

Finite-Length Analysis of BATS Codes

Tsz-Ching Ng

Department of Mathematics

The Chinese University of Hong Kong, Hong Kong, China

Email: tcng@math.cuhk.edu.hk

Shenghao Yang

Institute for Interdisciplinary Information Sciences

Tsinghua University, Beijing, China

Email: shyang@tsinghua.edu.cn

Abstract—In this paper, performance of finite-length batched sparse (BATS) codes with belief propagation (BP) decoding is analyzed. For fixed number of input symbols and fixed number of batches, a recursive formula is obtained to calculate the exact probability distribution of the stopping time of the BP decoder. When the number of batches follows a Poisson distribution, a recursive formula with lower computational complexity is derived. Inactivation decoding can be applied to reduce the receiving overhead of the BP decoder, where the number of inactive symbols determines the extra computation cost of inactivation decoding. Two more recursive formulas are derived to calculate the expected number of inactive symbols for fixed number of batches and for Poisson distributed number of batches, respectively. Since LT/Raptor codes are BATS codes with unit batch size, our results also provide new analytical tools for LT/Raptor codes.

Index Terms—BATS codes, finite-length analysis, belief propagation, inactivation decoding

I. INTRODUCTION

Batched sparse (BATS) codes [1], [2] have been proposed for file transmission through networks employing linear network coding [3]. BATS codes generalize LT codes [4] and Raptor codes [5] by generating coded symbols in batches. Coded symbols of the same batch are encoded using the same subset of input symbols, and a BATS code with unit batch size is just an LT/Raptor code. An advantage of using batch size larger than one is that linear network coding can be applied on symbols of the same batch during the network transmission, while the degree distribution of the received batches remains unchanged. This property guarantees that with proper designed degree distribution, efficient belief propagation (BP) decoding can be applied to decode BATS codes. BATS codes with linear network coding achieve higher rate than fountain codes with routing, and have lower complexity than the normal random linear network coding scheme that combines all symbols together during the network transmission.

The existing works only analyzed the asymptotic performance of BATS codes. The analysis of BATS codes in [2] provides a sufficient condition for the BP decoder to recover a fixed fraction of the input symbols with high probability when both the number K of input symbols and the number n of batches received tend to infinity. The sufficient condition induces an optimization problem, the solution of which gives a degree distribution that demonstrates nearly optimal performance when K goes to infinity.

Though the asymptotic analysis demonstrates that BATS codes can be nearly capacity achieving, the performance of

BATS codes when K is relatively small is of more practical interests. The error bound obtained in the asymptotic analysis is rather loose for small K . In this paper, we provide analysis of BATS codes with finite number of input symbols. Since LT/Raptor codes are BATS codes with unit batch size, our results also provide new analytical tools for LT/Raptor codes.

Particularly, for given values of K and n , we derive the distribution of the stopping time of the BP decoder using a recursive formula, which can be used to calculate the error probability of the BP decoder (decoding error occurs when the BP decoder stops before a target number of input symbols are recovered). The computational complexity of evaluating the above recursive formulas is $\mathcal{O}(K^2 n^2 M)$, where M is the batch size. Our formula applies to LT codes by setting $M = 1$. Though the formula obtained by Karp et al. for LT codes [6] has slightly lower complexity than our formula with $M = 1$, directly extending the analysis in [6] to $M > 1$ results in much higher complexity since more variables should be tracked recursively.

Inactivation decoding [7] has been used for LT/Raptor codes to reduce the number of extra batches required for decoding (also known as *decoding overhead*). A similar method can be applied to BATS codes to reduce the decoding overhead. Inactivation decoding trades computation cost (decoding inactive input symbols using Gaussian elimination) to get low decoding overhead. To understand the tradeoff between computation cost and decoding overhead, for given values of K and n we derive a recursive formula to compute the expected number of inactive symbols required for inactivation decoding, which shares the same computational complexity of the first recursive formula.

The number of batches received in a time interval is random in general. For a given distribution of the number of batches, directly using the above formulas to accurately compute the error probability and the expected number of inactive symbols can have high computation cost. But for Poisson distributed number of batches, we show that those values can be computed using simpler recursive formulas, which have complexity $\mathcal{O}(K\bar{n}M^2D)$, where \bar{n} is the mean of the number of batches, and D is the maximum degree. Our Poisson model is different from the model used to analyze LT codes by Maneva and Shokrollahi [8], where the number of received coded symbols is the summation of random variables of binomial distributions.

Due to the page limit, only parts of the proofs are given

to demonstrate our techniques used to obtain these recursive formulas. Readers are referred to [9] for the full proofs.

II. BATS CODES

In this section, the encoding and decoding processes of BATS codes will be briefly introduced. Readers may refer to [2] for more discussion.

