this section, we discuss some important issues concerning the choice of the precode, in the perspective of designing efficient Raptor codes for small to moderate lengths. Indeed, the limitations in designing good high-rate precode for considered code lengths (i.e., with good girth properties) imposes the consideration of lower rate precodes. Using our asymptotic optimization method, we show that the choice of a rate lower than usually proposed for precodes enables to design good Raptor codes. We obtain raptor codes which perform well at small lengths, with almost no asymptotic loss. We show in particular that the error floor can be greatly reduced by properly choosing the rate splitting between the precode and the LT code.

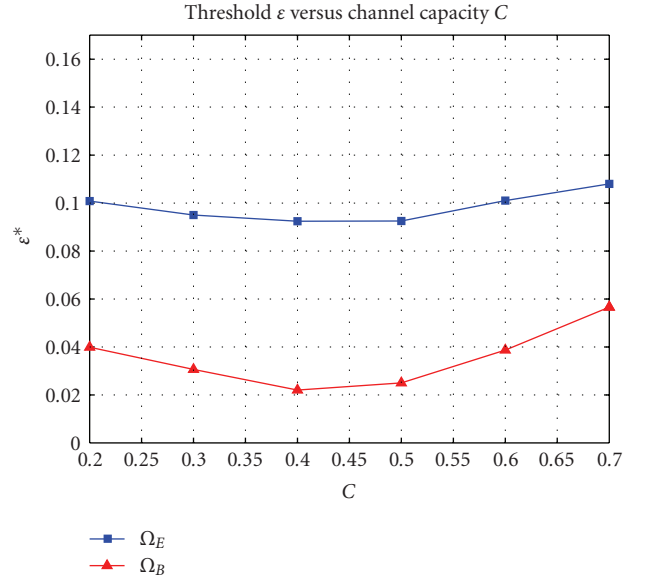


FIGURE 5: Thresholds of two distributions optimized for  $C = 0.5$ , for different channel capacities. We compare  $\Omega_B(x)$ , a distribution that we optimized for joint decoding, to  $\Omega_E(x)$  proposed in [5] decoded under joint decoding.

**4.1. The Rate Splitting Issue.** In literature, the rate of the precode is usually chosen very close to 1, for the following reason. The optimization of output degree distributions allows to design LT codes such that the fraction of unrecovered input symbols is extremely low. Choosing a very high rate precode is a valid strategy when the two components of the Raptor code are decoded sequentially, and when the information block length is sufficiently high so that the asymptotic analysis holds.

The choice of a high-rate precode could nevertheless be a suboptimal choice when we consider iterative joint decoding of the precode and the LT code and/or the block length is small. Indeed, for short to moderate lengths, the topology

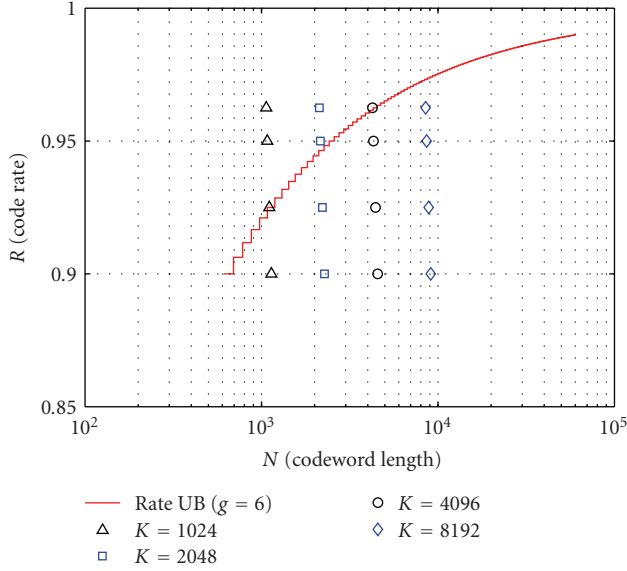


FIGURE 6: Upper bound on the code rate (Rate UB) such that a regular  $(3, d_c)$  LDPC code of girth 6 (no 4-cycles) and size  $N$  exists.

