Quasi-Systematic Doped LT Codes

Xiaojun Yuan and Li Ping
Department of Electronic Engineering, City University of Hong Kong, Hong Kong
Email: xjyuan@cityu.edu.hk, eeliping@cityu.edu.hk

***Abstract*—We propose a family of binary erasure codes, namely, quasi-systematic doped LT (QS-DLT) codes that are almost systematic, universal, and asymptotically capacity-achieving with encoding and decoding complexity $O(K\log(1/\varepsilon))$, where K is the information length, and ε is the overhead. Finite-length analysis is carried out to study the error-floor behavior of our proposed codes. Numerical results verify that our proposed codes provide a low-complexity alternative to systematic Raptor codes with comparable performance.**

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

I. Introduction

Properly designed low-density parity-check (LDPC) codes can achieve performance arbitrarily close to the capacity of binary erasure channels (BECs) with reasonable complexity [1]-[4]. In a conventional approach, a code is designed for a specific channel erasure rate (denoted by δ) that is assumed known by the encoder. A coding scheme is said to be rateless if it can generate coded bits potentially limitlessly. A rateless scheme is said to be universal if it is capacity-achieving without requiring the knowledge of δ in encoding. Universal codes are useful, e.g., in digital fountain applications [5].

Luby Transform (LT) codes [6] are a well-known family of universal codes. 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 [1] [7], where K is the information length. In practice, linear complexity $O(K)$ is mostly preferable. An LT code can be designed with linear complexity by reducing the average degree of information bits, but it then suffers from a high-error-floor problem. This problem can be solved using the Raptor code approach [7] by serially concatenating a conventional LDPC code with an LT code. Raptor codes can reliably recover all of the information bits with complexity $O(K\log(1/\varepsilon))$ [7], where ε is the overhead, i.e., the normalized difference between the numbers of the received and 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 can greatly reduce the cost. LT and Raptor codes are, in their straightforward forms, non-systematic. A technique to design systematic Raptor codes was proposed in [7] with encoding complexity roughly αK^2, where α is a relatively small positive number independent of K.

This paper is concerned with a family of quasi-systematic doped LT (QS-DLT) codes that are almost systematic and universally capacity-approaching with encoding and decoding complexity of $O(K\log(1/\varepsilon))$. The key idea is the parallel 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, one major advantage of this parallel structure is that it allows the use of the belief propagation (BP) algorithm in quasi-systematic encoding, which ensures linear encoding complexity. [1] Stopping-set analysis [10] is presented to study the error-floor behavior of QS-DLT codes with finite length. Numerical results demonstrate that QS-DLT codes provide a promising low-complexity alternative to systematic Raptor codes.

II. Preliminaries

We begin with an outline of LT codes. Denote by G the graphic representation of an LT code. An example of G is shown in Fig. 1(a), where each black node represents a state bit z_i, each white node represents an output bit y_j, and each square node represents a parity check. The encoding rule is

$$y_j = \sum_{\{z_i\} \text{ connected to } y_j \text{ via a parity check}} z_i \tag{1}$$

where the summation is binary. In a conventional LT code, state bits represent information, but in this paper state bits will be formed differently, as detailed later.

The degree of a black node is defined as the number of connected white nodes (via parity checks); and that of a white node as the number of connected state bits. The edges are randomly connected to the black nodes, and so the left degrees (the degrees of black nodes) follow a Poisson distribution [7]. The right-degree distribution (i.e., the degree distribution of white nodes) is specified by an optimized polynomial $P(x) = \Sigma_i P_i x^i$, where P_i is the proportion of degree-i white nodes.

The code defined on G can be equivalently represented in an algebraic form. Let $\boldsymbol{z} \equiv [z_0, z_1, \ldots, z_{N-1}]^{\mathrm{T}}$ and $\boldsymbol{y} \equiv [y_0, y_1, \ldots, y_{M-1}]^{\mathrm{T}}$ be the state-bit and output-bit vectors, respectively, where N (or M) is the number of state (or output) bits. Each y_i can be represented as the inner product of two vectors, i.e., $y_i = \boldsymbol{g}_i^{\mathrm{T}}\boldsymbol{z}$, where $\boldsymbol{g}_i$ is the binary vector with its "1"s corresponding to the state bits connected to y_i. Then

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{z} \tag{2}$$

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

The output bits are transmitted over the erasure channel. The receiver is to recover the state bits from a partially recon-

[1] Other advantages include that this parallelism facilitates the density evolution analysis [1] and the optimization based on the overall code structure. We will not discuss details in this paper due to space limitation.

structed coding graph with some output bits lost. The belief propagation (BP) algorithm can be employed for this purpose.

