Analysis of the Second Moment of the LT Decoder

Ghid Maatouk, *Student Member, IEEE,* and Amin Shokrollahi, *Fellow, IEEE*

***Abstract*—In this paper, the second moment of the ripple size during the LT decoding process is analyzed. Combined with a result by Karp et. al (2004) stating that the expectation of the ripple size is of the order of k, our work gives bounds on the error probability of the LT decoder. Further, an analytic expression for the variance of the ripple size up to terms of constant order is given, and the expression of Karp et. al for the expectation of the ripple size is refined up to terms of the order of $1/k$. This provides a first step towards an analytic finite-length analysis of LT decoding.**

***Index Terms*—Finite-length analysis, LT decoder, ripple, second moment.**

I. Introduction

We start with a brief introduction to Fountain codes, LT codes, and belief propagation (BP) decoding. For details, the reader is referred to [3], [6].

LT codes belong to the class of Fountain codes. A Fountain code generates for a given set of k input symbols a potentially limitless stream of output symbols $z_1, z_2, \ldots$, where each output symbol is produced independently by the addition of some subset of the input symbols, chosen according to a probability distribution on $\mathbb{F}_2^k$. We assume that the symbols can be either bits or binary vectors, and that there is a way for the receiver to know, for each output symbol, which input symbols it is produced from. A number of such ways are described in [3]. A good Fountain code is one for which the receiver can decode the k input symbols with high probability after collecting n output symbols, with n close to k.

Such codes are well-suited for reliable transmission of data packets over the Internet. The Internet can be modeled as an erasure channel, where each packet is either lost, discarded or delivered to the receiver. Fountain codes are well adapted for many transmission scenarios, for example when there are multiple receivers on one or multiple channels. Then each receiver can recover the k input symbols independently of the other receivers, as soon as it has collected n output symbols.

LT codes are *universal* Fountain codes [3] in that the decoding process can recover with high probability a set of k input symbols from n output symbols with n arbitrarily close to k. Universality refers to the fact that an LT code with this property achieves the capacity of *any* binary erasure channel with erasure probability p as long as $p < 1$. LT codes have the following probability distribution for generating output symbols: for each output symbol, first sample a number d (the "degree" of this symbol) from a probability distribution $\Omega = (\Omega_1, \cdots, \Omega_k)$ on the integers $1, \ldots, k$. Then pick d distinct input symbols uniformly at random and XOR them to produce the corresponding output symbol. An LT code that encodes k input symbols and uses a distribution Ω with generating function $\Omega(x) = \sum_i \Omega_i x^i$ is said to have parameters $(k, \Omega(x))$.

LT codes are decoded using BP decoding. The decoding starts when the receiver has gathered $n = (1+\epsilon)k$ output symbols, for some predetermined (relative) *overhead* ϵ. Define the *decoding graph* [6] to be an undirected bipartite graph with k nodes on one side, representing the k input symbols, and n nodes on the other, representing the output symbols. An input node is connected to an output node if the corresponding input symbol contributes to the value of the output symbol. The decoding process is as follows: the receiver randomly chooses an output symbol among the set of output symbols connected to only one input symbol. Then the value of the corresponding input symbol can be recovered directly. This value is XORed to the value of any other output symbols connected to this input symbol, then the input symbol and all its outgoing edges are removed from the graph. The decoder then repeats the operation by randomly picking another output symbol connected to only one input symbol. If at any stage before the recovery of all symbols no such output symbol is found, the decoder reports an error. The reader is referred to [3] and [6] for more details.

An important set to consider is the set of output symbols of degree 1 (the *ripple*) [3]. The size of the ripple varies during the decoding process, as high-degree output symbols become of degree 1 after the removal of their edges, and as ripple elements become useless after the recovering of their unique neighbor. The decoding is in error if and only if the ripple becomes empty before all the input symbols are recovered. A natural question is thus whether we can track the size of the ripple, in the expectation, during the decoding process. Karp et al. [2] proved that the expected ripple size is linear in k throughout most of the decoding process. Their asymptotic analytic expressions for the expected ripple size can be found in Section II. They also derive an expression for the expected *cloud* size throughout decoding, where the cloud is defined at each decoding step as the set of output symbols of degree strictly higher than 1. We are interested in the cloud size inasmuch as the cloud "feeds" the ripple during the decoding process, as higher-degree symbols lose edges. Thus, expressions for the expectation and higher moments of

This research was supported by Grant 228021-ECCSciEng of the European Research Council.

The authors are with the School of Computer and Communications Sciences, EPFL, 1015 Lausanne, Switzerland (email: {amin.shokrollahi, ghid.maatouk}@epfl.ch).

A preliminary version of this work has appeared at the International Symposium on Information Theory, Seoul, 2009.

the ripple size depend on the corresponding expressions for the cloud size.

In this paper, we extend the analysis of [2] in two ways. First, we consider higher moments of the cloud and the ripple size in order to upper bound the error probability of the LT decoder. More specifically, we use methods similar to those of [2] to derive an expression for the variance of the ripple size and prove that it is also linear in k throughout most of the decoding process. We can then use this expression together with the expression for the expectation of the ripple size to offer a guarantee for successful decoding, as follows: if, for fixed LT code parameters, $R(u)$ is the expectation and $\sigma_R(u)$ is the standard deviation of the ripple size when u symbols are unrecovered, then, if the function

$$h_\alpha(u) = R(u) - \alpha \cdot \sigma_R(u) \tag{1}$$

for some parameter α never takes negative values, we can upper bound the error probability of the LT decoder by the probability that the ripple size deviates from its mean by more than α standard deviations. This is easily done using Chebyshev's inequality.

Second, we take the first step towards an analytic finite-length analysis of the LT decoder, by providing exact expressions for the expectation (variance) of the ripple size up to $O(1/k)$ (constant) terms. This is done by considering lower-order terms in the difference equations, but also by getting tight bounds on the discrepancy introduced by approximating difference equations by differential equations.

It is worthy to note that the expressions we deal with are valid for "most of the decoding process," that is, the analysis breaks down when the number of unrecovered symbols is no longer a constant fraction of k. This is no issue, however, when one considers Raptor codes, which need only a constant fraction of the input symbols to be recovered by the LT decoder [6].

Note that the problem of considering the second moment of the size of the set of nodes of residual degree 1 in the decoding graph of LDPC codes has been studied before (see for instance [1], [4] and [5]). Our method is different from previously used methods in that we consider the state generating function of the LT decoder (defined below in (2)) in order to analyze moments of the ripple size. The reason such an analysis works for LT codes is the special structure that these codes offer: namely, the fact that the degree of output symbols is chosen independently and randomly makes a local analysis of the LT decoder much easier. A similar analysis for LDPC codes would be much more complicated, because the degree distributions on *both* sides of the decoding graph of such codes are fixed, which introduces global dependencies between the variables that describe the decoder, making such a local analysis very hard.

The rest of this paper is organized as follows. In Section II, we define the state generating function of the LT decoder and give an overview of the work of Karp et al. [2], in which they give a recursion for this generating function and use it to derive closed-form expressions for the expected size of the ripple and the cloud. In Section III, we state and prove a technical lemma on which our moment derivations will rely. Then in Section IV, we derive a closed-form expression for the second moment of the ripple size, thus obtaining an expression for the variance up to terms of constant order. We conclude in Section V.

II. Preliminaries - an expression for the expected ripple size

Let u be the number of unrecovered (*undecoded*) input symbols at a given decoding step. Define the decoder to be in state (c, r, u) if the cloud size is c and the ripple size is r at this decoding step. To each state (c, r, u), we can associate the (conditional) probability $p_{c,r,u}$ of the decoder being in this state, given that u symbols are undecoded. Define the *state generating function* of the LT decoder when u symbols are undecoded as

$$P_u(x, y) = \sum_{c \geq 0, r \geq 1} p_{c,r,u} x^c y^{r-1}. \tag{2}$$

Note that since we count only the case $r \geq 1$, $P_u(x, y)$ does not satisfy $P_u(1, 1) = 1$. Rather,

$$P_u(1, 1) = 1 - \sum_{c \geq 0} p_{c,0,u}.$$

The following theorem by Karp et al. gives a recursion for the state generating function of the LT decoder.