A. Encoding

Suppose K input symbols will be transmitted from a source node to a sink node through a network employing linear network coding, where each input symbol is an element¹ of the finite field \mathbb{F}_q of size q . For the purpose of analysis, we describe a random encoding procedure of BATS codes. A *batch* is a row vector of M symbols. The encoder of a BATS code generates a potentially unlimited sequence of batches $\mathbf{X}_1, \mathbf{X}_2, \dots$ using the K input symbols and a degree distribution $\Psi = \{\Psi_d, d = 1, \dots, D\}$, where D is the maximum degree. Each batch \mathbf{X}_i is generated independently using the same procedure as follows.

First, choose a degree d_i by sampling the distribution Ψ . Second, uniformly at random choose an index set A_i of d_i integers ranging from 1 to K , and form a row vector \mathbf{B}_i by using the input symbols with indices in A_i . Third, set $\mathbf{X}_i = \mathbf{B}_i \mathbf{G}_i$, where \mathbf{G}_i is a $d_i \times M$ matrix with all components being uniformly i.i.d. from the field \mathbb{F}_q . We call the symbols with indices in A_i the contributors of the batch \mathbf{X}_i .

Symbols in batches are sent out by the source node using certain scheduling scheme. By assuming that the network only applies linear operations on each batch and does not mix different batches together, the received form of batch \mathbf{X}_i is $\mathbf{Y}_i = \mathbf{X}_i \mathbf{H}_i$, where \mathbf{H}_i , called a *transfer matrix*, is determined by the network operations including linear network coding and packet loss. We assume that $\mathbf{H}_i, i = 1, 2, \dots$ are mutually independent and following the same distribution of \mathbf{H} , and $\mathbf{H}_i, i = 1, 2, \dots$ are also independent to the encoding process.

The network output \mathbf{Y}_i of a batch can be expressed as

$$\mathbf{Y}_i = \mathbf{B}_i \cdot \mathbf{G}_i \cdot \mathbf{H}_i. \quad (1)$$

We call \mathbf{Y}_i a received batch, or a batch for simplicity, and we call (1) the associated linear system of batch i .

Similar to Raptor codes, precoding can be applied to the input symbols before applying the above encoding procedure. A BATS code with batch size $M = 1$ is just a Raptor code or an LT code, depending on whether precoding is applied or not.

B. Belief propagation decoding

Consider n batches $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_n$ are received. We assume that the sink knows $\mathbf{G}_i \mathbf{H}_i$ and A_i for $i = 1, \dots, n$. The decoding of BATS codes is actually to solve the linear systems of the form in (1), where \mathbf{B}_i is the variable, to recover the input symbols. We use the following belief propagation (BP) decoding algorithm, which is also illustrated in Fig. 1.

¹In general, each symbol can be a vector of elements, but the generalization does not affect the analysis.

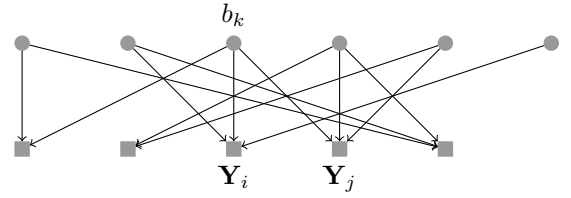


Fig. 1. A decoding graph. Nodes in the first row represent the input symbols. Nodes in the second row represents the batches.

A received batch \mathbf{Y}_i is called *decodable* if $\mathbf{G}_i \mathbf{H}_i$ has rank d_i . If so, then \mathbf{B}_i is recovered by solving the linear system (1), which has a unique solution since $\text{rk}(\mathbf{G}_i \mathbf{H}_i) = d_i$. The symbols in \mathbf{B}_i , once decoded, can be substituted into and hence simplify the associated linear systems of batches \mathbf{Y}_j with $A_j \cap A_i \neq \emptyset$. For example in Fig. 1, if \mathbf{Y}_i is decoded, b_k can be recovered. Since b_k is a contributor of \mathbf{Y}_j , the value of b_k can be substituted into the linear system associated with \mathbf{Y}_j . After the substitution, some previously undecodable batches may become decodable. We repeat the above decoding and substituting procedure until there are no decodable batches.

The goal of this paper is to analyze the number of input symbols recovered when the BP decoding stops. However, directly analyzing the above decoding procedure is difficult. We instead use a decoding process that in each decoding step only one input symbol is decoded, which is described as follows.

The time starts at 0 and increases by 1 after each decoding step. We say an input symbol is *decodable* if it contributes to a decodable batch. In each decoding step, we first pick a decodable input symbol², and mark it as decoded. Then we substitute the input symbol to the associated linear systems of the batches it contributes to. The decoding stops when there is no decodable input symbols.

