Batched Sparse Codes

Shenghao Yang, *Member, IEEE* and Raymond W. Yeung, *Fellow, IEEE*

***Abstract*—Network coding can significantly improve the transmission rate of communication networks with packet loss compared with routing. However, using network coding usually incurs high computational and storage costs in the network devices and terminals. For example, some network coding schemes require the computational and/or storage capacities of an intermediate network node to increase linearly with the number of packets for transmission, making such schemes difficult to be implemented in a router-like device that has only constant computational and storage capacities. In this paper, we introduce BATched Sparse code (BATS code), which enables a digital fountain approach to resolve the above issue. BATS code is a coding scheme that consists of an outer code and an inner code. The outer code is a matrix generation of a fountain code. It works with the inner code that comprises random linear coding at the intermediate network nodes. BATS codes preserve such desirable properties of fountain codes as ratelessness and low encoding/decoding complexity. The computational and storage capacities of the intermediate network nodes required for applying BATS codes are independent of the number of packets for transmission. Almost capacity-achieving BATS code schemes are devised for unicast networks and certain multicast networks. For general multicast networks, under different optimization criteria, guaranteed decoding rates for the destination nodes can be obtained.**

***Index Terms*—Network coding, fountain codes, sparse graph codes, erasure network.**

I. Introduction

One fundamental task of communication networks is to distribute a bulk of digital data, called a *file*, from a source node to a set of destination nodes. We consider this file distribution problem, called *multicast*, in *packet networks*, in which data packets transmitted on the network links can be lost due to channel noise, congestion, faulty network hardware, and so on.

Existing network protocols, for example TCP, mostly use retransmission to guarantee reliable transmission of individual packets. Retransmission relies on feedback and is not scalable for multicast transmission. On the other hand, fountain codes, including LT codes [1], Raptor codes [2] and online codes [3], provide a good solution without relying on feedback for routing networks, where the intermediate nodes apply store-and-forward. When using fountain codes, the source node keeps transmitting coded packets generated by a fountain code encoder and a destination node can decode the original file after receiving n coded packets, where n typically is only slightly larger than the number of the input packets, regardless of which n packets are received. Fountain codes have the advantages of ratelessness, universality, and low encoding/decoding complexity. Taking Raptor codes as an example, both the encoding and decoding of a packet has constant complexity.

Routing, however, is not an optimal operation at the intermediate nodes for multicast. For a general network, the maximum multicast rate can be achieved only by *network coding* [4]. Network coding allows an intermediate node to generate and transmit new packets using the packets it has received. Linear network coding was proved to be sufficient for multicast communications [5], [6] and can be realized distributedly by random linear network coding [7]–[10].

Moreover, routing is not optimal in the presence of packet loss from the throughput point of view, *even for unicast*. For example, the routing capacity of the network in Fig. 1 is 0.64 packet per use.[1] If we allow decoding and encoding operations at the intermediate node and treat the network as a concatenation of two erasure channels, we can achieve the rate 0.8 packet per use by using erasure codes on both links.

The following network coding method has been proved to achieve the multicast capacity for networks with packet loss in a wide range of scenarios [11]–[13]. The source node transmits random linear combinations of the input packets and an intermediate node transmits random linear combinations of the packets it has received. Note that no erasure codes are required for each link though packet loss is allowed. Network coding itself plays the role of end-to-end erasure codes. A destination node can decode the input packets when it receives enough coded packets with linearly independent coding vectors. This scheme is referred to as the *baseline random linear network coding scheme (baseline RLNC scheme)*.

The baseline RLNC scheme has been implemented for small number of input packets, e.g., 32 [14], but the scheme is difficult to be implemented efficiently when the number of input packets is large due to the computational and storage complexities and the coefficient vector overhead. Consider transmitting K packets where each packet consists of T symbols in a finite field. The encoding of a packet at the source node takes $\mathcal{O}(TK)$ finite field operations, where T is given and K goes to infinity. A finite field operation refers

This paper was presented in part at the IEEE International Symposium on Information Theory, Saint Petersburg, Russia, August 2011.

This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003. This work was partially funded by a grant from the University Grants Committee of the Hong Kong Special Administrative Region (Project No. AoE/E-02/08) and Key Laboratory of Network Coding, Shenzhen, China (ZSDY20120619151314964).

S. Yang is with the Institute for Theoretical Computer Science, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China (email: shyang@tsinghua.edu.cn).

R. W. Yeung is with the Institute of Network Coding and the Department of Information Engineering, The Chinese University of Hong Kong, N.T., Hong Kong, and with the Key Laboratory of Network Coding Key Technology and Application and Shenzhen Research Institute, The Chinese University of Hong Kong, Shenzhen, China (e-mail: whyeung@ie.cuhk.edu.hk).

[1] One use of a network means the use of all network links at most once.

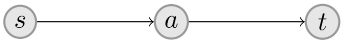


Fig. 1: Three-node network. Node s is the source node, node t is the destination node, and node a is the intermediate node that does not demand the file. Both links are capable of transmitting one packet per use and have a packet loss rate 0.2.

to the addition or multiplication of two field elements. An intermediate node needs to buffer all the packets it has received for network coding, so in the worst case, the storage cost at an intermediate node is K packets and the computational cost of encoding a packet at an intermediate node is $\mathcal{O}(TK)$ finite field operations. Decoding using Gaussian elimination costs on average $\mathcal{O}(K^2 + TK)$ finite field operations per packet. Though these complexities are polynomials in K, the baseline RLNC scheme is still difficult to implement for large K.

Coefficient vectors are used in random linear network coding to recover the linear transformation induced by network coding [7]. For transmitting K input packets, the baseline RLNC scheme requires that each packet includes a coefficient vector of K symbols. Hence, the coefficient vector overhead is K symbols per packet of T symbols. Network communication systems usually have a maximum value for T, e.g., several thousands of symbols. Therefore, for large values of K, the coding vector overhead is significant.

In this paper, we study file transmission through networks with packet loss using network coding. We hope to build network coding enabled devices with limited storage and computational capabilities. Accordingly, it is desirable for a network coding scheme to have i) low encoding complexity in the source node and low decoding complexity in the destination nodes, ii) constant computational complexity for encoding a packet at an intermediate node and constant buffer requirement at an intermediate node,[2] iii) small protocol and coefficient vector overhead, and iv) high transmission rate.

A. Related Works

Existing works mostly use one of the following two approaches to reduce the computation/storage complexity and the coefficient vector overhead. These approaches apply in the presence or absence of packet loss. The first approach is to use chunks, each of which is a subset of the input packets. A large file can be separated into a number of small chunks, and network coding is applied to each chunk [8]. The use of small chunks can effectively reduce the computational complexity and coefficient vector overhead. Therefore this idea is used in many implementations of random linear network coding in both wireline networks [15], [16] and wireless networks [14], [17]. However, the use of chunks introduces the scheduling issue of chunks since all or a large fraction of the chunks are required to be decoded individually. Specifically, sequential scheduling of chunks requires feedback and is not scalable for multicast, while random scheduling of chunks requires the intermediate nodes to cache all the chunks [18]. A detailed discussion of the scheduling issues of chunks can be found in [19].

Further, random scheduling of chunks becomes less efficient when a fraction of chunks have been decoded. To resolve this issue of the random scheduling of chunks, both precoding [18] and chunks with overlap [20]–[23] have been considered. Precoding allows the input packets to be recovered when only a fraction of all the chunks have been successfully decoded. Chunked codes use the already decoded chunks to help the decoding of the other chunks. Instead of overlapping, a more general approach to connect multiple chunks is to add some parity check constraints between chunks [24], [25]. For a given number of input packets, those chunk based codes mentioned above can only generate a fixed number of chunks.

The second approach is to use fountain codes for networks with coding at the intermediate nodes. The low complexity belief propagation decoding algorithm of LT/Raptor codes depends on a suitably chosen degree distribution. Since coding at the intermediate nodes changes the degrees of the coded packets, it is difficult to guarantee that the degrees of the received packets follow a specific distribution. Heuristic algorithms have been proposed for special network topologies (e.g., line networks [26], [27]) and special communication scenarios (e.g., peer-to-peer file sharing [28], [29]), but these solutions are difficult to be extended to general network settings and require the intermediate nodes to have a buffer size that increases linearly with the number of packets for transmission.

In addition to the above two approaches, there are techniques focusing on certain specific issues or scenarios. For example, an error correction code based approach is proposed by Jafari et al. to reduce the coefficient vector overhead [30]. This approach puts a limit on the number of packets that can be combined together, but does not take the decoding complexity into consideration. Link-by-link feedback can be used to reduce the storage at the intermediate nodes [31]–[33]. Jaggi *et al.* have proposed a binary permutation matrix based approach to reduce the complexity of the finite field operations in linear network coding [34].

B. Our Solution

In this paper, we propose an efficient linear network coding solution based on a new class of codes called *BATched Sparse (BATS) codes*, which extend fountain codes to incorporate random linear network coding. A BATS code consists of an inner code and an outer code over a finite field. The outer code is a matrix generalization of a fountain code, and hence rateless. The outer code encodes the file to be transmitted into *batches*, each containing M packets. When the batch size M is equal to 1, the outer code reduces to a fountain code. The inner code applies a linear transformation on each batch and is represented by the linear transfer matrices of the batches. The inner code is formed by *linear network coding* performed at the intermediate network nodes with the constraint that only packets belonging to the same batch can be combined inside the network. The property of the inner code preserves the degrees of the batches so that an efficient belief propagation (BP) decoding algorithm can be used to jointly decode the outer code and the inner code.

[2] A constant buffer requirement is desirable because one may not know ahead of time the size of the file to be transmitted.

BATS codes are suitable for any network that allows linear network coding at the intermediate nodes. BATS codes are robust against dynamical network topology and packet loss since the end-to-end operation remains linear. Moreover, BATS codes can operate with small finite fields. In contrast, most existing random linear network coding schemes require a large field size to guarantee a full rank for the transfer matrix. For BATS codes, the transfer matrices of the batches are allowed to have arbitrary rank deficiency. We demonstrate the applications of BATS codes in line networks, unicast networks, and some multicast networks.

BATS codes resolve the feedback issue of sequential scheduling of the chunk-based approach: Feedback is not required for sequential scheduling of batches due to the rateless property. BATS codes also resolve the degree distribution issue of the fountain-code-based approach since the inner code of BATS codes (induced by linear network coding) does not change the degrees of the batches. When applying BATS codes, the encoding of a packet by the outer code costs $\mathcal{O}(TM)$ finite field operations, where T and M are given and K goes to infinity. An intermediate node uses $\mathcal{O}(TM)$ finite field operations to recode a packet, and an intermediate node is required to buffer only $\mathcal{O}(M)$ packets for tree networks, including the three-node network in Fig. 1. BP decoding of BATS codes costs on average $\mathcal{O}(M^2 + TM)$ finite field operations per packet. The coefficient vector overhead of a BATS code is M symbols per packet. Note that all these requirements for BATS codes are constant in K, the total number of packets for transmission.

The (empirical) rank distribution of the transfer matrices of the batches plays an important role in BATS codes. The optimization of the outer code depends only on the rank distribution. We use density evolution to analyze the BP decoding process of BATS codes, and obtain a sufficient and a necessary condition for BP decoding recovering a given fraction of the input packets with high probability. For given rank distributions, a degree distribution for a BATS code can be obtained by solving an optimization problem induced by the sufficient condition.

For any inner code with rank distribution $(h_0, h_1, \ldots, h_M)$, we verify theoretically for certain cases and demonstrate numerically for general cases that the BATS code with BP decoding achieves rates very close to the expected rank $\sum_i ih_i$, the theoretical upper bound on the achievable rate of the code in packets per batch. For unicast erasure networks, BATS codes with BP decoding can achieve the min-cut capacity asymptotically when both M and T tend to infinity. This can be extended to multicast erasure networks when all the destination nodes have the same empirical rank distribution (which may be rare in practice).

When the destination nodes have different empirical rank distributions, we can optimize the degree distribution for various criteria, and obtain a set of guaranteed rate tuples for BP decoding. However, there is no guarantee in general that with this degree distribution, the rate of BP decoding at each destination node can achieve the expected rank for that node. For a given batch size, we can obtain numerically the percentage of the expected rank that is achievable for all possible rank distributions by using one degree distribution. For example, the percentage is at least 52.74 for batch size 16. When the possible empirical rank distributions are in a smaller set, a better degree distribution achieving higher rates can be found.

C. Organization of this Paper

BATS codes are formally introduced in Section II. The belief propagation decoding of BATS codes is analyzed in Section III. A necessary and a sufficient condition such that the BP decoding stops with a given fraction of the input packets recovered is obtained in Theorem 1, which is proved in Section IV. The degree distribution optimizations and the achievable rates of BATS codes are discussed in Section V. The degree distribution optimizations of BATS codes for multiple rank distributions is discussed in Section V-C. The necessary techniques for the design of the outer codes and decoding algorithms with good finite length performance are discussed in Section VI. Examples of how to use BATS codes in networks, as well as the design of the inner code of a BATS code, are given in Section VII. Concluding remarks are in Section VIII.

II. BATS Codes

In this section, we discuss the encoding and decoding of BATS codes. Consider encoding K input packets, each of which has T symbols in a finite field $\mathbb{F}$ with size q. A packet is denoted by a column vector in $\mathbb{F}^T$. The rank of a matrix $\mathbf{A}$ is denoted by $\mathrm{rk}(\mathbf{A})$. In the following discussion, we equate a set of packets to a matrix formed by juxtaposing the packets in this set. For example, we denote the set of the input packets by the matrix

$$\mathbf{B} = \begin{bmatrix} b_1, b_2, \cdots, b_K \end{bmatrix},$$

where b_i is the ith input packet. On the other hand, we also regard $\mathbf{B}$ as a set of packets, and so, with an abuse of notation, we also write $b_i \in \mathbf{B}$, $\mathbf{B}' \subset \mathbf{B}$, etc.

A. Encoding of Batches

Let us first describe the outer code of a BATS code, which generates coded packets in batches. (We also call the outer code itself the BATS code when the meaning is clear from the context.) A *batch* is a set of M coded packets generated from a subset of the K input packets. For $i = 1, 2, \ldots$, the ith batch $\mathbf{X}_i$ is generated from a subset $\mathbf{B}_i \subset \mathbf{B}$ of the input packets by the operation

$$\mathbf{X}_i = \mathbf{B}_i \mathbf{G}_i,$$

where $\mathbf{G}_i$, a matrix with M columns, is called the *generator matrix* of the ith batch. We call the packets in $\mathbf{B}_i$ the contributors of the ith batch. The formation of $\mathbf{B}_i$ is specified by a *degree distribution* $\Psi = (\Psi_0, \Psi_1, \cdots, \Psi_K)$ as follows: 1) sample the distribution Ψ which returns a *degree* d_i with probability Ψ_{d_i}; 2) uniformly at random choose d_i input packets to form $\mathbf{B}_i$. The design of Ψ is crucial for the performance of BATS code, which will be discussed in details in this paper.

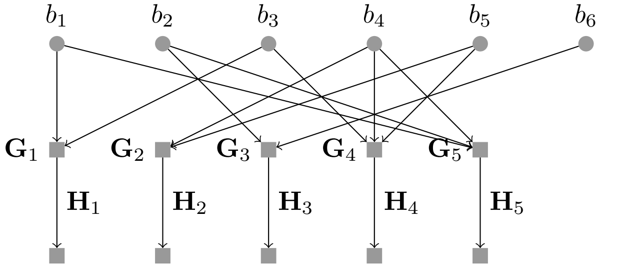


Fig. 2: Tanner graph for the inner and the outer code of a BATS code. Nodes in the first row are the variable nodes representing the input packets. Nodes in the second row are the check nodes representing the batches generated by the outer code. Nodes in the third row are the check nodes representing the batches processed by the inner code.

The generator matrix $\mathbf{G}_i$ has dimension $d_i \times M$ and can be generated randomly. Specifically, $\mathbf{G}_i$ is the instance of a $d_i \times M$ random matrix G_i, in which all the components are independently and uniformly chosen at random. Such a random matrix is also called a *totally random matrix*. We analyze BATS codes with random generator matrices in this paper. Random generator matrices do not only facilitate analysis but are also readily implementable. For example, $\mathbf{G}_i$, $i = 1, 2, \cdots$ can be generated by a pseudorandom number generator and can be recovered at the destination nodes by the same pseudorandom number generator.

The generator matrices can also be designed deterministically. For example, when $d_i \leq M$, we can pick $\mathbf{G}_i$ such that $\mathrm{rk}(\mathbf{G}_i) = d_i$. When $d_i > M$, we can use the generator matrix of an MDS code as the generator matrix of the ith batch. But we would not analyze the performance of such transfer matrices in this paper.

When $M = 1$ and the generator matrices have no zero components, the above batch encoding process becomes the encoding of LT codes. We are interested in this paper in the case $M > 1$. There are no limits on the number of batches that can be generated. So BATS code can be used as a rateless code.

The batch encoding process can be described by a Tanner graph. The Tanner graph has K *variable nodes*, where variable node i corresponds to the ith input packet b_i, and n *check nodes*, where check node j corresponds to the jth batch $\mathbf{X}_j$. Check node j is connected to variable node i if b_i is a contributor of $\mathbf{X}_j$. Associated with each check node j is the generator matrix $\mathbf{G}_j$. Fig. 2 illustrates an example of a Tanner graph for encoding batches.

B. Transmission of Batches

Now we turn to the inner code of a BATS code. The batches generated by the outer code are transmitted in a network employing network coding to multiple destination nodes. We assume that the end-to-end transformation of each batch is a linear operation. Fix a destination node. Let $\mathbf{H}_i$ be the transfer matrix of the ith batch and $\mathbf{Y}_i$ be the output (received) packets of the ith batch. We have

$$\mathbf{Y}_i = \mathbf{X}_i\mathbf{H}_i = \mathbf{B}_i\mathbf{G}_i\mathbf{H}_i. \tag{1}$$

The number of rows of $\mathbf{H}_i$ is M. The number of columns of $\mathbf{H}_i$ corresponds to the number of packets received for the ith batch, which may vary for different batches and is finite. We assume that $\mathbf{H}_i$ is known for decoding. In linear network coding, this knowledge can be obtained at the destination nodes through the coefficient vectors in the packet headers.

In other words, we assume that a received packet of a destination node cannot be the linear combinations of the packets of more than one batch from the same BATS code. To obtain such received packets, we may assume that an intermediate node can only apply network coding on packets of the same batch.[3] Packet loss and dynamical network topology are allowed during the network transmission. The benefits of applying network coding within batches includes

- The network coding complexity at an intermediate node is $\mathcal{O}(MT)$ finite field operations per packet, which does not depend on K.
- The coefficient vector overhead is bounded by M. When the packet length T is sufficiently larger than M, this overhead is negligible.

Moreover, since packets from different batches will not be encoded together, it is not necessary to keep all the batches in an intermediate node for the purpose of network coding.

We call the the network coding scheme at the intermediate network nodes the *inner code* of a BATS code. The transfer matrices of batches are determined jointly by the inner code and the network topology between the source node and the destination node. Under the principle that only packets of the same batch can be recoded, we have a lot of freedom in designing the inner code, including how to manage the buffer content, how to schedule the transmission of batches/packets, and how to use the feedback messages. The design of the inner code is closely related to the network topology. We will use several typical network topologies to demonstrate how to design the inner code such that the benefit of BATS codes is maximized (see Section VII).

The empirical rank distribution of the transfer matrices is an important parameter for the design of BATS codes. The empirical rank distribution determines the maximum achievable rate of the outer code and provides sufficient information to design nearly optimal outer codes. Since many network operations are random, e.g., random linear network coding, random packet loss pattern and network topology dynamics, the transfer matrices are also random matrices. Consider $\mathbf{H}_i$ as the instance of a random matrix H_i. The operation of the network on the batches in (1) can be modeled as a channel with input X_i and output $Y_i = X_iH_i$, $i = 1, 2, \ldots$, where the instance of H_i, regarded as the state of the channel, is known by the receiver. This channel model is called a *linear operator channel (LOC) with receiver side channel state information*. Similar channel models have been studied without the channel state information [35], [36]. Unless otherwise specified, receiver side channel state information is assumed for all the LOCs discussed in this paper. The LOC is not necessary to

[3] It is possible that network coding between packets of different batches is applied locally so that the coded packets of different batches in an intermediate node can be decoded directly at the nodes in the next hop.