Theorem 1: [2] Suppose that the original code has k input symbols and that $n = k(1 + \epsilon)$ output symbols have been collected for decoding. Further, denote by Ω_i, $i = 1, \ldots, D$, the probability that an output symbol is of degree i, where D is the maximum degree of an output symbol. Then we have for $u = k + 1, k, \ldots, 1$

$$P_{u-1}(x, y) = \frac{1}{y}\left[P_u\left(x(1 - p_u) + yp_u, \frac{1}{u} + y\left(1 - \frac{1}{u}\right)\right) - P_u\left(x(1 - p_u), \frac{1}{u}\right)\right], \tag{3}$$

where for $u \leq k$,

$$p_u = \frac{\frac{u-1}{k(k-1)} \sum_{d=1}^{D} \Omega_d d(d-1) \frac{\left[\begin{array}{c} k-u \\ d-2 \end{array}\right]}{\left[\begin{array}{c} k \\ d \end{array}\right]}}{1 - u\sum_{d=1}^{D} \Omega_d d \frac{\left[\begin{array}{c} k-u \\ d-1 \end{array}\right]}{\left[\begin{array}{c} k \\ d \end{array}\right]} - \sum_{d=1}^{D} \Omega_d \frac{\left[\begin{array}{c} k-u \\ d \end{array}\right]}{\left[\begin{array}{c} k \\ d \end{array}\right]}} \tag{4}$$

and

$$\left[\begin{array}{c} a \\ b \end{array}\right] := \binom{a}{b} b!,$$

and $p_{k+1} := \Omega_1$. Further, $P_{k+1}(x, y) := x^n$.[1]

A proof of this theorem can be found in [7].

This recursion gives a way to compute the probability of a decoding error at each step of the BP decoding. As the decoding is in error at a given step if and only if the ripple at this step is empty, the error states are those of the form $(c, 0, u)$, where $u > 0$. Thus the probability of error at a given decoding step $u > 0$ is computed as

$$P_{\text{err}}(u) = \sum_{c \geq 0} p_{c,0,u} = 1 - \sum_{c \geq 0, r \geq 1} p_{c,r,u} = 1 - P_u(1, 1),$$

and the overall error probability of the decoder as

$$P_{\text{err}} = \sum_{u=1}^{k} P_{\text{err}}(u).$$

Karp et al. [2] consider the following approximation of the LT process: suppose output symbols are allowed to choose their neighbors with replacement during encoding. Then the expression in (4) for p_u is replaced by

$$p_u = \frac{1}{k} f\left(\frac{u}{k}\right) - \frac{1}{k^2} g\left(\frac{u}{k}\right) = \frac{1}{k} f\left(\frac{u}{k}\right) + O(1/k^2),$$

where

$$f(x) := \frac{x\Omega''(1-x)}{1 - x\Omega'(1-x) - \Omega(1-x)} \tag{5}$$

and

$$g(x) := \frac{f(x)}{x}. \tag{6}$$

Note that $f(x)$ is also given by

$$f(x) = \frac{d}{dx} \log\left(1 - x\Omega'(1-x) - \Omega(1-x)\right).$$

Also note that the expression for $P_{u-1}(x, y)$ in (3) is still valid with this new expression for p_u.

For simplicity, the process that we analyze in what follows is this modified LT process. Intuitively, the modified process is "worse" than the original LT process in that it allows for multiple, "useless" edges in the decoding graph. With this assumption, Karp et al. [2] use the recursion given by Equation (3) to derive difference equations for the expected size of the ripple and the cloud, and further approximate these difference equations by differential equations that they solve to get closed-form expressions for the expected ripple and cloud size. Formally, let

$$R(u) := \sum_{c \geq 0, r \geq 1} (r-1) p_{c,r,u}$$

denote the expected number of output symbols in the ripple when u symbols are undecoded, right after an output symbol is chosen for the next decoding step (the fact that this chosen output symbol leaves the ripple with probability 1 explains why the expected ripple size is not defined as $\sum r p_{c,r,u}$). Similarly, let

$$C(u) := \sum_{c \geq 0, r \geq 1} c p_{c,r,u}$$

denote the expected number of output symbols in the cloud when u input symbols are undecoded. Then Karp et al. [2] derive closed-form expressions for continuous approximations of $R(u)$ and $C(u)$. More precisely, let $\xi := u/k$ denote the *fraction* of undecoded symbols. For $\delta \in \mathbb{R}$ smaller than 1, define the set

$$S_\delta := \left\{ \frac{i}{k} \;\middle|\; i = \lfloor \delta k \rfloor, \ldots, k \right\}. \tag{7}$$

From now on, we always make the assumption that $\xi \in S_\delta$ for some strictly positive δ, so that our analysis only applies to all but the very end of the decoding process. Let $C(\xi) := C(u)/n$ be the normalized version of $C(u)$. Then the continuous function $\hat{C}(x)$ given by

$$\hat{C}(x) = c_0 \left(1 - x\Omega'(1-x) - \Omega(1-x)\right), \tag{8}$$

with

$$c_0 = 1 - (1 - \Omega_1)^{n-1}, \tag{9}$$

is a "good" approximation for $C(\xi)$ on S_δ, where the notion of "good" is made more precise in the upcoming Theorem 2.

Similarly, let $R(\xi) := R(u)/n$ be the normalized version of $R(u)$. Then the continuous function $\hat{R}(x)$ given by

$$\hat{R}(x) = x \left(c_0 \Omega'(1-x) + \frac{1}{1+\epsilon} \ln x + r_0 \right), \tag{10}$$

with

$$r_0 = \Omega_1 (1 - \Omega_1)^{n-1} - \frac{1 - (1 - \Omega_1)^n}{n}, \tag{11}$$

is a "good" approximation for $R(\xi)$ on S_δ.[2] Theorem 2 formalizes this notion of a "good" approximation.

Theorem 2: [2] Consider an LT code with parameters $(k, \Omega(x))$ and assume $n = (1+\epsilon)k$ symbols have been collected for decoding. During BP decoding, let $C(u)$ and $R(u)$ be, respectively, the expected size of the cloud and ripple as a function of the number u of undecoded input symbols. Then, under the assumptions that u is a constant fraction of k and $\Omega_1 > 0$, we have

$$\begin{aligned} C(u) &= n\hat{C}(u/k) + O(1) \\ &= n\left(1 - \frac{u}{k}\Omega'(1 - u/k) - \Omega(1 - u/k)\right) + O(1) \end{aligned}$$

and

$$\begin{aligned} R(u) &= n\hat{R}(u/k) + O(1) \\ &= (1+\epsilon)u \left(\Omega'(1 - u/k) + \frac{1}{1+\epsilon} \ln \frac{u}{k} \right) + O(1). \end{aligned}$$

[1] The expression we present here for p_u corrects some slight typos in [2].

[2] Again, these expressions for $\hat{C}(x)$ and $\hat{R}(x)$ correct some slight typos in [2].

III. Approximating solutions of difference equations

The core idea behind the analysis in [2], which also comes into play in this work, is to first express $C(u)$ and $R(u)$ as first-order derivatives of the state generating function. More precisely, note that

$$C(u) = \frac{\partial P_u}{\partial x}(1,1), \quad R(u) = \frac{\partial P_u}{\partial y}(1,1).$$

This allows the authors of [2] to use the recursion (3) to deduce difference equations for $C(u)$ and $R(u)$. These difference equations can then be approximated by differential equations whose solutions are analytic approximations for the quantities $C(u)$ and $R(u)$.

We follow the same method to find analytic approximations for the second-order derivatives of $P_u(x,y)$ that will arise during the variance analysis. The following technical lemma unifies the machinery at work in all of these derivations. It also allows us to carefully bound the discrepancy terms that arise from the continuous approximations of the quantities we are interested in.

Before we state the lemma, we point out that the functions we consider are dependent on the input length k, and we remind the reader that $O(a(k))$ for some function $a(k)$ denotes the set of functions $b(k)$ whose growth is "at most as fast" as that of $a(k)$, i.e., the set of functions $b(k)$ for which there exists a constant c and an integer k_0 such that

$$|b(k)| \leq c|a(k)| \text{ for } k \geq k_0.$$

In what follows it will be convenient to write, by an abuse of notation, $b(k) = O(a(k))$ to mean $b(k) \in O(a(k))$, or to simply say that $b(k)$ is of the order of $a(k)$. If $b(k) = O(1)$, we say that $b(k)$ is of constant order.

