Quasi-Systematic Doped LT Codes

Xiaojun Yuan, *Member, IEEE,* and Li Ping, *Senior Member, IEEE*

***Abstract*—We propose a family of binary erasure codes, namely, quasi-systematic doped Luby-Transform (QS-DLT) codes, that are rateless, almost systematic, and universally capacity-achieving without the prior knowledge of channel erasure rate. The encoding and decoding complexities of QS-DLT codes are $O(K\log(1/\epsilon))$, where K is the information length, and ϵ is the overhead. Stopping-set analysis is carried out to study the error-floor behavior of QS-DLT codes. Analysis and numerical results demonstrate that QS-DLT codes provide a low-complexity alternative to systematic Raptor codes with comparable performance.**

***Index Terms*—LT codes, Raptor codes, quasi-systematic doped Luby-Transform (QS-DLT) codes.**

I. Introduction

IT IS WELL known that properly designed low-density parity-check (LDPC) codes can achieve performance close to the capacity of binary erasure channels (BECs) with reasonable complexity [1]-[9]. In a conventional approach, a code is designed for a specific channel erasure rate (denoted by δ below) that is assumed known at the transmitter side. A coding scheme is said to be rateless if it can potentially generate coded bits limitlessly. A rateless scheme is said to be universal if the coding scheme can deliver good[1] performance without the knowledge of δ at the transmitter. Universal codes are useful, e.g., in digital fountain applications [10].

Luby Transform (LT) codes [11] are a well-known family of universal codes in the erasure channels. A standard LT code requires encoding and decoding complexity $O(K\log K)$ for reliable decoding, i.e., to ensure an error probability diminishing at least in a polynomial order of K [12], where K is the information length. In practice, linear complexity $O(K)$ is preferable. An LT code can be designed with linear complexity by reducing the degree density of information bits, but it then suffers from a high-error-floor problem. This problem can be solved using the Raptor-code approach [12] by serially concatenating a conventional LDPC code with an LT code. It was shown in [12] that a Raptor code with an overhead ϵ and complexity $O(K\log(1/\epsilon))$ can reliably recover all of the information bits.

Systematic codes, a class of codes with information bits included in transmission, are preferable in many practical situations. The encoding and decoding of a systematic code is not necessary when no erasure occurs in a transmission block, which greatly reduces the cost. LT and Raptor codes are, in their straightforward forms, non-systematic. A technique to design systematic Raptor codes was proposed in [12] with complexity roughly αK^2, where α is a small positive number independent of K.

This paper is concerned with a family of universal erasure codes, namely, quasi-systematic doped LT (QS-DLT) codes. We will say that a code C is quasi-systematic if C can be decoded using the first $(1+\eta)K$ bits in C with per bit complexity $O(\eta)$, where $0 \leq \eta \ll 1$. In other words, a quasi-systematic code can be decoded with negligible cost (in terms of redundant bits and decoding complexity) when the first $(1+\eta)K$ bits are perfectly received. Here "negligible cost" may be measured based on some practical measures, e.g., $\eta < \epsilon$. The key idea of our approach is the "parallel-type" concatenation of an LT code and a doping code. The coded bits of the doping code are randomly sampled and "doped" into the LT output bits, and hence the name. Compared to the serial concatenation structure of Raptor codes, the parallel concatenation structure has two advantages. First, it facilitates the density evolution analysis [1] [9] and the degree-distribution optimization based on the overall code structure. Second, it allows the use of the belief propagation (BP) algorithm in quasi-systematic encoding, which ensures linear encoding complexity. We show that capacity-approaching QS-DLT codes can be designed at encoding and decoding complexity of $O(K\log(1/\epsilon))$. We also present stopping-set analysis [18] to study the error-floor behavior of QS-DLT codes with finite length. Both analysis and simulation shows that QS-DLT codes are a promising low-complexity alternative to systematic Raptor codes.

II. Preliminaries

A. LT Codes

We start with a brief review of LT codes. The bipartite-graph representation of LT codes is illustrated in Fig. 1(a), where each circle represents a variable node (or equivalently, a bit), and each square represents a parity check node. For each parity check node, the value of a connected bit is equal to the binary sum of the other connected bits. The degree of a node is defined as the number of its connected edges.[2] Let $\{P_1, P_2, P_3, \cdots\}$ be a distribution over $\{1, 2, 3, \cdots\}$ which can be represented by a polynomial $P(x)$ with P_i being the

Manuscript received 14 October 2008; revised 15 February 2008. This work was fully supported by a grant from the Research Grant Council of the Hong Kong SAR, China [Project No. CUHK 2/06C].

Xiaojun Yuan was with the Department of Electrical Engineering, City University of Hong Kong, HK SAR. He is now with the Department of Electrical Engineering, University of Hawaii at Manoa, Hawaii, USA, 96822 (e-mail: xyuan24@hawaii.edu).

Li Ping is with the Department of Electrical Engineering, City University of Hong Kong, HK SAR (e-mail: eeliping@cityu.edu.hk).

Digital Object Identifier 10.1109/JSAC.2009.090805.

[1]Loosely speaking, "good" means reasonably near-capacity (but not necessarily capacity-achieving) performance. Examples of good codes include turbo and LDPC codes.

[2]Only the edges connecting the information bits are counted in determining the degree of a parity check node for LT codes, which follows the convention in [11] and [12].

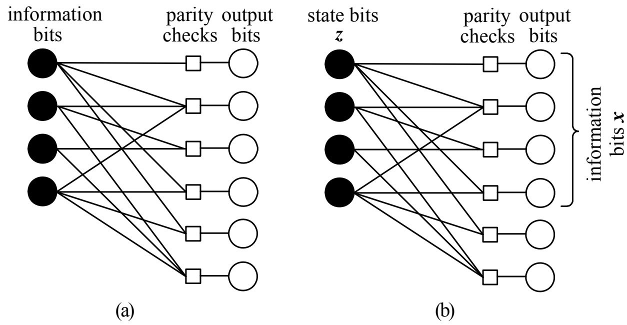

Fig. 1. (a) The bipartite graph of an LT code. (b) The structure of a systematic LT code. The code in (b) has exactly the same structure as the one in (a), with the only difference being that the information of the former is carried by a fraction of the LT output bits.

coefficient before x^i. An output (or coded) bit can be generated as follows: (i) randomly draw an integer d from the distribution $P(x)$; (ii) then generate an output bit using the sum of d randomly selected information (or input) bits. An LT code can be generated entirely based on $P(x)$, and so $P(x)$ is called the generation distribution. From Fig. 1(a), $P(x)$ is also the degree distribution of the parity check nodes.

Codes defined on a bipartite graph can also be represented in an algebraic form. Let $\boldsymbol{x} = [x_0, x_1, \cdots, x_{K-1}]^T$ and $\boldsymbol{y} = [y_0, y_1, \cdots, y_{J-1}]^T$ be the vectors of the information bits and output bits, respectively, where K is the number of information bits, J is the number of output bits, and "^{T}" represents the transpose operation. Each y_i can be represented as the inner product of two vectors, i.e., $y_i = \boldsymbol{g}_i^T \boldsymbol{x}$, where $\boldsymbol{g}_i$ is the binary vector with its 1's corresponding to the information bits connected to y_i. Then

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} \tag{1}$$

where $\boldsymbol{G} \equiv [\boldsymbol{g}_0, \boldsymbol{g}_1, \cdots, \boldsymbol{g}_{J-1}]^T$ is the LT generation matrix.

During transmission, only output bits are sent over the channel. An LT decoder tries to recover the information bits based on the received output bits using the belief propagation (BP) algorithm. Specifically, let the decoding graph be initialized as the bipartite graph induced by the received bits. The BP decoder recursively identifies a degree-1 parity check node, recovers the connected information bit, and then removes it (together with its connected edges) from the decoding graph. We will say that a bipartite graph is BP-decodable if and only if BP decoding on this graph ends with all information bits recovered.