For each batch i and time t , let A_i^t be the indices of the contributors of batch i that have not been decoded. Note that $A_i^0 = A_i$. Let j be the index of the input symbol decoded in the decoding step right after time t . Then $A_i^{t+1} = A_i^t \setminus \{j\}$ if $j \in A_i^t$, and $A_i^{t+1} = A_i^t$ otherwise. Define $\mathbf{B}_i^t, \mathbf{G}_i^t$ and \mathbf{Y}_i^t for each batch i and time t as follows. First, $\mathbf{B}_i^0 = \mathbf{B}_i, \mathbf{G}_i^0 = \mathbf{G}_i$ and $\mathbf{Y}_i^0 = \mathbf{Y}_i$. For $t \geq 0$, $\mathbf{B}_i^{t+1} = \mathbf{B}_i^t, \mathbf{G}_i^{t+1} = \mathbf{G}_i^t$ and $\mathbf{Y}_i^{t+1} = \mathbf{Y}_i^t$ if $A_i^{t+1} = A_i^t$. Otherwise, let j be the index in $A_i^t \setminus A_i^{t+1}$. Then \mathbf{B}_i^{t+1} is formed by removing the component of \mathbf{B}_i^t corresponding to b_j , \mathbf{G}_i^{t+1} is formed by removing the row g of \mathbf{G}_i^t corresponding to b_j , and $\mathbf{Y}_i^{t+1} = \mathbf{Y}_i^t - b_j g$. The associated linear system of batch i at time t can be denoted by $\mathbf{Y}_i^t = \mathbf{B}_i^t \cdot \mathbf{G}_i^t \cdot \mathbf{H}_i$.

C. Solvability of a batch

At time t of the decoding procedure, the degree of a batch i is $|A_i^t|$, and define the *rank* of the batch to be $\text{rk}(\mathbf{G}_i^t \mathbf{H}_i)$. A batch becomes decodable only when its degree equals to its

²How to choose the input symbol does not affect the time when the decoding stops.

rank. Let us check the probability that a batch is decodable when its degree is s .

Let $\mathbf{G}_{[s]}$ be an $s \times M$ random matrix with uniformly i.i.d. components in \mathbb{F}_q , and $\mathbf{G}'_{[s]}$ be the submatrix of $\mathbf{G}_{[s]}$ obtained by removing one row. Define

$$\tilde{h}_s := \Pr\{\text{rk}(\mathbf{G}_{[s+1]}\mathbf{H}) = \text{rk}(\mathbf{G}'_{[s+1]}\mathbf{H}) = s\}, \quad (2)$$

$$\tilde{h}'_s := \Pr\{\text{rk}(\mathbf{G}_{[s]}\mathbf{H}) = s\}. \quad (3)$$

We can check that \tilde{h}_s is the probability that a batch is decodable for the first time when its degree is s . Once a batch becomes decodable, it keeps to be decodable until all its contributors are decoded. Similarly, we see that \tilde{h}'_s is the probability that a batch is decodable when its degree is s . Note that $\tilde{h}'_s = \sum_{k \geq s} \tilde{h}_k$ and for $s > M$, $\tilde{h}_s = 0$.

The explicit forms of \tilde{h}_s and \tilde{h}'_s will not be directly used in the analysis, but are useful in the numerical evaluation. According to Appendix II of [2],

$$\tilde{h}_s = \sum_{i=s}^M \frac{\zeta_s^i}{q^{i-s}} h_i \quad \text{and} \quad \tilde{h}'_s = \sum_{k=s}^M \zeta_s^k h_k,$$

where

$$\zeta_r^m := \begin{cases} (1 - q^{-m})(1 - q^{-m+1}) \cdots (1 - q^{-m+r-1}) & r > 0 \\ 1 & r = 0. \end{cases}$$

and $h_r := \Pr\{\text{rk}(\mathbf{H}) = r\}$ is the rank distribution of \mathbf{H} .

When applying the analysis to LT/Raptor codes, i.e. $M = 1$, we can use deterministic generator matrices with all components being the identity of the field, and hence $\tilde{h}_1 = \tilde{h}'_1 = h_1$. In the case of $M = 1$, h_0 can be regarded as the erasure rate.

III. STOPPING TIME OF BP DECODER

In this section, we fix a number K of input symbols and compute the distribution of the stopping time of the BP decoder. In general, the number of (received) batches can be random. Let N be the random variable of the number of batches used in the decoder.

A. Fixed number of batches

We start with the performance of BP decoding for a fixed number n of batches, i.e., the condition $N = n$ is assumed.

Let R^t be the number of decodable input symbols at time t . The probability that the decoder stops at time t is

$$\Pr\{R^t = 0, R^\tau > 0, \tau < t | N = n\}.$$

The stopping time of the BP decoder is equal to the number of packets that can be decoded.

Let C^t be the number of batches with its degree strictly larger than its rank at time t , i.e., the number of undecodable batches at time t . We are interested in the probabilities

$$\Lambda_{c,r|n}^t := \Pr\{C^t = c, R^t = r, R^\tau > 0, \tau < t | N = n\}.$$

We will express $\Lambda_{c,r|n}^t$ in terms of $\Lambda_{c',r'|n}^{t'}$'s for $t' < t$, so that we can calculate $\Lambda_{c,r|n}^t$'s recursively.