Lemma 1: Let $\delta \in \mathbb{R}$ be a positive constant and L a positive integer, and for $\ell = 1, \ldots, L$ and $k \in \mathbb{N}$, let $f_{\ell,k}(x)$ and $g_{\ell,k}(x)$ be continuous functions on $[\delta, 1]$ such that

$$f_{\ell,k}(x) = O(1/k), \quad \ell = 1, \ldots, L$$
$$g_{\ell,k}(x) = O(1/k^2), \quad \ell = 1, \ldots, L.$$

Furthermore, for $\ell = 2, \ldots, L$, let $A_{\ell,k}(\xi)$ be a function defined on S_δ (as defined in (7)), such that $A_{\ell,k}(\xi)$ is of constant order. Assume $A_{\ell,k}$ can be approximated by a function $\hat{A}_{\ell,k}$ continuous on $[0,1]$, with a discrepancy $d_{\ell,k}(\xi)$ of the order of $1/k$, i.e., $A_{\ell,k}(\xi) = \hat{A}_{\ell,k}(\xi) - d_{\ell,k}(\xi)$.

Now let $A_k(\xi)$ be a function defined on S_δ that satisfies the difference equation

$$\begin{aligned} A_k(\xi) - A_k(\xi - 1/k) &= f_{1,k}(\xi)A_k(\xi) + g_{1,k}(\xi)A_k(\xi) \\ &\quad + \sum_{\ell=2}^{L} f_{\ell,k}(\xi)A_{\ell,k}(\xi) + \sum_{\ell=2}^{L} g_{\ell,k}(\xi)A_{\ell,k}(\xi) \\ &\quad + O(1/k^3), \end{aligned}$$

and let $\hat{A}_k(x)$ be a twice-differentiable function on $[0,1]$ that is the solution of the differential equation

$$\hat{A}_k'(x) = k f_{1,k}(x)\hat{A}_k(x) + k \sum_{\ell=2}^{L} f_{\ell,k}(x)\hat{A}_{\ell,k}(x)$$

with initial conditions $\hat{A}_k(1) = A_k(1)$.
Then $A_k(\xi)$ can be approximated by $\hat{A}_k(\xi)$ with a discrepancy $d_k(\xi) := \hat{A}_k(\xi) - A_k(\xi)$ of the order of $1/k$ and given precisely by

$$d_k(\xi) = \frac{1}{k^2} \sum_{i=0}^{k(1-\xi)-1} D_{i,k} \prod_{j=i+1}^{k(1-\xi)-1} d_{j,k} + O(1/k^2), \quad (12)$$

where

$$\begin{aligned} D_{i,k} =& \frac{1}{2}\hat{A}_k''(1 - i/k) + k^2 g_{1,k}(1 - i/k)\hat{A}_k(1 - i/k) \\ &+ k^2 \sum_{\ell=2}^{L} g_{\ell,k}(1 - i/k)\hat{A}_{\ell,k}(1 - i/k) \\ &- k^2 \sum_{\ell=2}^{L} f_{\ell,k}(1 - i/k)d_{\ell,k}(1 - i/k), \\ d_{j,k} =& 1 - f_{1,k}(1 - j/k). \end{aligned}$$

Proof: To ease notation, we will suppress the variable k from the indices of the various functions involved. $A(\xi)$ satisfies the recursion

$$\begin{aligned} A(\xi - 1/k) =& (1 - f_1(\xi))A(\xi) - g_1(\xi)A(\xi) \\ &- \sum_{\ell=2}^{L} f_\ell(\xi)\hat{A}_\ell(\xi) + \sum_{\ell=2}^{L} f_\ell(\xi)d_\ell(\xi) \\ &- \sum_{\ell=2}^{L} g_\ell(\xi)\hat{A}_\ell(\xi) + O(1/k^3). \end{aligned} \quad (13)$$

We can write the Taylor series expansion of $\hat{A}(x)$ around $\xi - 1/k$ as

$$\hat{A}(\xi - 1/k) = \hat{A}(\xi) - \frac{1}{k}\hat{A}'(\xi) + \frac{1}{2k^2}\hat{A}''(\xi) + O(1/k^3).$$

By assumption,

$$\hat{A}'(\xi) = k f_1(\xi)\hat{A}(\xi) + k \sum_{\ell=2}^{L} f_\ell(\xi)\hat{A}_\ell(\xi),$$

therefore we have

$$\begin{aligned} \hat{A}(\xi - 1/k) =& (1 - f_1(\xi))\hat{A}(\xi) - \sum_{\ell=2}^{L} f_\ell(\xi)\hat{A}_\ell(\xi) \\ &+ \frac{1}{2k^2}\hat{A}''(\xi) + O(1/k^3). \end{aligned} \quad (14)$$

We wish to bound the discrepancy term $d(\xi) = \hat{A}(\xi) - A(\xi)$. For this, we find a recursive expression for it. We know that $d(1) = 0$ and that $d(\xi - 1/k)$ can be related to $d(\xi)$ by

subtracting the recursion (13) for $A(\xi)$ from the recursion (14) for $\hat{A}(\xi)$:

$$\begin{aligned} d(\xi - 1/k) =& \hat{A}(\xi - 1/k) - A(\xi - 1/k) \\ =& (1 - f_1(\xi))d(\xi) + \frac{1}{2k^2}\hat{A}''(\xi) + g_1(\xi)\hat{A}(\xi) \\ & - \sum_{\ell=2}^{L} f_\ell(\xi) d_\ell(\xi) + \sum_{\ell=2}^{L} g_\ell(\xi)\hat{A}_\ell(\xi) + O(1/k^3). \end{aligned}$$

This recursion readily gives the closed-form expression of Equation (12) for the discrepancy term $d(\xi)$. ∎

IV. An Expression for the Variance of the Ripple Size

We now turn to finding an analytic approximation for the variance of the ripple size by following similar methods to those of [2]. Our main result is given by the following theorem.

Theorem 3: Consider an LT code with parameters $(k, \Omega(x))$ and overhead ϵ and let $\sigma_R^2(u)$ be the variance of the ripple size throughout BP decoding, as a function of the number of undecoded symbols u. Then

$$\begin{aligned} \sigma_R^2(u) = (1+\epsilon)u & \left(\Omega'(1 - u/k) + \frac{1}{1+\epsilon} \ln \frac{u}{k} \right) \\ & \cdot \left(1 + 2(1+\epsilon)k d_R(u/k)\right) \\ & - (1+\epsilon)\frac{u^2}{k}\Omega'(1-u/k)^2 \\ & - (1+\epsilon)^2 k^2 d_N(u/k), \end{aligned} \tag{15}$$

with $d_R(\xi)$ and $d_N(\xi)$ given by (18) and (24), respectively.

The remainder of this section is devoted to proving Theorem 3.

By definition, $\sigma_R^2(u)$ is given by

$$\sigma_R^2(u) = \sum_{c \geq 0, r \geq 1} (r-1)^2 p_{c,r,u} - R(u)^2.$$

If we define

$$\begin{aligned} N(u) :=& \frac{\partial^2 P_u}{\partial y^2}(1,1) \\ =& \sum_{c \geq 0, r \geq 1} (r-1)(r-2) p_{c,r,u} \\ =& \sum_{c \geq 0, r \geq 1} (r-1)^2 p_{c,r,u} - R(u), \end{aligned} \tag{16}$$

we can relate $\sigma_R^2(u)$, $N(u)$ and $R(u)$ as follows:

$$\sigma_R^2(u) = N(u) - R(u)^2 + R(u).$$

It is thus enough to find an expression for $N(u)$ to get an expression for $\sigma_R^2(u)$. In what follows, we show that a "good" approximation for $N(u)$ is $n^2\hat{N}(u/k)$, where the function $\hat{N}(x)$ is given by Equation (27). For convenience, the analytic approximations for (a normalized version of) $N(u)$ as well as for other moments of the cloud and ripple sizes appear in Table I at the end of the section.

The goal is now to find a difference equation for a normalized version of $N(u)$ so that we can apply Lemma 1. To this end, we start by differentiating both sides of the recursion (3) twice with respect to y and evaluating at $(1,1)$. This gives us a recursion for $N(u)$:

$$\begin{aligned} N(u-1) = & \left(1 - \frac{1}{u}\right)^2 N(u) - 2p_u C(u) - 2\left(1 - \frac{1}{u}\right) R(u) \\ & + p_u^2 \frac{\partial^2 P_u}{\partial x^2}(1,1) + 2p_u \left(1 - \frac{1}{u}\right) \frac{\partial^2 P_u}{\partial x \partial y}(1,1) \\ & - 2\left[-P_u(1,1) + P_u\left(1 - p_u, \frac{1}{u}\right)\right]. \end{aligned} \tag{17}$$

Before we can proceed with solving the resulting difference equation, we first need to handle the "residual" term

$$-2\left[-P_u(1,1) + P_u\left(1 - p_u, \frac{1}{u}\right)\right],$$

which does not involve derivatives and for which we cannot find an expression independent of the state generating function. However, we can bound it under an assumption on the ripple size, as Lemma 2 shows.