B. Density Evolution Analysis

The density evolution technique can be used to analyze LT codes. Consider the code G defined in Fig. 1(a). Let $\Lambda(x) \equiv \sum_i \Lambda_i x^i$ be the degree distribution polynomial of the information bits, where Λ_i is the fraction of the information bits with degree *i*. Similarly, let $\lambda(x) \equiv \sum_i \lambda_i x^{i-1}$ and $\rho(x) \equiv \sum_i \rho_i x^{i-1}$ be the edge degree distributions, where λ_i (or, ρ_i) is the fraction of edges that are connected to an information bit (or, a parity check node) with degree i. Then

$$\lambda(x) = \frac{d\Lambda(x)}{dx} \Big/ \frac{d\Lambda(x)}{dx}\Big|_{x=1}, \text{ and}$$
$$\rho(x) = \frac{dP(x)}{dx} \Big/ \frac{dP(x)}{dx}\Big|_{x=1}.$$

Let E be the number of edges (excluding those connecting the output bits). Then

$$K = E\sum_i \lambda_i/i = E\int_0^1 \lambda(x)dx, \text{ and}$$
$$J = E\sum_i \rho_i/i = E\int_0^1 \rho(x)dx.$$

The overhead ϵ is defined as the normalized difference between the received bits and the information bits (normalized by the latter). For the LT code above, we have

$$\epsilon = \frac{J-K}{K} = \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} - 1. \tag{2}$$

It was shown in [1] that capacity-achieving erasure codes have a matched distribution pair, i.e.

$$\rho(x) = 1 - \lambda^{-1}(1-x) \tag{3}$$

where $\lambda^{-1}(x)$ is the inverse function of $\lambda(x)$, which exists since $\lambda(x)$ is monotonic. Then

$$\int_0^1 \rho(x)dx \overset{(a)}{=} 1 - \int_0^1 \lambda^{-1}(x)dx = \int_0^1 \lambda(x)dx \tag{4}$$

where step (a) follows from (3). Thus, together with (2), ϵ tends to zero for capacity-achieving codes.

Let d_l and d_r be the average degree of information bit and parity check nodes, respectively.[3] Then $d_l = Jd_r/K$. Considering the fact that edges are randomly connected to information bits, the degree distribution of information bits can be expressed as [12]

$$\Lambda_K(x) = (1 - d_r(1-x)/K)^{Kd_l/d_r}.$$

When $K \to \infty$ and d_l is kept as a constant, $\Lambda_K(x)$ tends to a Poisson distribution, i.e.

$$\Lambda(x) = \lim_{K\to\infty} \Lambda_K(x) = e^{d_l(x-1)} \tag{5a}$$

and so

$$\lambda(x) = \frac{d\Lambda(x)}{dx} \Big/ \frac{d\Lambda(x)}{dx}\Big|_{x=1} = e^{d_l(x-1)}. \tag{5b}$$

We will henceforth assume that K is sufficiently large, so that asymptotic distributions can be used in the density evolution analysis. The effect of a finite K will be studied in Section IV. From (3) and (5), the optimum $\rho(x)$ and $P(x)$ for LT codes are given respectively by

$$\hat{\rho}_{LT}(x) = -d_l^{-1}\ln(1-x) \tag{6a}$$

[3] d_l is a measure of per-bit complexity since the complexity of encoding and BP decoding is determined by the total number of edges on the coding graph.

and

$$\hat{P}_{LT}(x) = x + (1-x)\ln(1-x) = \sum_{i=1}^{\infty} \frac{x^{i+1}}{i(i+1)}. \quad (6b)$$

Note that (6b) is the well-known Soliton distribution [11] with the maximum degree tending to infinity. It can be seen that $P_1 = 0$ for the optimum distribution in (6b). However, BP decoding cannot start if $P_1 = 0$. Thus, we have the following (cf., Theorem 10 in [14]).

Remark I: For capacity-achieving LT codes, P_1 approaches, but never equals 0.

A matched pair of λ and ρ in (3) are not necessarily realizable. In particular, when d_l is finite, we can see from (6a) that $\hat{\rho}_{LT}(1) = \infty$, and so, $\hat{\rho}_{LT}(x)$ is not a valid distribution. Alternatively, we can use a distribution $\rho(x)$ with

$$\rho(x) > \hat{\rho}_{LT}(x), \quad 0 \le x < 1 - P_e. \quad (7)$$

Then the code based on $\rho(x)$ can reliably recover at least a proportion $1 - P_e$ of information bits, where P_e is the residual loss rate after BP decoding. It was shown in [12] that (7) can be satisfied with properly chosen P_e and $\rho(x)$. Summarizing the above discussions, we have the following remark.

Remark II: LT codes with a finite d_l (implying linear complexity in K) can only reliably recover a fixed fraction of information bits.

C. Systematic LT Codes

The above LT codes are non-systematic. Accumulate LT (ALT) codes were proposed in [13] to achieve systematic encoding. ALT codes inherit some interesting properties from LT codes, e.g., ratelessness. However, the optimum distribution of an ALT code depends on the channel erasure probability [13], and so ALT codes are not universal.

A systematic LT code can be constructed by the technique illustrated in Fig. 1(b) [12]. It has the same structure as Fig. 1(a), except that the information bits form part of the output bits. Define

$$\boldsymbol{x} = \boldsymbol{G}^{\text{ENC}}\boldsymbol{z} \quad (8)$$

where the vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ represent the information bits and the state bits, respectively, and $\boldsymbol{G}^{\text{ENC}}$ is part of the generation matrix associated with $\boldsymbol{x}$. The encoding process consists of two steps: (i) first determine $\boldsymbol{z}$ from $\boldsymbol{x}$ via BP decoding of $\boldsymbol{G}^{\text{ENC}}$; (ii) and then generate redundant LT output bits based on $\boldsymbol{z}$. The decoding process also consists of two steps: (i) first apply LT decoding to recover $\boldsymbol{z}$ from the received bits; (ii) and then recover the lost bits in $\boldsymbol{x}$ from $\boldsymbol{z}$. The above codes are universal, provided that both $\boldsymbol{x}$ and the other output bits follow the degree distribution of an LT code.

The main difficulty in realizing the above systematic LT codes is that a high average degree of $O(\log K)$ is required to ensure good performance, which leads to complexity $O(K \log K)$. The complexity can be reduced to $O(K)$ using a low average degree, but then an error-floor problem (see Remark II) occurs, which causes failure in both encoding (i.e., in constructing $\boldsymbol{G}^{\text{ENC}}$) and decoding [17]. The precoding technique used in Raptor codes has complexity αK^2 [7], which is also high (though α can be quite small). In the sequel, we develop an alternative solution with complexity $O(K)$.

III. Quasi-Systematic Doped LT (QS-DLT) Codes

In this section, we first introduce the doping technique that can relieve the error-floor problem of LT codes. The resulting codes are called the doped LT (DLT) codes, based on which we develop QS-DLT codes. We prove that these codes can perform reliable decoding with linear complexity.

A. Doping Code

The structure of the doping code is illustrated in Fig. 2(a). The doping encoder is initialized as follows: (i) repeat each of the *K* state bits *r* times, with the resulting bits called "encoding bits"; (ii) interleave the *rK* encoding bits to form an encoding line. Denote the *j*th encoding bit along the encoding line by e_j. A doping bit is generated as follows: (i) randomly select an integer *i* in $\{1, 2, ..., rK\}$, and (ii) set b_i as the doping bit, where b_i is the binary sum of $\{e_j | 1 \le j \le i\}$. In the above, $\{b_i\}$ can be calculated in batch by accumulating addition along the encoding line; each b_i may be selected with repetition, and so the doping bits can be selected independently and potentially limitlessly.