For $c \leq n$, define

$$\Lambda_{c|n}^t := (\Lambda_{c,0|n}^t, \Lambda_{c,1|n}^t, \dots, \Lambda_{c,K-t|n}^t),$$

and let $(\Lambda_{c|n}^t)^{\setminus 1}$ be the sub-vector of $\Lambda_{c|n}^t$ without the first component. Let

$$\text{Bi}(n, k; p) := \binom{n}{k} (p)^k (1-p)^{n-k}$$

and let

$$\text{hyge}(n, i, j, k) := \begin{cases} \frac{\binom{i}{k} \binom{n-i}{j-k}}{\binom{n}{j}} & k \leq \min\{i, j\} \\ 0 & o.w. \end{cases}$$

be the p.m.f. of hyper-geometric distribution.

Theorem 1. *Given the number K of input symbols, the number n of batches, the degree distribution $\{\Psi_d\}$, the rank distribution $\{h_r\}$ of the transfer matrix, the maximum degree D and the batch size M , we have*

$$\Lambda_{c|n}^0 = \text{Bi}(n, c; 1 - \rho^0) \mathbf{e}_1 \mathbf{Q}_0^{n-c}, \quad (4)$$

and for $t > 0$,

$$\Lambda_{c|n}^t = \sum_{c' \geq c} \text{Bi}(c', c; 1 - \rho^t) (\Lambda_{c'|n}^{t-1})^{\setminus 1} \mathbf{Q}_t^{c'-c}$$

where the notations are defined as follows:

- 1) $\mathbf{e}_1 = (1, 0, \dots, 0)$.
- 2) $\rho^0 = \sum_s p_s^0$, where $p_s^0 = \Psi_s \tilde{h}'_s$.
- 3) For $t > 0$,

$$\rho^t = \frac{\sum_s p_s^t}{1 - \sum_{\tau=0}^{t-1} \sum_s p_s^\tau}$$

where

$$p_s^t = \tilde{h}_s \sum_{d=s+1}^D \Psi_d \frac{d}{K} \text{hyge}(K-1, d-1, t-1, d-s-1).$$

- 4) For $t \geq 0$, \mathbf{Q}_t is a $(K-t+1) \times (K-t+1)$ matrix with

$$\mathbf{Q}_t(i+1, j+1) = \sum_{s=j-i}^j \frac{p_s^t}{\sum_s p_s^t} \text{hyge}(K-t, i, s, i+s-j)$$

if $j - M \leq i \leq j$, and $\mathbf{Q}_t(i+1, j+1) = 0$ otherwise, where $\mathbf{Q}_t(i+1, j+1)$ is the component of \mathbf{Q}_t on the $(i+1)$ th row and $(j+1)$ th column.

The derivation of the initial status (4) is given in Appendix. The probability that BP decoding stops at time t is $\sum_c \Lambda_{c,0|n}^t$, and hence the error probability of BP decoding can be calculated. When there is no precoding, the BP decoder must recover all input symbols, so the error probability is $\sum_{t=0}^{K-1} \sum_c \Lambda_{c,0|n}^t$. When precoding is applied, the BP decoder only needs to recover a fraction of the input symbols, so the error probability is $\sum_{t=0}^{K'-1} \sum_c \Lambda_{c,0|n}^t$, where K' is the number of input symbols to be recovered.

The computational complexity to evaluate the recursive formula in Theorem 1 is $\mathcal{O}(K^2 n^2 M)$, where K is the number of input symbols, n is the number of batches and M is the batch size. The complexity can be shown by first noting that the quantities $\{p_s^t, \rho^t, \mathbf{Q}_t\}_{t \leq K, s \leq M}$ can be computed in $\mathcal{O}(K^2 M D)$ steps using recursive formulas (e.g., $\binom{n}{k} =$

$\frac{n}{n-k} \binom{n-1}{k}$) to compute binomial coefficients. Secondly, the matrix \mathbf{Q}_t has at most $K+1$ columns, and each column has most $M+1$ non-zero elements. Therefore, the complexity for multiplying a vector to \mathbf{Q}_t is $\mathcal{O}(KM)$. For fixed $t \leq K$, $c \leq n$, the vectors $\{(\mathbf{\Lambda}_{c|n}^{t-1})^{\setminus 1} \mathbf{Q}_t^{c'-c}\}_{c \leq c' \leq n}$ can be computed recursively in $\mathcal{O}(KnM)$ steps. Hence, the total complexity is $\mathcal{O}(K^2 n^2 M)$, where we assume $D = \mathcal{O}(n^2)$. Since M is small (e.g., 32) and n is usually linear with K , we can also say that the complexity is $\mathcal{O}(K^4)$.

The approach we use to prove Theorem 1 is different from the one used to analyze LT codes by Karp et al. [6], where the number of decodable received symbols (called *output ripple*) is calculated recursively. Directly extending their approach for $M > 1$ requires us to calculate the number of decodable batches. But since decodable batches, when $M > 1$, can have different degrees, different recursive formula must be provided for each degree value of decodable batches. So directly generalizing their approach will result in much higher complexity. Our approach, instead, tracks the number of decodable input symbols. Though our formula has slightly higher complexity ($\mathcal{O}(K^4)$) when $M = 1$ than the formula of LT codes in [6] ($\mathcal{O}(K^3 \log^2(K) \log \log(K))$), our approach is simpler to apply for cases with $M > 1$.