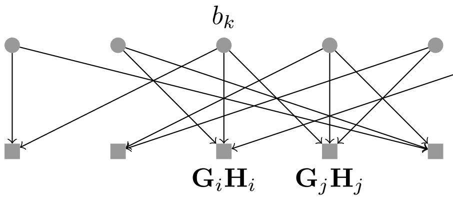


Fig. 3: A decoding graph. Nodes in the first row are the variable nodes representing the input packets. Nodes in the second row are the check nodes representing the batches.

be memoryless since H_i, $i = 1, 2, \ldots$ are not assumed to be independent. With receiver side channel state information, the capacity of the LOC can be easily characterized. Consider that

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} \mathrm{rk}(H_i)}{n} \xrightarrow{P} \bar{h}.$$

We can check that channel capacity of the above channel is upper bounded by $\bar{h}$ and the upper bound can be achieved by random linear codes [37]. As a channel code for the LOC, the maximum achievable rate of a BATS code is bounded by $\bar{h}$ for any inner code with average rank of the transfer matrices converging to $\bar{h}$. From the above analysis, we should design the inner code to maximize $\bar{h}$. Define the *design coding rate* of a BATS code as K/n. As we will show in Section V, for a given empirical rank distribution $(h_0, h_1, \ldots, h_M)$, we have a BATS code that can achieve a rate very close to $\sum_i ih_i$.

C. Belief Propagation Decoding

A destination node tries to decode the input packets using $\mathbf{Y}_i$ and the knowledge of $\mathbf{G}_i$ and $\mathbf{H}_i$ for $i = 1, 2, \ldots, n$. The decoding is equivalent to solving the system of linear equations formed by (1) for $i = 1, \ldots, n$. Solving the system of linear equations using Gaussian elimination has high computational cost when K is large. We propose a belief propagation (BP) based low complexity decoding algorithm for BATS codes. The BP decoding process is better described using the bipartite graph in Fig. 3, which is the same as the encoding graph in Fig. 2 except that the two stages of the encoding are combined together and the overall transfer matrix $\mathbf{G}_i\mathbf{H}_i$ is associated with each check node i.

A check node i is called decodable if $\mathrm{rk}(\mathbf{G}_i\mathbf{H}_i)$ is equal to the degree of the ith batch d_i. If check node i is decodable, then $\mathbf{B}_i$ can be recovered by solving the linear system of equations $\mathbf{Y}_i = \mathbf{B}_i\mathbf{G}_i\mathbf{H}_i$, which has a unique solution since $\mathrm{rk}(\mathbf{G}_i\mathbf{H}_i) = d_i$. After decoding the ith batch, we recover the d_i input packets in $\mathbf{B}_i$. Next, we substitute the values of these input packets in $\mathbf{B}_i$ in the undecoded batches. Consider that b_k is in $\mathbf{B}_i$. If variable node k has only one edge that connects with check node i, just remove variable node k. If variable node k also connects check node $j \neq i$, then we further reduce the degree of check node j by one and remove the row in $\mathbf{G}_j$ corresponding to variable node k. In the decoding graph, this is equivalent to first removing check node i and its neighboring variable nodes, and then for each removed variable node update its neighboring check nodes. We repeat this decoding-substitution procedure on the new graph until no more check nodes are decodable.

One of the main tasks of this paper is to understand the performance of BATS code under BP decoding, which will be discussed in Section III-V.

D. Computational Complexity

In the following computational complexity, the unit is a finite field operation. Suppose that T and M are given, and K and n are the variables that tend to infinity in the big O notation.

To generate a batch of degree d, we combine d packets together M times. So generating a batch with degree d costs $\mathcal{O}(TMd)$ finite field operations. Thus the encoding complexity of n batches is $\mathcal{O}(TM\sum_{i=1}^{n} d_i)$, which converges to $\mathcal{O}(TMn\bar{\Psi})$ finite field operations when n is large, where $\bar{\Psi} = \sum_d d\Psi_d$ is the average degree.

Let $k_i = \mathrm{rk}(\mathbf{H}_i)$ and let k_i' be the rank of $\mathbf{G}_i\mathbf{H}_i$ when check node i is decodable. It is clear that $k_i' \leq k_i \leq M$. By the definition of the decodability of a check node, k_i' is also the degree of check node i when it is decodable. Since the degree of a check node tends to decrease at each step of the decoding process, we have $k_i' \leq d_i$. The decoding processing involves two parts: the first part is the decoding of the decodable check nodes, which costs $\mathcal{O}(\sum_i k_i'^3 + T\sum_i k_i'^2)$ finite field operations; the second part is the updating of the decoding graph, which costs $\mathcal{O}(T\sum_i (d_i - k_i')M)$ finite field operations. So the total complexity is $\mathcal{O}(\sum_i k_i'^3 + T\sum_i k_i'^2 + T\sum_i (d_i - k_i')M)$, which can be simplified to $\mathcal{O}(nM^3 + TM\sum_i d_i)$. When n is large, the complexity converges to $\mathcal{O}(M^3 n + TMn\bar{\Psi})$ finite field operations. Usually, T and $\bar{\Psi}$ is larger than M and the second term is dominant.

III. Analysis of BP Decoding

In this section, we study that when the BP decoding stops, how much input packets have been decoded for a given degree distribution. Some existing methods for analyzing the BP decoding of erasure codes can be modified to analyze the BP decoding of BATS codes. In this paper, we adopt the differential equation approach by Wormald [38] that has been used by Luby et al. [39] to analyze Tornado codes (see also [40] for an analysis of LDPC codes over erasure channel).

Compared with the analysis of fountain codes, BATS codes have a relatively complex decoding criteria that involves both the degree and the rank value of a check node. In addition to the evolution of the degrees of the check nodes, the evolution of the ranks of the check nodes also needs to be tracked in the decoding analysis.

A. Random Decoding Graph

Consider a BATS code with K input symbols and n batches. Fix a degree distribution $\Psi = (\Psi_0, \Psi_1, \cdots, \Psi_D)$, where D is the maximum integer such that Ψ_D is nonzero. Assume that D does not change with K. The decoder observes a random graph as well as the associated generator and transfer matrices. The probability model of these objects is implied in the encoding of BATS codes described in the last section. Here we explicitly describe this model for the sake of the analysis.

Denote by $\mathrm{dg}_i, i = 1, \dots, n$ a sequence of i.i.d random variables each of which follows the distribution Ψ. Denote by $\mathcal{T}$ a Tanner graph with K variable nodes and n check nodes. The ith check node of $\mathcal{T}$ has degree dg_i. For a given degree d, a check node connects to d variable nodes chosen uniformly at random. Therefore the probability $\Pr\{\mathcal{T} = T | \{\mathrm{dg}_i = d_i\}\}$ can be fixed. The generator matrix G_i of check node i is a $\mathrm{dg}_i \times M$ totally random matrix, i.e., its components are uniformly i.i.d. Conditioning on a sequence of degrees, the Tanner graph and the generator matrices of the BATS code are obtained independently.

Let H_i be the transfer matrix associated with check node i. Note that $H_i, i = 1, \dots, n$ may not be independent. We do not need to make any assumption on the distribution of $H_i, i = 1, \dots, n$, except that the empirical distributions of the transfer matrix ranks converge in probability to a probability vector $h = (h_0, \dots, h_M)$. Specifically, for $k = 0, \dots, M$ let

$$\pi_k \triangleq \frac{|\{i : \mathrm{rk}(H_i) = k\}|}{n}.$$

Note that π_k depends on n. We assume that the convergence of the matrix ranks satisfies

$$|\pi_k - h_k| = \mathcal{O}(n^{-1/6}), \quad 0 \leq k \leq M, \tag{2}$$

with probability at least $1 - \gamma(n)$, where $\gamma(n) = o(1)$, i.e., there exists a constant c such that for all sufficiently large n,

$$\Pr\{|\pi_k - h_k| < cn^{-1/6},\ 0 \leq k \leq M\} \geq 1 - \gamma(n),$$

and

$$\lim_{n\to\infty} \gamma(n) = 0.$$

As an example of valid transfer matrices, $\{H_i\}$ are i.i.d. and $\mathrm{rk}(H_i)$ follows the distribution h. Hereafter, we call the probability vector h the rank distribution (of the transfer matrix). We assume that the transfer matrices are independent of the generation of batches.

Denote by $\mathrm{BATS}(K, n, \Psi, h)$ the random vector $(\{\mathrm{dg}_i, G_i, H_i\}_{i=1}^n, \mathcal{T})$. The decoder observes an instance of $\mathrm{BATS}(K, n, \Psi, h)$ with probability

$$\begin{aligned} &\Pr\left\{\{\mathrm{dg}_i = d_i, G_i = \mathbf{G}_i, H_i = \mathbf{H}_i\}_{i=1}^n, \mathcal{T} = T\right\} \\ &= \left(\prod_i \Psi_{d_i}\right) \Pr\left\{\mathcal{T} = T | \mathrm{dg}_i = d_i, i = 1, \dots, n\right\} \\ &\quad \times \left(\prod_i \Pr\{G_i = \mathbf{G}_i | \mathrm{dg}_i = d_i\}\right) \\ &\quad \times \Pr\left\{H_i = \mathbf{H}_i, i = 1, \dots, n\right\}. \end{aligned}$$

The decoding runs on an instance of $\mathrm{BATS}(K, n, \Psi, h)$ and we will look at the convergence of the decoding performance.

We will analyze the decoding performance of $\mathrm{BATS}(K, n, \Psi, h)$ with a random decoding strategy. We call $\mathrm{rk}(G_i H_i)$ the *rank* of check node i. In each decoding step, an edge (U, V) with degree equal to the rank is uniformly chosen, where U is a check node and V is a variable node. Since check node U has degree equal to the rank, variable node V is decodable. Variable node V, as well as all the edges connected to it, are removed from the decoding graph. For each check node connected to variable node V, three operations are applied: 1) the degree is reduced by 1; 2) the row in the generator matrix corresponding to the variable node V is removed; and 3) the rank is updated accordingly. The decoding process stops when there is no edge with degree equal to the rank. The following decoding analysis is based on this random decoding strategy. In the decoding process described in the last section, decoding a check node with degree equal to the rank can recover several variable nodes simultaneously. Note that for a given instance of the decoding graph, both strategies will reduce the decoding graph to the same residual graph when they stop (see the discussion in Appendix B).

B. Edge Perspective

An edge is said to be of degree d and rank r if it is connected to a check node with degree d and rank r. Let $R_{d,r}$ be the number of edges of degree d and rank r. Define the following two regions of the degree-rank pair:

$$\begin{aligned} \bar{\mathcal{F}} &\triangleq \{(d, r) : 1 \leq r \leq M, r \leq d \leq D\}, \\ \mathcal{F} &\triangleq \{(d, r) : 1 \leq r \leq M, r < d \leq D\}. \end{aligned}$$

We see that $\bar{\mathcal{F}} = \mathcal{F} \cup \{(r, r), r = 1, \dots, M\}$. A check node with rank zero does not help the decoding, so we do not include $(d, 0)$ in $\bar{\mathcal{F}}$ and $\mathcal{F}$. Define the *degree-rank distribution of the edges* as

$$\bar{R} \triangleq (R_{d,r}, (d, r) \in \bar{\mathcal{F}}).$$

Note that $R_{d,r}/d$ gives the number of nodes with degree d and rank r.

Using the property of totally random matrix and some counting techniques in projective space [41], [42], we have

$$\Pr\{\mathrm{rk}(G_i H_i) = r | \mathrm{dg}_i = d, \mathrm{rk}(H_i) = k\} = \frac{\zeta_r^d \zeta_r^k}{\zeta_r^r q^{(d-r)(k-r)}}, \tag{3}$$

where

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

Define

$$\zeta_r^{d,k} \triangleq \frac{\zeta_r^d \zeta_r^k}{\zeta_r^r q^{(d-r)(k-r)}}.$$

Let

$$\rho_{d,r} = d\Psi_d \sum_{k=r}^{M} \zeta_r^{d,k} h_k. \tag{4}$$

The value $n\rho_{d,r}$ is the expected number of edges of degree d and rank r in the decoding graph when the rank of a transfer matrix is chosen according to the probability vector h independently. The following lemma shows that $R_{d,r}/n$ converges in probability to $\rho_{d,r}$ as n goes to infinity.

Lemma 1. *With probability at least* $1 - (\gamma(n) + 2MD\exp(-2n^{2/3}))$,

$$\left|\frac{R_{d,r}}{n} - \rho_{d,r}\right| = \mathcal{O}(n^{-1/6}), \quad (d, r) \in \bar{\mathcal{F}}.$$

Proof: Consider the instances of decoding graphs with $\{\pi_k\}$ satisfying (2). By the assumption on $\{\pi_k\}$, this will decrease the bound by at most $\gamma(n)$. With an abuse of notation, we treat $\{\pi_k\}$ as an instance satisfying (2) in the following of this proof, i.e., the decoding graph has $n\pi_k$ check nodes with transfer matrix rank k.

By (3), the expected number of check nodes with degree d and rank r is

$$\sum_{k=r}^{M} n\pi_k\Psi_d\zeta_r^{d,k} = n\Psi_d\sum_{k=r}^{M}\pi_k\zeta_r^{d,k}.$$

Applying Hoeffding's inequality, with probability at least $1-2MD\exp(-2n^{2/3})$,

$$\left|\frac{R_{d,r}}{dn} - \Psi_d\sum_{k=r}^{M}\pi_k\zeta_r^{d,k}\right| < n^{-1/6}, \quad (d,r)\in\bar{\mathcal{F}}. \qquad (5)$$

Then,

$$\begin{aligned}&\left|\frac{R_{d,r}}{n}-\rho_{d,r}\right| \\ &= \left|\frac{R_{d,r}}{n} - d\Psi_d\sum_{k=r}^{M}\pi_k\zeta_r^{d,k} + d\Psi_d\sum_{k=r}^{M}\pi_k\zeta_r^{d,k} - \rho_{d,r}\right| \\ &\le \left|\frac{R_{d,r}}{n} - d\Psi_d\sum_{k=r}^{M}\pi_k\zeta_r^{d,k}\right| + d\Psi_d\sum_{k=r}^{M}|\pi_k-h_k|\zeta_r^{d,k}.\end{aligned}$$

By (5), under the condition in (2), we have

$$\left|\frac{R_{d,r}}{n}-\rho_{d,r}\right| = \mathcal{O}(n^{-1/6})$$

with probability at least $1-2MD\exp(-2n^{2/3})$.

The proof is completed by substracting the probability that $\{\pi_k\}$ does not satisfy (2). ■

C. Density Evolution

Consider the evolution of BATS(K,n,Ψ,h) during the decoding process. Time t starts at zero and increases by one for each variable node removed by the decoder. During the decoding, some of the random variables we defined in the previous two subsections will be analyzed as random processes. We denote by $\mathrm{dg}_i(t)$ the degree of the ith check node in the residual graph at time t, and $G_i(t)$ the corresponding generator matrix, where $\mathrm{dg}_i(0)=\mathrm{dg}_i$ and $G_i(0)=G_i$. For $(d,r)\in\bar{\mathcal{F}}$ let $R_{d,r}(t)$ denote the number of edges in the residual graph of degree d and rank r at time $t\ge 0$ with $R_{d,r}(0)=R_{d,r}$.

Upon removing a neighboring variable node of a check node with degree d and rank r, the degree of the check node will change to $d-1$. The rank of the check node may remain unchanged or may change to $r-1$. Regarding a degree-rank pair as a state, the state transition of a check node during the decoding process is illustrated in Fig. 4, where the transition probability is characterized in the following lemma.

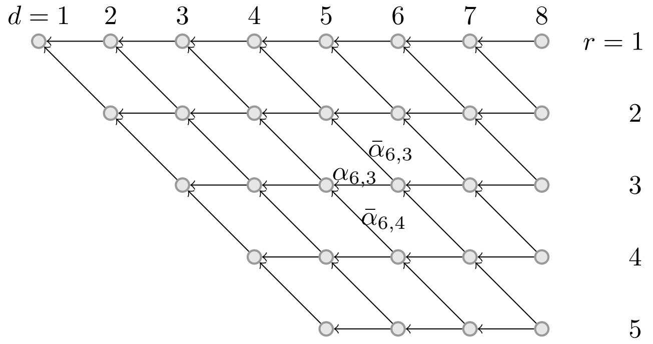


Fig. 4: State transition diagram for $M=5$ and $D=8$. Each node in the graph represent a degree-rank pair. In each step, if the check node connects to the decoded variable node, its state changes according to the direction of the outgoing edges of its current state. The label on an edge shows the probability that a direction is chosen.

Lemma 2. *For any check node i and any $(d,r)\in\bar{\mathcal{F}}$,*

$$\begin{aligned}&\Pr\{\mathrm{rk}(G_i(t+1)H_i)=r\big|\mathrm{rk}(G_i(t)H_i)=r,\\ &\quad \mathrm{dg}_i(t+1)=d-1,\mathrm{dg}_i(t)=d\} = \frac{1-q^{-d+r}}{1-q^{-d}} \triangleq \alpha_{d,r},\\ &\Pr\{\mathrm{rk}(G_i(t+1)H_i)=r-1\big|\mathrm{rk}(G_i(t)H_i)=r,\\ &\quad \mathrm{dg}_i(t+1)=d-1,\mathrm{dg}_i(t)=d\} = 1-\alpha_{d,r}\triangleq\bar{\alpha}_{d,r}.\end{aligned}$$

Proof: We omit the index i in the proof to simplify the notation. By (3) and the fact that G_i is totally random, we have for $k\ge r$,

$$\begin{aligned}&\Pr\{\mathrm{rk}(G(t)H)=r|\mathrm{rk}(G(t+1)H)=r,\\ &\quad \mathrm{dg}(t+1)=d-1,\mathrm{dg}(t)=d,\mathrm{rk}(H)=k\} = q^{r-k},\\ &\Pr\{\mathrm{rk}(G(t+1)H)=r|\mathrm{dg}(t+1)=d-1,\\ &\qquad \mathrm{dg}(t)=d,\mathrm{rk}(H)=k\} = \zeta_r^{d-1,k},\\ &\Pr\{\mathrm{rk}(G(t)H)=r|\mathrm{dg}(t+1)=d-1,\\ &\qquad \mathrm{dg}(t)=d,\mathrm{rk}(H)=k\} = \zeta_r^{d,k}.\end{aligned}$$

Hence,

$$\begin{aligned}&\Pr\{\mathrm{rk}(G(t+1)H)=r\big|\mathrm{rk}(G(t)H)=r,\\ &\quad \mathrm{dg}(t+1)=d-1,\mathrm{dg}(t)=d,\mathrm{rk}(H)=k\} = q^{r-k}\frac{\zeta_r^{d-1,k}}{\zeta_r^{d,k}}\\ &\qquad = \frac{1-q^{-d+r}}{1-q^{-d}}.\end{aligned}$$

The proof is completed by multiplying $\Pr\{\mathrm{rk}(H)=k|\mathrm{rk}(G(t)H)=r,\mathrm{dg}(t+1)=d-1,\mathrm{dg}(t)=d\}$ on both sides of the above equality and taking summation over all $k\ge r$. ■

Assume that the decoding process has not stopped. At time t, we have $K-t$ variable nodes left in the residual graph, and an edge with degree equal to the rank is uniformly chosen to be removed. Let

$$\bar{R}(t)\triangleq(R_{d,r}(t):(d,r)\in\bar{\mathcal{F}}).$$

The random process $\{\bar{R}(t)\}$ is a Markov chain, which suggests a straightforward approach to compute all the transition probabilities in the Markov chain. However, this approach leads to a complicated formula. Instead of taking this approach, we

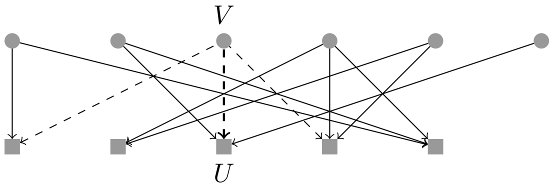


Fig. 5: A decoding graph. Edge (U, V) is to be removed at time t.

work out the expected change $R_{d,r}(t+1) - R_{d,r}(t)$ explicitly for all $t \geq 0$. Let

$$R_0(t) = \sum_{r=1}^{M} R_{r,r}(t).$$

We do not need to study the behavior of $R_{r,r}(t)$ for individual values of r since $R_0(t)$ is sufficient to determine when the decoding process stops. Specifically, the decoding process stops as soon as $R_0(t)$ becomes zero.