The following is the decoding process. Let $\boldsymbol{y}_{\text{d}} \equiv [y_{\text{d}}(1), y_{\text{d}}(2), y_{\text{d}}(3), ...]^T$ be the received doping bits (ordered by their positions along the encoding line) that can be represented by

$$\boldsymbol{y}_{\text{d}} = \boldsymbol{G}_{\text{d}}\boldsymbol{z} \quad (9)$$

where $\boldsymbol{z}$ is the vector of state bits, and $\boldsymbol{G}_{\text{d}}$ is the generation matrix of the doping code. $\boldsymbol{G}_{\text{d}}$ is not a sparse matrix due to the accumulating nature of the doping bits. Therefore, it is difficult to apply BP decoding directly in solving (9) for $\boldsymbol{z}$. However, $\boldsymbol{G}_{\text{d}}$ can be converted into a sparse matrix by calculating the differential of $\boldsymbol{y}_{\text{d}}$, yielding

$$\boldsymbol{y}'_{\text{d}} = \boldsymbol{R}\boldsymbol{y}_{\text{d}} = \boldsymbol{R}\boldsymbol{G}_{\text{d}}\boldsymbol{z} \quad (10a)$$

where $\boldsymbol{R}$ is a bi-diagonal matrix defined as

$$\boldsymbol{R} = \begin{pmatrix} 1 & & & 0 \\ 1 & 1 & & \\ & \ddots & \ddots & \\ 0 & & 1 & 1 \end{pmatrix}. \quad (10b)$$

The bits in $\boldsymbol{y}'_{\text{d}}$ are illustrated in Fig. 2(b). (Note: The code in Fig. 2(b) is equivalent to the one in Fig. 2(a).) Clearly, the number of "1"s in $\boldsymbol{R}\boldsymbol{G}_{\text{d}}$ is roughly *rK*, i.e., the length of the encoding line. Thus, BP decoding based on $\boldsymbol{R}\boldsymbol{G}_{\text{d}}$ requires *r* additions per bit. The doping code above is similar to RA codes [15] and semi-random LDPC codes [16] (as well as the concatenated zigzag codes [20]). However, we emphasize that the doping (coded) bits of the doping code can be selected randomly and independently (which differs from other schemes). This random selection ensures the rateless property of the doping code.

We next derive the degree distributions of the doping code. Clearly, the state-bit degree distribution of the doping code

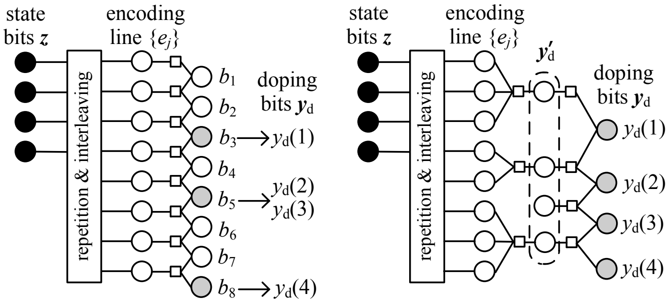

Fig. 2. (a) The structure of the doping code with K = 4 and r = 2. Note that b_5 is selected as a doping bit twice as $y_{\rm d}(2)$ and $y_{\rm d}(3)$. The degree of $y_{\rm d}(3)$ is zero. (b) An equivalent representation of the doping code in (a).

is $\Lambda_{\rm d}(x) = x^r$, and so the corresponding edge degree-distribution is $\lambda_{\rm d}(x) = x^{r-1}$. Now consider the doping bits. Let two adjacent doping bits, namely, $y_{\rm d}(i)$ and $y_{\rm d}(i-1)$, correspond to b_j and $b_{j'}$, respectively (where $y_{\rm d}(0) = b_0 = 0$). Then, according to (10a), we obtain

$$y'_{\rm d}(i) = y_{\rm d}(i) - y_{\rm d}(i-1) = b_j - b_{j'} = \sum_{j'<k\le j} e_k.$$

Define the degree of $y_{\rm d}(i)$ (and respectively $y'_{\rm d}(i)$) as the number of $\{e_k\}$ involved in the above summation. For example, in Fig. 2(a), the degrees of $y_{\rm d}(2)$ and $y_{\rm d}(3)$ are, respectively, 2 and 0.

Proposition III: Let $P_{\rm d}(x)$ be the asymptotic degree distribution of $\boldsymbol{y}_{\rm d}$ when $K \to \infty$. Then

$$P_{\rm d}(0) = re^{-\hat{p}/r} - r + \hat{p} = O\left(\hat{p}^2/r\right), \text{ and} \tag{11a}$$

$$\rho_{\rm d}(x) = \frac{dP_{\rm d}(x)}{dx} \Big/ \frac{dP_{\rm d}(x)}{dx}\bigg|_{x=1} = \frac{(1-e^{-\hat{p}/r})^2}{(1-xe^{-\hat{p}/r})^2} \tag{11b}$$

where $\hat{p}$ is the ratio of the number of the doping bits to that of the state bits.

The proof for Proposition III can be found in Appendix A. Note that $P_{\rm d}(0)$ in (11a) is the expected proportion of degree-0 (i.e., repeated) bits in $\boldsymbol{y}_{\rm d}$. (See $y_{\rm d}(3)$ in Fig. 2(a) for an example of repeated bits.) After channel erasure, the received bits contain the same proportion (i.e., $P_{\rm d}(0)$) of repeated bits that are redundant for decoding. This redundancy is the price paid for ratelessness. Later in Subsection C, we show that the performance loss related to $P_{\rm d}(0)$ is asymptotically negligible.

B. DLT Codes

A DLT code is obtained by mixing a doping code with an LT code. As illustrated in Fig. 3(a), the overall code can be regarded as the parallel concatenation of a doping sub-code and an LT sub-code with respect to the state bits. Let p be the doping ratio, i.e., the proportion of the doping bits in the overall output bits (including LT output bits and doping bits). Then

$$p \equiv \frac{\text{number of doping bits}}{\text{number of doping bits and LT output bits}} = \frac{\hat{p}}{1+\epsilon} \tag{12}$$

where ϵ is the overhead. In DLT encoding, each output bit is generated as follows: randomly draw a real number l between 0 and 1; if $l > p$, generate an LT output bit; otherwise, generate a doping bit.

The doping bits and the LT output bits are mixed in transmission, though they are drawn separately in Fig. 3(a). By abuse of notation, let $\boldsymbol{y}_{\rm d}$ be the received doping bit vector. A DLT decoder performs the following: (i) calculate $\boldsymbol{y}'_{\rm d} = \boldsymbol{R}\boldsymbol{y}_{\rm d}$; (ii) recover $\boldsymbol{z}$ based on the LT output bits and $\boldsymbol{y}'_{\rm d}$ using BP decoding. The complexity of either DLT encoding or decoding is $d_l + r$ additions per bit, where d_l is the average state-bit degree of the LT sub-code, and r, that of the doping sub-code.

It is worth mentioning that DLT decoding can be alternatively scheduled in a slightly suboptimal two-stage way: first decode the LT sub-code and then the doping sub-code. The similarity between the doping code and the precode used in Raptor codes is then obvious: they achieve the same objective, i.e., to reduce the residual errors left by LT decoding. This similarity allows us to borrow some results from [12] in proving the asymptotic property of DLT codes, as detailed below.

C. Asymptotic Performance Analysis

In this subsection, we show that the DLT codes are asymptotically capacity-achieving with per-bit complexity $O(\log(1/\epsilon))$. Let

$$P_M(x) = \frac{1}{\mu+1}\left(\mu x + \sum_{i=2}^{M} \frac{x^i}{(i-1)i} + \frac{x^{M+1}}{M}\right) \tag{13}$$

be a truncated version of the Soliton distribution in (6b), where $M = \lceil 4(1+\epsilon)/\epsilon \rceil$, and $\mu = (\epsilon/2) + (\epsilon/2)^2$. We cite the following result from [12] (cf., Lemma 4 and Theorem 5 in [12]) without proof.