BP Algorithm: Identify a degree-1 output bit y_i. Recover the unique state bit z_j connected to y_i via a parity check. Delete z_j, y_i, and the edges connected to z_j and y_i. Repeat the above steps until all state bits are recovered (decoding success) or until no degree-1 output bits can be found (decoding failure).

We say that a graph G is *BP-decodable* if the above algorithm ends at "decoding success". The complexity of the BP algorithm is proportional to the average degree of the state bits (denoted by d). This complexity is generally much lower than other alternatives, such as the Gaussian elimination. The overhead ε of the LT code can be expressed as $\varepsilon = M/N-1$. LT codes with a small ε and d suffer from an error-floor problem, i.e., some portion of bits cannot be recovered by the BP algorithm. For example, for the LT code with [7]

$$P(x) = 0.008x + 0.494x^2 + 0.166x^3 + 0.073x^4 + 0.083x^5 + 0.056x^8 + 0.037x^9 + 0.056x^{19} + 0.025x^{65} + 0.003x^{66}, \quad (3)$$

$N = 65536$ and $\varepsilon = 0.03$, the number of residue erasures after BP decoding ranges from 200 to 500 with high probability.

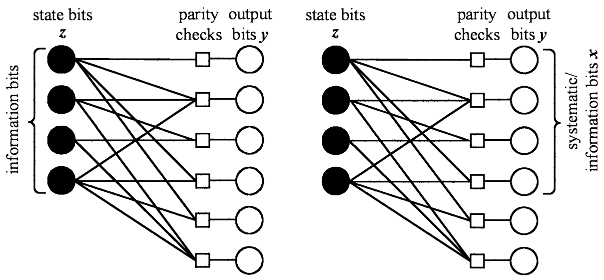

Fig. 1. (a) The coding structure of an LT code. (b) The systematic LT code formed by the code in (a). A portion of output bits represent the information.

Table 1. The average number of trials to successfully construct G^{ENC} for the LT codes with $P(x)$ in (3) and $N = 65536$.

ε	0.50	0.56	0.64	0.75	1.00
Average number of trials	$\approx \infty$	800	74	10	1.7

III. Systematic LT Codes

A. The Basic Idea

A systematic approach to LT encoding is illustrated in Fig. 1(b). The code therein has the same structure as the LT code in Fig. 1(a). The only difference is that in Fig. 1(b) the information bits form part of the output bits. Information bits here are also called *systematic* bits. Denote by G^{ENC} the sub-graph induced by the information bits and still by G the overall coding graph. The encoding process of systematic LT codes consists of two steps: determine the state bits from the information bits; then determine the other output bits from the state bits. The first step can be accomplished by LT decoding on G^{ENC}. To reduce complexity, we need to ensure that G^{ENC} is BP-decodable. As shown later, this is one of the key difficulties in realizing systematic LT codes with linear complexity.

B. Optimal Number of State Bits

Let K be the number of information bits. We show that the best choice of N (the number of state bits) is $N = K$. To see this, first suppose $K > N$. From (1), each output bit can be viewed as a linear equation. The first step of the systematic LT encoding process corresponds to solving N unknown variables from K linear equations. Since the values of the information bits are arbitrary, the solution may not exit if $K > N$.

On the other hand, suppose $K < N$. The codes in Fig. 1 have exactly the same decoding performance in terms of frame error rate (FER) since they share the same structure. Let M be the number of received output bits required to achieve a certain FER performance for the code in Fig. 1(a). The overhead of this code is $M/N-1$. To achieve the same performance for the code in Fig. 1(b), the required overhead is $M/K-1$ ($> M/N-1$). Thus, $N > K$ leads to inefficiency.