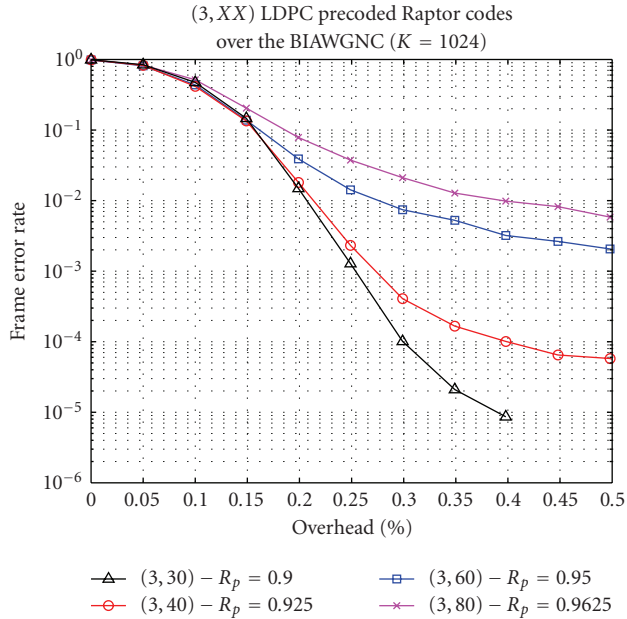


FIGURE 7: Performance of LDPC precoded Raptor codes of size  $K = 1024$ . Only the lowest considered rate  $R_p = 0.9$  shows good performance. For all other precode rates, the code exhibits an error floor, which can be explained by the large number of small cycles in the precode Tanner graphs.

of the overall Tanner graph in terms of short cycles and subsequent stopping/trapping sets needs to be considered for the optimization. Using graph theoretic argument, it can be shown that, using a very high rate LDPC precode can introduce a large number of length-4 cycles. More precisely, the code length such that a LDPC codes of girth 6 (no length-4 cycles) exist grows exponentially with the check node degree  $d_c$  [10], hence grows with the code rate

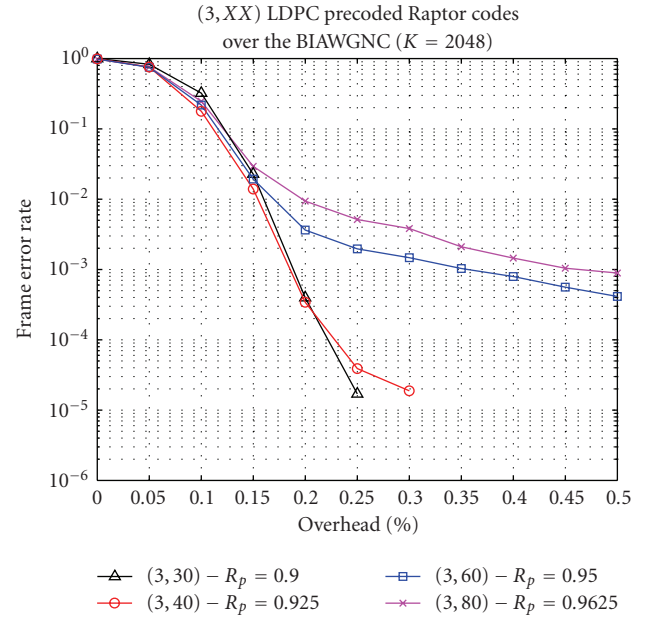


FIGURE 8: Performance of LDPC precoded Raptor codes of size  $K = 2048$ . The lowest rates  $R = 0.9$  and  $R_p = 0.925$  show good performance, whereas the Raptor codes with precodes of higher rates exhibit a severe error floor.

(cf. e.g., the upper bound in Figure 6). Having unavoidable short cycles results in error floors which are unacceptably high, as demonstrated by our simulations. So, we need to take this fact into consideration when performing the optimization, since asymptotic arguments only are not sufficient anymore. Therefore, considering a lower rate precode has the main objective of improving the Raptor code in the error floor region for finite block lengths, by allowing LDPC precodes with girth 6.

We show in this section that if the output degree distribution is matched—with proper optimization—to the EXIT chart of a *lower rate* precode, there is almost no asymptotic loss, that is, no loss in the waterfall region, but one can obtain Raptor codes which have better error floors at finite lengths. By lower rate, we mean rates that are between  $R_p = 0.9$  and  $R_p = 0.95$ , whereas typically in the existing literature, very high rate codes, for example,  $R_p = 0.98$ , are considered. Indeed, as pointed out in the remark in Section 3.2.2  $R_{LT}$  is upper bounded by a rate greater than  $C$ . When performing joint decoding, the optimized output degree distributions tends effectively to have a rate  $R_{LT} > C$ . In fact, through the objective function of the optimization, one intends to minimize the overhead: the code will have a global rate  $R$  close to the capacity. It easily allows to consider precodes with lower rates and to raise the issue of the repartition of the overall rate between the LT code and the precode.