Lemma IV: Any set of $(1+\epsilon/2)K$ output bits of the LT code with $P_M(x)$ can reliably recover at least $(1-\epsilon/4(1+\epsilon))K$ information bits via BP decoding at a per-bit cost $O(\log(1/\epsilon))$.

We next show that the doping code has the following property, which, together with Lemma IV, leads to Theorem VI below.

Lemma V: The doping code with $4 \le r \le 10$ can reliably recover a proportion $\hat{p}/2$ of lost state bits, where $\hat{p}$, defined in Proposition III, satisfies $\hat{p} \le p_{max} \equiv 0.5$.

Proof: The doping code can reliably recover a proportion ω of lost state bits provided that [1]

$$\rho_{\rm d}(x) > 1 - \lambda_{\rm d}^{-1}\left(\frac{1-x}{\omega}\right), \text{ for } 1-\omega \le x < 1. \tag{14}$$

Substitute $\lambda_{\rm d}(x) = x^{r-1}$, $\omega = \hat{p}/2$, and (11b) into (14), and let $y = (1-x)/(\hat{p}/2)$. Then (14) becomes

$$\frac{(1-e^{-\hat{p}/r})^2}{(1-(1-y\hat{p}/2)e^{-\hat{p}/r})^2} > 1 - y^{\frac{1}{r-1}}, \text{for } 0 < y \le 1. \tag{15}$$

Considering the fact that $e^{-z} \leq 1 - z + z^2/2$ for $z \geq 0$, we obtain

$$\begin{aligned}
&\frac{(1-e^{-\hat{p}/r})^2}{(1-(1-y\hat{p}/2)e^{-\hat{p}/r})^2} \\
&\overset{(a)}{\geq} \frac{(\hat{p}/r - \hat{p}^2/2r^2)^2}{(1-(1-y\hat{p}/2)(1-\hat{p}/r+\hat{p}^2/2r^2))^2} \\
&= \frac{(1-\hat{p}/2r)^2}{(yr/2+1-\hat{p}/2r-y\hat{p}(1-\hat{p}/2r)/2)^2} \\
&\overset{(b)}{\geq} \frac{(1-\hat{p}/2r)^2}{(yr/2+1-\hat{p}/2r)^2} \overset{(c)}{\geq} \frac{(1-p_{max}/2r)^2}{(yr/2+1-p_{max}/2r)^2}
\end{aligned}$$

where steps (*a*) and (*c*) follow from the fact that $(1-z)^2/(c-z)^2$ is monotonically decreasing in $z \in [0,1]$ for an arbitrary constant $c > 1$, and step (*b*) from the fact that ignoring the term $y\hat{p}(1-\hat{p}/2r)/2$ increases the denominator. It then suffices to show that

$$\frac{(1-p_{max}/2r)^2}{(yr/2+1-p_{max}/2r)^2} > 1 - y^{\frac{1}{r-1}}, \text{ for } 0 < y \leq 1$$

which can be verified easily for r = 4, 5, $\cdots$, 10, respectively. This concludes the proof. □

Theorem VI: A DLT code consisting of an LT sub-code with generation distribution $P_M(x)$ and a doping sub-code with $4 \leq r \leq 10$ and $p = (\epsilon/2)/(1+\epsilon)$ can perform reliable decoding at a per-bit cost of $O(\log(1/\epsilon))$, where the overhead ϵ satisfies $\epsilon < 2p_{max} = 1$.

Proof: Consider the decoding of the above DLT code scheduled as follows: first decode the LT sub-code and then the doping sub-code. From Lemma IV, given $(1+\epsilon/2)K$ LT output bits, the number of un-recovered state bits after LT decoding is no more than $\epsilon K/4(1+\epsilon)$ with high probability. On the other hand, according to Lemma V, the doping code can reliably recover $\hat{p}K/2 = (1+\epsilon)pK/2 = \epsilon K/4$ lost state bits, which is greater than $\epsilon K/4(1+\epsilon)$. Thus, the DLT code can perform reliable decoding at overhead ϵ. The per-bit complexity of the doping code is $O(1)$ and, again from Lemma IV, that of the LT sub-code is $O(\log(1/\epsilon))$. Thus, the overall complexity is $O(\log(1/\epsilon))$. □

In the above analysis (particularly in (14)), we have ignored the impact of $P_{\text{d}}(0)$, as explained below. Following Theorem VI and substituting $\hat{p} = p(1+\epsilon)$ (see (12)) together with $p = (\epsilon/2)/(1+\epsilon)$ into (11a), we have

$$P_{\text{d}}(0) = O\left(\hat{p}^2/r\right) = O\left(\epsilon^2/4r\right).$$

Thus, $P_{\text{d}}(0)$ approaches to zero faster than ϵ, and hence does not affect the asymptotic behavior of the overall code. From Theorem VI, the DLT code with $P_M(x)$ in (13) is asymptotically capacity-achieving at a per-bit cost of $O(\log(1/\epsilon))$. The related analysis is based on a suboptimal two-stage decoding strategy, and so $P_M(x)$ is not necessarily the best choice in practice. We next consider joint decoding of the LT and doping sub-codes, and develop an optimization technique based on the density evolution on the overall coding graph.

D. Degree Distribution Optimization

The LT sub-code and the doping sub-code are concatenated in parallel with respect to the state bits, which distinguishes our proposed scheme from the Raptor code. An advantage of this parallel structure is the density evolution analysis below (that is difficult for the Raptor code due to its serial structure).

Let $\Lambda(u)$ and $\Lambda_{\text{d}}(v)$ be the state-bit degree distribution polynomials of the LT and doping sub-codes, respectively, where u and v are dummy variables. The joint state-bit degree distribution is given by

$$\Lambda(u,v) = \Lambda(u)\Lambda_{\text{d}}(v). \tag{16}$$

Then the joint edge degree distribution from the perspective of the LT sub-code is

$$\lambda_{\text{LT}}(u,v) = \frac{\partial\Lambda(u,v)}{\partial u} \Big/ \frac{\partial\Lambda(u,v)}{\partial u}\Big|_{u=v=1} = \lambda(u)\Lambda_{\text{d}}(v) \tag{17a}$$

and the one from the perspective of the doping sub-code is

$$\lambda_{\text{DP}}(u,v) = \frac{\partial\Lambda(u,v)}{\partial v} \Big/ \frac{\partial\Lambda(u,v)}{\partial v}\Big|_{u=v=1} = \Lambda(u)\lambda_{\text{d}}(v). \tag{17b}$$

Density evolution shows that the fixed point is given by

$$1 - v = \rho_{\text{d}}(1 - \lambda_{\text{DP}}(u,v)), \text{ and} \tag{18}$$
$$1 - u = \rho(1 - \lambda_{\text{LT}}(u,v)). \tag{19}$$

Let $v(u)$ be the fixed point of (18) for a given u, and $u(v)$ be that of (19) for a given v. We introduce the following definition that is an extension of the matching principle in (3).

Definition VII: λ_{LT}, λ_{DP}, ρ and ρ_{d} are matched if $v(u)$ and $u(v)$ represent the same curve in the (u,v) plane that starts from $(u,v) = (0,0)$ and ends at (1, 1).