Therefore, the optimal choice is $N = K$, i.e., the numbers of black and white nodes in G^{ENC} should be the same. We henceforth always assume $N = K$, except explicitly specified.

C. Construction of G^{ENC}

Now consider the construction of G^{ENC}. We need to ensure that G^{ENC} is BP-decodable, and that the degree distributions on G^{ENC} resemble those of a typical LT code, i.e., Poisson on the left and $P(x)$ on the right. A brute-force approach to constructing G^{ENC} is to repeatedly generate G^{ENC} using LT encoding and check whether it is BP-decodable. The checking process is equivalent to LT decoding with zero overhead, which fails with high probability. This implies a time-consuming design procedure. On the other hand, even if it succeeds, the realized distribution may not match $P(x)$ since G^{ENC} is selected from an event with a very small probability.

The author in [7] suggested to construct G^{ENC} by identifying K information bits from a larger LT coding graph G' with $K(1+\varepsilon)$ output bits. The BP algorithm can be applied to G' for this purpose. If the BP algorithm ends successfully, G^{ENC} can be formed by removing the $K\varepsilon$ redundant rows in G' (or equivalently speaking, by the coding graph induced by the set of K deleted $\{y_j\}$ in the BP algorithm in Section II). Clearly, G^{ENC} so constructed is BP-decodable. However, the error-floor problem of LT codes causes difficulties. We explain this for small and large ε values separately below.

When ε is small, a large number of trials is required (cf., Table 1). The characteristics of a small probability event may significantly deviate from the average behavior. This means, the more trials are required, the more deviation of the realized right degree distribution from $P(x)$. On the other hand, suppose that ε is large enough to ensure a reasonable success rate of constructing G^{ENC} using BP decoding. Clearly, the degree-1 output bits in G' have a high probability of being selected in constructing G^{ENC}. Let P_1 be the coefficient of x in $P(x)$. The total number of degree-1 output bits is $(1+\varepsilon)P_1K$ in G' and so the portion of degree-1 output bits in G^{ENC} is close to $(1+\varepsilon)P_1$. This can be quite different from the desired value of P_1 when ε is large. Similarly, considerable changes occur for other coefficients when ε is large. Thus, the realized right degree distribution may still considerably deviate from $P(x)$.

The precoding technique used in Raptor codes can be employed to overcome the error-floor problem. Similarly to systematic LT codes, the information positions of a systematic Raptor code need to be identified in the LT output bits. However, the serial concatenation nature of the precode and LT sub-code makes the overall code not directly BP-decodable. As a result, the complexity involved in systematic raptor encoding is significantly increased to $O(K^2)$ [7].

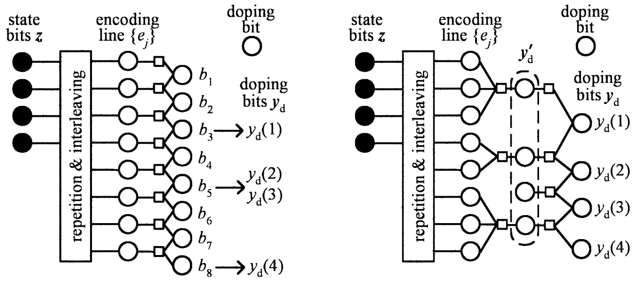

Fig. 2. (a) The structure of the doping code with $K = 4$ and $r = 2$. Note that b_5 is selected as a doping bit twice as $y_d(2)$ and $y_d(3)$. (b) An equivalent representation of the code in (a).

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

We propose the use of the doping technique [9] to solve the error-floor problem of systematic LT codes, while maintaining linear encoding and decoding complexity.

A. The Doping Technique

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 by r times to generate rK encoding bits; (ii) randomly interleave these encoding bits to form an encoding line. A doping bit is generated as follows: (i) randomly select an integer i in $\{1, 2, \ldots, rK\}$, and (ii) set the doping bit as the binary sum of the first i bits in the encoding line (denoted by b_i, where $\{b_i\}$ can be calculated in batch by accumulating addition along the encoding line). In the above encoding, the repeated selection of a same b_i is allowed, and so the doping bits can be generated independently and potentially limitlessly.