**4.2. On Cycle Spectrum of Finite Length LDPC Precoder.** For our purpose, we will consider Raptor codes of size



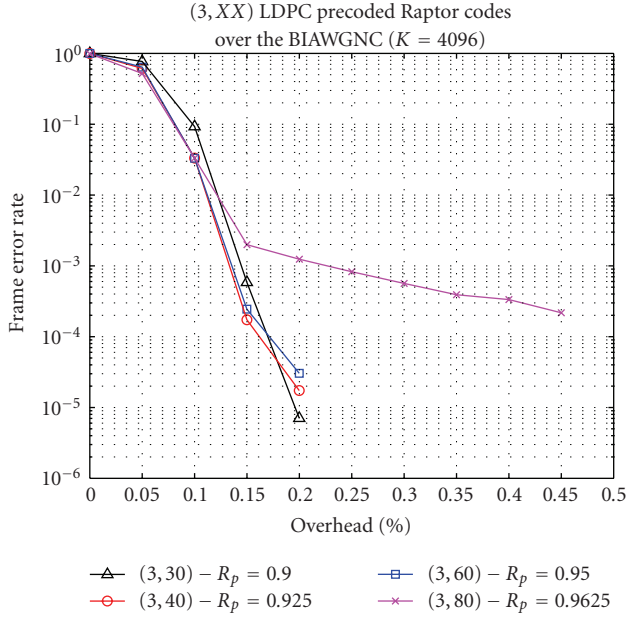


FIGURE 9: Performance of LDPC precoded Raptor codes of size  $K = 4096$ . The precoder of highest rate  $R_p = 0.9625$  exhibits a severe error floor behavior.

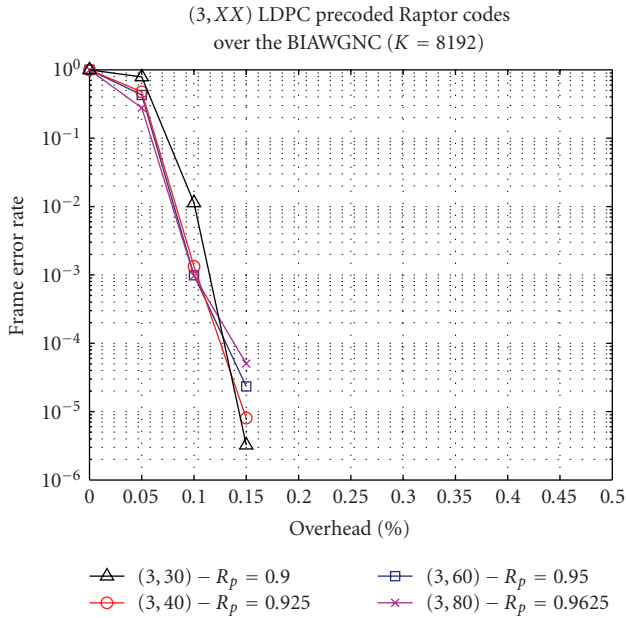


FIGURE 10: Performance of LDPC precoded Raptor codes of size  $K = 8192$ . With very little loss in the waterfall region, the precoder of rate  $R_p = 0.9$  does not exhibit an error floor.

$K = 1024, 2048, 4096$  and  $8192$ . We restricted ourselves to regular LDPC precodes because for high rates, regular codes are known to have good thresholds, close to the irregular thresholds. We considered regular LDPC precodes with the following parameters:

- (i)  $(d_v, d_c) = (3, 30)$  regular LDPC code of rate  $R_p = 0.9$ ;
- (ii)  $(d_v, d_c) = (3, 40)$  regular LDPC code of rate  $R_p = 0.925$ ;
- (iii)  $(d_v, d_c) = (3, 60)$  regular LDPC code of rate  $R_p = 0.95$ ;
- (iv)  $(d_v, d_c) = (3, 80)$  regular LDPC code of rate  $R_p = 0.9625$ .

The different LDPC precodes (one for each rate and size) were constructed with a PEG-based algorithm that minimizes the multiplicity of the girth [18]. We denote by  $X$ -cycle a cycle of length  $X$ . All the precodes of size  $K = 8192$  are of girth 6 (i.e., they have no 4-cycles in their associated Tanner graph). The other  $(d_v, d_c)$  LDPC precodes have the following cycle spectrums in Table 1.