The overhead of a DLT code can be expressed as

$$\epsilon = \frac{J + J_{\text{d}}}{K} - 1 = \frac{\int_0^1 \rho(u)du}{\int_0^1 \lambda_{\text{LT}}(u,1)du} + \frac{\int_0^1 \rho_{\text{d}}(v)dv}{\int_0^1 \lambda_{\text{DP}}(1,v)dv} - 1 \tag{20}$$

where J is the number of LT output bits, and J_{d} is the number of the doping bits. Then

Theorem VIII: $\epsilon = 0$ provided that λ_{LT}, λ_{DP}, ρ and ρ_{d} are matched.

The proof for Theorem VIII is given in Appendix B. Since λ_{LT}, λ_{DP} and ρ_{d} are determined in (17) and (11b), the matched ρ for DLT codes can be expressed as

$$\hat{\rho}_{\text{DLT}}(u) \equiv 1 - \tilde{\lambda}^{-1}(1-u) \tag{21}$$

where $\tilde{\lambda}(u) \equiv \lambda_{\text{LT}}(u, v(u))$, and the inverse is taken with respect to u. However, $\hat{\rho}_{\text{DLT}}(u)$ given by (21) is not necessarily a valid distribution, as it may not be expressed by a power series with non-negative coefficients. To avoid this difficulty, we search for realizable $\rho(u)$ that best matches $\hat{\rho}_{\text{DLT}}(u)$, as formulated below.

$$\begin{aligned}
&\min_{\{\rho_i\}} \int_0^1 \rho(u)du \\
\text{subject to} \quad &\rho(u) > \hat{\rho}_{\text{DLT}}(u), u \in [0,1), \\
&\rho(1) = 1, \rho_i \geq 0, \text{ for } 1 \leq i \leq M, \\
&\text{and } \rho_i = 0, \text{ for } i > M
\end{aligned} \tag{22}$$

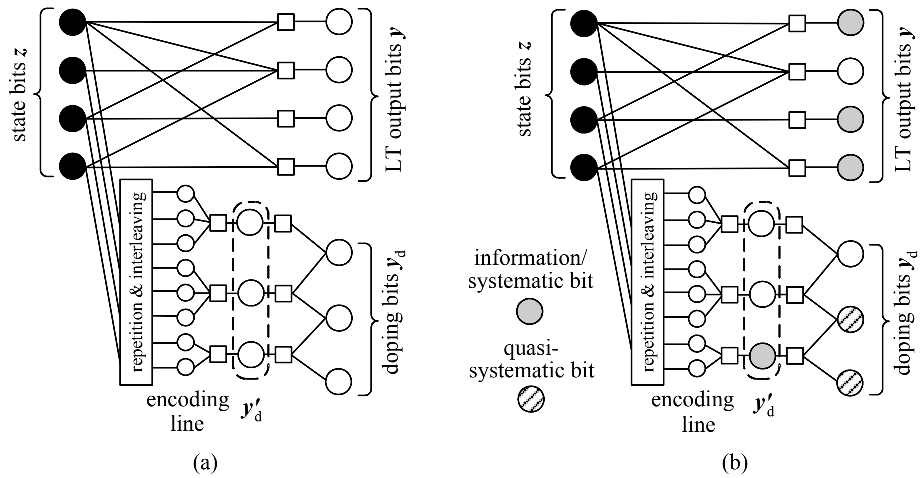

Fig. 3. (a) The structure of a DLT code. (b) The QS-DLT code constructed based on the DLT code in (a).

where M is an integer representing the maximum degree in ρ. Minimizing $\int_0^1 \rho(u)du$ is equivalent to minimizing ϵ in (20). The inequality $\rho(u) > \hat{\rho}_{\text{DLT}}(u)$ is required to hold in [0, 1), implying that all state bits need to be recovered. Letting this inequality holds on discretized values of u, we can solve (22) standardly using linear programming .

E. QS-DLT Codes

Next we discuss a quasi-systematic encoding technique. The systematic LT encoding technique described in Section II.C involves solving z from x based on (8). However, this approach suffers from an error-floor problem. The following is an alternative solution. Recall from Fig. 3(a). Let y and y'_d be the vectors of LT output bits and differentiated doping bits, respectively. Then

$$\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{y}'_{\text{d}} \end{pmatrix} = \boldsymbol{G}''\boldsymbol{z} \tag{23a}$$

where

$$\boldsymbol{G}'' \equiv \begin{pmatrix} \boldsymbol{G}' \\ \boldsymbol{R}\boldsymbol{G}_{\text{d}} \end{pmatrix} \tag{23b}$$

with G' and G_d being the generation matrices of the LT and doping sub-codes, respectively.

From the previous discussions, the DLT code based on G'' (with a small overhead) can perform reliable BP decoding, during which K rows in G'' can be selected to form a BP-decodable G^{ENC}. This implies that the K bits in $\{y, y'_d\}$ corresponding to the rows of G^{ENC} are used to solve z, and so they can be used to form the information vector x. Hence, $x = G^{\text{ENC}}z$. The encoding process involves solving this equation based on BP decoding. It is possible that some information bits in x are contained in y'_d. Direct transmission of these bits is not desirable since they will destroy the structure of the doping code (and so its error-correcting capability). Instead, for each information bit in y'_d, we can transmit the two connected doping bits. These doping bits are called the quasi-systematic bits. A quasi-systematic bit is transmitted only once even if it is connected to two information bits in y'_d. The resulting coding structure is illustrated in Fig. 3(b). As a summary, in encoding, a QS-DLT encoder solves z from x by BP decoding on G^{ENC}, and then generate other LT output bits and doping bits based on z following the DLT encoding principle. The information bits in y and the quasi-systematic bits are first transmitted over the channel and then the other LT output and doping bits. In decoding, a QS-DLT decoder first recovers z from the received bits, and then recovers x from z.

Note that, since y'_d is not transmitted, the overall code is not strictly systematic. However, if y_d is correctly received, generating y'_d from y_d becomes trivial. Thus the code meets the complexity criteria for the quasi-systematic property mentioned in the introduction. Furthermore, the number of quasi-systematic bits is bounded by $(1+\epsilon)pK$ (i.e., the number of the doping bits in G'') that is small compared with K. Thus, the above coding scheme is indeed quasi-systematic.

Similarly to Raptor codes, a QS-DLT decoder can operate in an incremental manner. Specifically, the receiver tries to decode using currently received bits. Upon decoding failure, the receiver collects more received bits. If the extra bits are all LT output bits, the decoder performs the next round decoding directly; if a new doping bit is received, the decoder treats this bit as recovered in the coding graph (cf., Fig. 2(a)) and recovers its neighbors in $\{e_j\}$ and $\{b_j\}$.

Code I in Table I is an example of QS-DLT codes obtained by solving (22). Fig. 4 shows the performance of Code I with $K = 524288$. At least 500 coding blocks are simulated for each point on the performance curves. From Fig. 4, the performance of Code I is close to the capacity, but varies slightly with δ. This jittering effect is due to the fact that the performance is evaluated based on the residual loss rate of information bits. There will be no such jittering if the performance is evaluated in terms of the residual loss rate of state bits. The performance curve of the Raptor code proposed in [12] with $K = 524288$ and precode rate ≈ 0.985 (using the same $P(x)$ as in Code II in Table I) is also included for reference. From Fig. 4, the QS-DLT code slightly outperforms the Raptor code, partly due to the fact that the QS-DLT code is optimized based on the entire coding graph, while such optimization is difficult for Raptor codes. We emphasize

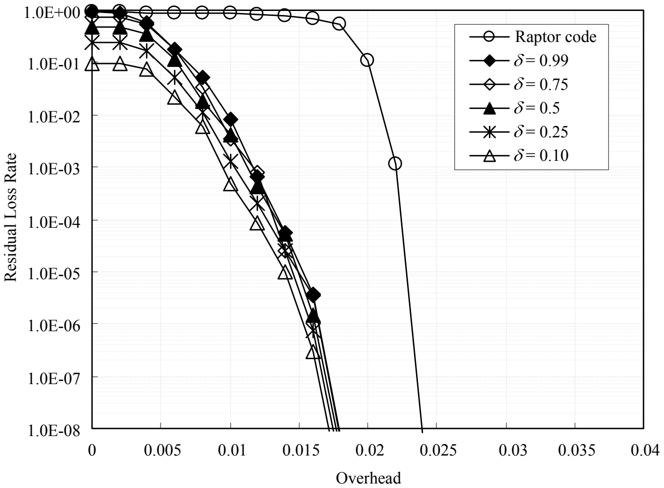

Fig. 4. The performance of QS-DLT codes with information length 524288 for various channel erasure rates. The performance curve of the Raptor code in [12] is also included for comparison.