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

$$\boldsymbol{y}_\mathrm{d} = \boldsymbol{G}_\mathrm{d}\boldsymbol{z} \tag{4}$$

where $\boldsymbol{z}$ is the state-bit vector, and $\boldsymbol{G}_\mathrm{d}$ is the generation matrix of the doping code. $\boldsymbol{G}_\mathrm{d}$ is not a sparse matrix due to the accumulating nature of the doping bits, but it can be converted to a sparse matrix by calculating the differential of $\boldsymbol{y}_\mathrm{d}$ as

$$\boldsymbol{y}'_\mathrm{d} = \boldsymbol{R}\boldsymbol{y}_\mathrm{d} = \boldsymbol{R}\boldsymbol{G}_\mathrm{d}\boldsymbol{z} \tag{5a}$$

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

$$\boldsymbol{R} \equiv \begin{bmatrix} 1 & & & 0 \\ 1 & 1 & & \\ & \ddots & \ddots & \\ 0 & & 1 & 1 \end{bmatrix}. \tag{5b}$$

The bits in $\boldsymbol{y}'_\mathrm{d}$ are illustrated in Fig. 2(b). Clearly, the number of "1"s in $\boldsymbol{R}\boldsymbol{G}_\mathrm{d}$ is roughly the length of the encoding line. Thus, BP decoding on $\boldsymbol{R}\boldsymbol{G}_\mathrm{d}$ requires r additions per bit.

The above doping code falls into the family of repeat accumulate (RA) codes [8]. It can be treated as a randomly punctured RA code, with some of $\{b_j\}$ (see Fig. 2(a)) punctured. However, we emphasize that the doping code differs from the conventional RA codes in that the doping bits can be generated independently and limitlessly, so as to satisfy the requirement of ratelessness. We note that this independency implies a non-zero probability of the repeated reception of a same doping bit (that can provide no extra information in decoding). However, it can be shown that the related performance loss is marginal. We omit details here due to space limitation.

B. DLT Codes

The parallel concatenation of an LT code with the above doping code results in the so-called doped LT (DLT) codes, as illustrated in Fig. 3(a). Let p be the doping ratio, i.e., the proportion of the doping bits in the overall output bits (including both LT output bits and doping bits). A DLT encoder generates each output bit as follows: randomly draw a number l between 0 and 1; if $l > p$, generate an LT output bit; otherwise, generate a doping bit. The doping and LT output bits are mixed in transmission, though they are drawn separately in Fig. 3(a). Let $\boldsymbol{y}_\mathrm{d}$ be the received doping-bit vector. A DLT decoder performs the following steps: (i) calculate $\boldsymbol{y}'_\mathrm{d} = \boldsymbol{R}\boldsymbol{y}_\mathrm{d}$; (ii) recover $\boldsymbol{z}$ from the LT output bits and $\boldsymbol{y}'_\mathrm{d}$ using BP decoding. The complexity of either DLT encoding or decoding is $d + r$ additions per bit.

Asymptotic analysis shows that, similarly to Raptor codes, DLT codes can perform reliable decoding at a cost of $O(K\log(1/\varepsilon))$, where ε is the overhead. We omit details here due to space limitation.

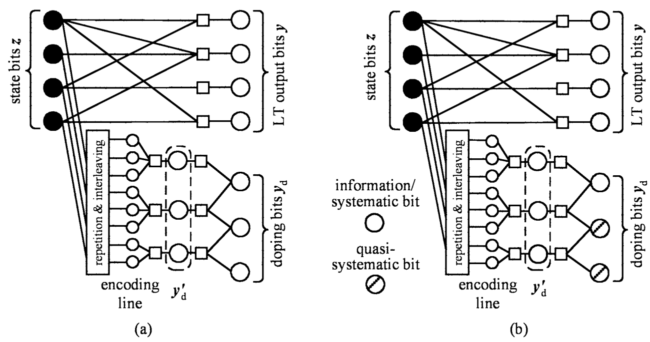

Fig. 3. (a) The structure of a DLT code. (b) A QS-DLT code constructed based on the DLT code in (a).

