Fountain Codes with Multiplicatively Repeated Non-Binary LDPC Codes

Kenta KASAI and Kohichi SAKANIWA,
Tokyo Institute of Technology, 152-8550 Tokyo, JAPAN, {kenta, sakaniwa}@comm.ss.titech.ac.jp

Abstract—We study fountain codes transmitted over the binary-input symmetric-output channel. For channels with small capacity, receivers needs to collects many channel outputs to recover information bits. Since a collected channel output yields a check node in the decoding Tanner graph, the channel with small capacity leads to large decoding complexity. In this paper, we introduce a novel fountain coding scheme with non-binary LDPC codes. The decoding complexity of the proposed fountain code does not depend on the channel. Numerical experiments show that the proposed codes exhibit better performance than conventional fountain codes, especially for small number of information bits.

Index Terms—fountain codes, rateless codes, non-binary LDPC codes

I. INTRODUCTION

Fountain codes are a class of erasure-recovering or error-correcting codes which produce limitless sequence of encoded bits from $k$ information bits where receivers can recover the $k$ information bits from any $(1 + \epsilon)k/C$ encoded bits, where $C$ is the channel capacity and $\epsilon$ is referred to as overhead. The name is after water fountains which endlessly produce water drops to entertain people. Designing fountain codes with small overhead is desirable. LT codes [1] and Raptor codes [2] are fountain codes which achieves vanishing overhead $\epsilon \to 0$ in the limit of large information size over the channel with $C = 1$, i.e., the binary erasure channel (BEC). By a nice analogy between the BEC and the packet erasure channel, fountain codes successfully adopted by several industry standards.

In [3], Etesami et al. investigated Raptor codes used over the memoryless binary-input output-symmetric (MBIOS) channels. And they showed that over the AWGN channels with capacity $C \geq 0.49$, Raptor codes achieve overhead $\epsilon \leq 0.08$ at BER $10^{-7}$ with information size $k = 65536$. A Raptor code can be viewed as concatenation of an outer high-rate LDPC code and infinitely many single parity-check codes of length $d$, where $d$ is chosen randomly with probability $\Omega_{d}$ for $d \geq 1$. In [4], Venkiah et al. proposed a joint decoding of the concatenated codes and an optimization method for output degree distributions $\Omega(x) := \sum_{d \geq 1} \Omega_{d} x^{d}$ and showed that the optimized codes outperform the conventional ones.

The problems for constructing fountain codes used for general channels with finite inputs are summarized as follows.

- **Problem 1**: The output degree distribution $\Omega(x)$ needs to be optimized for each $k$. And large check node degree $d$ leads to the large encoding and decoding complexity.

- **Problem 2**: The number of check nodes in the inner codes is given by $(1 + \epsilon)k/C$. This increases as the channel capacity $C$ decreases. Since check node computation is dominant in decoding, the decoding complexity is high for small $C$.

- **Problem 3**: Large size of information and vanishing overhead are often considered. This leads to large size of memory devices and transmission latency.

In this paper, we will propose a novel fountain coding scheme which is free of those drawbacks.

In this paper, we consider non-binary LDPC codes defined by sparse parity-check matrices over $GF(2^m)$ for $2^m > 2$. Non-binary LDPC codes are invented by Gallager [5] and, Davey and MacKay [6] found non-binary LDPC codes can outperform binary ones. Non-binary LDPC codes have captured much attention recently due to their decoding performance.

It is known that the irregularity of Tanner graphs helps improve the decoding performance of binary LDPC codes. While, it is not the case for the non-binary LDPC codes. Interestingly, the $(2, d_c)$-regular non-binary LDPC codes over $GF(2^m)$ are empirically known [7] as the best performing codes for $2^m \geq 64$, especially for short code length. This means that, for designing non-binary LDPC codes, one does not need to optimize the degree distributions of Tanner graphs, since $(2, d_c)$-regular non-binary LDPC codes are best. Furthermore, the sparsity of $(2, d_c)$-regular Tanner graph makes efficient decoding possible.

II. FOUNTAIN CODING WITH MULTIPLICATIVELY REPEATED NON-BINARY LDPC CODES

In this section we explain a new fountain coding scheme. The new coding scheme uses a non-binary LDPC code as a pre-code.