Lemma 3. *For any constant $c \in (0,1)$, as long as $t \leq cK$ and $R_0(t) > 0$, we have*

$$\begin{aligned}&\mathbb{E}[R_{d,r}(t+1) - R_{d,r}(t)|\bar{R}(t)] \\ &\quad = (\alpha_{d+1,r}R_{d+1,r}(t) + \bar{\alpha}_{d+1,r+1}R_{d+1,r+1}(t) - R_{d,r}(t)) \\ &\qquad \times \frac{d}{K-t},\ (d,r) \in \mathcal{F}, \end{aligned} \tag{6}$$

and

$$\begin{aligned}&\mathbb{E}[R_0(t+1) - R_0(t)|\bar{R}(t)] \\ &\quad = \frac{\sum_r r\alpha_{r+1,r}R_{r+1,r}(t)}{K-t} - \frac{R_0(t)}{K-t} - 1 + \mathcal{O}(1/K). \end{aligned} \tag{7}$$

Proof: Fix a time $t \geq 0$. With an abuse of notation, we treat $\bar{R}(0), \ldots, \bar{R}(t)$ as instances in the proof, i.e., the values of these random vectors are fixed. Let (U, V) be the edge chosen to be removed at time t, where V is the variable node and U is the check node, according to the random decoding algorithm described in Section III-A. Note that V is uniformly distributed among all variable nodes and U must be a check node with degree equal to the rank at time t. See the illustration in Fig. 5.

Let $N_{d,r}$ be the number of check nodes which has degree d and rank r at time t and has degree $d-1$ at time $t+1$. Let $N_{d,r}^{+}$ (resp. $N_{d,r}^{-}$) be the number of check nodes which has degree d and rank r at time t and has degree $d-1$ and rank r (resp. $r-1$) at time $t+1$. Clearly, $N_{d,r}^{+} + N_{d,r}^{-} = N_{d,r}$. The difference $R_{d,r}(t+1) - R_{d,r}(t)$ can then be expressed as

$$R_{d,r}(t+1) - R_{d,r}(t) = d(N_{d+1,r}^{+} + N_{d+1,r+1}^{-} - N_{d,r}). \tag{8}$$

The probability that a check node with degree d and rank r, $d > r$, connects to the variable node V at time t is $d/(K-t)$. Therefore, when $d > r$,

$$N_{d,r} \sim \text{Binom}\left(\frac{R_{d,r}(t)}{d}, \frac{d}{K-t}\right).$$

As we characterize in Lemma 2, for a check node with degree d and rank r connecting to the variable node V at time t, its rank will become r (resp. $r-1$) with probability $\alpha_{d,r}$ (resp. $\bar{\alpha}_{d,r}$) at time $t+1$. So when $d > r$,

$$\begin{aligned} N_{d,r}^{+} &\sim \text{Binom}\left(\frac{R_{d,r}(t)}{d}, \alpha_{d,r}\frac{d}{K-t}\right), \\ N_{d,r}^{-} &\sim \text{Binom}\left(\frac{R_{d,r}(t)}{d}, \bar{\alpha}_{d,r}\frac{d}{K-t}\right). \end{aligned}$$

The expectation in (6) is obtained by taking expectation on (8).

To verify (7), note that $N_{r,r}^{+} = 0$ and hence $N_{r,r}^{-} = N_{r,r}$. Then we have

$$\begin{aligned} R_0(t+1) - R_0(t) &= \sum_r (R_{r,r}(t+1) - R_{r,r}(t)) \\ &= \sum_r rN_{r+1,r}^{+} - \sum_r N_{r,r}. \end{aligned} \tag{9}$$

For a check node with degree r and rank r, with probability $r/R_0(t)$ it is U, and hence connects to V, otherwise, with probability $r/(K-t)$ it connects to V. Therefore,

$$N_{r,r} \sim \text{Binom}\left(\frac{R_{r,r}(t)}{r}, \frac{r}{R_0(t)} + \left(1 - \frac{r}{R_0(t)}\right)\frac{r}{K-t}\right).$$

Taking expectation on (9), we have

$$\begin{aligned} &\mathbb{E}[R_0(t+1) - R_0(t)|\bar{R}(t)] \\ &= \sum_r r\alpha_{r+1,r}\frac{R_{r+1,r}(t)}{K-t} \\ &\quad - \sum_r \left(\frac{R_{r,r}(t)}{R_0(t)} + \left(1 - \frac{r}{R_0(t)}\right)\frac{R_{r,r}(t)}{K-t}\right) \\ &= \sum_r r\alpha_{r+1,r}\frac{R_{r+1,r}(t)}{K-t} - \frac{R_0(t)}{K-t} - 1 + \sum_r \frac{r}{R_0(t)}\frac{R_{r,r}(t)}{K-t}. \end{aligned}$$

The expectation in (7) is obtained by noting that $\sum_r \frac{r}{R_0(t)}\frac{R_{r,r}(t)}{K-t} < \frac{M^2}{K(1-c)}$ since $t \leq cK$. ■

D. Sufficient and Necessary Conditions

We care about when $R_0(t)$ goes to zero for the first time. The evolution of $R_0(t)$ depends on that of $R_{d,r}(t)$, $(d,r) \in \mathcal{F}$. Consider the system of differential equations

$$\begin{aligned} \frac{d\rho_{d,r}(\tau)}{d\tau} &= \big(\alpha_{d+1,r}\rho_{d+1,r}(\tau) + \bar{\alpha}_{d+1,r+1}\rho_{d+1,r+1}(\tau) \\ &\qquad -\rho_{d,r}(\tau)\big)\frac{d}{\theta-\tau}, \quad (d,r) \in \mathcal{F}, \end{aligned} \tag{10}$$

$$\frac{d\rho_0(\tau)}{d\tau} = \frac{\sum_{r=1}^{D-1} r\alpha_{r+1,r}\rho_{r+1,r}(\tau) - \rho_0(\tau)}{\theta-\tau} - 1 \tag{11}$$

with initial values $\rho_{d,r}(0) = \rho_{d,r}$, $(d,r) \in \mathcal{F}$, and $\rho_0(0) = \sum_r \rho_{r,r}$, where $\theta = K/n$ is the design rate of the BATS code.

We can get some intuition about how the system of differential equations is obtained by substituting $R_{d,r}(t)$ and $R_0(t)$ with $n\rho_{d,r}(t/n)$ and $n\rho_0(t/n)$, respectively, in (6) and (7). Defining $\tau = t/n$ and letting $n \to \infty$, we obtain the system of differential equations in (10) and (11). The expectation operations are ignored because $\rho_{d,r}(\tau)$ and $\rho_0(\tau)$ are deterministic functions. Theorem 2 in Section IV makes the above intuition rigorous.

The system of differential equations in (10) and (11) is solved in Appendix C for $0 \leq \tau < \theta$. In particular, the solution for $\rho_0(\tau)$ is

$$\rho_0(\tau) = \left(1 - \frac{\tau}{\theta}\right)\left(\sum_{r=1}^{M}\sum_{d=r+1}^{D} d\Psi_d \hbar_r \mathrm{I}_{d-r,r}\left(\frac{\tau}{\theta}\right) + \sum_{r=1}^{M} r\Psi_r \sum_{s=r}^{M} \hbar_s + \theta \ln(1-\tau/\theta)\right), \tag{12}$$

where

$$\hbar_r = \hbar_r(h) \triangleq \sum_{i=r}^{M} \frac{\zeta_r^i}{q^{i-r}} h_i. \tag{13}$$

and

$$\mathrm{I}_{a,b}(x) \triangleq \sum_{j=a}^{a+b-1} \binom{a+b-1}{j} x^j (1-x)^{a+b-1-j}$$

is called the *regularized incomplete beta function*. For $\bar{\eta} \in (0,1)$, the following theorem shows that if $\rho_0(\tau) > 0$ for $\tau \in [0, \bar{\eta}\theta]$, then the decoding does not stop until $t > \bar{\eta}K$ with high probability, and $R_{d,r}(t)$ and $R_0(t)$ can be approximated by $n\rho_{d,r}(t/n)$ and $n\rho_0(t/n)$, respectively.

Theorem 1. *Consider a sequence of decoding graphs BATS(K, n, Ψ, h), $n = 1, 2, \ldots$ with fixed $\theta = K/n$, and the empirical rank distribution of transfer matrices $(\pi_0, \ldots, \pi_M)$ satisfying*

$$|\pi_i - h_i| = \mathcal{O}(n^{-1/6}), \quad 0 \leq i \leq M, \tag{14}$$

with probability at least $1 - \gamma(n)$, where $\gamma(n) = o(1)$. For $\bar{\eta} \in (0,1)$,

(i) *if $\rho_0(\tau) > 0$ for $\tau \in [0, \bar{\eta}\theta]$, then for sufficiently large K, with probability $1 - \mathcal{O}(n^{7/24}\exp(-n^{1/8})) - \gamma(n)$, the decoding terminates with at least $\bar{\eta}K$ variable nodes decoded, and*

$$|R_{d,r}(t) - n\rho_{d,r}(t/n)| = \mathcal{O}(n^{5/6}),\ (d,r) \in \mathcal{F}$$
$$|R_0(t) - n\rho_0(t/n)| = \mathcal{O}(n^{5/6})$$

uniformly for $t \in [0, \bar{\eta}K]$;

(ii) *if $\rho_0(\tau) < 0$ for some $\tau \in [0, \bar{\eta}\theta]$, then for sufficiently large K, with probability $1 - \mathcal{O}(n^{7/24}\exp(-n^{1/8})) - \gamma(n)$, the decoding terminates before $\bar{\eta}K$ variable nodes are decoded.*

When $M = 1$, the above theorem does not exactly reduce to the analysis of the BP decoding of Raptor codes since random generator matrices are used to generate batches. If we instead use the generator matrices with all ones when $M = 1$, the above theorem still holds with $\hbar$ replaced by h in the formula of $\rho_0(\tau)$ and becomes exactly the analysis of the BP decoding of Raptor codes. Note that when $M > 1$, using generator matrices with all ones does not give a good performance.

IV. Proof of Theorem 1

A. A General Theorem

The main technique to prove Theorem 1 is a general theorem by Wormald [38], [43] with a small modification. The statement of the next theorem follows that of [38, Theorem 5.1] with an extra initial condition. A similar version is provided in [40, Theorem C.28] with a deterministic boundedness condition.

We say a function $f(u_1, \ldots, u_j)$ satisfies a *Lipschitz condition* on $\mathcal{D} \subset \mathbb{R}^j$ if there exists a constant C_L such that

$$|f(u_1, \cdots, u_j) - f(v_1, \cdots, v_j)| \leq C_L \max_{1 \leq i \leq j} |u_i - v_i|$$

for all $(u_1, \cdots, u_j)$ and $(v_1, \cdots, v_j)$ in $\mathcal{D}$. We call C_L the Lipschitz constant for f. Note that $\max_{1 \leq i \leq j} |u_i - v_i|$ is the distance between $(u_1, \cdots, u_j)$ and $(v_1, \cdots, v_j)$ in the l^∞-norm.

Theorem 2. *Let $\mathcal{G}_0, \mathcal{G}_1, \ldots$ be a random process with a positive integer parameter n, and let $(Y_l(t))_{l=0}^L$ be a random vector determined by $\mathcal{G}_0, \ldots, \mathcal{G}_t$. For some constant C_0 and all l, $|Y_l(t)| < C_0 n$ for $t \geq 0$ and all n. Let $\mathcal{D}$ be some bounded connected open set containing the closure of*

$$\{(0, z_1, \ldots, z_L) : \exists n, \Pr\{Y_l(0) = z_l n, 1 \leq l \leq L\} \neq 0\}.$$

Define the stopping time *$T_\mathcal{D}$ to be the minimum t such that $(t/n, Y_1(t)/n, \ldots, Y_L(t)/n) \notin \mathcal{D}$. Assume the following conditions hold.*

(i) *(Boundedness) For some functions $\beta = \beta(n) \geq 1$ and $\gamma = \gamma(n)$, the probability that*

$$\max_l |Y_l(t+1) - Y_l(t)| \leq \beta,$$

is at least $1 - \gamma$ for $t < T_\mathcal{D}$.

(ii) *(Trend) For some function $\lambda_1 = \lambda_1(n) = o(1)$, if $t < T_\mathcal{D}$,*

$$\mathbb{E}[Y_l(t+1) - Y_l(t) | \mathcal{G}_1, \ldots, \mathcal{G}_t] = f_l\left(\frac{t}{n}, \left(\frac{Y_i(t)}{n}\right)_{i=0}^{L}\right) + \mathcal{O}(\lambda_1),$$

for $1 \leq l \leq L$.

(iii) *(Lipschitz) Each function f_l satisfies a Lipschitz condition on $\mathcal{D} \cap \{(t, z_1, \ldots, z_L), t \geq 0\}$ with the same Lipschitz constant C_L for each l.*

(iv) *(Initial condition) For some point $(0, z_1^0, \ldots, z_l^0) \in \mathcal{D}$,*

$$|Y_l(0)/n - z_l^0| \leq \sigma = o(1), 0 \leq l \leq L.$$

Then the following are true.

(a) *For $(0, (\hat{z}_l)_{l=1}^L) \in \mathcal{D}$, the system of differential equations*

$$\frac{dz_l(\tau)}{d\tau} = f_l(\tau, (z_{l'}(\tau))_{l'=1}^L), \quad l = 1, \ldots, L,$$

has a unique solution in $\mathcal{D}$ for $z_l : \mathbb{R} \to \mathbb{R}$ passing through $z_l(0) = \hat{z}_l$, $l = 1, \ldots, L$, and this solution extends to points arbitrarily close to the boundary of $\mathcal{D}$.

(b) *Let $\lambda > \max\{\sigma, \lambda_1 + C_0 n\gamma\}$ with $\lambda = o(1)$. There exists a sufficiently large constant C_1 such that when n is sufficiently large, with probability $1 - \mathcal{O}(n\gamma + \frac{\beta}{\lambda}\exp(-\frac{n\lambda^3}{\beta^3}))$,*

$$|Y_l(t) - nz_l(t/n)| = \mathcal{O}(\lambda n) \tag{15}$$

uniformly for $0 \leq t \leq \bar{\tau}n$ and for each l, where $\hat{z}_l = z_l^0$, and $\bar{\tau} = \bar{\tau}(n)$ is the supremum of those τ to which the solution of the system of differential equations in (a) can *be extended before reaching within l^∞-distance $C_1\lambda$ of the boundary of $\mathcal{D}$.*

Proof: The proof follows exactly the proof of [38, Theorem 5.1] except for the place where we need to handle the initial condition (iv). We only have to modify the definition of B_j (below (5.9) in [38]) in the original proof to

$$B_j = (n\lambda + \omega)\left(\left(1+\frac{B\omega}{n}\right)^j - 1\right) + B_0\left(1+\frac{B\omega}{n}\right)^j,$$

where $B_0 = n\lambda$. The induction in the original proof now begins by the fact that $|z_l(0) - Y_l(0)/n| \le \sigma < \mathcal{O}(\lambda)$. The other part of the proof stays the same as that of [38, Theorem 5.1]. ■

B. Completing the Proof

We first prove two technical lemmas. For BATS(K, n, Ψ, h), the degrees of the variable nodes are not independent but follow the same distribution. The following lemma shows that the degree of a variable node is not likely to be much larger than its expectation.

Lemma 4. *Let V be the degree of a variable node of BATS(K, n, Ψ, h). For any $\alpha > 0$,*

$$\Pr\{V > (1+\alpha)\bar{\Psi}/\theta\} < \left(\frac{e^\alpha}{(1+\alpha)^{(1+\alpha)}}\right)^{\bar{\Psi}/\theta},$$

where $\theta = K/n$.

Proof: Fix a variable node. Let X_i be the indicator random variable of the ith check node being the neighbor of the specific variable node. Then $V = \sum_i X_i$. We have $\mathbb{E}[V] = \sum_i \mathbb{E}[X_i] = \sum_i \sum_d \frac{d}{K}\Psi_d = \frac{n}{K}\bar{\Psi} = \frac{\bar{\Psi}}{\theta}$. Since X_i, $i = 1, \dots, n$, are mutually independent, the lemma is proved by applying the Chernoff bound. ■

The following lemma verifies the boundedness condition of Theorem 2.

Lemma 5. *When $\beta/D > \bar{\Psi}/\theta$, the probability that*

$$\max_{\iota \in \mathcal{F}\cup\{0\}} |R_\iota(t+1) - R_\iota(t)| \le \beta,$$

is at least

$$1 - \theta n \exp\left(-\frac{\beta}{D}(\ln(\beta/D) - \ln(\bar{\Psi}/\theta) - 1) - \frac{\bar{\Psi}}{\theta}\right).$$

Proof: Let V be the degree of the variable node to be removed at the beginning of time $t+1$. By (8), we have for $(d, r) \in \mathcal{F}$,

$$|R_{d,r}(t+1) - R_{d,r}(t)| \le DV,$$

and by (9), we have

$$|R_0(t+1) - R_0(t)| \le DV.$$

Hence when $\beta/D > \bar{\Psi}/\theta$,

$$\begin{aligned}
&\Pr\left\{\max_{\iota\in\mathcal{F}\cup\{0\}} |R_\iota(t+1) - R_\iota(t)| \le \beta\right\} \\
&\ge \Pr\{VD \le \beta\} \\
&\ge \Pr\{\text{degrees of all variable nodes at time zero} \le \beta/D\} \\
&> 1 - \theta n \exp\left(-\frac{\beta}{D}(\ln(\beta/D) - \ln(\bar{\Psi}/\theta) - 1) - \frac{\bar{\Psi}}{\theta}\right),
\end{aligned}$$

where the last inequality follows from Lemma 4 and the union bound. ■

Proof of Theorem 1: We consider in the proof only the instances of BATS(K, n, Ψ, h) satisfying

$$\left|\frac{R_{d,r}}{n} - \rho_{d,r}\right| = \mathcal{O}(n^{-1/6}), \quad (d, r) \in \bar{\mathcal{F}}. \tag{16}$$

By Lemma 1 this will decrease the probability bounds we will obtained by at most $\gamma(n) + 2MD\exp(-2n^{2/3})$.

Define the stopping time T_0 as the first time t such that $R_0(t) = 0$. By defining suitable functions $f_{d,r}, (d, r) \in \mathcal{F}$ and f_0 we can rewrite (6) and (7) as

$$\begin{aligned}
&\mathbb{E}[R_{d,r}(t+1) - R_{d,r}(t)|\bar{R}(t)] \\
&\quad = f_{d,r}\left(\frac{t}{n}, \left(\frac{R_0(t)}{n}\right), \left(\frac{R_{d',r'}(t)}{n}\right)_{(d',r')\in\mathcal{F}}\right), \ (d, r) \in \mathcal{F} \\
&\mathbb{E}[R_0(t+1) - R_0(t)|\bar{R}(t)] \\
&\quad = f_0\left(\frac{t}{n}, \left(\frac{R_0(t)}{n}\right), \left(\frac{R_{d',r'}(t)}{n}\right)_{(d',r')\in\mathcal{F}}\right) + \mathcal{O}\left(\frac{1}{n}\right),
\end{aligned}$$

for $t < T_0$. For $\iota \in \mathcal{F} \cup \{0\}$, define random variable $\hat{R}_\iota$ as $\hat{R}_\iota(0) = R_\iota(0)$ and for $t \ge 0$,

$$\begin{aligned}
&\hat{R}_\iota(t+1) \\
&= \begin{cases} R_\iota(t+1) & t < T_0 \\ \hat{R}_\iota(t) + f_\iota\left(\frac{t}{n}, \left(\frac{R_0(t)}{n}\right), \left(\frac{R_{\tilde{d},\tilde{r}}(t)}{n}\right)_{(\tilde{d},\tilde{r})\in\mathcal{F}}\right) & t \ge T_0. \end{cases}
\end{aligned}$$

Note that T_0 is also the first time that $\hat{R}_0(t)$ becomes zeros.