C. QS-DLT Codes

We next describe the construction of QS-DLT codes based on DLT codes, as illustrated in Fig. 3(b). With abuse of notation, let $\boldsymbol{y}$ be the LT output bit vector (cf., (2)), and $\boldsymbol{y}'_\mathrm{d}$ be the differentiated doping bit vector (cf., (5)). Then, we can represent the DLT code by

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}'_\mathrm{d} \end{bmatrix} = \boldsymbol{G}''\boldsymbol{z} \quad \text{with} \quad \boldsymbol{G}'' \equiv \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{R}\boldsymbol{G}_\mathrm{d} \end{bmatrix} \tag{6}$$

where $\boldsymbol{G}$ and $\boldsymbol{G}_\mathrm{d}$ are the generation matrices of the LT sub-code and the doping sub-code, respectively. As mentioned before, DLT codes can perform reliable decoding at a small overhead.

This is equivalent to say that $\boldsymbol{G}''$ with a small overhead is BP-decodable with high probability.

A square and BP-decodable $\boldsymbol{G}^{\mathrm{ENC}}$ can be constructed by removing redundant rows of $\boldsymbol{G}''$ using BP decoding. The bits in $\{\boldsymbol{y}, \boldsymbol{y}'_{\mathrm{d}}\}$ corresponding to the rows of $\boldsymbol{G}^{\mathrm{ENC}}$ are selected as the information bits. It is possible that some rows of $\boldsymbol{RG}_{\mathrm{d}}$ are selected in forming $\boldsymbol{G}^{\mathrm{ENC}}$, or equivalently, some of the information bits are selected from $\boldsymbol{y}'_{\mathrm{d}}$. Direct transmission of these bits is not desirable since they will destroy the structure of the doping code (and so its error-correcting capability). To avoid this, we define the *quasi-systematic* bits to be the doping bits connected to the information bits in $\boldsymbol{y}'_{\mathrm{d}}$, as illustrated in Fig. 3(b). Quasi-systematic bits, together with the information bits in $\boldsymbol{y}$, are first transmitted over the channel.

We note that a small proportion of information bits (those identified in $\boldsymbol{y}'_{\mathrm{d}}$) are not transmitted. They can be decoded from the quasi-systematic bits (if received) using differentiation. The total number of the systematic and quasi-systematic bits is larger than that of the information bits. Therefore, strictly speaking, QS-DLT codes are not systematic. However, the portion of information bits identified in $\boldsymbol{y}'_{\mathrm{d}}$ is bounded by $p(1+\varepsilon)$, and the decoding cost related to these bits using quasi-systematic bits is trivial. Thus, we say that QS-DLT codes are *quasi-systematic*.

The overall encoding and decoding process is outlined below. $\boldsymbol{G}^{\mathrm{ENC}}$ can be constructed as: (i) generate $\boldsymbol{G}''$ of $K(1+\varepsilon)$ rows, and apply BP decoding to $\boldsymbol{G}''$; (ii) if $\boldsymbol{G}''$ is BP-decodable, identify the information bits $\boldsymbol{x}$ in $\{\boldsymbol{y}, \boldsymbol{y}'_{\mathrm{d}}\}$ and the systematic bits in $\{\boldsymbol{y}, \boldsymbol{y}_{\mathrm{d}}\}$; (iii) otherwise, return to step (i). In encoding, the QS-DLT encoder solves $\boldsymbol{z}$ from $\boldsymbol{x}$ by BP decoding, and then generate other LT output bits and doping bits based on $\boldsymbol{z}$. The LT output bits and the doping bits are transmitted over the channel. In decoding, the QS-DLT decoder first recovers $\boldsymbol{z}$ from the received bits, and then recovers $\boldsymbol{x}$ from $\boldsymbol{z}$.

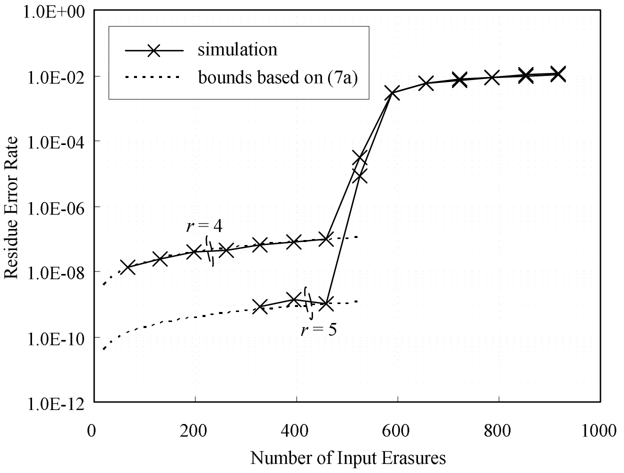

Fig. 4. The error transfer functions of the doping code with $K = 65536$, and $r =$ 4 and 5, respectively.