In [8], the authors presented low-rate non-binary codes. The code is a concatenation of $(2, 3)$-regular non-binary LDPC code and inner multiplicative repetition codes. In general, low-rate LDPC codes have many check nodes and suffer from the high decoding complexity than high rate codes. One of the remarkable features of the code is that the decoding complexity does not depend on the coding rate. The code exhibits excellent decoding performance for small code length and is rate-compatible. We will use the low-rate code [8] with vanishing rate as a fountain code.

We fix a Galois field $GF(2^m)$ with a primitive element $\alpha$ and its primitive polynomial $\pi(x)$. Once the primitive element is fixed, one can represent each symbol in the Galois field as a binary sequence of length $m$ [9]. For example, with a primitive...
by giving the encoding procedure as follows.

LDPC pre-code defined over check matrix have degree or equivalently GF(2) follows. The pre-code referred to as Tanner graphs. For a given sparse parity-check code of rate 1/3 over GF(2), defined over each bit in symbols, i.e., \( x_i \) for \( i \geq 1 \). We refer to elements in GF(2) as symbols for \( m \geq 2 \) and bits for \( m = 1 \).

A non-binary LDPC code \( C \) over GF(2) is defined by the null space of a sparse \( M \times N \) parity-check matrix \( H = \{ h_{i,j} \} \) defined over GF(2).

\[
C = \{ x \in \text{GF}(2)^N \mid Hx^T = 0 \in \text{GF}(2)^M \}
\]

The \( c \)-th parity-check equation for \( c = 1, \ldots, M \) is written as

\[
h_{c,1}x_1 + \cdots + h_{c,N}x_N = 0 \in \text{GF}(2)^M,
\]

where \( h_{c,1}, \ldots, h_{c,N} \in \text{GF}(2)^M \) and \( x_1, \ldots, x_N \in \text{GF}(2)^M \).

The binary LDPC codes are represented by Tanner graphs with variable and check nodes [10, pp. 75]. The non-binary LDPC codes, in this paper, are also represented by bipartite graphs with variable nodes and check nodes, which are also referred to as Tanner graphs. For a given sparse parity-check matrix \( H = \{ h_{c,v} \} \) over GF(2), the graph is defined as follows. The \( v \)-th variable node and \( c \)-th check node are connected if \( h_{c,v} \neq 0 \). By \( v = 1, \ldots, N \) and \( c = 1, \ldots, M \), we also denote the \( v \)-th variable node and \( c \)-th check node, respectively.

A non-binary LDPC code with a parity-check matrix over GF(2) is called \((d_v, d_c)\)-regular if all the columns and all the rows of the parity-check matrix have weight \( d_v \) and \( d_c \), respectively, or equivalently all the variable and check nodes have degree \( d_v \) and \( d_c \), respectively. Let \( C_1 \) be a \((2, 3)\)-regular LDPC pre-code defined over GF(2) of length \( N \) symbols or equivalently \( mN \) bits and of rate 1/3. It can be seen that \( N = 3k/m \). The pre-code \( C_1 \) has a \( 2N/3 \times N \) sparse parity-check matrix \( H = \{ h_{i,j} \} \) over GF(2). The matrix \( H \) has row weight 3 and column weight 2. Fig. 1 shows the Tanner graph of \( C_1 \) of length \( N = 18 \) symbols. It can be shown that \((2, d_c)\)-regular non-binary LDPC codes is linear-encodable by using a non-singular zig-zag subgraph.

We define a new fountain code \( C_\infty : \text{GF}(2)^k \rightarrow \text{GF}(2)^\infty \) by giving the encoding procedure as follows.

1. First, map the \( k \) information bits to \( k/m \) information GF(2)-symbol.
2. By the pre-code \( C_1 \), encode the \( k/m \) information symbols to \( N \) symbols \( x_1, \ldots, x_N \in \text{GF}(2)^N \).
3. Repeat the followings endlessly from \( i = 1 \) to \( \infty \).
   a) Pick randomly \( v_i \in [1, N] \), \( w_i \in [1, m] \) and \( h_i \in \text{GF}(2)^{mN} \setminus \{0\} \).
   b) Transmit \( w_i \)-th bit of \( h_i x_{v_i} \in \text{GF}(2)^m \).