We now apply Theorem 2 with $(\hat{R}_0(t), (\hat{R}_{d,r}(t))_{(d,r)\in\mathcal{F}})$ in place of $(Y_l(t))_{l=1}^L$. The region $\mathcal{D}$ is defined as

$$\mathcal{D} = (-\eta, (1-\eta/2)\theta) \times (-M, M+\eta) \times (-\eta, d)^{|\mathcal{F}|}.$$

So 1) t/n is in the interval $(-\eta, (1-\eta/2)\theta)$; 2) $\hat{R}_0(t)/n$ is in the interval $(-M, M+\eta)$; and 3) $\hat{R}_{d,r}(t)/n$, $(d, r) \in \mathcal{F}$, is in the interval $(-\eta, d)$. As required, $\mathcal{D}$ is a bounded connected open set and containing all the possible initial state $(0, \hat{R}_0(0)/n, (\hat{R}_{d,r}(0)/n)_{(d,r)\in\mathcal{F}})$.

The conditions of Theorem 2 can readily be verified. When $t \ge T_0$, the change $|\hat{R}_\iota(t+1) - \hat{R}_\iota(t)|$ for $\iota \in \mathcal{F} \cup \{0\}$ is deterministic and upper bounded. When $t < T_0$, by Lemma 5 with $\beta = n^{1/8}$, the boundedness condition (i) holds with

$$\gamma = n\exp\left(-n^{1/8}(c_{1,3}\ln n - c_{1,1}) - c_{1,2}\right),$$

where $c_{1,1}$, $c_{1,2}$, and $c_{1,3}$ are only related to $\bar{\Psi}$ and θ. The trend condition (ii) is satisfied with $\lambda_1 = \mathcal{O}(1/n)$. By definition, it can be verified that f_ι, $\iota \in \mathcal{F} \cup \{0\}$ satisfy the Lipschitz condition (iii). The initial condition (iv) holds with $\sigma = \mathcal{O}(n^{-1/6})$.

Wormald's method leads us to consider the system of differential equations

$$\begin{aligned}
\frac{d\rho_{d,r}(\tau)}{d\tau} &= f_{d,r}(\tau, \rho_0(\tau), (\rho_{d',r'}(\tau))_{(d',r')\in\mathcal{F}}), \quad (d, r) \in \mathcal{F} \\
\frac{d\rho_0(\tau)}{d\tau} &= f_0(\tau, \rho_0(\tau), (\rho_{d',r'}(\tau))_{(d',r')\in\mathcal{F}})
\end{aligned}$$

with the initial condition $\rho_{d,r}(0) = \rho_{d,r}$, $(d, r) \in \mathcal{F}$, and $\rho_0(0) = \sum_r \rho_{r,r}$. The conclusion (a) of Theorem 2 shows the existence and uniqueness of the solution of the above system of differential equations. We solve the system of differential equations explicitly in Appendix C.

Let $\lambda = \mathcal{O}(n^{-1/6})$. By the conclusion (b) of Theorem 2, we know that for a sufficiently large constant C_1, with probability $1 - \mathcal{O}(n\gamma + \frac{\beta}{\lambda}\exp(-\frac{n\lambda^3}{\beta^3}))$,

$$\begin{aligned} |\hat{R}_{d,r}(t) - n\rho_{d,r}(t/n)| &= \mathcal{O}(n^{5/6}),\ (d,r) \in \mathcal{F}, \\ |\hat{R}_0(t) - n\rho_0(t/n)| &= \mathcal{O}(n^{5/6}) \end{aligned}$$

uniformly for $0 \le t \le \bar{\tau} n$, where $\bar{\tau}$ is defined in Theorem 2. Increase n if necessary so that $\frac{\beta}{\lambda}\exp(-\frac{n\lambda^3}{\beta^3}) = n^{7/24}\exp(-n^{-1/8}) > n\gamma$ and $C_1\lambda < \frac{\eta}{2}\theta$, which implies $\bar{\tau} \ge (1-\eta)\theta$. So there exists constants c_0 and c_0' such that the event

$$E_0 = \{|\hat{R}_0(t)/n - \rho_0(t/n)| \le c_0 n^{-1/6},\ 0 \le t \le (1-\eta)K\}$$

holds with probability at least $1 - c_0' n^{7/24}\exp(-n^{-1/8})$.

Now we consider the two cases in the theorem to prove. (i) If $\rho_0(\tau) > 0$ for $\tau \in [0, (1-\eta)\theta]$, then there exists $\epsilon > 0$ such that $\rho_0(\tau) \ge \epsilon$ for $\tau \in [0, (1-\eta)\theta]$. Increase n if necessary so that $c_0 n^{-1/6} < \epsilon$. Then, we have

$$\begin{aligned} \Pr\{T_0 > (1-\eta)K\} &= \Pr\{\hat{R}(t) > 0, 0 \le t \le (1-\eta)K\} \\ &\ge \Pr\{E_0\} \qquad\qquad (17) \\ &\ge 1 - c_0' n^{7/24}\exp(-n^{-1/8}), \end{aligned}$$

where (17) follows that under the condition E_0, for all $t \in [0, (1-\eta)K]$, $\hat{R}_0(t)/n \ge \rho_0(t/n) - c_0 n^{-1/6} > 0$. Since $\hat{R}_\iota = R_\iota$, $\iota \in \mathcal{F} \cup \{0\}$, when $t < T_0$, the first part of the theorem is proved.

(ii) Consider $\rho_0(\tau_0) < 0$ for $\tau_0 \in [0, (1-\eta)\theta]$. There exists $\epsilon > 0$ such that $\rho_0(\tau) \le -\epsilon$ for all $\tau \in [\tau_0 - \epsilon, \tau_0 + \epsilon] \cap [0, (1-\eta)\theta]$. Increase n if necessary so that $c_0 n^{-1/6} < \epsilon$ and $n\epsilon > 1$. Then, we have

$$\begin{aligned} &\Pr\{T_0 \le (1-\eta)K\} \\ &\quad = \Pr\{\hat{R}_0(t) < 0,\ \text{for some } t \in [0, (1-\eta)K]\} \\ &\quad \ge \Pr\{E_0\} \qquad\qquad (18) \\ &\quad \ge 1 - c_0' n^{7/24}\exp(-n^{-1/8}), \end{aligned}$$

where (18) can be shown as follows. Since $n\epsilon > 1$, there exists t_0 such that $t_0/n \in [\tau_0 - \epsilon, \tau_0 + \epsilon] \cap [0, (1-\eta)\theta]$. Hence, under the condition E_0, $\hat{R}_0(t_0)/n \le c_0 n^{-1/6} + \rho_0(t_0/n) < 0$.

The proof of the theorem is completed by subtracting the probability that (16) does not hold. ■

V. Degree Distribution Optimizations

Following the discussion of BATS codes in Section II, we now study the design of degree distributions based on the sufficient condition in Theorem 1.

A. Optimization for single rank distribution

We start with a single destination node with rank distribution $h = (h_0, h_1, \ldots, h_M)$. For $\bar{\eta} \in [0, 1)$, we say a rate R is *$\bar{\eta}$-achievable by BATS codes using BP decoding* if for every $\epsilon > 0$ and every sufficiently large K, there exists a BATS code with K input packets such that for $n \le \bar{\eta}K/(R - \epsilon)$ received batches, the BP decoding recovers at least $\bar{\eta}K$ input packets with probability at least $1 - \epsilon$.

For a given rank distribution h, the following optimization problem maximizes the $\bar{\eta}$-achievable rate with the degree distribution as the variable:

$$\begin{aligned} &\max \theta \\ &\quad \text{s.t. } \Omega(x; \hbar(h), \Psi) + \theta \ln(1-x) \ge 0, \quad 0 \le x \le \bar{\eta}, \\ &\qquad\quad \sum_d \Psi_d = 1 \ \text{ and } \ \Psi_d \ge 0,\ d = 1, \cdots, D, \end{aligned} \qquad \text{(P1)}$$

where

$$\Omega(x; \hbar, \Psi) \triangleq \sum_{r=1}^{M} \hbar_r \sum_{d=r+1}^{D} d\Psi_d \mathrm{I}_{d-r,r}(x) + \sum_{r=1}^{M} r\Psi_r \sum_{s=r}^{M} \hbar_s, \qquad (19)$$

When the context is clear, we also write $\Omega(x; \Psi)$, $\Omega(x; \hbar)$ or $\Omega(x)$ to simplify the notation.

To compare with the degree distribution optimization of the Raptor codes, we see that when $M = 1$, $\Omega(x; \hbar, \Psi) = \hbar_1 \sum_{d=1}^{D} d\Psi_d x^{d-1}$. The optimization P1 reduces to the optimization of the Raptor codes if $\hbar_1$ is replaced by h_1. The reason why we have $\hbar$ instead of h in $\Omega(x; \hbar, \Psi)$ is that random generator matrices are used for the batches. When $M = 1$, if generator matrices with all 1's are used, $\Omega(x; h, \Psi) = h_1 \sum_{d=1}^{D} d\Psi_d x^{d-1}$ will be obtained instead in the analysis, and the optimization (P1) reduces exactly to the optimization of the Raptor codes.

Lemma 6. *Let $\hat{\theta}$ be the optimal value in* (P1). *When the empirical rank distribution of the transfer matrices converges in probability to $h = (h_0, \ldots, h_M)$ (in the sense of* (14)*), any rate less than or equal to $\bar{\eta}\hat{\theta}$ is $\bar{\eta}$-achievable by BATS codes using BP decoding.*

Proof: By (12), we can write

$$\rho_0(\tau) = (1 - \tau/\theta)\left(\Omega(\tau/\theta) + \theta \ln(1 - \tau/\theta)\right). \qquad (20)$$

To show that $\bar{\eta}\hat{\theta}$ is $\bar{\eta}$-achievable, by Theorem 1, we only need to show that there exists a degree distribution such that for any $\epsilon > 0$,

$$\Omega(x) + (\hat{\theta} - \epsilon)\ln(1-x) > 0, \quad 0 \le x \le \bar{\eta}. \qquad (21)$$

Since the proof for $\bar{\eta} = 0$ is trivial, assume that $\bar{\eta} > 0$. For the degree distribution Ψ that achieves $\hat{\theta}$ in (P1), we have

$$\Omega(x; \Psi) + \hat{\theta}\ln(1-x) \ge 0, \quad 0 \le x \le \bar{\eta}.$$

Multiplying by $\frac{\hat{\theta}-\epsilon}{\hat{\theta}}$ on both sides, we have

$$\frac{\hat{\theta} - \epsilon}{\hat{\theta}}\Omega(x; \Psi) + (\hat{\theta} - \epsilon)\ln(1-x) \ge 0, \quad 0 \le x \le \bar{\eta}. \qquad (22)$$

Since $\Omega(x; \Psi) > 0$ for $x > 0$, (22) implies that Ψ satisfies (21) except possibly for $x = 0$. If $\Omega(0; \Psi) > 0$, which implies Ψ satisfies (21), we are done.

In the following, we consider the case with $\Omega(0; \Psi) = 0$. Checking the definition of Ω in (19), we have

$$\Omega(0; \Psi) = \sum_{r=1}^{M} r\Psi_r \sum_{s=r}^{M} \hbar_s.$$

Let r^* be the largest integer r such that $h_r > 0$. Since $\Omega(0; \Psi) = 0$, we know that $\sum_{d \leq r^*} \Psi_d = 0$. Define a new degree distribution Ψ' by $\Psi'_d = \Psi_d \frac{\hat{\theta} - \epsilon}{\hat{\theta}}$ for $d > r^*$ and $\Psi'_d = \Delta$ for $d \leq r^*$, where $\Delta > 0$ can be determined by the constraint $\sum_d \Psi'_d = 1$. Then we can check that Ψ' satisfies (21). ■

The converse of Lemma 6 is that "a rate larger than $\bar{\eta}\hat{\theta}$ is not $\bar{\eta}$-achievable". Intuitively, for any $\epsilon > 0$, we cannot have a degree distribution such that

$$\Omega(x) + (\hat{\theta} + \epsilon) \ln(1 - x) \geq 0, \quad 0 \leq x \leq \bar{\eta},$$

where $\hat{\theta}$ is the maxima of (P1). Thus, with $\hat{\theta} + \epsilon$ in place of θ in the expression of ρ_0 in (20), for any degree distribution we have $\rho_0(\tau) < 0$ for some $\tau \in [0, \bar{\eta}(\hat{\theta} + \epsilon)]$. By Theorem 1, for any degree distribution there exists K_0 such that when the number of input packets $K \geq K_0$, with probability approaching 1 the BATS code cannot recover $\bar{\eta}K$ input packets. To prove this converse, however, we need a uniform bound K_0 for all degree distributions such that the second part of Theorem 1 holds, which is difficult to obtain. Instead, we demonstrate that $\hat{\theta}$ is close to the capacity of the underlying linear operator channel (cf. Section II-B).

Before analyzing the achievable rate, we determine the maximum degree D, which affects the encoding/decoding complexity. The next theorem shows that it is optimal to choose $D = \lceil M/\eta \rceil - 1$, where $\eta = 1 - \bar{\eta}$.

Theorem 3. *Using $D > \lceil M/\eta \rceil - 1$ does not give a better optimal value in* (P1)*, where $\eta = 1 - \bar{\eta}$.*

Proof: Consider an integer Δ such that $\eta \geq \frac{M}{\Delta+1}$. Let Ψ be a degree distribution with $\sum_{d > \Delta} \Psi_d > 0$. Construct a new degree distribution $\tilde{\Psi}$ with

$$\tilde{\Psi}_d = \begin{cases} \Psi_d & \text{if } d < \Delta, \\ \sum_{k \geq \Delta} \Psi_k & \text{if } d = \Delta, \\ 0 & \text{if } d > \Delta. \end{cases}$$

Write

$$\begin{aligned} &\Omega(x; \tilde{\Psi}) - \Omega(x; \Psi) \\ &\quad = \sum_{d=\Delta+1}^{\infty} \Psi_d \sum_{r=1}^{M} \hbar_r (\Delta \mathrm{I}_{\Delta-r,r}(x) - d\mathrm{I}_{d-r,r}(x)). \end{aligned}$$

For $d \geq \Delta + 1$,

$$\frac{r-1}{d-r} \leq \frac{M-1}{d-M} < \frac{M}{\Delta - M + 1} \leq \frac{\eta}{1-\eta}.$$

So we can apply the properties of the incomplete beta function (Lemma 9 in Appendix A) to show that, for any x with $0 < x \leq 1 - \eta$,

$$\begin{aligned} \frac{d\mathrm{I}_{d-r,r}(x)}{(d-1)\mathrm{I}_{d-1-r,r}(x)} &< \frac{d}{d-1}\left(1 - \frac{\eta}{r}\right) \\ &\leq \frac{d}{d-1}\left(1 - \frac{\eta}{M}\right) \\ &\leq \frac{\Delta+1}{\Delta}\left(1 - \frac{1}{\Delta+1}\right) \\ &= 1, \end{aligned}$$

which gives $\Omega(x; \tilde{\Psi}) > \Omega(x; \Psi)$ for $0 < x \leq 1 - \eta$. This means that using only degree distributions Ψ with $\sum_{d > \Delta} \Psi_d = 0$, we can get the same optimal value as using all degree distributions. Therefore, it is sufficient to take the maximum degree $D \leq \min_{\eta \geq \frac{M}{\Delta+1}} \Delta = \lceil M/\eta \rceil - 1$. ■

To solve (P1) numerically, we can relax it as a linear programming by only considering x in a linearly sampled set of values between 0 and $1 - \eta$. Let $x_i = (1 - \eta)\frac{i}{N}$ for some integer N. We relax (P1) by considering only $x = x_i$, $i = 1, \ldots, N$, where N can be chosen to be 100 or even smaller.

For many cases, we can directly use the degree distribution Ψ obtained by solving (P1). But it is possible that $\Omega(0; \Psi) = 0$, so that the degree distribution Ψ does not guarantee that decoding can start. We can then modify Ψ as we do in the proof of Lemma 6 by increasing the probability masses Ψ_d, $d \leq M$ by a small amount to make sure that decoding can start.

B. Achievable Rates

The first upper bound on the optimal value $\hat{\theta}$ of (P1) is given by the capacity of LOCs with receiver side channel state information. When the empirical rank distribution of the transfer matrices converging to $h = (h_0, \ldots, h_M)$, the capacity is $\sum_r rh_r$ packets per batch. The BP decoding algorithm recovers at least a fraction $\bar{\eta}$ of all the input packets with high probability. So asymptotically BATS codes under BP decoding can recover at least a fraction $\bar{\eta}\hat{\theta}$ of the input packets. Thus, we have $\bar{\eta}\hat{\theta} \leq \sum_r rh_r$.

A tighter upper bound can be obtained by analyzing (P1) directly. Rewrite

$$\Omega(x; \hbar, \Psi) = \sum_{r=1}^{M} \hbar_r S_r(x; \Psi), \tag{23}$$

where

$$S_r(x; \Psi) = S_r(x) \triangleq \sum_{d=r+1}^{D} d\Psi_d \mathrm{I}_{d-r,r}(x) + \sum_{d=1}^{r} d\Psi_d. \tag{24}$$

This form of $\Omega(x; \hbar, \Psi)$ will be used in the subsequent proofs.

Theorem 4. *The optimal value $\hat{\theta}$ of* (P1) *satisfies*

$$\bar{\eta}\hat{\theta} \leq \sum_{r=1}^{M} r\hbar_r.$$

Proof: Fix a degree distribution that achieves the optimal value of (P1). Using (35) in Appendix A, we have

$$\begin{aligned} \int_0^1 S_r(x)dx &= \sum_{d=r+1}^{D} d\Psi_d \int_0^1 \mathrm{I}_{d-r,r}(x)dx + \sum_{d=1}^{r} d\Psi_d \\ &= \sum_{d=r+1}^{D} r\Psi_d + \sum_{d=1}^{r} d\Psi_d \\ &\leq r \sum_{d=1}^{D} \Psi_d \\ &= r. \end{aligned}$$

Hence,

$$\int_0^1 \Omega(x)dx = \int_0^1 \sum_{r=1}^{M} \hbar_r S_r(x)dx \leq \sum_{r=1}^{M} r\hbar_r. \tag{25}$$

Since $\Omega(x)$ is an increasing function,

$$\int_{1-\eta}^{1} \Omega(x)dx \geq \eta\Omega(1-\eta) \geq -\eta\hat{\theta}\ln\eta. \tag{26}$$

Since $\Omega(x) + \hat{\theta}\ln(1-x) \geq 0$ for $0 < x \leq 1-\eta$,

$$\begin{aligned}&\int_0^{1-\eta} \Omega(x)dx - \hat{\theta}(\eta\ln\eta + 1 - \eta) \\ &\quad = \int_0^{1-\eta} \Omega(x)dx + \hat{\theta}\int_0^{1-\eta} \ln(1-x)dx \geq 0. \end{aligned}\tag{27}$$

Therefore, by (25)-(27), we have

$$\begin{aligned}\sum_{r=1}^{M} r\hbar_r &\geq \int_0^1 \Omega(x)dx \\ &= \int_0^{1-\eta} \Omega(x)dx + \int_{1-\eta}^{1} \Omega(x)dx \\ &\geq \hat{\theta}(\eta\ln\eta + 1 - \eta) - \eta\hat{\theta}\ln\eta \\ &= \hat{\theta}(1-\eta).\end{aligned}$$

The proof is completed. ■

By (49) in Appendix C, $\sum_{k=r}^{M} \hbar_k = \sum_{i=r}^{M} h_i\zeta_r^i \leq \sum_{k=r}^{M} h_k$, where the last inequality follows from $\zeta_r^i < 1$. Hence,

$$\begin{aligned}\sum_r r\hbar_r &= \sum_{r=1}^{M}\sum_{k=r}^{M} \hbar_k \\ &\leq \sum_{r=1}^{M}\sum_{k=r}^{M} h_k \\ &= \sum_r rh_r.\end{aligned}$$

Therefore, Theorem 4 gives a strictly better upper bound than $\sum_r rh_r$. When $q \to \infty$, $\sum_r r\hbar_r \to \sum_r rh_r$. Even for small finite fields, $\sum_r r\hbar_r$ and $\sum_r rh_r$ are very close.