V. Finite Length Analysis

We next analyze the error-floor behavior of QS-DLT codes. Let l be the number of residue erasures in state bits after LT decoding, $q(l)$ be the probability density distribution of l, and $p(l)$ be the transfer function of the residue erasure rate against l for decoding the doping code. The residue erasure rate of the overall code is then bounded by $\Sigma_l q(l)p(l)$. It can be shown that, with a proper selection of the doping ratio, the error floor of a QS-DLT code is mainly determined by the doping code.

The error-floor behavior of a code can be estimated by enumerating stopping sets [10] [11]. This motivates us to determine $p(l)$ by analyzing the stopping-set distribution of the doping code. The main result is summarized below.

Proposition 1: For the doping code, the residue erasure probability of the state bits is bounded by

$$p(l) \le \sum_n \left(\frac{l}{K}\right)^n \binom{K-1}{n-1} f(N_{\mathrm{d}}, nr)\,; \qquad (7a)$$

where N_{d} is the number of doping bits, and

$$f(N,m) \equiv \sum_{\sum_{k=2}^{N} k w_k = m} \prod_{i=2}^{N} \binom{N-\sum_{j=2}^{i-1} w_j}{w_i} \Big/ \binom{N+m}{m} \qquad (7b)$$

with the summation taken over all possible non-negative integers $\{w_k\}_{k=2}^{N}$ satisfying $\sum_{k=2}^{N} k w_k = m$.

The proof of Proposition 1 can be found in the appendix. Fig. 4 shows the transfer function of the doping code with various values of r. The bounds calculated using (7a) are also included for comparison. From Fig. 4, the error floor reduces with the increase of r. This implies that one can meet the error-floor requirement (if exists) by increasing r, at the expense of a slight increase in complexity.

Similarly to Raptor codes, we can also apply extended Hamming coding to QS-DLT codes, so as to remove the stopping set of size less than 4. Specifically, we first determined the value of the K state bits from the information bits by BP decoding on $\boldsymbol{G}^{\mathrm{ENC}}$; apply extended Hamming coding to the state bits, which increases the number of state bits to $K+\lceil \log_2 K \rceil$, with the extra $\lceil \log_2 K \rceil$ state bits being the extended hamming coded bits; then generate other LT output and doping bits based on the expanded set of state bits. The above approach may lead to certain performance loss by noting the fact that the number of state bits now exceeds K (see Section III.B). However, such loss is negligible since $\lceil \log_2 K \rceil \ll K$.

The remaining problem is that, for each extra state bit produced by the extended hamming encoding, we need to insert r extra encoding bits into the encoding line. Random insertion is not feasible. To see this, we divide the encoding line into segments with the encoding bits in each segment connected to one bit in $\boldsymbol{y}'_{\mathrm{d}}$ (cf., Fig. 3(b)). An extra encoding bit cannot be inserted into a segment connected to an information bit in $\boldsymbol{y}'_{\mathrm{d}}$ since the values of both the information bit and the existing encoding bits in this segment are already determined at this stage, and so such an insertion may invalidate the parity check. However, this can be avoided since there are sufficient non-information bits in $\boldsymbol{y}'_{\mathrm{d}}$. For example, for the QS-DLT code based on (3) with $K = 65536$, $r = 5$, $p = 0.015$, and $\varepsilon = 0.035$, the doping code has about 1000 doping bits and needs to recover at most 500 residue erasures left by LT decoding, implying that the number of information bits identified in $\boldsymbol{y}'_{\mathrm{d}}$ is at most 500. Thus, we can insert $r\lceil \log_2 K \rceil = 80$ extra encoding bits randomly into the other (no less than 500) segments. This approach can significantly reduce the error-floor. Numerical results show that, for the QS-DLT code considered, the residue erasure rate of state bits is bounded by 6.3×10^{-10}; after extended Hamming coding, this rate can be reduced to 2.4×10^{-14}.