Lemma 2: Assume that the ripple size r is at least 4. Then

$$-2\left[-P_u(1,1) + P_u\left(1 - p_u, \frac{1}{u}\right)\right] = 2 + O(1/k).$$

Proof: See Appendix A. ∎

In what follows, we assume that the size of the ripple is at least equal to the constant 4.[3]

Next, we give finer approximations for $C(u)$ and $R(u)$ than the ones provided in [2]. Recall that Theorem 2 approximated $C(u)$ by $n\hat{C}(u/k)$ and $R(u)$ by $n\hat{R}(u/k)$ up to a constant-order term, where $\hat{C}(x)$ and $\hat{R}(x)$ are given by (8) and (10), respectively. The following lemma gives a precise expression for the constant-order discrepancy term.

Lemma 3: The expected cloud and ripple sizes when u symbols are undecoded, where u is a constant fraction of k, can be approximated by

$$C(u) = n\hat{C}(u/k) - n d_C(u/k)$$

and

$$R(u) = n\hat{R}(u/k) - n d_R(u/k),$$

where the discrepancy terms are given by

$$\begin{aligned} d_C(\xi) &= \frac{1}{k^2} \sum_{i=0}^{k(1-\xi)-1} C_i \prod_{j=i+1}^{k(1-\xi)-1} \left(1 - \frac{c_j}{k}\right) + O(1/k^2) \\ d_R(\xi) &= \frac{1}{k^2} \sum_{i=0}^{k(1-\xi)-1} R_i \prod_{j=i+1}^{k(1-\xi)-1} \left(1 - \frac{r_j}{k}\right) + O(1/k^2), \end{aligned} \tag{18}$$

[3]It is not difficult to check at the end of the analysis, and using an inductive reasoning, that this assumption holds with high probability.

with

$$\begin{aligned} C_i =& \frac{1}{2}\hat{C}''(1-i/k) - g(1-i/k)\hat{C}(1-i/k), \\ c_j =& f(1-j/k), \end{aligned} \tag{19}$$

$$\begin{aligned} R_i =& \frac{1}{2}\hat{R}''(1-i/k) + g(1-i/k)\hat{C}(1-i/k) \\ &+ kf(1-i/k)d_C(1-i/k), \\ r_j =& \frac{1}{1-j/k}. \end{aligned} \tag{20}$$

Proof: See Appendix B. ∎

We also need to find expressions for the second-order derivatives $\frac{\partial^2 P_u}{\partial x^2}(1,1)$ and $\frac{\partial^2 P_u}{\partial x \partial y}(1,1)$. To do so, we define

$$\begin{aligned} M(u) &:= \frac{\partial^2 P_u}{\partial x^2}(1,1), \\ L(u) &:= \frac{\partial^2 P_u}{\partial x \partial y}(1,1). \end{aligned}$$

Then Lemmas 4 and 5 give closed-form expressions for $M(u)$ and $L(u)$, respectively.

Lemma 4: Let $M(\xi) := M(u)/n^2$ be the normalized version of $M(u)$, where ξ denotes the fraction u/k of undecoded symbols. Then

$$M(\xi) = \hat{M}(\xi) - d_M(\xi),$$

where

$$\hat{M}(x) = m_0 \left(1 - x\Omega'(1-x) - \Omega(1-x)\right)^2,$$

with

$$m_0 = \left(1 - \frac{1}{n}\right)\left(1 - (1-\Omega_1)^{n-2}\right), \tag{21}$$

and the discrepancy term is given by

$$d_M(x) = \frac{1}{k^2}\sum_{i=0}^{k(1-x)-1} M_i \prod_{j=i+1}^{k(1-x)-1}\left(1 - \frac{2c_j}{k}\right) + O(1/k^2),$$

with

$$\begin{aligned} M_i &= \frac{1}{2}\hat{M}''(1-i/k) \\ &\quad - \left(2g(1-i/k) + f(1-i/k)^2\right)\hat{M}(1-i/k) \end{aligned}$$

and c_j as given by (19).

Proof: See Appendix C. ∎

Lemma 5: Let $L(\xi) := L(u)/n^2$ be the normalized version of $L(u)$. Then

$$L(\xi) = \hat{L}(\xi) - d_L(\xi),$$

where

$$\begin{aligned} \hat{L}(x) = x\left(1 - x\Omega'(1-x) - \Omega(1-x)\right) \\ \cdot \left(m_0\Omega'(1-x) + \frac{c_0}{1+\epsilon}\ln x + l_0\right), \end{aligned}$$

with

$$l_0 = \frac{-1}{n} + (1-\Omega_1)^{n-2}\left(\Omega_1 + \frac{1-2\Omega_1}{n}\right), \tag{22}$$

and the discrepancy term is given by

$$\begin{aligned} d_L(x) =& \frac{1}{k^2}\sum_{i=0}^{k(1-x)-1} L_i \prod_{j=i+1}^{k(1-x)-1}\left(1 - \frac{r_j + c_j}{k}\right) \\ &+ O(1/k^2), \end{aligned}$$

with

$$\begin{aligned} L_i &= \frac{1}{2}\hat{L}''(1-i/k) - 2g(1-i/k)\hat{L}(1-i/k) \\ &\quad + \left(g(1-i/k) + f(1-i/k)^2\right)\hat{M}(1-i/k) \\ &\quad + kf(1-i/k)d_M(1-i/k) \\ &\quad - \frac{1}{1+\epsilon}f(1-i/k)\hat{C}(1-i/k) \\ &\quad - \frac{k}{1+\epsilon}d_C(1-i/k) \end{aligned}$$

and c_j and r_j as given by (19) and (20), respectively.

Proof: See Appendix D. ∎

Replacing $M(u)$ and $L(u)$ by the above expressions and using the bound of Lemma 2 for the residual term in the recursion (17), we obtain the following difference equation for $N(u)$:

$$\begin{aligned} N(u) - N(u-1) =& \left(\frac{2}{u} - \frac{1}{u^2}\right)N(u) - p_u^2 M(u) \\ &- 2p_u\left(1 - \frac{1}{u}\right)L(u) + 2p_u C(u) \\ &+ 2\left(1 - \frac{1}{u}\right)R(u) + O(1). \end{aligned} \tag{23}$$

Note that $N(u)$ as defined in Equation (16) can be as large as a constant fraction of k^2. We thus need to normalize this quantity if we are to say something meaningful about the difference $N(u) - N(u-1)$. We let $N(\xi) := N(u)/n^2$ be the normalized version of $N(u)$, where ξ denotes, as before, the fraction u/k of undecoded symbols.

Normalizing (23) and replacing the functions $M(\xi), L(\xi), C(\xi)$ and $R(\xi)$ by the approximations given by Lemmas 3, 4 and 5, we obtain

$$\begin{aligned} &N(\xi) - N(\xi - 1/k) \\ &\quad = \left(\frac{2}{k\xi} - \frac{1}{k^2\xi^2}\right)N(\xi) - \frac{1}{k^2}f(\xi)^2\hat{M}(\xi) \\ &\qquad + \left(-\frac{2}{k}f(\xi) + \frac{4}{k^2}g(\xi)\right)\hat{L}(\xi) \\ &\qquad + \left(\frac{2}{(1+\epsilon)k} - \frac{2}{(1+\epsilon)k^2\xi}\right)\hat{R}(\xi) \\ &\qquad + \frac{2}{(1+\epsilon)k^2}f(\xi)\hat{C}(\xi) + \frac{2}{k}f(\xi)d_L(\xi) \\ &\qquad - \frac{2}{(1+\epsilon)k}d_R(\xi) - \frac{2}{(1+\epsilon)^2k^2} + O(1/k^3). \end{aligned}$$

This difference equation satisfies the conditions of Lemma 1, so that a straightforward application of the lemma allows us to approximate $N(\xi)$ by $\hat{N}(\xi)$, where $\hat{N}(x)$ is the solution of the differential equation

$$\hat{N}'(x) = \frac{2}{x}\hat{N}(x) - 2f(x)\hat{L}(x) + \frac{2}{1+\epsilon}\hat{R}(x)$$

with initial condition $\hat{N}(1) = N(\xi{=}1)$.