The proposed fountain code \( C_\infty \) can be viewed as a non-binary Raptor code with a non-binary \((2, 3)\)-regular LDPC pre-code and an output degree distribution \( \Omega(x) = x[2] \).

Note that \( \Omega(x) = x \) does not mean simple repetition of bits but multiplicative repetition of symbols in GF(2) for the proposed non-binary setting.

### III. Decoding Scheme

We assume that transmission takes place over the MBIOS channel. Specifically, the channel is specified by the transition probability \( P(y|x) = \text{Pr}(Y = y|X = x) \) where \( X \) and \( Y \) are the random variable of an input bit \( x \) and the channel output \( y \), respectively. And we assume that the information bits are chosen with uniform probability.

The most important feature of the fountain coding system is that the decoder does not receive all the channel output but collects \( n \) channel outputs. The decoder recovers the \( k \) information bits from the \( n \) collected channel outputs. The overheard \( \epsilon \) is defined by

\[
\epsilon = C/R - 1, \quad R = k/n,
\]

where \( C \) is the channel capacity. Then, the decoder has \( n = (1 + \epsilon)k/C \) collected channel outputs. Note that, in the original setting of fountain codes as in [1, 2], the capacity is set \( C = 1 \), i.e., all the collected bits are uncorrupted. The aim of the fountain coding in this paper is to reliably recover the information bits with small overhead. The overheard \( \epsilon = 0 \) implies that the information bits are transmitted at rate \( R = C \), which is our extreme aim. With infinitely many information bits, Raptor codes can achieve \( \epsilon = 0 \) for the channel with \( C = 1 \), i.e., the BEC. And Raptor codes optimized for the BEC exhibit a quite good performance for large information bits with \( k = 65536 \). However, for both the BEC and the general MBIOS channels with \( C < 1 \), Raptor codes exhibit high error floors [3, 11, 12, 4] for small information bits with \( k \sim 1024 \).

For the \( i \)-th transmitting bit, the sender picked randomly \( v_i \in [1, N] \), \( w_i \in [1, m] \) and \( h_i \in \text{GF}(2)^{mN} \setminus \{0\} \) and transmitted \( w_i \)-th bit of \( h_i x_{v_i} \in \text{GF}(2)^m \). Let \( I \) be the set of transmitting indices that the receiver collected. It follows \#\( I = n \). In other words, for \( i \in I \), the receiver collects \( y_i \) that is the corrupted version of the \( i \)-th transmitted bits. We assume that the decoder knows not only \( y_i \) but also the indices \( v_i, w_i \) and the multiplicative coefficients \( h_i \) for \( i \in I \). In practice, this is realized by embedding the indices in the header of packets or synchronization between the sender and the receivers [2].

The proposed code \( C_\infty \) can be decoded by the sum-product (SP) decoding algorithm on the Tanner graphs. The SP decoder for the non-binary LDPC codes exchanges probability vectors in \( \mathbb{R}^{2m} \), called messages, between variable nodes and check nodes [13]. An example of the Tanner graph used by the decoder is shown in Fig. 2. The variable nodes of degree
one with white dots in Fig. 2 represent collected channel outputs. If the SP decoding algorithm is immediately applied to the proposed codes, all the variable nodes and check nodes, including the variable nodes of those multiplicative repetition symbols, are activated, i.e. exchange the messages. However, the messages reached at the variable nodes of degree one do not change messages that sent back from the nodes. Therefore, after the initialization, the decoder does not need to pass the messages all the way to those variable nodes of degree 1 and their adjacent check nodes of degree 2. Consequently, the decoder uses only the Tanner graph of the pre-code $C_1$. It follows that the complexity of the decoding algorithm does not depend on the number $n$ of collected channel outputs and the channel capacity $C$. In contrast the decoding complexity of the conventional fountain codes largely depends on $n$ and $C$ as explained in Section I.