V. Conclusions

In this paper, we have proposed QS-DLT codes that are a family of quasi-systematic universal erasure codes with capacity-achieving property at a cost of $O(K\log(1/\varepsilon))$. Stopping-set analysis is carried out to analyze the error-floor behavior of QS-DLT codes. Numerical results demonstrate that our proposed codes can perform as well as systematic Raptor codes, but with linear encoding and decoding complexity.

Appendix Proof of Proposition 1

Consider K state bits with each repeated by r times to form the encoding bits. Randomly interleave the rK encoding bits to form an encoding line. Select n state bits and mark their nr repeats in the encoding line as *interested bits*, and the others are marked as *uninterested bits*. N_d doping bits are randomly and independently inserted into the encoding line. An example of the overall encoding line is illustrated in the upper part of Fig. 5. There are three types of bits in the encoding line and

$$\binom{rK+N_d}{N_d}\binom{rK}{rn} \tag{A1}$$

different encoding lines (without distinguishing the bits of a same type). These encoding lines are equally probable by assuming $rK \gg N_d$. 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

$$\binom{rK+N_d}{rn+N_d}$$

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

$$\binom{rK+N_d}{N_d}\binom{rK}{rn}\Big/\binom{rK+N_d}{rn+N_d}=\binom{N_d+nr}{nr}. \tag{A2}$$

As shown in Fig. 5, 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 n.

We can calculate the total number of patterns that form a stopping set of size n as follows. Let the segment-length distribution $\boldsymbol{w} \equiv \{w_2, w_3, \ldots, w_{nr}\}$, 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

$$\binom{N_d}{w_2}\binom{N_d-w_2}{w_3}\cdots\binom{N_d-\sum_{j=2}^{nr-1}w_j}{w_{nr}}=\prod_{i=2}^{nr}\binom{N_d-\sum_{j=2}^{i-1}w_j}{w_i}. \tag{A3}$$

Thus, the probability that n state bits form a stopping set can be expressed as $f(N_d, nr)$, where f is defined in (7b).

We next determine the error 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 the number of combinations for selecting n–1 elements from a set of size K–1. This bit produces an error if the n state bits forming any of the above stopping sets are un-recovered. The probability of this event is bounded by

$$\left(\frac{l}{K}\right)^n\binom{K-1}{n-1}f(N_d, nr). \tag{A4}$$

Considering the stopping sets of various sizes, we obtain (7a).

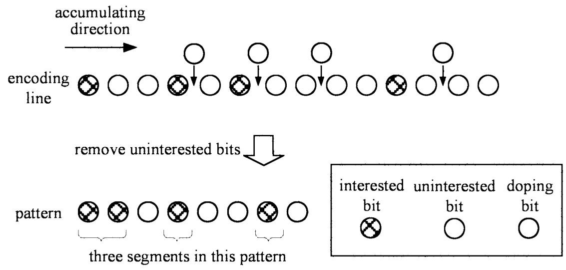

Fig. 5. The pattern of an encoding line.

Acknowledgement

This work was fully supported by CityU Applied R&D Centre under Project Number 9681001.

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] 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.

[3] 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.

[4] 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, July 2005.

[5] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACMSIGCOMM'98*, Vancouver, BC, Canada, Jan. 1998, pp. 56–67.

[6] M. Luby, "LT-codes," in *Proc. 43rd Annu. IEEE Symp. Foundations of Computer Science (FOCS),* Vancouver, BC, Canada, Nov. 2002, pp. 271-280.

[7] A. Shokrollahi, "Raptor codes", *IEEE Trans. Inform. Theory,* vol. 52, no. 6, June 2006.

[8] 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.

[9] X. Yuan and Li Ping, "Doped accumulate LT codes," *IEEE Int. Symp. Inform. Theory (ISIT)*, Nice, France, June 24-29, 2007.

[10] C. Di, D. Proietti, I. 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.

[11] R. Karp, M. Luby, and A. Shokrollahi, "Finite length analysis of LT codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Chicago, IL, Jun./Jul. 2004.