The approximation introduces a discrepancy term $d_N(\xi) := \hat{N}(\xi) - N(\xi)$ given by

$$d_N(\xi) = \frac{1}{k^2}\sum_{i=0}^{k(1-\xi)-1} N_i \prod_{j=i+1}^{k(1-\xi)-1} n_j + O(1/k^2), \qquad (24)$$

with

$$\begin{aligned} N_i =& \frac{1}{2}\hat{N}''(1-i/k) - \frac{1}{(1-i/k)^2}\hat{N}(1-i/k) \\ & - f(1-i/k)^2\hat{M}(1-i/k) + 4g(1-i/k)\hat{L}(1-i/k) \\ & + 2kf(1-i/k)d_L(1-i/k) \\ & - \frac{2}{(1+\epsilon)(1-i/k)}\hat{R}(1-i/k) - \frac{2k}{1+\epsilon}d_R(1-i/k) \\ & + \frac{2f(1-i/k)}{(1+\epsilon)}\hat{C}(1-i/k) - \frac{2}{(1+\epsilon)^2}, \\ n_j =& 1 - \frac{2}{k(1-j/k)}. \qquad (25) \end{aligned}$$

(26)

Finally, the following theorem gives a closed-form expression for the approximation $\hat{N}(x)$.

Theorem 4: Let $\hat{N}(x)$ satisfy the differential equation

$$\hat{N}'(x) = \frac{2}{x}\hat{N}(x) - 2f(x)\hat{L}(x) + \frac{2}{1+\epsilon}\hat{R}(x)$$

with initial condition $\hat{N}(1) = N(\xi = 1)$. Then an analytic expression for $\hat{N}(x)$ is

$$\begin{aligned} \hat{N}(x) =& x^2\Big(m_0\Omega'(1-x)^2 + 2l_0\Omega'(1-x) \\ & + \frac{2c_0}{1+\epsilon}\Omega'(1-x)\ln x + \frac{2r_0}{1+\epsilon}\ln x \\ & + \frac{1}{(1+\epsilon)^2}(\ln x)^2 + n_0\Big), \qquad (27) \end{aligned}$$

where the constants c_0, m_0 and l_0 are given by (9), (21) and (22), respectively, and the value of the constant n_0 is

$$n_0 = \frac{2}{n^2}\left(1-(1-\Omega_1)^n\right) - (1-\Omega_1)^{n-2}\left(\Omega_1^2 + \frac{2\Omega_1 - 3\Omega_1^2}{n}\right).$$

Proof: See Appendix E. ∎

We are now ready to conclude the proof of Theorem 3. Plugging the expression for

$$N(u) = n^2(\hat{N}(\xi) - \hat{d}_N(\xi) + O(1/k^2))$$

given by (27) and (24) and the expression for

$$R(u) = n(\hat{R}(\xi) - d_R(\xi))$$

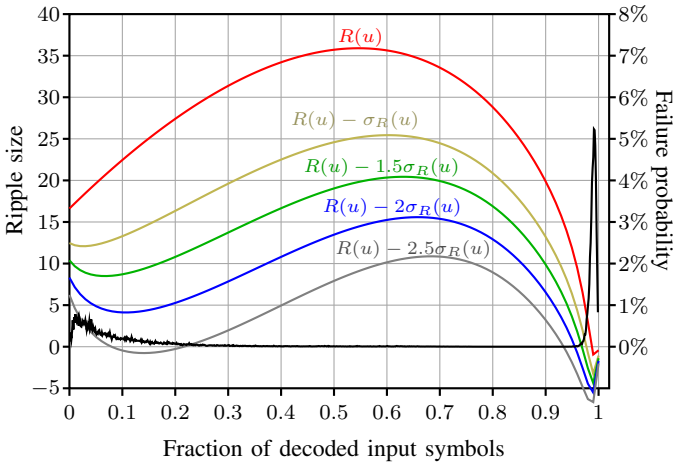

Fig. 1. Ripple size expectation and standard deviation versus the fraction of decoded input symbols. The black line is the empirical failure probability of the decoder based on 100 million simulations. It confirms that the "problem zones" of the decoder are the ones predicted by the second moment method.

given by Lemma 3 into the relation

$$\sigma_R^2(u) = N(u) - R(u)^2 + R(u)$$

immediately yields Equation (15).

Theorem 3 implies the following order estimate for $\sigma_R(u)$, which can be found using general principles as well (see for example [8]).

Corollary 1: Consider an LT code with parameters $(k, \Omega(x))$ and let $\sigma_R(u)$ be the standard deviation of the ripple size throughout BP decoding. Then

$$\sigma_R(u) = O(\sqrt{k}).$$

Figure 1 shows a plot of the expected ripple size and the functions $h_i(u)$ given by (1) for $i \in \{1, 1.5, 2, 2.5\}$, throughout the decoding process, for an LT code with $k = 800$ and $\epsilon = 0.1$, and with the degree distribution

$$\Omega(x) = \frac{1}{\frac{1}{50} + \sum_{i=2}^{50}\frac{1}{i(i-1)}}\left[\frac{1}{50}x + \sum_{i=2}^{50}\frac{1}{i(i-1)}x^i\right],$$

inspired from Luby's Ideal Soliton distribution [3]. The plot also shows the result of real simulations of this code, and confirms that the "problem zones" of the decoder are those predicted by the functions $h_i(u)$: the closer they are to the horizontal axis, the more probable it is that the decoder fails. As can be seen, there is a fair chance that the decoder fails when the fraction of decoded input symbols $1-u/k$ is between 0 and 0.2, and there is a very good chance that the decoder fails when the fraction of decoded input symbols is close to 0.95.

V. Conclusion

We have given an analytic expression for the variance of the ripple size throughout the LT decoding process. This

TABLE I
EXPRESSIONS FOR THE CONTINUOUS APPROXIMATIONS

$\hat{C}(x)$	$c_0\left(1-x\Omega'(1-x)-\Omega(1-x)\right)$
$\hat{R}(x)$	$x\left(c_0\Omega'(1-x)+\frac{1}{1+\epsilon}\ln x+r_0\right)$
$\hat{M}(x)$	$m_0\left(1-x\Omega'(1-x)-\Omega(1-x)\right)^2$
$\hat{L}(x)$	$x\left(1-x\Omega'(1-x)-\Omega(1-x)\right)$ $\cdot\left(m_0\Omega'(1-x)+\frac{c_0}{1+\epsilon}\ln x+l_0\right)$
$\hat{N}(x)$	$x^2\Big(m_0\Omega'(1-x)^2+2l_0\Omega'(1-x)+\frac{2c_0}{1+\epsilon}\Omega'(1-x)\ln x$ $+\frac{2r_0}{1+\epsilon}\ln x+\frac{1}{(1+\epsilon)^2}(\ln x)^2+n_0\Big)$
c_0	$1-(1-\Omega_1)^{n-1}$
r_0	$\Omega_1(1-\Omega_1)^{n-1}-\frac{1-(1-\Omega_1)^n}{n}$
m_0	$\left(1-\frac{1}{n}\right)\left(1-(1-\Omega_1)^{n-2}\right)$
l_0	$\frac{-1}{n}+(1-\Omega_1)^{n-2}\left(\Omega_1+\frac{1-2\Omega_1}{n}\right)$
n_0	$\frac{2}{n^2}\left(1-(1-\Omega_1)^n\right)-(1-\Omega_1)^{n-2}\left(\Omega_1^2+\frac{2\Omega_1-3\Omega_1^2}{n}\right)$

expression is asymptotically of the order of k, and we have expressed it as a function of k as a first step towards a finite-length analysis of the LT decoding. The next step is to work around the assumption that u is a "constant fraction" of k. Then we would obtain a guarantee for successful decoding as a function of the LT code parameters and overhead for practical values of k. This would then allow us to solve the corresponding design problem, namely to choose degree distributions that would make the function $h_\alpha(u)$ stay positive for as large a value of α as possible, for a fixed code length k.

VI. ACKNOWLEDGMENTS

The authors would like to warmly thank the anonymous reviewers, as well as the associate editor Pascal Vontobel, for a number of comments which has helped improve the presentation of the paper.