The SP decoding involves mainly 4 parts, i.e. the initialization, the check to variable computation, the variable to check computation, and the tentative decision parts. Let $X$ be the random variable of a transmitted bit $x$, and let $Y$ be the random variable of the corresponding channel output $y$. The a posteriori probability $Q(x|y) := P_r(x = x|y = y)$, for $x = 0, 1$ and $y \in \mathcal{A}$ is assumed to be known to the decoder, where $\mathcal{A}$ is the receiving alphabet.

initialization:

The decoders collected $n = (1 + \epsilon)k/C$ channel outputs, $y_i$ for $i \in I$, where $\#I = n$. Define $I_v := \{i \in I \mid v_i = v\}$. It follows that $I = \bigcup_{v=1}^{N} I_v$. For each variable node $v$ in $C_1$ for $v = 1, \ldots, N$, calculate $p_v^{(0)}(x)$ for $x \in \mathbb{GF}(2^n)$ as follows,

$$p_v^{(0)}(x) = \xi \prod_{i \in I_v} \tilde{p}_i^{(0)}(h_i x) \quad (1)$$

where $\xi$ is the normalized factor such that $\sum_{x \in \mathbb{GF}(2^n)} p_v^{(0)}(x) = 1$. Each variable node $v = 1, \ldots, N$ in $C_1$ sends the initial message $p_v^{(0)} = p_v^{(0)} \in \mathbb{R}^{2^n}$ to each adjacent check node $c$. Set the iteration round as $\ell := 0$.

check to variable output:

For each check node $c = 1, \ldots, M$ in $C_1$, let $V_c$ be the set of the adjacent variable nodes. It holds that $\#V_c = 3$, since the pre-code $C_1$ is $(2, 3)$-regular. Each $c$ has 3 incoming messages $p_{vc}^{(0)}$ for $v \in V_c$ from the 3 adjacent variable nodes. The check node $c$ sends the following message $p_{vc}^{(0+1)} \in \mathbb{R}^{2^n}$ to each adjacent variable node $v \in V_c$.

$$p_{vc}^{(0+1)}(x) = p_{vc}^{(0)}(h_{vc} x) \text{ for } x \in \mathbb{GF}(2^n),$$

where $p_1 \otimes p_2 \in \mathbb{R}^{2^n}$ is convolution of $p_1 \in \mathbb{R}^{2^n}$ and $p_2 \in \mathbb{R}^{2^n}$. To be precise,

$$(p_1 \otimes p_2)(x) = \sum_{y, z \in \mathbb{GF}(2^n)} p_1(y)p_2(z) \text{ for } x \in \mathbb{GF}(2^n).$$

The convolution seems the most complex part of the decoding. Indeed, the convolutions are efficiently calculated via FFT and IFFT [14], [13]. Increment the iteration round as $\ell := \ell + 1$.

tentative decision

For each $v = 1, \ldots, N$, the tentatively estimated $v$-th transmitted symbol is given as

$$\hat{x}_v^{(0)} = \arg \max_{x \in \mathbb{GF}(2^n)} \prod_{c \in C_v} \tilde{p}_c^{(0)}(x) p_{vc}^{(0)}(x).$$

If $\tilde{x}_v^{(0)} := (\hat{x}_1^{(0)}, \ldots, \hat{x}_N^{(0)})$ forms a codeword of $C_1$, i.e. $\tilde{x}_v^{(0)}$ satisfies every parity-check equation of $C_1$

$$\sum_{v \in V_c} h_{vc} \hat{x}_v^{(0)} = 0 \in \mathbb{GF}(2^n)$$

for all $c = 1, \ldots, M$, the decoder outputs $\hat{x}_v^{(0)}$ as the estimated codeword. Otherwise repeat the check to variable, variable to check and tentative steps. If the iteration round $\ell$ reaches at a pre-determined number, the decoder collects more channel outputs and start over the decoding.

IV. ANALYSIS OF ASYMPTOTIC OVERHEAD

In this section, we investigate the overhead $\epsilon$ in the limit of many information bits $k \to \infty$ for the transmissions over the BEC, i.e., $C = 1$. Rathi developed the density evolution which enables the prediction of the decoding performance of the non-binary LDPC codes in the limit of large code length. The density evolution usually gives, for a given code ensemble, the maximum channel erasure probability, referred to as threshold, at which the average decoding erasure probability goes to zero. We will use the density evolution calculating the maximum overhead $\epsilon$ at which the average decoding erasure probability goes to zero in the limit of $k \to \infty$. 
The density evolution used in this section was originally developed for the non-binary LDPC code ensembles with parity-check matrices defined over the general linear group GL(GF(2), m). However, Rathi reported that the threshold for the code ensemble defined over GF(2^m) and GL(GF(2), m) also have approximately the same threshold within the order of 10^{-4}. Consequently, we shall evaluate the threshold of the proposed codes by the density evolution for GL(GF(2), m).