We emphasize that the 4-cycles in the other codes do not result from a poor construction, but from the fact that for the corresponding rates and sizes, it is not possible to construct regular  $(3, d_c)$  LDPC codes [10] of girth 6 (no 4-cycles). To illustrate this fact, the upper bound on the code rate such that a regular  $(3, d_c)$  LDPC code of girth 6 and size  $N$  exists is reported in Figure 6. The coding rates and sizes of the 16 precodes that we used are also reported in the figure. It appears that our constructions with 4-cycles all correspond to a size and coding rate that do not permit the construction of graphs with no 4-cycles [10]. Note that we have considered so far rates no lower than  $R = 0.9$ . According to the upper bound on the code rate, the minimum codeword length to have a code of rate  $R = 0.9$  with girth-6 is  $N = 600$  which is very short length for our purposes. As we will see later, considering shorter lengths for having lower rate is not a reasonable choice for practical reasons with regards to the resulting overhead.

**4.3. “Asymptotic Design” for Finite Length Distributions.** If we have to consider lower rate precodes to account for finite length design constraints, one also might question whether the asymptotic analysis of the joint decoder remains valid for finite length design. Indeed, in the asymptotic regime, the concentration theorem [16] ensures that the performance of a randomly sampled code converges to the expected performance as the codeword length increases. For EXIT charts, the  $x \rightarrow F(x, \sigma^2, T(\cdot))$  characterizes the expected IC evolution of the decoder in the asymptotic regime. In the asymptotic regime, that is, when the codeword length is infinite, the decoding trajectory in the EXIT chart will fit between the curves  $y = x$  and  $y = F(x, \sigma^2, T(\cdot))$ . However, the concentration to the expected performance does not hold for the finite length case, and one must account for a certain variance in the decoding trajectories. Following the steps of [3], we propose to use the following convergence constraint in the optimization problem for finite length

[C<sub>2</sub>] Convergence constraint is

$$F(x, \sigma^2, T(\cdot)) > x + \frac{c}{K} \sqrt{1-x} \quad \forall x \in [0; x_0 - \delta] \quad (18)$$

for some  $\delta > 0$ ,

where  $c$  is a (small) positive constant.

**4.4. Simulation Results.** We optimized output degree distributions for the 4 different precodes with different rates. Figures 7 to 10 show simulation results for Raptor codes of length  $K = 1024, 2048, 4096$  and  $8192$  respectively, constructed with precodes described in the previous section. All simulations were carried out on a BIAWGN channel of capacity  $C = 0.5$  with a maximum of 600 iterations. These results show that as long as joint optimization using the precode transfer function is performed, a lower rate precode does not significantly impact the performance of the Raptor code in the waterfall region, and that contrary to what is commonly believed, using a relatively low rate for the precode ( $R_p \simeq 0.9$ ), can improve greatly the error floor performance of the Raptor code, especially at very short lengths. In fact, according to the cycle spectrum given in Section 4.2, it appears that all curves that exhibit an error floor are associated with a precode with cycles of length 4.

## 5. Conclusion

In this paper, we developed an analytical asymptotic analysis of the joint decoding of Raptor codes on a BIAWGN channel, and derived the optimization problem for the design of efficient output degree distributions. Threshold computations and simulation results show that Raptor codes designed for joint decoding outperform the traditional tandem decoding scheme, both at long and short to moderate lengths. Even though Raptor codes are not universal on other channels than the BEC, we showed that a Raptor code optimized for joint decoding for a given channel capacity also performs well on a wide range of channel capacities when joint decoding is considered. Finally, we showed that as long as joint optimization using the precode transfer function is performed, a lower rate precode does not significantly impact the performance of the Raptor code in the waterfall region, and that contrary to what is commonly believed, using a relatively low rate for the precode ( $R_p \simeq 0.9$ ), can improve greatly the error floor performance of the Raptor code.

## Appendix

### Proof of Proposition 2

Let  $F$  be defined by  $F = (\psi \circ \phi)$ , where  $\phi$  is defined by (3)  $x_v^{(l)} = \phi(x_u^{(l-1)})$ :

$$\phi(x) = \sum_{i=1}^{d_v} \iota_i J((i-1)J^{-1}(x) + \tau(x)) \quad (\text{A.1})$$

with

$$\tau(x) = J^{-1} \left( T \left( \sum_{i=1}^{d_v} \iota_i J(iJ^{-1}(x)) \right) \right), \quad (\text{A.2})$$

and  $\psi$  is defined by the Check Node message update equation (4)  $x_u^{(l)} = \psi(x_v^{(l)})$ :

$$\psi(x) = 1 - \sum_{j=1}^{d_c} \omega_j J((j-1)J^{-1}(1-x) + f_0). \quad (\text{A.3})$$

It suffices to prove the following result:  $\lim_{x \rightarrow 0} F'(x) = \alpha \omega_2 e^{-f_0/4}$ .