We prove for a special case and demonstrate by simulation for general cases that the optimal value $\hat{\theta}$ of (P1) is very close to $\sum_r r\hbar_r$.

Theorem 5. *For $D = \lceil M/\eta \rceil - 1$, the optimal value $\hat{\theta}$ of* (P1) *satisfies*

$$\hat{\theta} \geq \max_{r=1,2,\cdots,M} r\sum_{i=r}^{M} \hbar_i.$$

Proof: Define a degree distribution Ψ^r by

$$\Psi_d^r = \begin{cases} 0 & \text{if } d \leq r, \\ \frac{r}{d(d-1)} & \text{if } d = r+1, \cdots, D-1, \\ \frac{r}{D-1} & \text{if } d = D. \end{cases} \tag{28}$$

Recall the definition of $S_r(x;\Psi)$ in (24). For $M \geq r' \geq r$, we will show that

$$S_{r'}(x;\Psi^r) + r\ln(1-x) > 0, \quad 0 \leq x \leq 1-\eta. \tag{29}$$

By Lemma 10 in Appendix A,

$$-r\ln(1-x) = r\sum_{d=r'+1}^{\infty} \frac{1}{d-1}\mathrm{I}_{d-r',r'}(x).$$

By (24) and (28),

$$\begin{aligned}&S_{r'}(x;\Psi^r) + r\ln(1-x) \\ &\quad \geq \sum_{d=r'+1}^{D} d\Psi_d^r \mathrm{I}_{d-r',r'}(x) - r\sum_{d=r'+1}^{\infty} \frac{1}{d-1}\mathrm{I}_{d-r',r'}(x) \\ &\quad = r\frac{D}{D-1}\mathrm{I}_{D-r',r'}(x) - r\sum_{d=D}^{\infty} \frac{1}{d-1}\mathrm{I}_{d-r',r'}(x) \\ &\quad = r\mathrm{I}_{D-r',r'}(x) - r\sum_{d=D+1}^{\infty} \frac{1}{d-1}\mathrm{I}_{d-r',r'}(x).\end{aligned}$$

To show $\mathrm{I}_{D-r',r'}(x) > \sum_{d=D+1}^{\infty} \frac{1}{d-1}\mathrm{I}_{d-r',r'}(x)$ for $x \in [0, 1-\eta]$, we prove that

$$\sum_{d=D+1}^{\infty} \frac{1}{d-1}\frac{\mathrm{I}_{d-r',r'}(x)}{\mathrm{I}_{D-r',r'}(x)} < 1 \quad \text{for } x \in [0, 1-\eta]. \tag{30}$$

By Lemma 8 in Appendix A, $\frac{\mathrm{I}_{d-r',r'}(x)}{\mathrm{I}_{D-r',r'}(x)}$ is monotonically increasing, so we only need to prove the above inequality for $x = 1-\eta$. By Lemma 9 in Appendix A, $\frac{\mathrm{I}_{d-r',r'}(1-\eta)}{\mathrm{I}_{D-r',r'}(1-\eta)} < (1-\frac{\eta}{M})^{d-D}$. Therefore,

$$\begin{aligned}\sum_{d=D+1}^{\infty} \frac{1}{d-1}\frac{\mathrm{I}_{d-r',r'}(x)}{\mathrm{I}_{D-r',r'}(x)} &\leq \frac{1}{D}\sum_{d=D+1}^{\infty} \frac{\mathrm{I}_{d-r',r'}(1-\eta)}{\mathrm{I}_{D-r',r'}(1-\eta)} \\ &< \frac{1}{D}\sum_{d=D+1}^{\infty} (1-\frac{\eta}{M})^{d-D} \\ &= \frac{M-\eta}{D\eta} \\ &\leq 1,\end{aligned}$$

where the last inequality follows from $D = \lceil M/\eta \rceil - 1$. So we have established (30) and hence (29).

Last, by (23) and (29), we have for $0 \leq x \leq 1-\eta$,

$$\begin{aligned}\Omega(x;\Psi^r) &\geq \sum_{r'\geq r} \hbar_{r'} S_{r'}(x;\Psi^r) \\ &> -\ln(1-x)r\sum_{r'\geq r} \hbar_{r'},\end{aligned}$$

or

$$\Omega(x;\Psi^r) + \left(r\sum_{r'\geq r} \hbar_{r'}\right)\ln(1-x) > 0.$$

We conclude that $\hat{\theta} \geq r\sum_{r'\geq r} \hbar_{r'}$. The proof is completed by considering all $r = 1, 2, \cdots, M$. ■

Though in general the lower bound in Theorem 5 is not tight, we can show for a special case that it converges asymptotically to the upper bound in Theorem 4. Consider a rank distribution $h = (h_0, h_1, \ldots, h_M)$ with $h_\kappa = 1$ for some $1 \leq \kappa \leq M$. Theorem 5 implies that $\hat{\theta} \geq \kappa\hbar_\kappa$. On the other hand, Theorem 4 says that $\bar{\eta}\hat{\theta} \leq \sum_r r\hbar_r = \kappa\hbar_\kappa + \sum_{r<\kappa} r\hbar_r$. Note that $\bar{\eta}$ can be arbitrarily close to 1, and $\sum_{r<\kappa} r\hbar_r \to 0$ and $\hbar_\kappa \to h_\kappa$ when the field size goes to infinity. Thus,

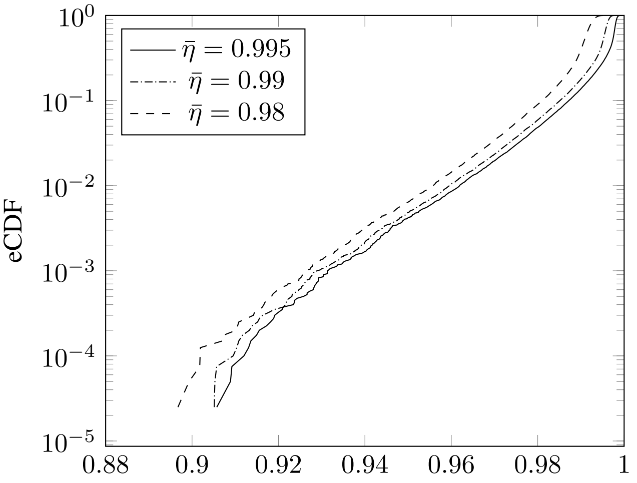


Fig. 6: The empirical cumulative distribution function (eCDF) of $\tilde{\theta} := \bar{\eta}\hat{\theta}/\sum_r r\hbar_r$ for 4×10^4 rank distributions. Here $q = 2^8$ and $M = 16$.

both the upper bound in Theorem 4 and the lower bound in Theorem 5 converge to κh_κ, the capacity of the LOC with empirical rank distribution converging to h.

We can compute the achievable rates of BATS codes numerically by solving (P1). Set $M = 16$ and $q = 2^8$. Totally 4×10^4 rank distributions are tested.[4] For each rank distribution h we solve (P1) for $\bar{\eta} = 0.98, 0.99$ and 0.995. The empirical distributions of $\tilde{\theta} \triangleq \bar{\eta}\hat{\theta}/\sum_r r\hbar_r$ are shown in Fig. 6. By Theorem 4, $\tilde{\theta} \leq 1$. The results show that when $\bar{\eta} = 0.995$, for more than 99.1% of the rank distributions, $\tilde{\theta}$ is larger than 0.96; for all the rank distributions the smallest $\tilde{\theta}$ is 0.9057. The figures in Fig. 6 clearly show the trend that when η becomes smaller, $\hat{\theta}$ becomes larger for the same rank distribution. Note that for these rank distributions, the ratio $\sum_r r\hbar_r/\sum_r rh_r$ are all larger than 0.999, so the upper bound in Theorem 4 is indeed very close to the capacity.

C. Optimizations for Multiple Rank Distributions

In the previous part of this section, we consider how to find an optimal degree distribution for a single rank distribution. For many scenarios, however, we need a degree distribution that is good for multiple rank distributions. In a general multicast problem, the rank distributions observed by the destination nodes can be different. Even for a single destination node, the empirical rank distribution may not always converge to the same value. We discuss the degree distributions for multiple rank distributions in the remaining part of this section.

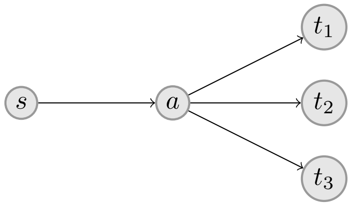


Fig. 7: In this network, node s is the source node. Node t_1, t_2 and t_3 are the destination nodes. Node a is the intermediate node that does not demand the file. All links are capable of transmitting one packet per use. The link (s, a) has packet loss rate 0.2. The links $(a, t_i), i = 1, 2, 3$ have packet loss rate 0.1, 0.2 and 0.3, respectively.

TABLE I
THE RANK DISTRIBUTIONS FOR THE THREE DESTINATION NODES IN FIG. 7.

rank	h^1	h^2	h^3
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0.0002
5	0	0.0001	0.0013
6	0.0002	0.0004	0.0058
7	0.0012	0.0025	0.0197
8	0.0056	0.011	0.0537
9	0.0201	0.0387	0.1165
10	0.0576	0.1041	0.1969
11	0.1306	0.2062	0.2468
12	0.2276	0.2795	0.2121
13	0.2796	0.2339	0.112
14	0.2052	0.1039	0.0312
15	0.0671	0.019	0.0036
16	0.0052	0.0008	0.0001

Let $\mathcal{H}$ be the set of rank distributions observed by the destination nodes of a multicast network. Consider a degree distribution Ψ and $\theta_h, h \in \mathcal{H}$ satisfying the following set of constraints:

$$\Omega(x; \Psi, \hbar(h)) + \theta_h \ln(1-x) \geq 0,\ \forall x \in [0, \bar{\eta}],\ \forall h \in \mathcal{H}, \quad (31)$$

where $\hbar(h) = (\hbar_i(h), i = 1, \dots, M)$. Then for each rank distribution $h \in \mathcal{H}$, rate $\bar{\eta}\theta_h$ is $\bar{\eta}$-achievable by the BATS code with degree distribution Ψ.

To illustrate the discussion, we extend the three-node network in Fig. 1 with two more destination nodes as shown in Fig. 7. In this network, node a transmits the same packets on its three outgoing links, but these links have different loss rates. Fixing $M = 16$, $q = 256$ and a certain inner code in node a (see the inner code to be defined in Section VII-A), we obtain the rank distributions h^i for node t_i, $i = 1, 2, 3$ in Table I. For the above example, see the maximum $\bar{\eta}$-achievable rates evaluated in Table II. The observation is that the degree distribution optimized for one rank distribution may not have a good performance for the other rank distributions: The degree distributions optimized for destination node t_1 and t_2 have poor performance for destination node t_3.

There are different criteria to optimize the degree distribution for a set of rank distributions. Here we discuss two of them as examples. One performance metric of interest is the multicast rate, which is a rate that is achievable by all the rank distributions. We can find the maximum multicast rate for all the rank distributions in $\mathcal{H}$ by solving the following

[4] A rank distribution is randomly generated as follows. First, select $x_1, x_2, \dots, x_{M-1}$ independently and uniformly at random in [0, 1]. Next, sort $\{x_i\}$ so that $x_1 \leq x_2 \leq \cdots \leq x_{M-1}$. Then, the rank distribution is given by $h_0 = 0$, and for $1 \leq r \leq M$, $h_r = x_r - x_{r-1}$, where $x_0 = 1$ and $x_M = 1$. This gives an almost uniform sampling among all the rank distributions with $\sum_{i=1}^{M} h_i = 1$ according to [44]. The reason that we choose $h_0 = 0$ is as follows. For a rank distribution $h = (h_0, \dots, h_M)$ with $h_0 > 0$, we obtain a new rank distribution $h' = (h'_0 = 0, h'_i = h_i/(1 - h_0) : i = 1, \dots, M)$. Optimization (P1) is equivalent for these two rank distributions except that the objective function is scaled by $1 - h_0$. Thus the values of $\tilde{\theta} := \bar{\eta}\hat{\theta}/\sum_r r\hbar_r$ for both h and h' are the same.

TABLE II

THE ACHIEVABLE RATES FOR DIFFERENT PAIRS OF RANK DISTRIBUTIONS AND DEGREE DISTRIBUTIONS. FOR EACH RANK DISTRIBUTION IN THE FIRST ROW AND EACH DEGREE DISTRIBUTION IN THE FIRST COLUMN, WE EVALUATE THE MAXIMUM 0.99-ACHIEVABLE RATE IN THE TABLE. FOR $i = 1, 2, 3$, Ψ^i IS OBTAINED BY SOLVING (P1) WITH h^i IN PLACE OF h. Ψ^3 CAN ALSO BE OBTAINED BY SOLVING (P2) WITH $\{h^1, h^2, h^3\}$ IN PLACE OF $\mathcal{H}$. $\Psi^{\text{MAX-PERC}}$ IS OBTAIN BY SOLVING (P3) WITH $\{h^1, h^2, h^3\}$ IN PLACE OF $\mathcal{H}$.

	h^1	h^2	h^3
$\sum_i i\hbar_i$	12.57	11.91	10.83
Ψ^1	12.55	6.10	1.77
Ψ^2	11.96	11.89	4.79
Ψ^3	10.99	10.95	10.81
$\Psi^{\text{max-perc}}$	11.94	11.35	10.28

optimization problem:

$$\begin{aligned} &\max \theta \\ &\text{s.t. } \Omega(x;\hbar(h),\Psi)+\theta\ln(1-x)\geq 0, \\ &\qquad\qquad \forall x\in[0,\bar{\eta}],\ \forall h\in\mathcal{H}, \\ &\Psi_i\geq 0,\ \sum_i \Psi_i = 1. \end{aligned} \tag{P2}$$

Denote by $\hat{\theta}_{\mathcal{H}}$ the maxima of (P2) w.r.t. $\mathcal{H}$. By the upper bound discussed in Section V-B, $\bar{\eta}\hat{\theta}_{\mathcal{H}}$ should be less than the minimum expected rank among all the rank distributions in $\mathcal{H}$, denoted by $\bar{h}_{\mathcal{H}}$. For the example that $\mathcal{H} = \{h^1, h^2, h^3\}$, the optimal degree distribution of (P2) is exactly Ψ^3. Since nodes t_1 and t_2 can emulate the packet loss rate of node t_3, the multicast rate of the BATS code is bounded by node t_3. So in this case, BATS codes can achieve a multicast rate very close to $\bar{h}_{\mathcal{H}}$.

In general, however, $\bar{\eta}\hat{\theta}_{\mathcal{H}}$ may not be very close to $\bar{h}_{\mathcal{H}}$. The maximum gap between $\bar{\eta}\hat{\theta}_{\mathcal{H}}$ and $\bar{h}_{\mathcal{H}}$ can be obtained numerically. For any real value μ, $0 \leq \mu \leq M$, define

$$\mathcal{B}(\mu) = \left\{ (h_0, \ldots, h_M) : \sum_{i=1}^{M} i h_i \geq \mu, \sum_{i=1}^{M} h_i = 1, h_i \geq 0 \right\}.$$

The set $\mathcal{B}(\mu)$ includes all the rank distributions that can potentially support rate μ. Since using more rank distributions can only give smaller optimal values, solving (P2) w.r.t. $\mathcal{B}(\mu)$ gives us a guaranteed multicast rate that is achievable by BATS codes with BP decoding for any set of rank distributions $\mathcal{H}$ with $\bar{h}_{\mathcal{H}} = \mu$. Directly solving (P2) w.r.t. $\mathcal{B}(\mu)$ is difficult since $\mathcal{B}(\mu)$ includes infinitely many of rank distributions. Using the techniques developed in [45], the set $\mathcal{B}(\mu)$ can be reduced to a finite set, and hence (P2) can be solved efficiently. See Fig. 8 for $\bar{\eta}\hat{\theta}_{\mathcal{B}(\mu)}$ when $M = 16$, $q = 256$ and $\bar{\eta} = 0.99$. For example, $\bar{\eta}\hat{\theta}_{\mathcal{B}(10)} = 8.10$.

The degree distribution obtained using (P2) may not be fair for all the destination nodes. For the degree distribution optimized using (P2), nodes t_1 and t_2 do not achieve a rate much higher than node t_3 though they have much lower loss rate than node t_3 (ref. Table II). To resolve this issue, we can find the percentage of $\sum_i i\hbar_i(h)$ that is achievable for all the rank distributions h in $\mathcal{H}$ using the following optimization:

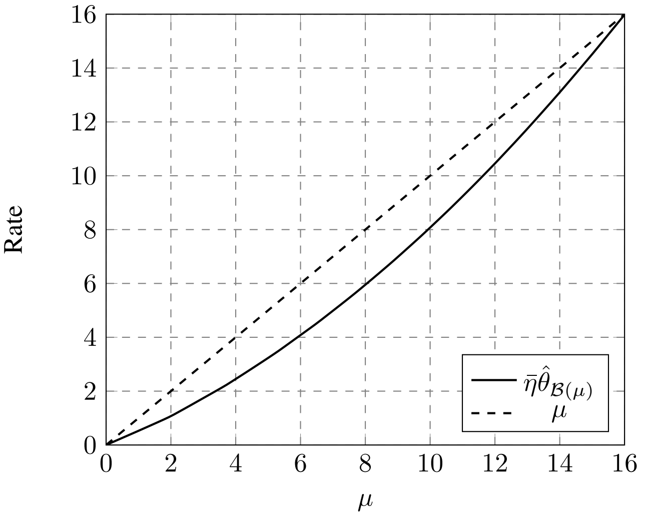


Fig. 8: The optimal values of (P2) w.r.t. $\mathcal{B}(\mu)$, where $M = 16$, $q = 2^8$ and $\bar{\eta} = 0.99$.

TABLE III

THE MAXIMUM VALUE $\hat{\alpha}$ OF (P3) WHEN $\mathcal{H}$ IS THE SET OF ALL RANK DISTRIBUTIONS FOR A GIVEN BATCH SIZE. HERE $\bar{\eta} = 0.99$

M	1	2	4	8	16	32
$\bar{\eta}\hat{\alpha}$	0.9942	0.8383	0.7068	0.6060	0.5274	0.4657

$$\begin{aligned} &\max \alpha \\ &\text{s.t. } \Omega(x;\hbar(h),\Psi)+\alpha\sum_i i\hbar_i(h)\ln(1-x)\geq 0, \\ &\qquad\qquad\qquad \forall x\in[0,\bar{\eta}],\ \forall h\in\mathcal{H}, \\ &\Psi_i\geq 0,\ \sum_i \Psi_i = 1. \end{aligned} \tag{P3}$$

Denote by $\hat{\alpha}$ the maxima of (P3). When $\mathcal{H} = \{h^1, h^2, h^3\}$ and $\bar{\eta} = 0.99$, the percentage is 94.9 (the optimal value of (P3) multiplied by $100\bar{\eta}$). The performance of the optimal degree distribution of (P3) is shown in the last row of Table II. BATS codes with this degree distribution achieves 95.0, 95.3 and 94.9 percentage of $\sum_i i\hbar_i$ for sink nodes t_1, t_2 and t_3, respectively.

In general, BATS codes are not universal. There does not exist a degree distribution that can achieve rates close to $\sum_i i\hbar(h)$ for all rank distributions for a given batch size M, except for $M = 1$, the case of LT/Raptor codes. In Table III, we give the optimal values of (P3) (multiplied by $\bar{\eta}$) with $\mathcal{H}$ being the set of all the rank distributions for batch size $1, 2, 4, \ldots, 64$. Take $M = 16$ as an example. The value $\bar{\eta}\hat{\alpha} = 0.5274$ implies a worst guaranteed rateless rate for an arbitrary number of destination nodes with arbitrary empirical rank distributions: A destination node can decode the original file with high probability after receiving n batches such that $0.5274\sum_{i=1}^{n}\mathrm{rk}(\mathbf{H}_i)$ is larger than the number of original input packets, where $\mathbf{H}_i$ is the transfer matrix of the ith batch. When the possible empirical rank distributions are in a smaller set, the optimal value of (P3) can be much larger, as in the network with three destination nodes.