In the binary case, we can predict the asymptotic decoding performance of LDPC codes transmitted over the general MBiOS channels in the large code length limit by density evolution [15]. Density evolution is possible also for the non-binary LDPC codes [16] but computationally intensive and tractable only for the BEC. The analysis for the BEC often helps us to capture the universal properties of LDPC codes.

When the transmission is taken place over the BEC and all-zero codewords are assumed to be sent, the messages, described by probability vectors \( p(x) \) where \( x \in \text{GF}(2^m) \) of length \( 2^m \) in general, can be reduced to linear subspaces of \( \text{GF}(2)^m \) [13]. To be precise, for each message in the SP decoding algorithm, a subset of \( \{ x \in \text{GF}(2)^m \mid p(x) \neq 0 \} \) forms a linear subspace of \( \text{GF}(2)^m \), where \( x \) is the binary representation of \( x \in \text{GF}(2^m) \).

For messages in SP decoding, probability vectors \( P = (P_0, \ldots, P_m) \) are used for the density evolution and referred to as densities. The i-th entry \( P_i \) is the probability that a message forms a subspace of dimension \( i \) for \( i = 1, \ldots, m \). Define two densities \( P^{(\ell)} \) and \( Q^{(\ell)} \) as the densities of messages sent from variable nodes and check nodes at the \( \ell \)-th iteration round, respectively. In [17], Rathi proved that the density that outgoing messages from a variable (resp. check) node of degree 3 with two incoming messages of density \( P \) and \( Q \) is given by \( P \boxtimes Q \) (reps. \( P \boxdot Q \)). The detailed calculation of \( P \boxtimes Q \) and \( P \boxdot Q \) are defined below. Using these 2 operations of 2 densities, the density evolution in [17] gives recursive update equations of \( P^{(\ell)} \) and \( Q^{(\ell)} \) for \( \ell \geq 0 \).

Rathi [13] developed the density evolution for the BEC that tracks probability densities of the dimension of the linear subspaces. For \( \ell \geq 0 \), the density evolution tracks the probability vectors \( E^{(\ell)} \) and \( Q^{(\ell)} \) which are referred to as densities. The initial messages in Eq. (1) can be seen as the intersection of \( d \) subspaces of the messages received as the channel outputs.

With \( \epsilon \) overhead, the decoder has \( k(1+\epsilon)/C \) channel outputs transmitted over the channel with capacity \( C \). The number of variable node in \( C_1 \) is \( N \). It holds that \( N = 3(N-M) = 3mk \), since \( C_1 \) is of rate 1/3 and defined over \( \text{GF}(2^m) \). The average number of collected channel outputs per variable node in \( C_1 \) is given by \( (1+\epsilon)/(3m) =: \beta \). It follows that the probability \( R_d \) that a randomly chosen variable node in \( C \) has \( d \) corresponding channel outputs is given by

\[
R_d = \left( \frac{N}{d} \right) \left( \frac{\beta}{N} \right)^d \left( 1 - \frac{\beta}{N} \right)^{N-d}.
\]

It follows that

\[
\sum_{d \geq 0} R_d x^d = \left( \frac{\beta}{N} x + 1 - \frac{\beta}{N} \right)^N \rightarrow_{N \rightarrow \infty} e^{-\beta(1-x)} = \sum_{d \geq 0} \frac{\beta^d e^{-\beta}}{d!} x^d.
\]

From this, we see the probability that a randomly chosen variable node in \( C \) has \( d \) corresponding channel outputs in the limit of \( k \rightarrow \infty \) is \( \frac{\beta^d e^{-\beta}}{d!} \). The density of the initial messages is given by \( P^{(0)} \) as follows,

\[
P^{(0)} = \sum_{d \geq 0} \frac{\beta^d e^{-\beta}}{d!} E \circ \cdots \circ E
\]

where \( E \) is a density such that the subspace is of dimension \( m-1 \) with probability 1. In precise, \( E := (E_0, \ldots, E_m) \),

\[
E_i := \begin{cases} 
1 & \text{if } i = m-1 \\
0 & \text{if } i \neq m-1.
\end{cases}
\]