B. Poisson number of batches

In network communications, the number of received packets in a time interval is usually random, and typically assumed to be Poisson distributed. When N follows a general distribution, we can calculate the error probability using the above formulas by first calculating the error probabilities for all possible instances n of N , and then combining these error probabilities according to the distribution of N , i.e., the error probability is

$$\sum_n \Pr\{N = n\} \sum_{t=0}^{K'-1} \sum_c \Lambda_{c,0|n}^t. \quad (5)$$

But this approach is not usable when the support of N is large. For example, when N has a Poisson distribution, the support of N is non-negative integers, and hence accurately computing the error probability using (5) directly is not possible.

In this subsection, we show that there exists a simpler recursive formula to compute the exact distribution of the stopping time of BP decoder when N has a Poisson distribution. We consider that the number N of batches is Poisson distributed with mean \bar{n} , that is,

$$\Pr\{N = n\} = \frac{\bar{n}^n}{n!} e^{-\bar{n}}.$$

For $0 \leq t \leq K$, define

$$\Lambda_r^t := \Pr\{R^t = r, R^\tau > 0, \tau < t\},$$

which is the probability that the BP decoder successfully decodes t input symbols and $R^t = r$. Define a vector of size $K-t+1$:

$$\mathbf{\Lambda}^t := (\Lambda_0^t, \Lambda_1^t, \dots, \Lambda_{K-t}^t).$$

Then, we have

$$\mathbf{\Lambda}^t = \sum_n \Pr\{N = n\} \sum_{c \leq n} \mathbf{\Lambda}_{c|n}^t.$$

Before giving the general result, we show $\mathbf{\Lambda}^0$ as an example. Substituting $\Pr\{N = n\}$ and $\mathbf{\Lambda}_{c|n}^0$ given in Theorem 1,

$$\begin{aligned} \mathbf{\Lambda}^0 &= \sum_n \frac{\bar{n}^n}{n!} e^{-\bar{n}} \sum_{c \leq n} \text{Bi}(n, c; 1 - \rho^0) \mathbf{e}_1 \mathbf{Q}_0^{n-c} \\ &= \sum_{c, n: c \leq n} \frac{\bar{n}^n}{n!} e^{-\bar{n}} \binom{n}{c} (1 - \rho^0)^c (\rho^0)^{n-c} \mathbf{e}_1 \mathbf{Q}_0^{n-c} \\ &= e^{-\bar{n}} \mathbf{e}_1 \sum_{c, n: c \leq n} \frac{(\bar{n}(1 - \rho^0))^c}{c!} \frac{(\bar{n}\rho^0 \mathbf{Q}_0)^{n-c}}{(n-c)!}. \end{aligned}$$

By defining $m = n - c$ and using matrix exponential defined for a square matrix \mathbf{A} as

$$\exp(\mathbf{A}) := \sum_{i=0}^{\infty} \frac{\mathbf{A}^i}{i!},$$

we can further simplify the above formula as

$$\begin{aligned} \mathbf{\Lambda}^0 &= e^{-\bar{n}} \mathbf{e}_1 \sum_c \frac{(\bar{n}(1 - \rho^0))^c}{c!} \sum_m \frac{(\bar{n}\rho^0 \mathbf{Q}_0)^m}{m!} \\ &= e^{-\bar{n}} \mathbf{e}_1 \exp(\bar{n}(1 - \rho^0)) \exp(\bar{n}\rho^0 \mathbf{Q}_0) \\ &= \mathbf{e}_1 \exp(-\bar{n}\rho^0) \exp(\bar{n}\rho^0 \mathbf{Q}_0). \end{aligned} \quad (6)$$

The general result is as follows.

Theorem 2. *Given the number K of input symbols, the degree distribution $\{\Psi_d\}$, the rank distribution $\{h_r\}$ of the transfer matrix, the maximum degree D , the batch size M , and the number of batches being Poisson distributed with the mean equal to \bar{n} , we have for $t \geq 0$,*

$$\mathbf{\Lambda}^t = \exp(-\bar{n} \sum_s p_s^t) (\mathbf{\Lambda}^{t-1})^{\setminus 1} \exp(\bar{n} \sum_s p_s^t \mathbf{Q}_t), \quad (7)$$

where $(\mathbf{\Lambda}^{-1})^{\setminus 1} := \mathbf{e}_1$, while p_s^t and \mathbf{Q}_t are defined as in Theorem 1.