TABLE I
DEGREE DISTRIBUTIONS OF DLT CODES FOR VARIOUS VALUES OF K, p, r AND d_l

Code I		Code II	
K	∞	K	65536
p	0.01	p	0.015
r	4	r	5
d_l	5.45	d_l	5.9
P_1	0.008001	P_1	0.007969
P_2	0.460778	P_2	0.493570
P_3	0.270578	P_3	0.166220
P_6	0.115605	P_4	0.072646
P_7	0.046850	P_5	0.082558
P_{14}	0.035912	P_8	0.056058
P_{15}	0.022356	P_9	0.037229
P_{30}	0.007295	P_{19}	0.055590
P_{31}	0.018887	P_{65}	0.025023
P_{79}	0.013738	P_{66}	0.003135

here that this does not necessarily mean that QS-DLT codes outperform Raptor codes in general.

IV. Finite-Length Analysis

Codes designed based on density evolution have good asymptotic performance. For moderate code lengths, however, special treatment is necessary. The following idea is borrowed from [11] and [12]: replace u in the right hand side of the inequality $\rho(u) > \hat{\rho}_{\mathrm{DLT}}(u)$ in (22) by $u + c\sqrt{(1-u)/K}$; then solve (22) for suitable c and K to obtain the optimized degree distribution. A heuristic explanation of this approach can be found in [12]. Note that the distribution $P(x)$ of Code II in Table I is directly cited from [12].

Next investigate the error-floor behavior of our proposed codes. To begin, we describe a heuristic approach to determining the doping ratio p. Suppose that DLT decoding is scheduled in a two-stage way: first decode the LT code, and then the doping code. Let l be the number of errors in state bits after LT decoding, $q(l)$ be the probability distribution of l. Also, let $p(l)$ be the transfer function of the residual loss rate against l for the doping code. Clearly, the residual loss rate of the overall code is bounded by $\sum_l q(l)p(l)$. Fig. 5 shows an example of the probability distribution of l for Code II (of length 65536) in Table I. We can see that l is less than 500 with high probability, implying that we need to select a doping ratio p to ensure that the doping code can reliably recover 500 erasures. Then, the error floor of the overall code is mainly determined by the doping code.

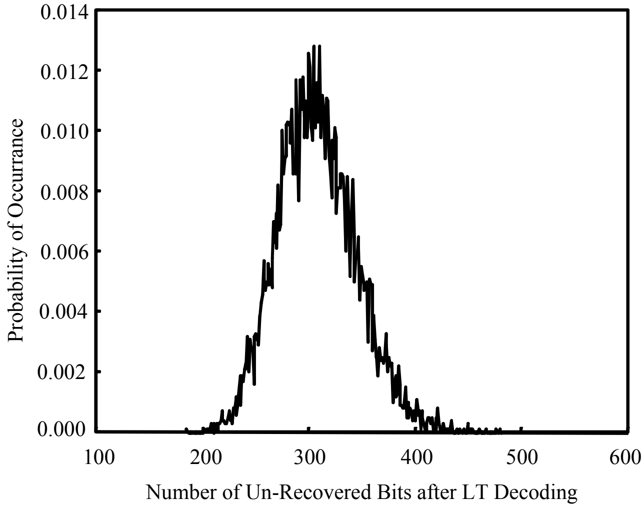

Fig. 5. The empirical probability distribution of un-recovered state bits after LT decoding of Code II in Table I. The overhead = 0.035, and 10000 code blocks are simulated.

It has been shown in [18] and [19] that the error-floor behavior of an erasure code can be estimated by enumerating the stopping sets. This motivates us to analyze the stopping-set distribution of the doping code. Two types of interleavers are considered in generating the encoding line: (i) a random inter-leaver; and (ii) a zigzag interleaver [20] that separates the encoding line into r sections (with each section containing exactly one copy of each state bit) and performs random inter-leaving within each section. The main results are summarized in the theorem below, with the proof provided in Appendix C.

Theorem IX: For the doping code using random interleaving, the residual erasure probability of state bits is bounded by

$$p_{\text{random}}(l) \le \sum_n \left(\frac{l}{K}\right)^n \binom{K-1}{n-1} f(N_{\mathrm{d}}, nr); \qquad (24)$$

for the one using zigzag interleaving, the residual erasure probability of state bits is bounded by

$$p_{\text{zigzag}}(l) \le \sum_n \left(\frac{l}{K}\right)^n \binom{K-1}{n-1} g(N_{\mathrm{d}}, n, r) \qquad (25)$$

where N_{d} is the number of doping bits, and n is the stopping-set size. In the above, f is defined as

$$f(N,m) \equiv \sum \frac{\prod_{i=2}^{N} \binom{N-\sum_{j=2}^{i-1} w_j}{w_i}}{\binom{N+m}{m}} \qquad (26)$$

where $\{w_j\}$ is a set of non-negative integers, and the outer summation in (26) is taken over all possible choices of $\{w_j\}$ satisfying $\sum_{j=2}^{N} j w_j = m$; and g is defined as

$$g(N,n,r) \equiv \frac{1}{r^N} \sum \prod_{i=1}^{r} \binom{N-\sum_{j=1}^{i-1} N(j)}{N(i)} f(N(i),n) \qquad (27)$$

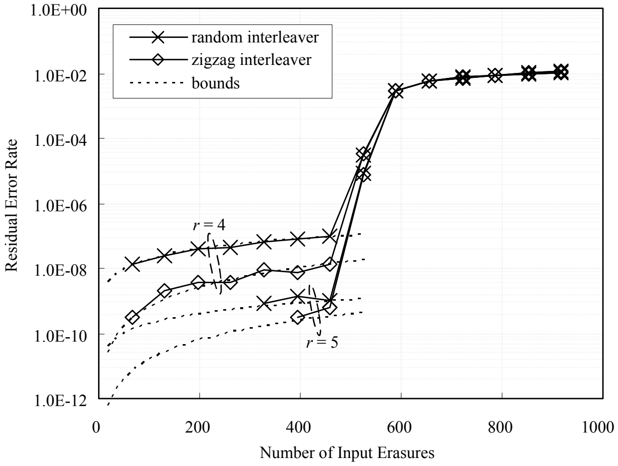

Fig. 6. The error transfer functions of the doping code with random interleaving and zigzag interleaving, respectively. $K = 65536$, and r is set to be 4 and 5, respectively.

where $\{N(i)\}$ are non-negative integers, and the outer summation in (27) is taken over all possible choices of $\{N(i)\}$ satisfying $\sum_{i=1}^{r} N(i) = N$.[4]

It is well-known that the error-floor behavior of LDPC-type codes is dominated by the stopping sets of small sizes. The zigzag interleaver reduces the occurrence probability of size-1 and size-2 stopping sets. Therefore, it leads to a much lower error-floor than the random interleaver, as verified by the numerical results in Fig. 6 (where the bounds calculated based on (24) and (25) are also included for comparison). We can see from Fig. 6 that the error floor reduces with increasing r. This allows us to meet the error-floor requirement (if it exists) by increasing r, at the expense of slight increase in complexity. It is also worth noting that, similarly to Raptor codes, we can apply extended Hamming coding to state bits, so as to remove the stopping set of size less than 4, which can significantly reduce the error-floor of QS-DLT codes. We omit details due to space limitation.