Since the pre-code is a (2,3)-regular LDPC codes, we have recursive update equations of densities as follows.

\[
Q^{(\ell+1)} = P^{(\ell)} \boxtimes Q^{(\ell)}, \quad P^{(\ell+1)} = P^{(0)} \boxdot Q^{(\ell+1)}.
\]

Since the messages of dimension 0 corresponds to the successful decoding, the asymptotic overhead \( \epsilon^* \) is defined as follows.

\[
\epsilon^* := \sup_{\epsilon \in [0,1]} \lim_{\ell \rightarrow \infty} P^{(\ell)} = 1.
\]

It follows that, in the limit of many information bits \( k \rightarrow \infty \), with overhead \( \epsilon < \epsilon^* \) the reliable transmissions are possible with the proposed \( C_\infty \).

Table II shows the asymptotic overhead \( \epsilon^* \) of the proposed code \( C_\infty \) over \( \text{GF}(2^m) \) for different \( m = 1, \ldots, 19 \). Table II also lists the asymptotic overheads with (2,dL)-regular non-binary LDPC pre-code for \( d_L = 4, 5 \) and 6. It can be seen that the best overhead \( \epsilon^* = 0.079 \) is attained at \( d_L = 3 \) and \( m = 9 \) and the fountain code \( C_\infty \) exhibit very poor overhead if defined on \( \text{GF}(2^m) \) with \( m = 1 \), i.e., the binary field. We will use \( m = 8 \), for its good asymptotic overhead \( \epsilon^* = 0.081 \) in Tab. II and friendliness for byte-oriented processors.

V. Numerical Results

In this section, we present demonstrations of \( C_\infty \) defined over \( \text{GF}(2^8) \) with small and moderate information bits. Transmission over the BEC and the AWGN channels are considered. Fig. 3 shows the histograms of overheads of \( C_\infty \) defined over \( \text{GF}(2^8) \). It seems that the asymptotic overhead is getting concentrated at 0.081 as predicted in Section IV.
TABLE I
ASYMPTOTIC OVERHEAD $\epsilon^*$ OF THE PROPOSED CODES $C_\infty$ WITH A
PRE-CODE $(2, d_c)$-REGULAR NON-BINARY LDPC CODES OVER GF$(2^m)$
TRANSMITTED OVER THE BEC, i.e., $C = 1.0$.

| $m$ | $d_c = 3$ | $d_c = 4$ | $d_c = 5$ | $d_c = 6$ |
|-----|-----------|-----------|-----------|-----------|
| 1   | 1.0799    | 3.3943    | 5.9311    | 8.6535    |
| 2   | 0.9748    | 3.2374    | 4.2477    | 6.3698    |
| 3   | 0.9295    | 1.8031    | 4.2128    | 5.1370    |
| 4   | 0.8797    | 1.3841    | 2.9732    | 4.5078    |
| 5   | 0.8123    | 1.8676    | 2.7151    | 4.1293    |
| 6   | 0.7999    | 1.2910    | 2.5356    | 3.8855    |
| 7   | 0.8089    | 1.2359    | 2.4487    | 3.7210    |
| 8   | 0.8089    | 1.2025    | 2.3786    | 3.6068    |
| 9   | 0.0792    | 1.1826    | 2.3132    | 3.5256    |
| 10  | 0.0813    | 1.1716    | 2.2987    | 3.4668    |
| 11  | 0.0856    | 1.1651    | 2.2865    | 3.4238    |
| 12  | 0.0911    | 1.1648    | 2.2613    | 3.3904    |
| 13  | 0.0977    | 1.1653    | 2.2511    | 3.3659    |
| 14  | 0.1044    | 1.1677    | 2.2445    | 3.3472    |
| 15  | 0.1111    | 1.1771    | 2.2393    | 3.3331    |
| 16  | 0.1179    | 1.1854    | 2.2333    | 3.3222    |
| 17  | 0.1245    | 1.1901    | 2.2278    | 3.3142    |
| 18  | 0.1309    | 1.1851    | 2.2230    | 3.3081    |
| 19  | 0.1371    | 1.1901    | 2.2392    | 3.3036    |