The computational complexity of evaluating the recursive formula in Theorem 2 is $\mathcal{O}(K\bar{n}M^2D/tol)$, where tol is the tolerable error in the computation. Assume the average number of symbols received is larger than the number of input symbols, i.e., $M\bar{n} \geq K$. Since the complexity of computing the quantities $\{p_s^t, \mathbf{Q}_t\}_{t \leq K, s \leq M}$ is $\mathcal{O}(K^2MD)$, to show the overall complexity it suffices to consider the cost of computing the action of matrix exponential $(\mathbf{\Lambda}^{t-1})^{\setminus 1} \exp(\bar{n} p^t \mathbf{Q}_t)$, which is usually faster than computing $\exp(\bar{n} p^t \mathbf{Q}_t)$, where $p^t = \sum_s p_s^t$. Using the algorithm in [10], the cost for computing an action of a matrix exponential e^A is $\mathcal{O}(\|A\|_1 \text{Mul}(A)/tol)$, where $\text{Mul}(A)$ is the cost for multiplying A with a vector. From the expression of $\bar{n} p^t \mathbf{Q}_t$, we have $\|\bar{n} p^t \mathbf{Q}_t\|_1 \leq \frac{\bar{n}D}{K}$ for $t > 0$ and $\|\bar{n} p^0 \mathbf{Q}_0\|_1 \leq \bar{n}$. Also, as in the discussion following Theorem 1, $\text{Mul}(\mathbf{Q}_t) = \mathcal{O}(KM)$. Therefore the complexity of calculating every $\mathbf{\Lambda}^t$ is

$$\mathcal{O}(K \times \bar{n}DM/tol + \bar{n}KM/tol) = \mathcal{O}(K\bar{n}MD/tol). \quad (8)$$

When precoding is applied, we know that $D = \mathcal{O}(M)$ is sufficient to achieve the maximum rate [2]. So the complexity in (8) becomes $\mathcal{O}(K\bar{n}M^3/\text{tol})$. When there is no precoding, we know the average degree must be of the order $\mathcal{O}(\log K)$ [5]. So the maximum degree D must also increase with K to get the optimal performance.

IV. INACTIVATION DECODING

BATS codes demonstrate nearly optimal asymptotic performance [2], but the achievable rates of BATS codes with finite input symbols are lower than the asymptotic optimal value since extra number of batches is required to guarantee the success of BP decoding. We call this number of extra batches received the *decoding overhead*. Though the decoding overhead is neglectable when K (the number of input symbols) is large, it reduces the rate of BATS codes significantly for small K . An effective method to reduce the decoding overhead is to use *inactivation decoding*, which was proposed for LT/Raptor codes [7], and can be similarly applied to BATS codes.

In the following of this section, after an introduction of inactivation decoding, we will modify Theorem 1 and 2 to calculate an important parameter of inactivation decoding.

In the BP decoding algorithm introduced in Section II-B, the decoding stops when there are no decodable input symbols. Though BP decoding stops, Gaussian elimination can still be used to decode the remaining input symbols (by combining the linear systems associated with the undecoded batches to a single linear system involving all undecoded input symbols). But the decoding complexity using Gaussian elimination is higher than BP decoding. Inactivation decoding combines BP decoding with Gaussian elimination in a more efficient way.

With inactivation decoding, when there are no decodable input symbols at time t , we instead randomly pick an undecoded symbol b_k and mark it as *inactive*. We substitute the inactive symbol b_k into the batches like a decoded symbol, except that b_k is an indeterminate. For example, if $k \in A_i^t$, each element of $\mathbf{Y}_i^{t+1} = \mathbf{Y}_i^t - b_k \mathbf{g}$ will be expressed as a linear polynomial of b_k . The decoding process is repeated until all input symbols are either decoded or inactive. The inactive input symbols can be recovered by solving a linear system of equations using Gaussian elimination, and then the values of inactive symbols are substituted into the decoded input symbols.

Inactivation decoding incurs extra computation cost that includes solving the inactive symbols using Gaussian elimination and substituting the values of the inactive symbols. Since both terms depend on the number of inactive symbols, knowing this number can help us to understand the tradeoff between computation cost and decoding overhead. In the following, we provide methods to compute the expected number of inactive symbols.

When the number of received batches is n , the expectation of the number of inactive symbols is given by

$$\sum_{t=0}^{K-1} \Pr\{R^t = 0 | N = n\}. \quad (9)$$

Theorem 1 can be modified to compute (9). For inactivation decoding, we define

$$\Gamma_{c,r|n}^t := \Pr\{C^t = c, R^t = r | N = n\}.$$

Let

$$\begin{aligned} \mathbf{\Gamma}_{c|n}^t &:= (\Gamma_{c,0|n}^t, \Gamma_{c,1|n}^t, \dots, \Gamma_{c,K-t|n}^t), \\ (\mathbf{\Gamma}_{c|n}^t)^{1+2} &:= (\Gamma_{c,0|n}^t + \Gamma_{c,1|n}^t, \Gamma_{c,2|n}^t, \dots, \Gamma_{c,K-t|n}^t), \end{aligned}$$

i.e., $(\mathbf{\Gamma}_{c|n}^t)^{1+2}$ is obtained by combining the first two components of $\mathbf{\Gamma}_{c|n}^t$.