APPENDIX A
PROOF OF LEMMA 2

We will prove that $1-P_u(1,1)+P_u\left(1-p_u,\frac{1}{u}\right)$ can be upper bounded by a term of the order of $1/k$. To see this, note that

$$\begin{aligned}1-&P_u(1,1)+P_u\left(1-p_u,\frac{1}{u}\right)\\&=\sum_{\substack{c\geq 0\\ r\geq 0}}p_{c,r,u}-\sum_{\substack{c\geq 0\\ r\geq 1}}p_{c,r,u}+\sum_{\substack{c\geq 0\\ r\geq 1}}p_{c,r,u}(1-p_u)^c\left(\frac{1}{u}\right)^{r-1}\\&=\sum_{\substack{c\geq 0\\ r=0}}p_{c,r,u}+\sum_{\substack{c\geq 0\\ r\geq 1}}p_{c,r,u}(1-p_u)^c\left(\frac{1}{u}\right)^{r-1}.\end{aligned}$$

If we assume that $p_{c,r,u}=0$ for $r<4$ (i.e., that the ripple has size at least 4), then we can write

$$\begin{aligned}1-P_u(1,1)+&P_u\left(1-p_u,\frac{1}{u}\right)\\&=\sum_{\substack{c\geq 0\\ r\geq 4}}p_{c,r,u}(1-p_u)^c\left(\frac{1}{u}\right)^{r-1}\\&\leq k\sum_{r\geq 4}\left(\frac{1}{u}\right)^3\left(\frac{1}{u}\right)^{r-4}.\end{aligned}$$

Note that $k\sum_{r\geq 4}(1/u)^3(1/u)^{r-4}$ is the sum of $O(k^2)$ terms, each of them of the order of $1/k^3$ (since u is assumed throughout the analysis to be a constant fraction of k), so that their sum is of the order of $1/k$.

This means that as long as the ripple size is at least 4, the contribution of the residual term $1-P_u(1,1)+P_u\left(1-p_u,\frac{1}{u}\right)$ is negligible, so that we can approximate $-P_u(1,1)+P_u\left(1-p_u,\frac{1}{u}\right)$ by $-1+O(1/k)$ and hence $-2\left(-P_u(1,1)+P_u\left(1-p_u,\frac{1}{u}\right)\right)$ by $2+O(1/k)$. ■

APPENDIX B
PROOF OF LEMMA 3

A. An approximation for $C(\xi)$

Recall that

$$C(u)=\sum_{c\geq 0,r\geq 1}cp_{c,r,u}=\frac{\partial P_u}{\partial x}(1,1).$$

Differentiating both sides of the recursion (3) with respect to x and evaluating at $(1,1)$, we obtain a recursion for $C(u)$, given by

$$C(u-1)=(1-p_u)C(u)-(1-p_u)\frac{\partial P_u}{\partial x}\left(1-p_u,\frac{1}{u}\right).$$

A simple analysis, similar to that of the proof of Lemma 2, shows that the residual term

$$(1-p_u)\frac{\partial P_u}{\partial x}\left(1-p_u,\frac{1}{u}\right)$$

can be bounded by a term of the order of $1/k^2$, for $r\geq 6$. Indeed, note that under this assumption,

$$\begin{aligned}(1-p_u)\frac{\partial P_u}{\partial x}\left(1-p_u,\frac{1}{u}\right)&=\sum_{\substack{c\geq 0\\ r\geq 6}}cp_{c,r,u}(1-p_u)^c\left(\frac{1}{u}\right)^{r-1}\\&\leq\sum_{\substack{c\geq 0\\ r\geq 6}}\frac{c}{u^5}\left(\frac{1}{u}\right)^{r-6}.\end{aligned}$$

In what follows, we thus make the assumption that the ripple size is at least 6, so that we can write

$$C(u-1)=(1-p_u)C(u)+O(1/k^2).$$

We normalize $C(u)$ by n so that we can work with expressions of constant order, and obtain the following difference equation for $C(\xi)$, where ξ denotes the fraction of undecoded symbols:

$$C(\xi) - C(\xi - 1/k) = \left(\frac{1}{k}f(\xi) - \frac{1}{k^2}g(\xi)\right) C(\xi) + O(1/k^3),$$

where expressions for the functions f and g are given by (5) and (6), respectively.

This difference equation satisfies the conditions of Lemma 1. The lemma allows us to approximate $C(\xi)$ by $\hat{C}(\xi)$, where $\hat{C}(x)$ is the solution of the differential equation

$$\hat{C}'(x) = f(x)\hat{C}(x) \tag{28}$$

with initial condition $\hat{C}(1) = C(\xi = 1)$. The discrepancy $d_C(\xi) = \hat{C}(\xi) - C(\xi)$ introduced by the approximation is equal to

$$d_C(\xi) = \frac{1}{k^2} \sum_{i=0}^{k(1-\xi)-1} C_i \prod_{j=i+1}^{k(1-\xi)-1} \left(1 - \frac{c_j}{k}\right) + O(1/k^2),$$

with

$$\begin{aligned} C_i &= \frac{1}{2}\hat{C}''(1 - i/k) - g(1 - i/k)\hat{C}(1 - i/k), \\ c_j &= f(1 - j/k). \end{aligned}$$

A closed-form expression for the solution of the differential equation (28) is readily seen to be of the form

$$\hat{C}(x) = c_0 \left(1 - x\Omega'(1 - x) - \Omega(1 - x)\right),$$

where the value of the constant c_0 is to be determined by the initial condition $c_0(1 - \Omega_1) = C(\xi = 1)$. To obtain an expression for $C(\xi = 1)$, we look at the beginning of the decoding process and note that

$$C(u{=}k) = \sum_{c\geq 0, r\geq 1} c p_{c,r,k}.$$

When there are k undecoded symbols, the coefficients $p_{c,r,k}$ are given by

$$p_{c,r,k} = \begin{cases} \binom{n}{c}\Omega_1^r(1 - \Omega_1)^c & \text{if } c + r = n \\ 0 & \text{otherwise.} \end{cases} \tag{29}$$

Then

$$\begin{aligned} C(u{=}k) &= \sum_{c=1}^{n-1} c\binom{n}{c}\Omega_1^{n-c}(1 - \Omega_1)^c \\ &= n(1 - \Omega_1)\sum_{c=0}^{n-2}\binom{n-1}{c}\Omega_1^{n-1-c}(1 - \Omega_1)^c \\ &= n(1 - \Omega_1)\left(1 - (1 - \Omega_1)^{n-1}\right), \end{aligned}$$

so that

$$C(\xi{=}1) = (1 - \Omega_1)\left(1 - (1 - \Omega_1)^{n-1}\right).$$

This gives us

$$c_0 = 1 - (1 - \Omega_1)^{n-1}.$$

B. An approximation for $R(\xi)$

By definition,

$$R(u) = \sum_{c\geq 0, r\geq 1} (r - 1)p_{c,r,u} = \frac{\partial P_u}{\partial y}(1, 1).$$

Differentiating both sides of the recursion (3) with respect to y and evaluating at $(1, 1)$, we obtain the recursion

$$\begin{aligned} R(u - 1) = &\left(1 - \frac{1}{u}\right) R(u) + p_u C(u) - P_u(1, 1) \\ &+ P_u\left(1 - p_u, \frac{1}{u}\right). \end{aligned}$$

A similar analysis to that of Appendix A shows that the residual term

$$-P_u(1, 1) + P_u\left(1 - p_u, \frac{1}{u}\right)$$

can be approximated by $-1 + O(1/k^2)$, for $r \geq 5$. We thus assume in what follows that r is at least 5, so that we can work with the following difference equation for $R(u)$:

$$R(u) - R(u - 1) = \frac{1}{u}R(u) - p_u C(u) + 1 + O(1/k^2).$$

Normalizing $R(u)$ by n, we obtain a difference equation for $R(\xi)$:

$$\begin{aligned} R(\xi) - R(\xi - 1/k) = &\frac{1}{k\xi}R(\xi) - \frac{1}{k}f(\xi)\hat{C}(\xi) + \frac{1}{k(1 + \epsilon)} \\ &+ \frac{1}{k^2}g(\xi)\hat{C}(\xi) + \frac{1}{k}f(\xi)d_C(\xi) \\ &+ O(1/k^3), \end{aligned}$$

where the functions f and g are as given by (5) and (6), respectively. Note that we replaced $C(\xi)$ by its approximation $\hat{C}(\xi)$ and accounted for the resulting error.