V. Conclusions

In this paper, we propose the use of the doping technique to treat the error-floor problem of LT codes. We have proposed QS-DLT codes that are quasi-systematic and universally capacity-approaching with complexity $O(K\log(1/\epsilon))$. Stopping-set analysis is carried out to analyze the error-floor behavior of QS-DLT codes with finite length. Numerical results show that the proposed codes can perform as well as systematic Raptor codes, but with linear encoding complexity.

Acknowledgment

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

[4]Generally speaking, the right hand side of (25) is computationally involving due to the triple summation (as seen by combining (25-27)). To reduce complexity, we can assume that each section of the encoding line contains the same number of doping bits. Then, $g(N_{\rm d}, n, r)$ reduces to $f(N_{\rm d}/r, n)^r$, which is a good approximation for practical usage.

Appendix A
Proof of Proposition III

Suppose that $\hat{p}K$ doping bits are randomly and independently inserted into the rK possible positions of the encoding line. For a given position, the probability that no doping bits are inserted into this position is $(1 - 1/rK)^{\hat{p}K}$. Then, the number of non-empty positions is $\left(1 - (1 - 1/rK)^{\hat{p}K}\right) rK$. These positions contains $\hat{p}K$ doping bits, and so the number of degree-0 differentiated bits is given by $\hat{p}K - \left(1 - (1 - 1/rK)^{\hat{p}K}\right) rK$. Thus

$$\begin{aligned} P_{\rm d}(0) &= \lim_{K\to\infty} \left(\hat{p} - \left(1 - (1 - 1/rK)^{\hat{p}K}\right) r\right) \\ &= re^{-\hat{p}/r} - r + \hat{p} = O\left(\hat{p}^2/r\right) \end{aligned}$$

where the last equality follows from the power series expansion.

Next prove (11b). We can express $P_{\rm d}(x)$ as

$$P_{\rm d}(x) = P_{\rm d}(0) + (1 - P_{\rm d}(0))\tilde{P}_{\rm d}(x)$$

where $\tilde{P}_{\rm d}(x)$ is the degree distribution of $\boldsymbol{y}'_{\rm d}$ after removing the degree-0 bits. The remaining thing is to find $\tilde{P}_{\rm d}(x)$. The probability that i consecutive positions are empty and the immediate next has at least one inserted doping bit is

$$\left[(1 - 1/rK)^{\hat{p}K}\right]^i \left[1 - (1 - 1/rK)^{\hat{p}K}\right].$$

The above event corresponds to a degree-$(i+1)$ differentiated bit in $\boldsymbol{y}'_{\rm d}$ for $i \geq 0$. Thus, $\tilde{P}_{\rm d}(x)$ is asymptotically given by

$$\begin{aligned} \tilde{P}_{\rm d}(x) &= \lim_{K\to\infty} \sum_{i=0}^{rK} \left[\left(1 - \frac{1}{rK}\right)^{\hat{p}K}\right]^i \left[1 - \left(1 - \frac{1}{rK}\right)^{\hat{p}K}\right] x^{i+1} \\ &= \frac{x(1 - e^{-\hat{p}/r})}{1 - xe^{-\hat{p}/r}}. \end{aligned}$$

Taking the derivative of $P_{\rm d}(x)$, we obtain $\rho_{\rm d}(x)$ in (11b). □

Appendix B
Proof of Thoerem VIII

From (18), we have

$$\int_0^1 \rho_{\rm d}(v)dv = 1 - \int_0^1 \lambda_{\rm DP}^{-1}(u(v), v)dv = \int_0^1 \lambda_{\rm DP}(u(v), v)dv$$

where $\lambda_{\rm DP}^{-1}(u(v), v)$ is the inverse of $\lambda_{\rm DP}(u(v), v)$ with respect to v, and the last equality utilizes the fact that

$$\int_0^1 \lambda_{\rm DP}(u(v), v)dv + \int_0^1 \lambda_{\rm DP}^{-1}(u(v), v)dv = 1.$$

Similarly, it can be shown that

$$\int_0^1 \rho(u)du = \int_0^1 \lambda_{\rm LT}(u, v(u))du.$$

Then

$$\begin{aligned}\epsilon &= \frac{\int_0^1 \rho(u)du}{\int_0^1 \lambda_{\rm LT}(u,1)du} + \frac{\int_0^1 \rho_{\rm d}(v)dv}{\int_0^1 \lambda_{\rm DP}(1,v)dv} - 1 \\ &= \frac{\int_0^1 \lambda_{\rm LT}(u,v(u))du}{\int_0^1 \lambda_{\rm LT}(u,1)du} + \frac{\int_0^1 \lambda_{\rm DP}(u(v),v)dv}{\int_0^1 \lambda_{\rm DP}(1,v)dv} - 1 \\ &\overset{(a)}{=} \frac{\int_0^1 \frac{\partial\Lambda(u,v)}{\partial u}\Big|_{v=v(u)} du \Big/ \frac{\partial\Lambda(u,v)}{\partial u}\Big|_{u=v=1}}{\int_0^1 \frac{\partial\Lambda(u,1)}{\partial u}du \Big/ \frac{\partial\Lambda(u,v)}{\partial u}\Big|_{u=v=1}} \\ &\quad + \frac{\int_0^1 \frac{\partial\Lambda(u,v)}{\partial v}\Big|_{u=u(v)} dv \Big/ \frac{\partial\Lambda(u,v)}{\partial v}\Big|_{u=v=1}}{\int_0^1 \frac{\partial\Lambda(1,v)}{\partial v}dv \Big/ \frac{\partial\Lambda(u,v)}{\partial v}\Big|_{u=v=1}} - 1 \\ &\overset{(b)}{=} \int_0^1 \frac{\partial\Lambda(u,v)}{\partial u}\Big|_{v=v(u)} du + \int_0^1 \frac{\partial\Lambda(u,v)}{\partial v}\Big|_{u=u(v)} dv - 1 \\ &= \int_0^1 d\Lambda - 1 = 0\end{aligned}$$

where step (*a*) follows from (17), and step (*b*) follows from Definition VII, together with the fact that

$$\int_0^1 \frac{\partial\Lambda(u,1)}{\partial u}du = \int_0^1 \frac{\partial\Lambda(1,v)}{\partial v}dv = 1. \qquad \square$$

Appendix C
Proof of Thoerem IX

We start with the doping code using random interleaving. Consider *K* state bits with each repeated by *r* times to produce *rK* encoding bits. Randomly interleave these encoding bits to form an encoding line. Select *n* state bits and mark the corresponding *nr* encoding bits as interested bits, and the others are marked as uninterested bits. $N_{\rm d}$ doping bits are randomly and independently inserted into the encoding line. A typical example of the overall encoding line (together with randomly inserted doping bits) is illustrated in Fig. 7. There are three types of bits in the encoding line, and there are in total

$$\binom{rK+N_{\rm d}}{N_{\rm d}}\binom{rK}{rn}$$

different encoding lines (without distinguishing the bits of a same type). These encoding lines are equiprobable when $rK \gg N_{\rm d}$. The pattern of an encoding line can be obtained by removing the uninterested bits from the encoding line. Clearly, each encoding line has a unique pattern, and each pattern corresponds to

$$\binom{rK+N_{\rm d}}{rn+N_{\rm d}}$$

different encoding lines. Thus, the total number of different patterns is given by

$$\binom{rK+N_{\rm d}}{N_{\rm d}}\binom{rK}{rn}\Big/\binom{rK+N_{\rm d}}{rn+N_{\rm d}} = \binom{N_{\rm d}+nr}{nr}.$$