Theorem 3. *Under the notations and assumption of Theorem 1, we have for inactivation decoding*

$$\mathbf{\Gamma}_{c|n}^0 = \text{Bi}(n, c; 1 - \rho^0) \mathbf{e}_1 \mathbf{Q}_0^{n-c},$$

and for $t > 0$,

$$\mathbf{\Gamma}_{c|n}^t = \sum_{c' \geq c} \text{Bi}(c', c; 1 - \rho^t) (\mathbf{\Gamma}_{c'|n}^{t-1})^{1+2} \mathbf{Q}_t^{c'-c}.$$

The expected number of inactive symbols for K input symbols and n batches is $\sum_{t=0}^{K-1} \sum_c \Gamma_{c,0|n}^t$.

When the number of received batches follows a Poisson distribution, the expected number of inactive symbols is given by

$$\sum_{t=0}^{K-1} \Pr\{R^t = 0\}. \quad (10)$$

We can modify Theorem 2 to compute (10). Define

$$\mathbf{\Gamma}^t := (\Gamma_0^t, \Gamma_1^t, \dots, \Gamma_{K-t}^t),$$

where

$$\Gamma_r^t = \Pr\{R^t = r\}.$$

Theorem 4. *Under the notations and assumption of Theorem 2, we have the following for inactivation decoding: For $t \geq 0$*

$$\mathbf{\Gamma}^t = \exp(-\bar{n} \sum_s p_s^t) (\mathbf{\Gamma}^{t-1})^{1+2} \exp(\bar{n} \sum_s p_s^t \mathbf{Q}_t),$$

where $(\mathbf{\Gamma}^{-1})^{1+2} := \mathbf{e}_1$.

The expected number of inactive symbols for Poisson distributed number of received batches is $\sum_{t=0}^{K-1} \Gamma_0^t$.

V. CONCLUDING REMARKS

The recursive formulas in this paper can be easily evaluated numerically using matrix operations. So without heavy simulation, we can directly calculate the error probability of BP decoder and the expected number of inactive symbols. Numerical results show that an asymptotically optimal degree distribution may not work well for small number of input symbols. For a given degree distribution, those recursive formulas can help us to determine the number of batches to received, and the distribution of stopping time also provides hints on how to tune the degree distribution to improve the performance. But more analytical tools are still desired to design optimal degree distribution for BATS codes with small number of input symbols.

ACKNOWLEDGEMENT

The work described in this paper was partially supported by a grant from University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-02/08).

REFERENCES

- [1] S. Yang and R. W. Yeung, "Coding for a network coded fountain," in *Proc. IEEE ISIT '11*, Saint Petersburg, Russia, Aug. 2011.
- [2] —, "Batched sparse codes," 2012. [Online]. Available: <http://arxiv.org/abs/1206.5365>
- [3] R. S.-Y. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, pp. 371–381, 2003.
- [4] M. Luby, "LT Codes," in *Proc. IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280.
- [5] A. Shokrollahi, "Raptor Code," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
- [6] R. Karp, M. Luby, and A. Shokrollahi, "Finite length analysis of LT-codes," in *Proc. IEEE ISIT '04*, 2004, p. 39.
- [7] A. Shokrollahi, S. Lassen, and R. Karp, "Systems and processes for decoding chain reaction codes through inactivation," U.S. Patent 6,856,263, Feb. 15, 2005.
- [8] E. Maneva and A. Shokrollahi, "New model for rigorous analysis of LT-codes," in *Proc. IEEE ISIT '06*, Jul. 2006, pp. 2677–2679.
- [9] T. C. Ng and S. Yang, "Finite-length analysis of BATS codes," 2013. [Online]. Available: http://iiis.tsinghua.edu.cn/~shenghao/pub/finite_length_full.pdf
- [10] A. H. Al-Mohy and N. J. Higham, "Computing the action of the matrix exponential, with an application to exponential integrators," *SIAM Journal on Scientific Computing*, vol. 33, no. 2, pp. 488–511, 2011.

APPENDIX

PROOF OF THE INITIAL STATUS OF THEOREM 1

We first calculate $\Lambda_{c,r|n}^0 = \Pr\{C^0 = c, R^0 = r | N = n\}$. The condition $N = n$ will be implied in the following of the proof. Let $\bar{\Theta}_s^t$ be the set of indices of batches that both the degree and the rank at time t equal to s . In other words, a batch with index in $\bar{\Theta}_s^t$, $s > 0$, is solvable and can decode s symbols. Let Θ^t be the set of indices of batches that are not in $\bar{\Theta}^t := \cup_{s=0}^M \bar{\Theta}_s^t$. We see that $R^t = |\cup_{i \in \bar{\Theta}^t} A_i^t|$, which is valid since $A_i^t = \emptyset$ for $i \in \Theta^t$. We see that $C^t = |\Theta^t|$.