First we give mention that  $J(0) = 0$ ,  $J'(0) \neq 0$ . Moreover,  $T'(0) = 0$  for an LDPC precode where  $\rho_2 = 0$  which is always true for practical codes, and simple calculus gives  $\tau'(0) = 0$ . First we compute  $\phi'(0)$ :

$$\begin{aligned} \phi'(x) &= \sum_{i=1}^{d_v} \iota_i \left( (i-1)(J^{-1})'(x) + \tau'(x) \right) J'[(i-1)J^{-1}(x)], \\ \lim_{x \rightarrow 0} \phi'(x) &= \lim_{x \rightarrow 0} \sum_{i=1}^{d_v} \iota_i (i-1) \frac{J'((i-1)J^{-1}(x))}{J'(J^{-1}(x))} \\ &= \sum_{i=1}^{d_v} \iota_i (i-1) = \alpha. \end{aligned} \quad (\text{A.4})$$

Then, we compute  $\psi'(0)$ :  $\psi'(x) = \sum_{j=1}^{d_c} \omega_j (j-1)(J^{-1})'(1-x) J'[(j-1)J^{-1}(1-x) + f_0]$ .

Let  $\mu$  be defined by  $\mu = \mu(x) = (J^{-1})(1-x)$ . Then, we obtain

$$\psi'(x) = \sum_{j=1}^{d_c} \omega_j (j-1) \frac{J'[(j-1)\mu + f_0]}{J'(\mu)}. \quad (\text{A.5})$$

Then, using the following approximation of  $J'(\mu)$  for  $\mu$  given in [12];  $J'(\mu) : \log_2(e)(\sqrt{\pi}e^{-\mu/4}/4\sqrt{\mu})$ ;

$$\begin{aligned} \lim_{x \rightarrow 0} \psi'(x) &= \lim_{\mu \rightarrow \infty} \sum_{j=1}^{d_c} \omega_j (j-1) \frac{J'[(j-1)\mu + f_0]}{J'(\mu)} \\ &= \lim_{\mu \rightarrow \infty} \sum_{j=1}^{d_c} \omega_j (j-1) \sqrt{\frac{\mu}{(j-1)\mu + f_0}} e^{-((j-2)\mu + f_0)/4} \\ &= \omega_2 e^{-f_0/4}. \end{aligned} \quad (\text{A.6})$$

Finally,  $\phi(0) = 0$ , and  $F'(x) = \phi'(x)(\psi' \circ \phi)(x)$  gives  $\lim_{x \rightarrow 0} F'(x) = \alpha \omega_2 e^{-f_0/4}$ .

## Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful suggestions and comments.

## References

- [1] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *Computer Communication Review*, vol. 28, no. 4, pp. 56–67, 1998.
- [2] M. Luby, "LT codes," in *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (STOC '02)*, pp. 271–280, 2002.
- [3] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
- [4] P. Maymounkov, "Online codes," Tech. Rep. TR2003-883, New York University, November 2002.

- [5] O. Etesami and A. Shokrollahi, "Raptor codes on binary memoryless symmetric channels," *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 2033–2051, 2006.
- [6] H. Pishro-Nik and F. Fekri, "On raptor codes," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, vol. 3, pp. 1137–1141, June 2006.
- [7] R. Karp, M. Luby, and A. Shokrollahi, "Finite length analysis of LT codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '04)*, p. 39, Chicago, Ill, USA, July 2004.
- [8] E. Maneva and A. Shokrollahi, "New model for rigorous analysis of LT-codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '06)*, pp. 2677–2679, Seattle, Wash, USA, July 2006.
- [9] P. Pakzad and A. Shokrollahi, "Design principles for raptor codes," in *Proceedings of the IEEE Information Theory Workshop (ITW '06)*, pp. 13–20, March 2006.
- [10] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, 2005.
- [11] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Transactions on Communications*, vol. 52, no. 8, pp. 1242–1247, 2004.
- [12] A. Roumy, S. Guemghar, G. Caire, and S. Verdú, "Design methods for irregular repeat accumulate codes," *IEEE Transactions on Information Theory*, vol. 50, pp. 1711–1727, 2004.
- [13] A. Venkiah, C. Poulliat, and D. Declercq, "Analysis and design of raptor codes for joint decoding using information content evolution," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '07)*, pp. 421–425, Nice, France, 2007.
- [14] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 657–670, 2001.
- [15] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.
- [16] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.
- [17] S. Ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.
- [18] A. Venkiah, D. Declercq, and C. Poulliat, "Design of cages with a randomized progressive edge-growth algorithm," *IEEE Communications Letters*, vol. 12, no. 4, pp. 301–303, 2008.