Fig. 3. Histograms of the overheads at which the proposed fountain code over GF$(2^m)$ successfully recovers $k$ information bits over the channel with $C = 1.0$. The number of the information bits is set $k = 192$, $512$, $1024$, $2048$, $8192$, $16384$ and $32768$ from the top to bottom. The horizontal axis describes the overhead $\epsilon$. It can be seen that it is getting concentrated at overhead 0.081 as predicted in Tab. I at $m = 8$.

shows the decoding performance of the proposed fountain code transmitted over the binary-input AWGN channels with capacity $C = 1.0$, 0.5 and 0.1. The horizontal axis describes the overhead and the vertical axis describes the block error rate. The proposed codes exhibit the better performance than the best-so-far Raptor codes.

VI. CONCLUSION

In this paper we propose a new simple fountain coding scheme whose decoding complexity does not depend on the number of collected channel outputs. No optimization of the output degree distribution is needed. Because of the non-binary property, we believe the proposed codes can be used for memory channel.

REFERENCES

[1] M. Luby, “LT codes,” in Proc. 40th Annual Allerton Conf. on Commun., Control and Computing, 2002, pp. 271 – 280.
[2] A. Shokrollahi, “Raptor codes,” IEEE Trans. Inf. Theory, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.
[3] O. Etesami and A. Shokrollahi, “Raptor codes on binary memoryless symmetric channels,” IEEE Trans. Inf. Theory, vol. 52, no. 5, pp. 2033 – 2051, May 2006.
[4] A. Venkiah, C. Poulliat, and D. Declercq, “Jointly decoded raptor codes: analysis and design for the BIAWGN channel,” EURASIP J. Wirel. Commun. Netw., vol. 2009, pp. 1–11, 2009.
[5] R. G. Gallager, Low Density Parity Check Codes. in Research Monograph series, MIT Press, Cambridge, 1963.
[6] M. Davy and D. MacKay, “Low-density parity check codes over GF$(q)$,” IEEE Commun. Lett., vol. 2, no. 6, pp. 165–167, Jun. 1998.
[7] C. Poulliat, M. Fossorier, and D. Declercq, “Design of regular $(2, d_c)$-LDPC codes over GF$(q)$ using their binary images,” IEEE Trans. Commun., vol. 56, no. 10, pp. 1626–1635, Oct. 2008.
[8] K. Kasai, D. Declercq, C. Poulliat, and K. Sakaniwa, “Multiplicatively repeated non-binary LDPC codes,” 2010, submitted for publication, [http://arxiv.org/abs/1004.5367].
[9] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error-Correcting Codes. Amsterdam: Elsevier, 1977.
[10] T. Richardson and R. Urbanke, Modern Coding Theory. Cambridge University Press, Mar. 2008.
[11] A. Venkiah, C. Poulliat, and D. Declercq, “Rate splitting issue for finite length Raptor codes,” in IEEE Sarnoff Symposium, Apr. 2008.
[12] A. Venkiah, “Analyse et optimisation de codes Raptor pour les canaux multicast sans fil,” Ph.D. dissertation, Universit de Cergy-Pontoise, Nov. 2008. [Online]. Available: [http://library.epfl.ch/theses/?nr=4111]
[13] V. Rathi and R. Urbanke, “Density Evolution, Threshold and the Stability Condition for non-binary LDPC Codes,” IEEE Proceedings - Communications, vol. 152, no. 6, pp. 1069–1074, 2005.
[14] D. Declercq and M. Fossorier, “Decoding algorithms for nonbinary LDPC codes over GF$(q)$,” IEEE Trans. Commun., vol. 55, no. 4, pp. 633–643, Apr. 2007.
[15] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” IEEE Trans. Inf. Theory, vol. 47, pp. 619–637, Feb. 2001.
[16] A. Bennatan and D. Brustolin, “On the application of LDPC codes to arbitrary discrete-memoryless channels,” IEEE Trans. Inf. Theory, vol. 50, no. 3, pp. 417–438, Mar. 2004.
[17] V. Rathi, “Non-binary LDPC codes and EXIT like functions,” Ph.D. dissertation, Lausanne, 2008. [Online]. Available: [http://library.epfl.ch/theses/?nr=4111].