This difference equation for $R(\xi)$ satisfies the conditions of Lemma 1, so that we can apply the lemma to approximate $R(\xi)$ by $\hat{R}(\xi)$, where the continuous function $\hat{R}(x)$ is the solution of the differential equation

$$\hat{R}'(x) = \frac{\hat{R}(x)}{x} - f(x)\hat{C}(x) + \frac{1}{1 + \epsilon}$$

with initial condition $\hat{R}(1) = R(\xi{=}1)$. The general solution of this differential equation can be easily found by standard techniques to be

$$\hat{R}(x) = x\left(c_0\Omega'(1 - x) + \frac{1}{1 + \epsilon}\ln x + r_0\right),$$

where c_0 is given by (9) and the value of r_0 can be determined by the initial condition $c_0\Omega_1 + r_0 = R(\xi{=}1)$. For the value of $R(\xi{=}1)$, we look at the beginning of the decoding process, as we did in the previous section in order to derive an expression for $C(\xi{=}1)$. By the same method we obtain

$$R(u{=}k) = n\Omega_1 - 1 + (1 - \Omega_1)^n,$$

so that

$$r_0 = \Omega_1(1 - \Omega_1)^{n-1} - \frac{1 - (1 - \Omega_1)^n}{n}.$$

The discrepancy $d_R(\xi) = \hat{R}(\xi) - R(\xi)$, introduced by approximating $R(\xi)$ by $\hat{R}(\xi)$, is given by Lemma 1 to be

$$d_R(\xi) = \frac{1}{k^2} \sum_{i=0}^{k(1-\xi)-1} R_i \prod_{j=i+1}^{k(1-\xi)-1} \left(1 - \frac{r_j}{k}\right) + O(1/k^2),$$

with

$$\begin{aligned} R_i =& \frac{1}{2}\hat{R}''(1-i/k) + g(1-i/k)\hat{C}(1-i/k) \\ &+ kf(1-i/k)d_C(1-i/k), \\ r_j =& \frac{1}{1-j/k}. \end{aligned}$$

Appendix C
Proof of Lemma 4

Recall that

$$M(u) = \frac{\partial^2 P_u}{\partial x^2}(1,1).$$

By differentiating both sides of the recursion (3) twice with respect to x and evaluating at $(1,1)$, we get the following recursion for $M(u)$:

$$M(u-1) = (1-p_u)^2 M(u) - (1-p_u)^2 \frac{\partial^2 P_u}{\partial x^2}\left(1 - p_u, \frac{1}{u}\right).$$

By a similar analysis to that of the proof of Lemma 2 (Appendix A), it can easily be shown that under the assumption $r \geq 6$, we can bound the residual term

$$(1-p_u)^2 \frac{\partial^2 P_u}{\partial x^2}\left(1 - p_u, \frac{1}{u}\right)$$

by a term of the order of $1/k$, thus obtaining the following difference equation for $M(u)$:

$$M(u) - M(u-1) = (2p_u - p_u^2)M(u) + O(1/k).$$

Normalizing by n^2, we obtain a difference equation for $M(\xi)$

$$\begin{aligned} M(\xi) - & M(\xi - 1/k) \\ =& \frac{2}{k} f(\xi) M(\xi) - \frac{1}{k^2}\left(2g(\xi) + f^2(\xi)\right) M(\xi) \\ &+ O(1/k^3), \end{aligned}$$

which satisfies the conditions of Lemma 1, so that we can apply the lemma to approximate $M(\xi)$ by $\hat{M}(\xi)$, where the continuous function $\hat{M}(x)$ is the solution of the differential equation

$$\hat{M}'(x) = 2f(x)\hat{M}(x)$$

with initial condition $\hat{M}(1) = M(\xi{=}1)$. The general solution of this differential equation is of the form

$$\hat{M}(x) = m_0 \left(1 - x\Omega'(1-x) - \Omega(1-x)\right)^2. \qquad (30)$$

The value of the constant m_0 can be found from the initial condition $\hat{M}(1) = M(\xi{=}1)$. Looking at the beginning of the decoding process, namely at the step $u = k$, we have

$$M(u{=}k) = \sum_{c \geq 0, r \geq 1} c(c-1) p_{c,r,u=k},$$

where the coefficients $p_{c,r,k}$ are given by Equation (29). Then

$$\begin{aligned} M(u{=}k) &= \sum_{c=0}^{n-1} c(c-1)\binom{n}{c}\Omega_1^{n-c}(1-\Omega_1)^c \\ &= n(n-1)\sum_{c=0}^{n-3}\binom{n-2}{c}\Omega_1^{n-2-c}(1-\Omega_1)^{c+2} \\ &= n(n-1)(1-\Omega_1)^2\left(1 - (1-\Omega_1)^{n-2}\right). \end{aligned}$$

We normalize to obtain

$$\hat{M}(1) = \left(1 - \frac{1}{n}\right)(1-\Omega_1)^2\left(1 - (1-\Omega_1)^{n-2}\right).$$

On the other hand, from (30),

$$\hat{M}(1) = m_0(1-\Omega_1)^2.$$

Equating the two expressions, we finally get

$$m_0 = \left(1 - \frac{1}{n}\right)\left(1 - (1-\Omega_1)^{n-2}\right).$$

As for the discrepancy term $d_M(\xi) = \hat{M}(\xi) - M(\xi)$ resulting from the approximation of $M(\xi)$ by $\hat{M}(\xi)$, Lemma 1 shows that it is equal to

$$d_M(\xi) = \frac{1}{k^2} \sum_{i=0}^{k(1-\xi)-1} M_i \prod_{j=i+1}^{k(1-\xi)-1} \left(1 - \frac{2c_j}{k}\right) + O(1/k^2),$$

with M_i and c_j as in the statement of Lemma 4. ■

Appendix D
Proof of Lemma 5

Recall that

$$L(u) = \frac{\partial^2 P_u}{\partial x \partial y}(1,1).$$

By differentiating both sides of the recursion (3) and evaluating at $(1,1)$, we obtain a recursion for $L(u)$:

$$\begin{aligned} L(u-1) =& p_u(1-p_u)M(u) + \left(1 - \frac{1}{u}\right)(1-p_u)L(u) \\ &- (1-p_u)C(u) + (1-p_u)\frac{\partial P_u}{\partial x}\left(1 - p_u, \frac{1}{u}\right). \end{aligned}$$

The residual term

$$(1-p_u)\frac{\partial P_u}{\partial x}\left(1 - p_u, \frac{1}{u}\right)$$

can be shown, in a similar proof to that in Appendix B, to be of the order of $1/k$ under the assumption $r \geq 5$, so that we have the following difference equation for $L(u)$:

$$\begin{aligned} L(u) - & L(u-1) \\ =& \left(\frac{1}{u} + p_u - \frac{p_u}{u}\right) L(u) - p_u(1-p_u)M(u) \\ &+ (1-p_u)C(u) + O(1/k). \end{aligned}$$

We normalize the difference equation above to obtain a difference equation for $L(\xi)$:

$$\begin{aligned} L(\xi) - L(\xi - 1/k) =& \left(\frac{1}{k\xi} + \frac{1}{k}f(\xi)\right) L(\xi) - \frac{1}{k}f(\xi)M(\xi) \\ &+ \frac{1}{k(1+\epsilon)}C(\xi) + O(1/k^3). \end{aligned}$$

Lemma 1 now yields the approximation $\hat{L}(\xi)$ for $L(\xi)$, where $\hat{L}(x)$ satisfies the differential equation

$$\hat{L}'(x) = \left(f(x) + \frac{1}{x}\right)\hat{L}(x) - f(x)\hat{M}(x) + \frac{1}{1+\epsilon}\hat{C}(x)$$

with initial condition $\hat{L}(1) = L(\xi=1)$. The general solution of this differential equation is of the form

$$\begin{aligned}\hat{L}(x) = x\,&(1 - x\Omega'(1-x) - \Omega(1-x)) \\ &\cdot \left(m_0\Omega'(1-x) + \frac{c_0}{1+\epsilon}\ln x + l_0\right),\end{aligned}$$

where the values of c_0 and m_0 are given by (9) and (21) respectively. The value of l_0 can be found using the initial condition $\hat{L}(1) = L(\xi=1)$. Looking at the initial step $u = k$ of the decoding process, a simple computation shows that

$$l_0 = \frac{-1}{n} + (1-\Omega_1)^{n-2}\left(\Omega_1 + \frac{1-2\Omega_1}{n}\right).$$

From Lemma 1, we can express the discrepancy term $d_L(\xi) = \hat{L}(\xi) - L(\xi)$ as

$$d_L(x) = \frac{1}{k^2}\sum_{i=0}^{k(1-x)-1} L_i \prod_{j=i+1}^{k(1-x)-1}\left(1 - \frac{r_j + c_j}{k}\right) + O(1/k^2),$$

with

$$\begin{aligned}L_i \;=\;& \frac{1}{2}\hat{L}''(1-i/k) - 2g(1-i/k)\hat{L}(1-i/k) \\ &+ \left(g(1-i/k) + f(1-i/k)^2\right)\hat{M}(1-i/k) \\ &+ kf(1-i/k)d_M(1-i/k) \\ &- \frac{1}{1+\epsilon}f(1-i/k)\hat{C}(1-i/k) - \frac{k}{1+\epsilon}d_C(1-i/k)\end{aligned}$$

and c_j and r_j as given by (19) and (20).