Using different objective functions and constraints, other optimization problems can be formulated to optimize a degree distribution for a set of rank distributions. For example, we can optimize the average rate and average completion time of all the destination nodes. Readers are referred to [45] for more degree distribution optimization problems and the techniques to solve these problems.

VI. Practical Batch Encoding and Decoding Designs

Asymptotic performance of BATS codes has been studied in the previous sections. Now we look at several practical issues about the design of BATS codes.

A. Overhead and Rate

We give an alternative and convenient way to evaluate the performance of a BATS code with finite block lengths. We define two kinds of overheads, which is related to the outer code and inner code respectively, and discuss their relationship with coding rate.

Suppose that a destination node decodes successfully after receiving n batches with transfer matrices $\{\mathbf{H}_i, i = 1, \ldots, n\}$. If the coefficient vector of a received packet is linearly dependent with those of the other received packets of the same batch, this packet is redundant and can be discarded by the decoder. Therefore we define the *receiving overhead* as RO $= \sum_{i=1}^{n}[\mathrm{col}(\mathbf{H}_i) - \mathrm{rk}(\mathbf{H}_i)]$, where $\mathrm{col}(\mathbf{H})$ is the number of columns of $\mathbf{H}$. The receiving overhead is generated inside the networks by the inner code, and hence cannot be reduced by the design of batch encoding and decoding. We should design the inner code to reduce the receiving overhead, but it may not always be necessary to reduce the receiving overhead to a value close to zero (see example designs in the next section).

Define the *coding overhead* as CO $= \sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i) - K$. We should design batch encoding and decoding schemes such that the coding overhead is as small as possible. The *coding rate observed by the destination node* is CR $= \frac{K}{\sum_{i=1}^{n} \mathrm{col}(\mathbf{H}_i)}$. For an optimal code, we have $\frac{\text{CO}}{K} \to 0$ and hence CR $\to \frac{\sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i)}{\sum_{i=1}^{n} \mathrm{col}(\mathbf{H}_i)}$, the normalized average rank of the transfer matrices.

B. Precoding

In Section V, we have discussed how to design BATS codes such that the BP decoding can stop with at least a given fraction of the input packets decoded. This fraction can be made arbitrarily close to 1 as long as the number of input packets is sufficiently large. After the BP decoding has stopped, we can try to decode the remaining input packets using Gaussian elimination. Can we guarantee that the Gaussian elimination succeeds with a small coding overhead? The answer is actually negative. Consider that we want to recover all the K input symbols using n batches with probability at least $1 - 1/K^c$ for some positive constant c. Similar to the analysis of LT codes (cf. [2, Proposition 1]), no matter what decoding algorithm is applied, the expected degree must be lower bounded by $c' \frac{K}{n} \log(K)$ for some positive constant c'. When K/n converges to a constant positive value, the expected degree is lower bounded by a function of $\log(K)$. However, the degree distribution obtained using (P1) with a fixed value of $\bar{\eta} = 1 - \eta$ has a constant expected rank, which is desired for low encoding/decoding complexity.

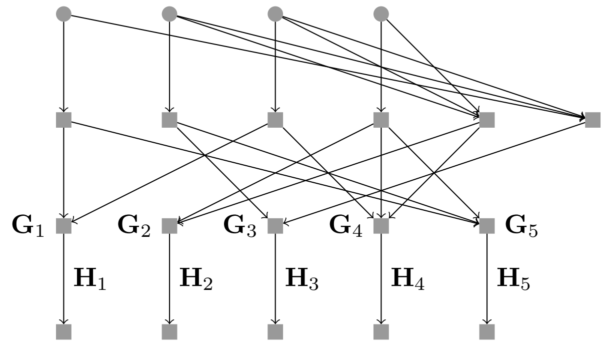


Fig. 9: Precoding of BATS codes. Nodes in the first row represent the input packets. Nodes in the second row represent the intermediate packets generated by the precode.

One way to resolve the above issue is to refine the analysis in Section III so that we allow $\bar{\eta} = o(K)$, e.g., $\bar{\eta} = (\log K)^{-1}$. However, this would incur a higher order of the encoding/decoding complexity. A better way to resolve the above issue is to use the precoding technique which has been used in Raptor codes. That is, before applying the batch encoding process in Section II-A, the input packets are first encoded using a traditional erasure code (called a *precode*). The batch encoding process is applied to the intermediate input packets generated by the precode. If the BP decoding of the BATS code can recover a given fraction of the intermediate input packets, the precode is able to recover the original input packets in face of a fixed fraction of erasures.

Due to similar requirements, the precode for Raptor codes in can be applied to BATS without much modifications. Readers can find the detailed discussion of these techniques in [46], [47].

C. Inactivation Decoding

BP decoding stops with high probability before the desired fraction of input packets are decoded when the number of input packets is small. When the BP decoding stops, a better way to continue the decoding process than Gaussian elimination is to use *inactivation*. Inactivation is an efficient way to solve sparse linear systems [48], [49], and it has been used for the decoding of LT/Raptor codes [46], [50] (a similar algorithm has been used for efficient encoding of LDPC codes [51]).

Recall that BP decoding stops when there are no decodable batches. In inactivation decoding, when there are no decodable batches at time t, we instead pick an undecoded input packet b_k and mark it as *inactive*. We substitute the inactive packet b_k into the batches like a decoded packet, except that b_k is an indeterminate. For example, if b_k is a contributor of batch i, we express the components of the updated $\mathbf{Y}_i$ as polynomials in b_k. The decoding process is repeated until all input packets are either decoded or inactive. The inactive input packets can be recovered by solving a linear system of equations using Gaussian elimination. In a nutshell, inactivation decoding trades computation cost (decoding inactive input symbols using Gaussian elimination) with coding overhead.

Inactivation decoding of BATS codes has been studied in [52], where a recursive formula is obtained to calculate the expected number of inactive packets when the inactive packets are chosen uniformly among all undecoded input packets. The degree distribution obtained using the optimization (P1) can be further fine-tuned to obtain better performance for inactivation decoding [53]. Using the techniques discussed in this section, it is possible to design BATS codes with very low coding overhead for finite block lengths. See the numerical results in Section VII-A and [53].

VII. Examples of BATS Code Applications

In this section, we use several examples to illustrate how to apply BATS codes in erasure networks, where each (network) link can transmit *one* packet in a time slot subject to a certain packet erasure probability. If not erased, the packet will be correctly received. We say a network has *homogeneous links* if all network links have the same erasure probability, and has *heterogeneous links* otherwise. Unless otherwise specified, network link transmission is instantaneous. We will focus on how to design the inner code including cache management and batch scheduling at the intermediate nodes.

A. Line Networks

A line network of length k is formed by a sequence of $k+1$ nodes $\{v_0, v_1, \dots, v_k\}$, where the first node v_0 is the source node and the last node v_k is the destination node. There are only network links between two consecutive nodes. The network in Fig. 1 is a line network of length 2. We first study line networks with *homogeneous links* and then extend the results to general line networks. Suppose that all the links in the line network have the same link erasure probability ϵ. When there is no computation and storage constraints at the intermediate network nodes, the min-cut capacity of the line network with length k is $1-\epsilon$ packet per use for any $k>0$. Here one use of the network means the use of each network link at most once; transmitting nothing on a network link in a particular time slot is allowed. We apply the following BATS code scheme for line networks.

Scheme 1 (Line network)**.** The source node generates batches and transmits a packet in each time slot. The M packets of a batch are transmitted in M consecutive time slots, and the batches are transmitted according to the order in which they are generated. The source node keeps transmitting batches until the destination node decodes successfully. No feedback is required except for the notification of successful decoding from the destination node.

In the first M time slots, node v_1 can potentially receive M packets of the first batch. In the first $M-1$ time slots, node v_1 saves the received packets in its buffer but transmits nothing. In the Mth time slot, node v_1 generates M coded packets using random linear coding on the packets in its buffer and the packet just received, if any, which are all in the same batch. After generating the M coded packets, the original received packets in the buffer are deleted. Node v_1 then transmits one of the coded packets and saves the remaining $M-1$ coded packets in its buffer. In each of the following $M-1$ time slots, node v_1 transmits one of the remaining coded packets of the first batch and then deletes in the buffer the transmitted packet immediately. During these time slots, if node v_1 receives a new packet (of the 2nd batch), the new packet is saved in the buffer. From the $2M$th to the $(3M-1)$th time slot, node v_1 repeats the above operations on the second batch, so on and so forth. All the other intermediate nodes apply the same operations as node v_1.

In the above scheme, each intermediate node caches at most $M-1$ packets in the buffer. There is a delay for each intermediate node: node v_i can only start to receive packets after $(i-1)(M-1)$ time slots. For a network of fixed length, the delay is neglectable compared with the total transmission time when the file size is large. The buffer at the intermediate nodes may be better managed to improve the rate and/or to reduce the delay, but the scheme we defined here is easy to analyze and is asymptotically optimal.

The transfer matrices of all batches are i.i.d. and can be expressed explicitly. The transmission of a batch through a network link can be modelled by an $M \times M$ random *diagonal* matrix E with independent components, where a diagonal component is 0 with probability ϵ and is 1 with probability $1-\epsilon$. The network coding at an intermediate node for a batch is given by a totally random $M \times M$ matrix Φ. The transfer matrix $H^{(1)}$ for the unit-length line network is $H^{(1)} = E_1$, where E_1 has the same distribution as E. For $k>1$, the transfer matrix $H^{(k)}$ for the k-length line network can be expressed as

$$H^{(k)} = H^{(k-1)} \Phi_{k-1} E_k,$$

where Φ_{k-1} has the same distribution as Φ and E_k has the same distribution as E. Further, $\Phi_1, \dots, \Phi_{k-1}, E_1, \dots, E_k$ are mutually independent.

The rank distribution of the transfer matrix $H^{(k)}$ can be calculated recursively. Let $h^{(k)} = (h_0^{(k)}, \dots, h_M^{(k)})$ be the rank distribution of $H^{(k)}$. First

$$h_r^{(1)} = \binom{M}{r}(1-\epsilon)^r \epsilon^{M-r}, \quad r = 0, 1, \dots, M.$$

Using (3) we obtain that for $k>1$,

$$h_r^{(k)} = \sum_{i=r}^{M} \sum_{j=r}^{M} h_i^{(k-1)} \binom{M}{j} (1-\epsilon)^j \epsilon^{M-j} \zeta_r^{i,j}, \ r = 0, \dots, M.$$

When $M=1$, the BATS code scheme for line networks degenerates to an LT/Raptor code scheme with forwarding at the intermediate nodes. The achievable rate for the length-k line network is $(1-\epsilon)^k$, i.e., the rate decreases exponentially fast with the network length.

We know from the previous discussion that the normalized expected rank $\sum_r r h_r^{(k)}/M$ can be approached by BATS codes. By the following lemma, when M tends to infinity, the normalized expected rank will converge to $1-\epsilon$. Therefore, for line networks with link erasure probability ϵ, Scheme 1 can achieve a normalized rate very close to $1-\epsilon$ when M is sufficiently large.

Lemma 7. $\lim_{M\to\infty} \sum_r r h_r^{(k)}/M = 1-\epsilon$.

Proof: First, we have $\sum_r rh_r^{(k)}/M = \mathbb{E}[\mathrm{rk}(H^{(k)})]/M \le \mathbb{E}[\mathrm{rk}(E_k)]/M \to 1-\epsilon$ as $M \to \infty$. We then prove by induction that for any $\delta > 0$,

$$\lim_{M\to\infty} \Pr\left\{\frac{\mathrm{rk}(H^{(k)})}{M} \ge 1-\epsilon-\delta\right\} = 1, \tag{32}$$

which impies $\lim_{M\to\infty}\mathbb{E}[\mathrm{rk}(H^{(k)})]/M \ge 1-\epsilon$.

Since $\mathrm{rk}(H^{(1)}) = \mathrm{rk}(E_1)$ follows a binomial distribution with mean $M(1-\epsilon)$, (32) with $k=1$ follows from the weak law of large numbers. Suppose that (32) holds for $H^{(i)}$, $i = 1,\ldots,k$, $k \ge 1$. We have for $t = 1-\epsilon-\delta$, $\delta > 0$,

$$\begin{aligned} &\Pr\left\{\frac{\mathrm{rk}(H^{(k+1)})}{M} \ge t\right\} \\ &= \Pr\left\{\frac{\mathrm{rk}(H^{(k)}\Phi E)}{M} \ge t\right\} \\ &\ge \Pr\left\{\frac{\mathrm{rk}(H^{(k)}\Phi E)}{M} \ge t \,\middle|\, \frac{\mathrm{rk}(H^{(k)})}{M} \ge t+\frac{\delta}{2}, \frac{\mathrm{rk}(E)}{M} \ge t\right\} \\ &\quad \times \Pr\left\{\frac{\mathrm{rk}(H^{(k)})}{M} \ge t+\delta/2\right\} \Pr\left\{\frac{\mathrm{rk}(E)}{M} \ge t\right\}, \end{aligned} \tag{33}$$

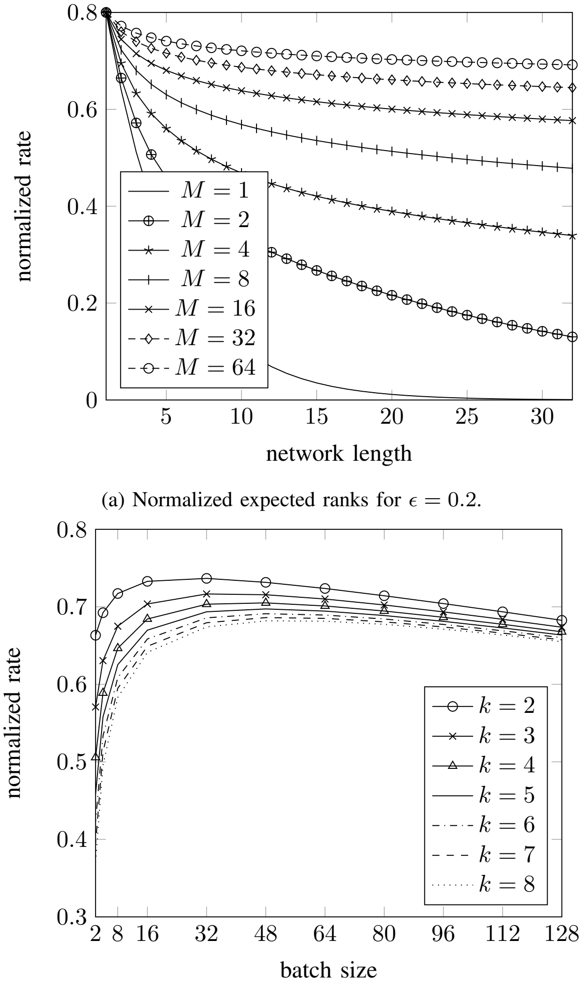
where the last two terms on the RHS of (33) converge to 1 as $M \to \infty$. Since

$$\begin{aligned} &\Pr\left\{\frac{\mathrm{rk}(H^{(k)}\Phi E)}{M} \ge t \,\middle|\, \frac{\mathrm{rk}(H^{(k)})}{M} \ge t+\frac{\delta}{2}, \frac{\mathrm{rk}(E)}{M} \ge t\right\} \\ &\ge \Pr\big\{\mathrm{rk}(H^{(k)}\Phi E) = \lceil Mt \rceil \,\big|\, \\ &\qquad \mathrm{rk}(H^{(k)}) = \lceil M(t+\delta/2) \rceil, \mathrm{rk}(E) = \lceil Mt \rceil\big\} \\ &= \zeta_{\lceil Mt \rceil}^{\lceil M(t+\delta/2) \rceil} \end{aligned}$$

where the equality follows from (3), and $\zeta_{\lceil Mt \rceil}^{\lceil M(t+\delta/2) \rceil} \to 1$, as $M \to \infty$, the RHS of (33) converges to 1 as $M \to \infty$. The proof is completed. ■

We, however, are more interested in the performance for small values of M, which can be characterized numerically. We calculate the normalized expected rank $\sum_r rh_r^{(k)}/M$ for $\epsilon = 0.2$ and field size $q = 256$ in Fig. 10a. Compared with $M = 1$, the normalized expected rank decreases slowly as the network length increases when $M \ge 2$. For a fixed network length, Fig. 10a also illustrates the tradeoff between the batch size and the maximum achievable rates of BATS codes (without considering the coefficient vector overhead, or assuming T is much larger than M). We see that when M is larger than 32, using a larger batch size only gives a marginal rate gain (but increases significantly the computation cost).

The gain by using a larger M can be offset by the coefficient vector overhead. If we include the coefficient vector overhead, the normalized rate of BATS codes should be multiplied by $(1 - M/T)$. In the extreme case that $M = T$, the achievable rate becomes zero. We calculate the value of $(1-M/T)\sum_r rh_r^{(k)}/M$ for $T = 1024$, $\epsilon = 0.2$ and $q = 256$ in Fig. 10b. These values of T and q correspond to a packets size of 1 KB. It can be seen from the plot that when $\epsilon = 0.2$, a small batch size roughly equal to 32 is almost rate-optimal for practical parameters.



(a) Normalized expected ranks for $\epsilon = 0.2$.

(b) Normalized expected ranks times $(1 - M/T)$ for $\epsilon = 0.2$.

Fig. 10: Numerical results for line networks. The field size q is 2^8.

If ϵ is large, e.g., 0.9, however, a much larger batch size, e.g., 200, will be required so that the normalized expected rank approaches $1-\epsilon$. But a large batch size results in a large coefficient vector overhead. We introduce a technique such that small batch sizes can still be used for high erasure probabilities. Suppose that M is a large enough batch size such that the normalized expected rank is close $1-\epsilon$. Let $\tilde{M} = M(1-\epsilon+\delta)$ for certain small positive value δ. For example, $\tilde{M} = 30$ when $\epsilon = 0.9$, $M = 200$ and $\delta = 0.05$. We modify Scheme 1 by using an outer code with batch size $\tilde{M}$ to replace the outer code with batch size M: For each batch X of size $\tilde{M}$, the source node generates M packets for transmission by multiplying X with an $\tilde{M} \times M$ totally random matrix $\Psi_{\tilde{M}\times M}$. The inner code does not change: the batch size is still M. The destination node decodes by using batch size $\tilde{M}$.

The effectiveness of the above technique is explained as follows. Let H be the transfer matrix of a batch for the outer code with batch size M. In the above modified Scheme 1, the transfer matrix of a batch for the outer code with batch size $\tilde{M}$ can be expressed as $\Psi_{\tilde{M}\times M}H$. We know that in Scheme 1, the rank of H is smaller than $\tilde{M} = M(1-\epsilon+\delta)$ with high probability when M is large. Thus, the expected ranks of H and $\Psi_{\tilde{M}\times M}H$ converges to the same value in probability as M tends to infinity. Therefore, the asymptotic performance of

TABLE IV

NUMERICAL RESULTS OF BATS CODES FOR THE LENGTH-4 LINE NETWORK WITH $M = 32$ AND $q = 256$.

K		1600	8000	16000
coding overhead	average	2.04	6.30	26.58
	max	16	77	1089
	min	0	0	0
number of inactivation	average	94.0	215.5	352.2
	max	119	268	379
	min	72	179	302
receiving overhead	average	599.5	3015.7	6041.6
	max	673	3183	6469
	min	532	2865	5788

the outer codes is not sacrificed by using a small batch size.

The above technique can in principle be applied for all values of ϵ. However, for small values of ϵ, the advantage of using the technique is small.

Scheme 1 does not depend on the erasure probability of network links, so it can also be applied to a line network with heterogeneous links. Consider a length-k line network where the maximum link erasure probability among all links is ϵ. The min-cut capacity of this network is $1 - \epsilon$. The expected rank of the transfer matrix of this network is less than the one of the length-k line netwrok where the erasure probabilities of all links are ϵ. Therefore, the normalized expected rank will converge to $1 - \epsilon$ when M tends to infinity, and hence Scheme 1 can achieve a normalized rate very close to $1 - \epsilon$ when M is sufficiently large for a line network.