When $t = 0$, the probability that a batch has degree s is Ψ_s and for a batch with degree s , it is decodable with probability h'_s (see (3)). Therefore, the probability that a batch is in $\bar{\Theta}_s^0$ is $\Psi_s h'_s$, i.e., for $i \leq n$ and $s \leq M$,

$$\Pr\{i \in \bar{\Theta}_s^0\} = p_s^0 := \Psi_s h'_s.$$

Hence,

$$\Pr\{i \in \bar{\Theta}^0\} = \sum_{s=0}^M p_s^0 := \rho^0.$$

Since all batches are independent, we obtain that

$$\Pr\{C^0 = k\} = \Pr\{|\Theta^0| = k\} = \text{Bi}(n, k; 1 - \rho^0). \quad (11)$$

Recall that \mathbf{Q}_0 is a $(K+1) \times (K+1)$ matrix defined as

$$\mathbf{Q}_0(i+1, j+1) = \sum_{k=j-i}^j \text{hyge}(K, i, k, i+k-j) \frac{p_k^0}{\rho^0}$$

if $j - M \leq i \leq j$, and $\mathbf{Q}_0(i+1, j+1) = 0$ otherwise.

Lemma 1. *We have*

$$(\Pr\{R^0 = j | C^0 = n - k\} : j = 0, \dots, K) = \mathbf{e}_1 \mathbf{Q}_0^k,$$

where $\mathbf{e}_1 = (1, 0, \dots, 0)$.

Proof: Fix n and k . If $k = 0$, then $\bar{\Theta}^0 = \emptyset$, and hence $\Pr\{R^0 = 0 | C^0 = n\} = 1$, i.e., the lemma with $k = 0$ is proved. In the following, we assume $k > 0$. We have

$$\begin{aligned} \Pr\{R^0 = j | C^0 = n - k\} \\ &= \Pr\{|\cup_{i \in \bar{\Theta}^0} A_i^0| = j | |\bar{\Theta}^0| = k\} \\ &= \Pr\{|\cup_{i=1}^k A_i^0| = j | \bar{\Theta}^0 = \{1, \dots, k\}\} \end{aligned}$$

where the second equality follows that since all batches are i.i.d., $\Pr\{|\cup_{i \in B} A_i^0| = j | \bar{\Theta}^0 = B\}$ is the same for any $B \subset \{1, 2, \dots, n\}$ with $|B| = k$.

Let

$$Q_{r|k}(j) = \Pr\{|\cup_{i=1}^r A_i^0| = j | \bar{\Theta}^0 = \{1, \dots, k\}\}.$$

We give a recursive formular to compute $Q_{r|k}(\cdot)$ for $r = 1, \dots, k$. Note that $Q_{k|k}(j) = \Pr\{R^0 = j | C^0 = n - k\}$. First,

$$\begin{aligned} Q_{1|k}(s) &= \Pr\{|A_1^0| = s | \bar{\Theta}^0 = \{1, \dots, k\}\} \\ &= \Pr\{1 \in \bar{\Theta}_s^0 | 1 \in \bar{\Theta}^0\} \\ &= \frac{p_s^0}{\rho^0}. \end{aligned}$$

Second, for $r > 1$,

$$\begin{aligned} Q_{r|k}(s) &= \Pr\{|\cup_{i=1}^r A_i^0| = s | \bar{\Theta}^0 = \{1, \dots, k\}\} \\ &= \sum_{i=0}^s Q_{i,s} Q_{r-1|k}(i), \end{aligned}$$

where

$$\begin{aligned} Q_{i,s} &= \Pr\{|\cup_{i=1}^r A_i^0| = s | |\cup_{i=1}^{r-1} A_i^0| = i, \bar{\Theta}^0 = \{1, \dots, k\}\} \\ &= \sum_{j=s-i}^s \Pr\{|\cup_{i=1}^r A_i^0| = s | |\cup_{i=1}^{r-1} A_i^0| = i, |A_r^0| = j\} \\ &\quad \times \Pr\{|A_r^0| = j | |\cup_{i=1}^{r-1} A_i^0| = i, \bar{\Theta}^0 = \{1, \dots, k\}\} \\ &= \sum_{j=s-i}^s \text{hyge}(K, i, j, i+j-s) \frac{p_j^0}{\rho^0}. \end{aligned}$$

The former part of each summand is a hypergeometric distribution; and the latter part is equal to $\Pr\{|A_r^0| = j | r \in \bar{\Theta}^0\}$, which can be obtained similar to $Q_{1|k}$.

Note that $Q_{i,s}$ does not depend on r . Let $\pi_{r|k} = (Q_{r|k}(0), Q_{r|k}(1), \dots, Q_{r|k}(K))$. We have

$$Q_{r|k} = \pi_{r-1|k} \mathbf{Q}_0 = \pi_{1|k} \mathbf{Q}_0^{r-1}.$$

Noting that $\pi_{1|k} = \frac{1}{\rho^0} (p_0^0, p_1^0, \dots, p_K^0)$ is the same as the first row of \mathbf{Q}_0 , the proof is completed. ■

By (11) and Lemma 1, we have

$$\begin{aligned} \Lambda_{c|n}^0 &= (\Pr\{C^0 = c, R^0 = j\} : j = 0, \dots, K) \\ &= \Pr\{C^0 = c\} (\Pr\{R^0 = j | C^0 = c\} : j = 0, \dots, K) \\ &= \text{Bi}(n, c; 1 - \rho^0) \mathbf{e}_1 \mathbf{Q}^{n-c}. \end{aligned}$$