Appendix E
Proof of Theorem 4

$\hat{N}(x)$ satisfies the differential equation

$$\hat{N}'(x) = \frac{2}{x}\hat{N}(x) - 2f(x)\hat{L}(x) + \frac{2}{1+\epsilon}\hat{R}(x)$$

with initial condition $\hat{N}(1) = N(\xi=1)$. The general solution of this differential equation can be easily found by standard methods to be

$$\begin{aligned}\hat{N}(x) =& x^2\Big(m_0\Omega'(1-x)^2 + 2l_0\Omega'(1-x) \\ &+ \frac{2c_0}{1+\epsilon}\Omega'(1-x)\ln x + \frac{2r_0}{1+\epsilon}\ln x \\ &+ \frac{1}{(1+\epsilon)^2}(\ln x)^2 + n_0\Big).\end{aligned} \tag{31}$$

To find the value of the constant n_0, we use the initial condition. The value of $N(\xi=1)$ can be found by looking at the beginning of the decoding process. $N(u)$ was defined as

$$N(u) := \frac{\partial^2 P_u}{\partial y^2}(1,1),$$

so that

$$N(u=k) = \sum_{c\geq 0, r\geq 1}(r-1)(r-2)p_{c,r,u=k}.$$

Then, using the expression for $p_{c,r,k}$ given by (29), we have that

$$\begin{aligned}N(u=k) \;&=\; \sum_{r=1}^{n}(r-1)(r-2)\binom{n}{r}\Omega_1^r(1-\Omega_1)^{n-r} \\ &=\; \sum_{r=2}^{n}r(r-1)\binom{n}{r}\Omega_1^r(1-\Omega_1)^{n-r} \\ &\quad -2\sum_{r=1}^{n}r\binom{n}{r}\Omega_1^r(1-\Omega_1)^{n-r} \\ &\quad +2\sum_{r=1}^{n}\binom{n}{r}\Omega_1^r(1-\Omega_1)^{n-r}.\end{aligned}$$

Now

$$\begin{aligned}&\sum_{r=2}^{n}r(r-1)\binom{n}{r}\Omega_1^r(1-\Omega_1)^{n-r} \\ &\qquad = n(n-1)\Omega_1^2\sum_{r=0}^{n-2}\binom{n-2}{r}\Omega_1^r(1-\Omega_1)^{n-2-r} \\ &\qquad = n(n-1)\Omega_1^2.\end{aligned}$$

Similarly,

$$\begin{aligned}&\sum_{r=1}^{n}r\binom{n}{r}\Omega_1^r(1-\Omega_1)^{n-r} \\ &\qquad = n\Omega_1\sum_{r=0}^{n-1}\binom{n-1}{r}\Omega_1^r(1-\Omega_1)^{n-1-r} \\ &\qquad = n\Omega_1,\end{aligned}$$

and

$$\sum_{r=1}^{n}\binom{n}{r}\Omega_1^r(1-\Omega_1)^{n-r} = 1 - (1-\Omega_1)^n.$$

Normalizing and using the initial condition $\hat{N}(\xi=1) = N(\xi=1)$, we get

$$\hat{N}(\xi=1) = \left(1 - \frac{1}{n}\right)\Omega_1^2 - \frac{2}{n}\Omega_1 + \frac{2}{n^2}\left(1 - (1-\Omega_1)^n\right).$$

Evaluating the expression for $\hat{N}(x)$ given by (31) at $\xi = 1$, and equating it to the above expression, we get

$$\begin{aligned}m_0\Omega_1^2 + 2l_0\Omega_1 + n_0 = &\left(1 - \frac{1}{n}\right)\Omega_1^2 - \frac{2}{n}\Omega_1 \\ &+ \frac{2}{n^2}\left(1 - (1-\Omega_1)^n\right),\end{aligned}$$

where the values of m_0 and l_0 were already found to be given by the expressions in (21) and (22). Solving for n_0, we finally obtain

$$n_0 = \frac{2}{n^2}\left(1 - (1-\Omega_1)^n\right) - (1-\Omega_1)^{n-2}\left(\Omega_1^2 + \frac{2\Omega_1 - 3\Omega_1^2}{n}\right).$$

■

REFERENCES

[1] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke, "Finite-length scaling for iteratively decoded LDPC ensembles," in *IEEE Transactions on Information Theory*, vol. 55 no. 2, February 2009, pp. 473–498.

[2] R. Karp, M. Luby, and A. Shokrollahi, "Finite length analysis of LT codes," in *Proceedings of the International Symposium on Information Theory (ISIT),* Chicago, Illinois, June 2004, p. 39.

[3] M. Luby, "LT codes," in *Proceedings of the 43rd Annual IEEE Symposium Foundations of Computer Science (FOCS)*, Vancouver, BC, Canada, Nov. 2002, pp. 271–280.

[4] T. Nozaki, K. Kasai, and K. Sakaniwa, "Analytical solution of covariance evolution for regular LDPC codes," in *Proceedings of the International Symposium on Information Theory (ISIT),* Seoul, South Korea, June 2009, pp. 2649–2653.

[5] T. Nozaki, K. Kasai, and K. Sakaniwa, "Analytical solution of covariance evolution for irregular LDPC codes," November 2010. Available: http://arxiv.org/pdf/1011.1701v2.

[6] A. Shokrollahi, "Raptor codes," in *IEEE Transactions on Information Theory*, vol. 52 no. 6, June 2006, pp. 2551–2567.

[7] A. Shokrollahi, "Theory and applications of Raptor codes," in *Mathknow: Mathematics, Applied Sciences and Real Life*, Springer, December 2009.

[8] R.W.R. Darling and J.R. Norris, "Differential equation approximations for Markov chains," in *Probab. Surveys*, vol. 5, 2008, pp. 37–79.

Ghid Maatouk is currently a graduate student at EPFL, Switzerland, where she is completing her Ph.D. under the supervision of Amin Shokrollahi. She expects to graduate in 2012. She received her B.E. in Computer and Communications Engineering from the American University of Beirut (AUB) in 2006 and her Masters in Communication Systems from EPFL in 2008. Her research interests include design and analysis of rateless codes, and locally testable codes.

Ghid was the recipient of an AUB merit scholarship and an EPFL excellency scholarship.

Amin Shokrollahi has worked and published on a variety of topics, including coding theory, computational number theory and algebra, and computational/algebraic complexity theory. He is best known for his work on iterative decoding algorithms of graph based codes, an area in which he has published several influential papers, and holds more than 20 granted and pending patents. He is the co-inventor of Tornado codes, and the inventor of Raptor codes. His codes have been standardized and successfully deployed in industrial applications involving data transmission over lossy networks.

Amin finished his PhD in 1991 at the University of Bonn. From 1995 to 1998 he was a Senior Researcher at the International Computer Science Institute in Berkeley. From 1998 to 2000 he was a Member of the Technical Staff at the Mathematical Sciences Research Center at Bell Laboratories. In 2000 he became the Chief Scientist of Digital Fountain, a company specializing on fast and reliable data transmission on unreliable networks. He held this position until early 2009 when the company was acquired by Qualcomm. In 2003 Amin joined the faculty of EPFL where he holds a position as a full professor jointly in the departments of Mathematics, and of Computer Science. He is the co-founder of Kandou Technologies, a company specializing in the design and implementation of high speed and energy efficient serial links of which is he currently the CEO.

Amin is a Fellow of the IEEE. He was awarded the best paper award of the IEEE IT Society in 2002 for his work on iterative decoding of LDPC codes, the IEEE Eric Sumner Award in 2007 for the development of Fountain Codes, and the joint Communication Society/Information Theory Society best paper award of 2007 for his paper on Raptor Codes. He is also a recipient of the prestigious 2009 Advanced Research Grant of the European Union Research Council. In addition, he is the co-recipient of the 2012 IEEE Hamming medal.