We use the length-4 line network as an example to evaluate the finite length performance of a BATS code with inactivation decoding. The results are in Table IV, where the average overhead is less than 3 packets per 1,600 packets.[5]

B. Unicast Networks

A unicast network is represented by a directed acyclic graph with one source node and one destination node. We first provide two BATS code schemes for unicast networks with *homogeneous links*, and then discuss how to extend the schemes to unicast networks with heterogeneous links. One way to apply Scheme 1 to unicast networks with homogeneous links is as follows.

Scheme 2 (Unicast)**.** Consider a unicast network with homogeneous links. Find L edge-disjoint paths from the source node to the destination node, and separate the input packets into L groups, each of which is associated with a path. The source node encodes each group of the input packets using a BATS code, and transmits all the batches on the associated path. An intermediate node on that path runs an instance of the inner code of the BATS code defined in Scheme 1. The destination node decodes the packets received from a path to recover a group of input packets.

For a unicast network with link erasure probability ϵ for all links, the min-cut capacity is $(1 - \epsilon)L^*$, where L^* is the maximum number of edge-disjoint paths from the source node to the destination node. Since Scheme 2 is equivalent to applying Scheme 1 on multiple line networks, the normalized expected rank of the transfer matrix for each path converges to $1 - \epsilon$ as M tends to infinity. Therefore, Scheme 2 can achieve a rate very close to the min-cut capacity by optimizing the degree distribution for each path separately. However, a better scheme can be obtained by encoding and decoding the batches for different paths jointly.

Scheme 3 (Unicast)**.** This scheme for a unicast network with homogeneous links is same as Scheme 2 except that the source node encodes all the input packets using a BATS code. The batches are grouped into sets of L sequentially. Each set is transmitted on the L paths in M time slots, with each batch in the set transmitted on a distinct path.

The rank distribution for the above scheme is the rank distributions averaged over all the paths. So, the normalized expected rank of the transfer matrix converges to $1 - \epsilon$ as M tends to infinity. Now we consider a general unicast network with heterogeneous links. We apply the above BATS code scheme to the unicast network in three steps:

1) Obtain a unicast network G^* with homogeneous links that has the same min-cut as the original unicast network G.
2) Apply Scheme 3 on network G^*.
3) Convert the scheme on G^* to one that can be used in network G while preserving the performance.

The second step is straightforward. The first and third steps are explained as follows.

In the first step, assume that the link erasure probability are all rational. Fix an integer N such that $(1 - \epsilon)N$ is an integer for any erasure probability ϵ in a link of the network. Network G^* has the same set of nodes as network G. For any link between nodes a and b in G with erasure probability ϵ, we have a set of $(1-\epsilon)N$ parallel links between nodes a and b in G^* with erasure probability $1-1/N$. We call network G^* the *homogenized network* of network G. We can check that the min-cut capacity of network G and G^* are the same. Use the three-node network in Fig. 1 as example. Suppose the two links (s, a) and (a, t) have erasure probabilities 0.2 and 0.1, respectively. Let $N = 10$. The homogenized network of the three-node network has 8 parallel links from node s to node a and 9 parallel links from node a to node t, where all the links have an erasure probability 0.9.

In the third step, we convert Scheme 3 on network G^* to one that can be used in the original network by emulating virtual links in the network nodes in G. In the three-node network example, for link (s, a), node s emulates 8 virtual outgoing links and node a emulates 8 virtual incoming links, each of which corresponds to a virtual outgoing link of node s; for link (a, t), node a emulates 9 virtual outgoing links and node t emulates 9 virtual incoming links, each of which corresponds to a virtual outgoing link of node a. In each time slot, nodes s and a randomly choose one of their virtual outgoing links, transmit the packet on that virtual link on the original outgoing link, and delete the packets on the other virtual outgoing links. We assume that the choice of virtual outgoing links in node s is known by node a so that the received packet of node a from link (s, a) can be associated with the corresponding

[5] The implementation of the BATS codes used for the generation of the results in Table IV can be found in https://github.com/shhyang/simbats.

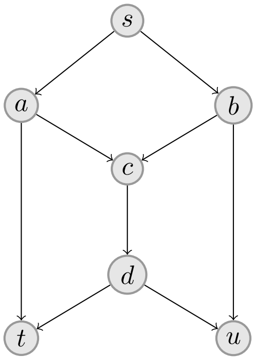


Fig. 11: Butterfly network. Node s is the source node. Node t and u are destination nodes.

virtual incoming link. The same is assumed for nodes a and t. In a general network topology, a network node needs to maintain a set of virtual outgoing (incoming) links for each original outgoing (incoming) link.

The rank distribution induced by Scheme 3 on network G^* is the same as the modified scheme on the original network G. Therefore, the BATS code scheme can achieve a rate very close to the min-cut capacity of a unicast network when M is sufficiently large.

C. Bufferfly Network

We use the butterfly network as an example to discuss how to design better BATS code schemes for general multicast networks. Suppose the butterfly network is homogeneous. We first propose a BATS code scheme and then discuss some improvements of the scheme for tackling a practical issue.

Scheme 4 (Butterfly network)**.** In this scheme for the butterfly network, the source node s generates batches of size $2M$ and transmits M packets of a batch on link (s, a) and the other M packets of the batch on link (s, b). Nodes a and b apply an inner code similar to that defined in Scheme 1 on batches of size M except that they generate $2M$ coded packets for each batch so that they can transmit different packets on two of their outgoing links. Node c applies the inner code defined in Scheme 1 with two exceptions: First node c has a buffer size $2M - 2$ packets since in each time slots it may receive two packets. Second, in each of the time slots $M, 2M, \ldots$, node c generates M coded packets using all the packets it has received in the current batch. Node d applies the inner code defined in Scheme 1 for line networks on batches of size M and transmits the same packets on both of their outgoing links.

Two aspects of the above scheme deserves some explanation. First, node a (or b) transmits different packets to its children, which is different from the operation at node d. Note that both outgoing links of node a can reach node t. Therefore, transmitting different packets on these two links can potentially increase the rank of the transfer matrix of a batch. Second, node c can potentially receive up to $2M$ packets of a batch, but it only generates M recoded packets. This is crucial for making the transmission on link (c, d) efficient for both destination nodes.

In Scheme 4, the two destination nodes have the same rank distribution for the transfer matrix of the batches. We can argue that for each destination node, the normalized expected rank of the transfer matrix induced by this scheme converges to $1 - \epsilon$ (using an approach similar to the one we have used for line networks and unicast networks with homogeneous links), where ϵ is the link erasure probability. Therefore, the BATS code scheme can achieve a rate very close to the multicast capacity of the butterfly network when M is sufficiently large.

Now we incorporate a practical issue in the butterfly network: consider that the two paths from node s to node c can have different latencies. With the latency issue, it is possible that node c receives packets from different batches on its incoming links at the same time slot. One approach to resolve the latency issue is to allow node c has a larger buffer so that the packets of the same batch can be aligned in node c for network coding. This approach is feasible when the latency difference is in certain measurable range. For example, if we know that the two paths from node s to node c have a maximum latency difference of 10 time slots, we can enlarge the buffer in node c to cache 20 more packets. A more robust scheme is still desired for the scenarios where the latency difference is large or not measurable.

Scheme 5 (Butterfly network)**.** Consider the butterfly network with possibly different latencies for different links. The source node separates its input packets into two groups A and B. The source node encodes packets in group A and in group B using two BATS codes of batch size M, and the batches generated using groups A and B are denoted by $X_A[i]$ and $X_B[i]$, $i = 1, 2, \ldots$, respectively. Batches $X_A[i]$, $i = 1, 2, \ldots$ are transmitted on link (s, a) and batches $X_B[i]$, $i = 1, 2, \ldots$ are transmitted on link (s, b). Nodes a, b and d apply the same inner code defined in Scheme 4.

Suppose note c is receiving packets of batch $X_A[i]$ from link (a, c). If node c recieves no packets from link (b, c) during the period of receiving the packets of $X_A[i]$, it will generate M coded packets of batch $X_A[i]$ and transmit them on M consecutive times slots on link (c, d). If node c receives some packets of batch $X_B[j]$ from link (b, c) during this period, where j may be different from i due to different latencies of different links, node c will align the transmission of the coded packets of $X_A[i]$ and $X_B[j]$. The M coded packets of $X_A[i]$ and the M coded packets of $X_B[i]$ generated by node c can be expressed as $X_A[i]H_A$ and $X_B[j]H_B$, respectively, for certain M-column matrices H_A and H_B. Node c then transmits the packets $X_A[i]H_A + X_B[j]H_B$.

Node t first decodes the packets in group A using the batches received from link (a, t). The packets received from link (d, t) are in batches of the form

$$Y = X_A[i]H_A[i] + X_B[j]H_B[j],$$

where H_A and H_B are the corresponding batch transfer matrices. Since group A has been decoded, node t can recover the batch $X_A[i]$ and cancel the effect of $X_A[i]$ from the received batch Y. Then node t decodes the packets in B. Node u applies a similar decoding procedure.

The butterfly network has two sub-trees with the node s

as the root and nodes t and u as the leaves: one sub-tree includes nodes a, c and d; and the other sub-tree includes nodes b, c and d. In the above scheme, for each group of the input packets, we apply a BATS code scheme for multicast in one of the two sub-trees. Since the two sub-trees share the network link (c, d), the batches of these two BATS codes are mixed together to share the network link (c, d). Note that we do not mix the batches of the same BATS code. The decoding in a destination node is a kind of successive cancellation: One group of the input packets is first decoded using BP decoding of BATS codes; The effect of this group is cancelled out from the mixed batches; The other group of the input packets is then decoded using BP decoding of BATS codes.

VIII. Concluding Remarks

Benefiting from previous research on network coding and fountain codes, BATS codes are proposed as a rateless code for transmitting files through multi-hop communication networks with packet loss. In addition to low encoding/decoding complexity, BATS codes can be realized with constant computation and storage complexity at the intermediate nodes. This desirable property makes BATS code a suitable candidate for the making of universal network coding based network devices that can potentially replace routers.

Our study in this paper provides the tools to optimize the performance of BATS code and the guidelines to design BATS code-based network communication schemes. Examples of BATS code applications are given for line networks and general unicast networks. For general multicast networks, schemes based on BATS codes can be developed as illustrated for the butterfly network. More work is expected to explore the applicability of BATS codes and to study the implementation of BATS code-based network communication systems.

Acknowledgements

We thank Prof. Rüdiger Urbanke for his insightful input to this work. We thank Cheuk Ting Li and Tom M. Y. Lam for their works in C/C++ implementation of BATS codes.

Appendix A
Incomplete Beta Function

Beta function with integer parameters is used extensively in this work. Related results are summarized here. For positive integer a and b, the *beta function* is defined by

$$B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt = \frac{(a-1)!(b-1)!}{(a+b-1)!}.$$

The *(regularized) incomplete beta function* is defined as

$$\begin{aligned} \mathrm{I}_{a,b}(x) &= \frac{\int_0^x t^{a-1}(1-t)^{b-1}dt}{B(a,b)} \qquad (34)\\ &= \sum_{j=a}^{a+b-1}\binom{a+b-1}{j}x^j(1-x)^{a+b-1-j}. \end{aligned}$$

For more general discussion of beta functions, as well as incomplete beta functions, please refer to [54].

Using the above definitions, we can easily show that

$$\int_0^1 \mathrm{I}_{a,b}(x)dx = \frac{b}{a+b}, \qquad (35)$$

and

$$\mathrm{I}_{a+1,b}(x) = \mathrm{I}_{a,b}(x) - \frac{x^a(1-x)^b}{aB(a,b)}. \qquad (36)$$

Lemma 8. $\frac{\mathrm{I}_{a+1,b}(x)}{\mathrm{I}_{a,b}(x)}$ *is monotonically increasing in* x.

Proof: By (36),

$$\begin{aligned} \frac{\mathrm{I}_{a+1,b}(x)}{\mathrm{I}_{a,b}(x)} &= 1 - \frac{x^a(1-x)^b}{aB(a,b)\mathrm{I}_{a,b}(x)}\\ &= 1 - \frac{1}{aB(a,b)\sum_{j=a}^{a+b-1}\binom{a+b-1}{j}x^{j-a}(1-x)^{a-1-j}}\\ &= 1 - \frac{1}{aB(a,b)\sum_{j=0}^{b-1}\binom{a+b-1}{j+a}x^j(1-x)^{-1-j}}, \end{aligned}$$

in which $x^j(1-x)^{-1-j}$ is monotonically increasing. ■

Lemma 9. *When* $\frac{b-1}{a+1} \le \frac{\eta}{1-\eta}$ *where* $0 < \eta < 1$, $\frac{\mathrm{I}_{a+1,b}(x)}{\mathrm{I}_{a,b}(x)} \le 1 - \frac{\eta}{b}$ *for* $0 < x \le 1-\eta$ *with equality when* $b = 1$ *and* $x = 1-\eta$.

Proof: Since $\frac{\mathrm{I}_{a+1,b}(x)}{\mathrm{I}_{a,b}(x)}$ is monotonically increasing in x (cf. Lemma 8), it is sufficient to show $\frac{\mathrm{I}_{a+1,b}(1-\eta)}{\mathrm{I}_{a,b}(1-\eta)} \le 1 - \frac{\eta}{b}$. Since $a+1 \ge (b-1)\frac{1-\eta}{\eta}$,

$$\begin{aligned} \mathrm{I}_{a,b}(1-\eta) &= \sum_{j=a}^{a+b-1}\binom{a+b-1}{j}(1-\eta)^j\eta^{a+b-1-j}\\ &\le b\binom{a+b-1}{a}(1-\eta)^a\eta^{b-1}, \end{aligned}$$

where the equality holds for $b = 1$. Thus,

$$\begin{aligned} \frac{\mathrm{I}_{a+1,b}(1-\eta)}{\mathrm{I}_{a,b}(1-\eta)} &= 1 - \frac{(1-\eta)^a\eta^b}{aB(a,b)\mathrm{I}_{a,b}(1-\eta)}\\ &\le 1 - \frac{(1-\eta)^a\eta^b}{abB(a,b)\binom{a+b-1}{a}(1-\eta)^a\eta^{b-1}}\\ &= 1 - \frac{\eta}{b}. \end{aligned}$$

■

We will use the following result about the summation of binomial coefficients:

$$\sum_{j=0}^{n}(-1)^{j-n}\binom{j+m}{n}\binom{n}{j} = 1, \quad m \ge n. \qquad (37)$$

The above equality can be verified as follows:

$$\begin{aligned}
&\sum_{j=0}^{n}(-1)^{j-n}\binom{j+m}{n}\binom{n}{j}\\
&\quad=\sum_{j=0}^{n}(-1)^{j-n}\binom{j+m}{j+m-n}\binom{n}{j}\\
&\quad=\sum_{j=0}^{n}(-1)^{j-n}(-1)^{j+m-n}\\
&\qquad\qquad\times\binom{-j-m+j+m-n-1}{j+m-n}\binom{n}{j} && (38)\\
&\quad=\sum_{j=0}^{n}(-1)^{m}\binom{-n-1}{j+m-n}\binom{n}{n-j}\\
&\quad=(-1)^m\binom{-1}{m} && (39)\\
&\quad=1, && (40)
\end{aligned}$$

where (39) follows from Vandermonde's identity; (38) and (40) use the relation between binomial coefficients with negative integers and positive integers.

Lemma 10. *For $r \geq 1$,*

$$\sum_{d=r+1}^{\infty}\frac{1}{d-1}\mathrm{I}_{d-r,r}(x) = -\ln(1-x), \quad x \in [0,\ 1).$$

Proof: As a special case, when $r = 1$, the equality becomes

$$\sum_{d=2}^{\infty}\frac{x^{d-1}}{d-1} = -\ln(1-x), \tag{41}$$

which is the Taylor expansion of $-\ln(1-x)$ for $x \in [0,1)$.

To prove the general case, let us first derive an alternative form of $\mathrm{I}_{d-r,r}(x)$. For $a > 0$,

$$\begin{aligned}
&\mathrm{I}_{a,b}(x)\\
&=\sum_{j=a}^{a+b-1}\binom{a+b-1}{j}x^j\sum_{i=0}^{a+b-1-j}(-1)^i\binom{a+b-1-j}{i}x^i\\
&=\sum_{m=a}^{a+b-1}x^m\sum_{j=a}^{m}\binom{a+b-1}{j}(-1)^{m-j}\binom{a+b-1-j}{m-j}\\
&=\sum_{m=a}^{a+b-1}(-x)^m\binom{a+b-1}{m}\sum_{j=a}^{m}\binom{m}{j}(-1)^j\\
&=\sum_{m=a}^{a+b-1}(-x)^m\binom{a+b-1}{m}\binom{m-1}{a-1}(-1)^a\\
&=b\binom{a+b-1}{b}(-1)^a\sum_{m=a}^{a+b-1}\frac{(-x)^m}{m}\binom{b-1}{m-a}.
\end{aligned}$$

Using this form for $\mathrm{I}_{d-r,r}(x)$, we have

$$\begin{aligned}
&\sum_{d=r+1}^{\infty}\frac{1}{d-1}\mathrm{I}_{d-r,r}(x)\\
&=\sum_{d=r+1}^{\infty}\frac{r}{d-1}\binom{d-1}{r}\sum_{m=d-r}^{d-1}\binom{r-1}{m-d+r}(-1)^{m-d+r}\frac{x^m}{m}\\
&=\sum_{m=1}^{\infty}\frac{x^m}{m}A_m, && (42)
\end{aligned}$$

where

$$A_m \triangleq \sum_{d=\max\{m,r\}+1}^{m+r}\frac{r}{d-1}\binom{d-1}{r}\binom{r-1}{m-d+r}(-1)^{m-d+r}.$$

For $m \leq r$,

$$\begin{aligned}
A_m &= \sum_{d=r+1}^{m+r}\frac{r}{d-1}\binom{d-1}{r}\binom{r-1}{m-d+r}(-1)^{m-d+r}\\
&= \sum_{d=r+1}^{m+r}\binom{d-2}{r-1}\binom{r-1}{m-d+r}(-1)^{m-d+r}\\
&= \sum_{j=0}^{m-1}\binom{j+r-1}{r-1}\binom{r-1}{m-j-1}(-1)^{m-j-1}\\
&= \sum_{j=0}^{m-1}\binom{j+r-1}{m-1}\binom{m-1}{m-j-1}(-1)^{m-j-1}\\
&= 1,
\end{aligned}$$

where the last equality follows from (37). Similarly, for $m > r$,

$$\begin{aligned}
A_m &= \sum_{d=m+1}^{m+r}\frac{r}{d-1}\binom{d-1}{r}\binom{r-1}{m-d+r}(-1)^{m-d+r}\\
&= \sum_{d=m+1}^{m+r}\binom{d-2}{r-1}\binom{r-1}{m-d+r}(-1)^{m-d+r}\\
&= \sum_{j=0}^{r-1}\binom{j+m-1}{r-1}\binom{r-1}{r-j-1}(-1)^{r-j-1}\\
&= 1.
\end{aligned}$$

The proof is completed by referring to (41) and (42) with $A_m = 1$. ■

Appendix B
Layered Decoding Graph

We have discussed different decoding strategies under the rule that a check node is decodable if and only if its rank equals its degree. We say a variable node is decodable if it is connected to a decodable check node. In Section II-C, a decodable check node is chosen and all its neighbors (variable nodes) are recovered simultaneously, while in Section III-A, a decodable variable node is uniformly chosen to be recovered. Here we show that under the decoding rule that a check node is decodable if and only if its rank equals its degree, both strategies stop with the same subset of the variable nodes undecoded.