As illustrated in the lower part of Fig. 7, the doping bits divide the pattern into segments of consecutive interested bits. If every segment contains no less than 2 interested bits, we say that the *n* selected state bits form a stopping set of size

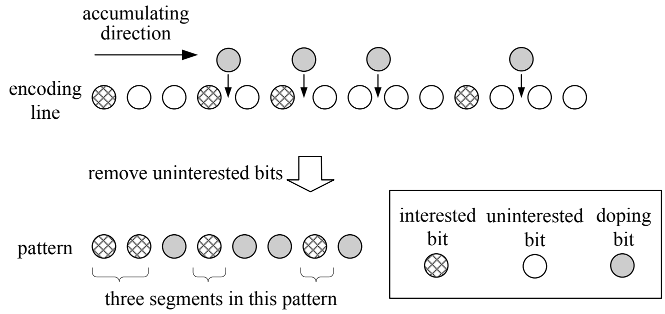

Fig. 7. The pattern of an encoding line.

n. We can calculate the total number of patterns that form a stopping set of size *n* as follows. Let the segment-length distribution $\boldsymbol{w} \equiv \{w_2, w_3, \cdots, w_{nr}\}$, where w_i denotes the number of length-*i* segments in the pattern. Then the number of stopping sets characterized by $\boldsymbol{w}$ is given by

$$\binom{N_{\rm d}}{w_2}\binom{N_{\rm d}-w_2}{w_3}\cdots = \prod_{i=2}^{nr}\binom{N_{\rm d}-\sum_{j=2}^{i-1}w_j}{w_i}.$$

Thus, the probability that *n* state bits form a stopping set is $f(N_{\rm d}, nr)$, where *f* is defined in (26).

We next determine the residual erasure probability of the state bits after DLT decoding. Suppose that LT decoding is applied, leaving *l* state bits un-recovered. The probability that a state bit has not been recovered is l/K. Suppose that this bit belongs to a size-*n* stopping set of the doping code. The total number of such stopping sets is equal to the number of $(n-1)$-combinations from a set of size $K-1$. This bit remains un-recovered if the *n* state bits forming any of the above stopping sets are un-recovered. The probability of this event is upper bounded by

$$\left(\frac{l}{K}\right)^n\binom{K-1}{n-1}f(N_{\rm d}, nr).$$

Considering the stopping sets of all possible sizes, we immediately obtain (24).

Now consider the doping code using zigzag interleaving. The only difference is that the encoding line is now divided into *r* sections and random interleaving is applied within each section. Let $N(i)$ be the number of doping bits inserted into the *i*th section. Then, given $\{N(i)\}$, $\prod_{i=1}^{r} f(N(i), n)$ is the probability that *n* randomly chosen state bits form a stopping set. Considering the various combinations of $N(i)$, we can express this probability by $g(N_{\rm d}, n, r)$, where g is defined in (27). Further considering the stopping sets of various sizes, we obtain (25). $\square$

References

[1] M. G. Luby, M. Mitzenmacher, A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 569-584, Feb. 2001.
[2] A. Shokrollahi, *Capacity-Achieving Sequences*. Minneapolis, MN: Inst. Mathematics and its Applications (IMA), 2000, vol. 123, IMA Volumes in Mathematics and its Applications, pp. 153-166.
[3] P. Oswald and A. Shokrollahi, "Capacity-achieving sequences for the erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, no. 12, pp. 3017-3028, Dec. 2002.

[4] H. D. Pfster and I. Sason, “Accumulate-repeat-accumulate codes: capacity-achieving ensembles of systematic codes for the erasure channel with bounded complexity,” *IEEE Trans. Inform. Theory*, vol. 53, no. 6, pp. 2088-2115, June 2007.
[5] C. H. Hsu and A. Anastasopoulos, ”Capacity-achieving codes with bounded graphical complexity on noisy channels,” in *Proc. 43rd Allerton Conf. Communication, Control and Computing*, Monticello, IL, USA, Sep. 2005.
[6] H. Jin, A. Khandekar, and R. J. McEliece, “Irregular repeat-accumulate codes,” in *Proc. 2nd Int. Symp. Turbo codes & related topics*, pp. 1-8, France, Sep. 2000.
[7] H. D. Pfister, I. Sason and R. Urbanke, “Capacity-achieving ensembles for the binary erasure channel with bounded complexity,” *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2352-2379, July 2005.
[8] I. Sason and R. Urbanke, “Complexity versus performance of capacity-achieving irregular repeat-accumulate codes on the binary erasure channel,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 1247-1256, June 2004.
[9] T. Richardson, A. Shokrollahi and R. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619-637, Feb. 2001.
[10] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, “A digital fountain approach to reliable distribution of bulk data,” in *Proc. ACMSIG-COMM’98*, Vancouver, BC, Canada, Jan. 1998, pp. 56-67.
[11] M. Luby, “LT-codes,” in *Proc. 43rd Annu. IEEE Symp. Foundations of Computer Science (FOCS)*, Vancouver, BC, Canada, Nov. 2002, pp. 271-280.
[12] A. Shokrollahi, “Raptor codes”, *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2551-2567, June 2006.
[13] X. Yuan and Li Ping, “Doped accumulate LT codes,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT) 2007*, Nice, France, June 2007, pp. 2001-2005.
[14] O. Etesami and A. Shokrollahi, “Raptor codes on binary memoryless symmetric channels,” *IEEE Trans. Inform. Theory*, vol. 52, no. 5, pp. 2033-2051, May 2006.
[15] D. Divsalar, H. Jin, and R. McEliece, “Coding theorems for ’turbo-like’ codes,” in *Proc. 1998 Allerton Conf. Communication, Control and Computing*, Monticello, IL, pp. 201-210, Oct. 1998.
[16] Li Ping and R. Sun, “Simple erasure correcting codes with capacity approaching performance,” *GLOBECOM ’02*, pp.1046-1050, 2002.
[17] X. Yuan and Li Ping, “On systematic LT codes”, *IEEE Commun. Letters*, vol. 12, no. 9, pp. 681-683, Sept. 2008.
[18] C. Di, D. Proietti, BI. E. Telatar, T. Richardson, and R. Urbanke, “Finite length analysis of low-density parity-check codes on the binary erasure channel,” *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1570-1579, Jun. 2002.
[19] R. Karp, M. Luby, and A. Shokrollahi, “Finite length analysis of LT codes,” in *Proc. IEEE Int. Symp. Inf. Theory*, Chicago, IL, Jun./Jul. 2004, p. 39.
[20] Li Ping, X. Huang and N. Phamdo, “Zigzag codes and concatenated zigzag codes,” *IEEE Trans. Inform. Theory special issue on codes on graphs*, vol., IT-47, no.2, pp., 800-807, Feb. 2001.

Xiaojun Yuan (S’04-M’08) received the B.S. degree in electronic and information systems from Shanghai Jiaotong University, the M.S. degree in circuit and systems from Fudan University, and the Ph.D degree at City University of Hong Kong. He is now with the Department of Electronic Engineering, University of Hawaii at Manoa, Hawaii, USA.

His research interests include signal processing, coding design, optimized communication systems design, and networks.

Li Ping (S’87-M’91-SM’06) received his Ph.D degree at Glasgow University in 1990. He lectured at Department of Electronic Engineering, Melbourne University, from 1990 to 1992, and worked as a member of research staff at Telecom Australia Research Laboratories from 1993 to 1995. He has been with the Department of Electronic Engineering, City University of Hong Kong, since January 1996, where he is now a chair professor.

His research interests are communications systems and coding theory. Dr. Li Ping was awarded a British Telecom-Royal Society Fellowship in 1986, the IEE J J Thomson premium in 1993 and a Croucher Senior Research Fellowship in 2005.