For a given decoding graph $\mathcal{G}$, let $\mathcal{G}^0 = \mathcal{G}$. Label by L_1 all the decodable check nodes in $\mathcal{G}^0$ and label by L_2 all the variable nodes in $\mathcal{G}^0$ connected to the check nodes with label L_1. We repeat the above procedure as follows. For $i = 1, 2, \ldots$, let $\mathcal{G}^i$ be the subgraph of $\mathcal{G}$ obtained by removing all the nodes with labels L_j for $j \leq 2i$, as well as the adjacent edges. (The generator matrices of the check nodes are also updated.) Label by L_{2i+1} all the decodable check nodes in $\mathcal{G}^i$

and label by L_{2i+2} all the variable nodes in $\mathcal{G}^i$ connected to the check nodes with label L_{2i+1}. This procedure stops when $\mathcal{G}^i$ has no more decodable check nodes. Let i_0 be the index where the procedure stops. The above labelling procedure is deterministic and generates unique labels for each decodable variable nodes and check nodes.

With the labels, we can generate a layered subgraph $\mathcal{G}'$ of $\mathcal{G}$. In $\mathcal{G}'$, layer j, $j = 1, 2, \ldots, 2i_0$, contains all the check/variable nodes with label L_j. Only the edges connecting two nodes belonging to two consecutive layers are preserved in $\mathcal{G}'$. By the assigning rule of the labels, it is clear that a variable node on layer $2i$ must connect to one check node on layer $2i-1$, $i = 1, \ldots, i_0$, because otherwise the variable node is not decodable. Further, a check node on layer $2i+1$ must connect to some variable nodes on layer $2i$, $i = 1, \ldots, i_0 - 1$, because otherwise the check node should be on layer $2i-1$.

By the definition of decodability, a decoding strategy must process the variable/check nodes in $\mathcal{G}'$ following an order such that a variable/check node is processed after all its lower layer descendant variable/check nodes have been processed. The two random decoding strategies we have discussed in Section II-C and Section III-A both can process all the nodes in $\mathcal{G}'$ before stopping.

Appendix C
Solving the system of differential equations

We solve the following system of differential equations given in (10) and (11), which is reproduced as follows:

$$\begin{aligned}\frac{d\rho_{d,r}(\tau)}{d\tau} &= (\alpha_{d+1,r}\rho_{d+1,r}(\tau) + \bar{\alpha}_{d+1,r+1}\rho_{d+1,r+1}(\tau)\\ &\quad -\rho_{d,r}(\tau))\frac{d}{\theta-\tau}, \quad 1 \le r \le M,\ r < d \le D,\\ \frac{d\rho_0(\tau)}{d\tau} &= \frac{\sum_{r=1}^{D-1} r\alpha_{r+1,r}\rho_{r+1,r}(\tau) - \rho_0(\tau)}{\theta-\tau} - 1\end{aligned}$$

with initial values $\rho_{d,r}(0) = \rho_{d,r}$ and $\rho_0(0) = \sum_r \rho_{r,r}$.

Let $y_{d,r}(\tau) = (1-\tau/\theta)^{-d}\rho_{d,r}(\tau)$. We have

$$\frac{dy_{d,r}(\tau)}{d\tau} = \frac{d}{\theta}(\alpha_{d+1,r}y_{d+1,r}(\tau) + \bar{\alpha}_{d+1,r+1}y_{d+1,r+1}(\tau)).$$

We see that $y_{d,r}(0) = \rho_{d,r}(0)$. Define

$$\rho_{d,r}^{(0)} \triangleq \rho_{d,r}, \tag{43}$$

$$\rho_{d,r}^{(i+1)} \triangleq \alpha_{d-i,r}\rho_{d,r}^{(i)} + \bar{\alpha}_{d-i,r+1}\rho_{d,r+1}^{(i)}; \tag{44}$$

We can verify that

$$y_{d,r}(\tau) = \sum_{j=d}^{D}\binom{j-1}{d-1}(\tau/\theta)^{j-d}\rho_{j,r}^{(j-d)}.$$

Thus

$$\rho_{d,r}(\tau) = (1-\tau/\theta)^d\sum_{j=d}^{D}\binom{j-1}{d-1}(\tau/\theta)^{j-d}\rho_{j,r}^{(j-d)}. \tag{45}$$

Using the general solution of linear differential equations, we obtain that

$$\begin{aligned}&\rho_0(\tau)\\ &= (1-\tau/\theta)\Bigg(\int_0^\tau \frac{\sum_{r=1}^{M} r\alpha_{r+1,r}\rho_{r+1,r}(t)}{\theta-t}(1-t/\theta)^{-1}dt\\ &\qquad +\theta\ln(1-\tau/\theta) + \rho_0(0)\Bigg)\\ &= (1-\tau/\theta)\Bigg(\sum_{r=1}^{M} r\alpha_{r+1,r}\int_0^\tau \frac{\rho_{r+1,r}(t)}{\theta-t}(1-t/\theta)^{-1}dt\\ &\qquad +\theta\ln(1-\tau/\theta) + \rho_0(0)\Bigg).\end{aligned} \tag{46}$$

The integral in (46) can be further calculated as follows:

$$\begin{aligned}&\int_0^\tau \frac{\rho_{r+1,r}(t)}{\theta-t}(1-t/\theta)^{-1}dt\\ &= \int_0^\tau \frac{\sum_{j=r+1}^{D}\rho_{j,r}^{(j-r-1)}\binom{j-1}{r}(1-t/\theta)^{r+1}(t/\theta)^{j-r-1}}{(\theta-t)(1-t/\theta)}dt\\ &= \int_0^\tau \sum_{j=r+1}^{D}\rho_{j,r}^{(j-r-1)}\binom{j-1}{r}(1-t/\theta)^{r-1}(t/\theta)^{j-r-1}\frac{dt}{\theta}\\ &= \sum_{j=r+1}^{D}\rho_{j,r}^{(j-r-1)}\binom{j-1}{r}\int_0^{\tau/\theta}(1-t)^{r-1}t^{j-r-1}dt\\ &= \sum_{j=r+1}^{D}\rho_{j,r}^{(j-r-1)}\binom{j-1}{r}\frac{(j-r-1)!(r-1)!}{(j-1)!}\mathrm{I}_{j-r,r}(\tau/\theta)\\ &= 1/r\sum_{j=r+1}^{D}\rho_{j,r}^{(j-r-1)}\mathrm{I}_{j-r,r}(\tau/\theta),\end{aligned}$$

where the first equality is obtained by substituting $\rho_{r+1,r}(t)$ in (45), and the second last equality is obtained by the definition of incomplete beta function (cf. (34)). Therefore, the solution for $\rho_0(\tau)$ is

$$\begin{aligned}\rho_0(\tau) = \left(1-\frac{\tau}{\theta}\right)&\left(\sum_{r=1}^{M}\alpha_{r+1,r}\sum_{d=r+1}^{D}\rho_{d,r}^{(d-r-1)}\mathrm{I}_{d-r,r}\left(\frac{\tau}{\theta}\right)\right.\\ &\left.+\sum_{r=1}^{M}\rho_{r,r} + \theta\ln(1-\tau/\theta)\right).\end{aligned}$$

The above formula of $\rho_0(\tau)$ can be simplified by substituting the quantities defined in (43) and (44) recursively. Denote by G_d a totally random $d \times M$ matrix and let H be an M-row random matrix with $\mathrm{rk}(H)$ following the distribution of the probability vector h. Let $G_d^{(0)}$ be G_d and for $t > 0$, $G_d^{(t)}$ be the submatrix of $G_d^{(t-1)}$ with a row deleted. We now show that

$$\rho_{d,r}^{(t)} = d\Psi_d \Pr\{\mathrm{rk}(G_d^{(t)}H) = r\}. \tag{47}$$

By (3), we have

$$\rho_{d,r}^{(0)} = \rho_{d,r} = d\Psi_d \Pr\{\mathrm{rk}(G_dH) = r\}.$$

Assume that (47) holds for all times up to $t-1$. By the definition of $\alpha_{d,r}$ in Lemma 2, we have

$$\begin{aligned}\alpha_{d-t+1,r} &= \Pr\{\mathrm{rk}(G_d^{(t)}H) = r|\mathrm{rk}(G_d^{(t-1)}H) = r\},\\ \bar{\alpha}_{d-t+1,r+1} &= \Pr\{\mathrm{rk}(G_d^{(t)}H) = r|\mathrm{rk}(G_d^{(t-1)}H) = r+1\}.\end{aligned}$$

Then,

$$\begin{aligned}\rho_{d,r}^{(t)} &= \alpha_{d-t+1,r}\rho_{d,r}^{(t-1)} + \bar{\alpha}_{d-t+1,r+1}\rho_{d,r+1}^{(t-1)}\\ &= \alpha_{d-t+1,r} d\Psi_d \Pr\{\mathrm{rk}(G_d^{(t-1)}H) = r\}\\ &\quad +\bar{\alpha}_{d-t+1,r+1} d\Psi_d \Pr\{\mathrm{rk}(G_d^{(t-1)}H) = r+1\} \quad (48)\\ &= d\Psi_d \Pr\{\mathrm{rk}(G_d^{(t)}H) = r\},\end{aligned}$$

where (48) follows from the induction hypothesis.

In the expression for $\rho_0(\tau)$, we have

$$\begin{aligned}\alpha_{r+1,r}\rho_{d,r}^{(d-r-1)} &= d\Psi_d\alpha_{r+1,r}\Pr\{\mathrm{rk}(G_d^{(d-r-1)}H) = r\}\\ &= d\Psi_d\alpha_{r+1,r}\Pr\{\mathrm{rk}(G_{r+1}H) = r\}\\ &= d\Psi_d\Pr\{\mathrm{rk}(G_{r+1}^{(1)}H) = \mathrm{rk}(G_{r+1}H) = r\}\\ &= d\Psi_d\sum_{i=r}^{M}\frac{\zeta_r^i}{q^{i-r}}h_i,\end{aligned}$$

where the last equality is obtained by

$$\begin{aligned}&\Pr\{\mathrm{rk}(G_{r+1}^{(1)}H) = \mathrm{rk}(G_{r+1}H) = r\}\\ &\quad = \sum_{k\geq r}\Pr\{\mathrm{rk}(G_{r+1}^{(1)}H) = \mathrm{rk}(G_{r+1}H) = r|\mathrm{rk}(H) = k\}h_k\\ &\quad = \sum_{k\geq r}\Pr\{\mathrm{rk}(G_{r+1}H) = r|\mathrm{rk}(G_{r+1}^{(1)}H) = r, \mathrm{rk}(H) = k\}\\ &\qquad\quad \times\Pr\{\mathrm{rk}(G_{r+1}^{(1)}H) = r|\mathrm{rk}(H) = k\}h_k\\ &\quad = \sum_{k\geq r}\frac{\zeta_r^k}{q^{k-r}}h_k.\end{aligned}$$

Define $(\hbar_r = \hbar_r(h), r = 1, \ldots, M)$ for a rank distribution h as

$$\hbar_r(h) = \sum_{i=r}^{M}\frac{\zeta_r^i}{q^{i-r}}h_i.$$

Since

$$\begin{aligned}\sum_{k=r}^{M}\zeta_r^k h_k &= \Pr\{\mathrm{rk}(G_rH) = r\}\\ &= \Pr\{\mathrm{rk}(G_{r+1}^{(1)}H) = r, \mathrm{rk}(G_{r+1}H) = r\}\\ &\quad + \Pr\{\mathrm{rk}(G_{r+1}^{(1)}H) = r, \mathrm{rk}(G_{r+1}H) = r+1\}\\ &= \hbar_r + \Pr\{\mathrm{rk}(G_{r+1}H) = r+1\}\\ &= \sum_{s=r}^{M}\hbar_s, \qquad (49)\end{aligned}$$

we can write $\rho_{r,r} = r\Psi_r\sum_{s=r}^{M}\hbar_s$. Using the above notations, we can simplify the expression for $\rho_0(\tau)$ as

$$\begin{aligned}\rho_0(\tau) = \left(1-\frac{\tau}{\theta}\right)&\left(\sum_{r=1}^{M}\sum_{d=r+1}^{D}d\Psi_d\hbar_r\mathrm{I}_{d-r,r}\left(\frac{\tau}{\theta}\right)\right.\\ &\left.+\sum_{r=1}^{M}r\Psi_r\sum_{s=r}^{M}\hbar_s + \theta\ln(1-\tau/\theta)\right).\end{aligned}$$

References

[1] M. Luby, "LT codes," in *Proc. 43rd Ann. IEEE Symp. on Foundations of Computer Science*, Nov. 2002, pp. 271–282.

[2] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[3] P. Maymounkov, "Online codes," NYU, Tech. Rep., Nov. 2002.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[5] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[6] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[7] T. Ho, B. Leong, M. Medard, R. Koetter, Y. Chang, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE ISIT '03*, Jun. 2003.

[8] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton Conf. Comm., Control, and Computing*, Oct. 2003.

[9] S. Jaggi, P. A. Chou, and K. Jain, "Low complexity optimal algebraic multicast codes," in *Proc. IEEE ISIT '03*, Jun. 2003.

[10] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *Proc. ACM SPAA '03*, New York, NY, USA, 2003, pp. 286–294.

[11] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20, 2008.

[12] Y. Wu, "A trellis connectivity analysis of random linear network coding with buffering," in *Proc. IEEE ISIT '06*, Seattle, USA, Jul. 2006.

[13] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 789–804, 2006.

[14] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 169–180, Aug. 2007.

[15] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE INFOCOM '05*, 2005.

[16] M.Wang and B.Li, "Lava: A reality check of network coding in peer-to-peer live streaming," in *Proc. IEEE INFOCOM '07*, 2007.

[17] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. SIGCOMM '06*, New York, NY, USA, 2006.

[18] P. Maymounkov, N. J. A. Harvey, and D. S. Lun, "Methods for efficient network coding," in *Proc. Allerton Conf. Comm., Control, and Computing*, Sep. 2006.

[19] S. Yang and R. W. Yeung, "Large file transmission in network-coded networks with packet loss – A performance perspective," in *Proc. ISABEL 2011*, Barcelona, Spain, Oct. 26-29 2011.

[20] D. Silva, W. Zeng, and F. R. Kschischang, "Sparse network coding with overlapping classes," in *Proc. NetCod '09*, 2009, pp. 74–79.

[21] A. Heidarzadeh and A. H. Banihashemi, "Overlapped chunked network coding," in *Proc. ITW '10*, 2010, pp. 1–5.

[22] Y. Li, E. Soljanin, and P. Spasojevic, "Effects of the generation size and overlap on throughput and complexity in randomized linear network coding," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 1111–1123, Feb. 2011.

[23] B. Tang, S. Yang, Y. Yin, B. Ye, and S. Lu, "Expander graph based overlapped chunked codes," in *Proc. IEEE ISIT '12*, Cambridge, MA, USA, Jul. 1-6 2012.

[24] K. Mahdaviani, M. Ardakani, H. Bagheri, and C. Tellambura, "Gamma codes: a low-overhead linear-complexity network coding solution," in *Proc. NetCod '12*, 2012, pp. 125–130.

[25] K. Mahdaviani, R. Yazdani, and M. Ardakani, "Overhead-optimized gamma network codes," in *Proc. NetCod '13*, 2013.

[26] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," in *Proc. IEEE ISIT '05*, 2005, pp. 1853–1857.

[27] R. Gummadi and R. Sreenivas, "Relaying a fountain code across multiple nodes," in *Proc. IEEE ITW '08*, May 2008, pp. 149 –153.

[28] M.-L. Champel, K. Huguenin, A.-M. Kermarrec, and N. Le Scouarnec, "LT network codes," in *Proc. IEEE ICDCS '10*, Jun. 2010.

[29] N. Thomos and P. Frossard, "Degree distribution optimization in Raptor network coding," in *Proc. IEEE ISIT '11*, Aug. 2011.

[30] M. Jafari, L. Keller, C. Fragouli, and K. Argyraki, "Compressed network coding vectors," in *Proc. IEEE ISIT '09*, 2009, pp. 109–113.

[31] C. Fragouli, D. Lun, M. Medard, and P. Pakzad, "On feedback for network coding," in *Proc. CISS '07*, 2007, pp. 248–252.

[32] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Proc. NetCod '08*, 2008, pp. 1–6.

[33] J. Sundararajan, D. Shah, and M. Medard, "ARQ for network coding," in *Proc. IEEE ISIT '08*, 2008, pp. 1651–1655.

[34] S. Jaggi, Y. Cassuto, and M. Effros, "Low complexity encoding for network codes," in *Proc. IEEE ISIT '06*, 2006, pp. 40–44.

[35] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.

[36] D. Silva, F. R. Kschischang, and R. Koetter, "Communication over finite-field matrix channels," *IEEE Trans. Inform. Theory*, vol. 56, no. 3, pp. 1296–1305, Mar. 2010.

[37] S. Yang, J. Meng, and E.-h. Yang, "Coding for linear operator channels over finite fields," in *Proc. IEEE ISIT '10*, Austin, USA, Jun.13-18 2010.

[38] N. C. Wormald, "The differential equation method for random graph processes and greedy algorithms," *Karonsky and Proemel, eds., Lectures on Approximation and Randomized Algorithms PWN, Warsaw*, pp. 73–155, 1999.

[39] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[40] T. Richardsan and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Cambridge, England, 2008.

[41] G. E. Andrews, *The theory of partitions*, ser. vol. 2, Encyclopedia of mathematics and its applications. Addison-Wesley Pub. Co., 1976.

[42] M. Gadouleau and Z. Yan, "Packing and covering properties of subspace codes for error control in random linear network coding," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2097 –2108, may 2010.

[43] N. C. Wormald, "Differential equations for random processes and random graphs," *Ann. Appl. Probab.*, no. 5, pp. 1217–1235, 1995.

[44] N. A. Smith and R. W. Tromble, "Sampling uniformly from the unit simplex." [Online]. Available: http://www.cs.cmu.edu/~nasmith/papers/smith+tromble.tr04.pdf

[45] S. Yang and Y. Chen, "Degree-distribution optimizations of BATS codes," 2013, [Online]. Available: http://iiis.tsinghua.edu.cn/~shenghao/pub/ddo.pdf

[46] A. Shokrollahi and M. Luby, *Raptor Codes*, ser. Foundations and Trends in Communications and Information Theory. now, 2011, vol. 6.

[47] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder, "RaptorQ forward error correction scheme for object delivery – rfc 6330," 2011. [Online]. Available: http://datatracker.ietf.org/doc/rfc6330/

[48] B. A. LaMacchia and A. M. Odlyzko, "Solving large sparse linear systems over finite fields," in *Proc. CRYPTO'90*. Springer, 1991, pp. 109–133.

[49] C. Pomerance and J. W. Smith, "Reduction of huge, sparse matrices over finite fields via created catastrophes," *Experimental Math*, vol. 1, no. 89-94, 1992.

[50] 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.

[51] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 638–656, 2001.

[52] T. C. Ng and S. Yang, "Finite length analysis of BATS codes," in *Proc. IEEE NetCod '13*, Calgary, Canada, Jun. 7-9 2013.

[53] ——, "Finite-length analysis of BATS codes," 2013, submitted for journal publication. [Online]. Available: http://arxiv.org/abs/1312.4811

[54] M. Zelen and N. C. Severo, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, M. Abramowitz and I. A. Stegun, Eds. New York: Dover, 1972.

Shenghao Yang (S'07-M'11) received the B.S. degree in Electronics Engineering from Nankai University in 2001, M.Eng. degree in Electronics Engineering from Peking University in 2004, and Ph.D. degree in Information Engineering from The Chinese University of Hong Kong in 2008.

He was a Postdoctoral Fellow in Department of Electrical and Computer Engineering, University of Waterloo from 2008 to 2009, and a Postdoctoral Fellow/Research Associate in Institute of Network Coding, The Chinese University of Hong Kong from 2010 to 2012. He is currently an Assistant Professor with Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China.

Raymond W. Yeung (S'85-M'88-SM'92-F'03) was born in Hong Kong on June 3, 1962. He received the B.S., M.Eng., and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY, in 1984, 1985, and 1988, respectively.

He was on leave at Ecole Nationale Supérieure des Télécommunications, Paris, France, during fall 1986. He was a Member of Technical Staff of AT&T Bell Laboratories from 1988 to 1991. Since 1991, he has been with the Department of Information Engineering, The Chinese University of Hong Kong, where he is now a chair professor. He is also a Changjiang Chair Professor at Xidian University (2009-12) and an Advisory Professor at Beijing University of Post and Telecommunications (2008-11). He has held visiting positions at Cornell University, Nankai University, the University of Bielefeld, the University of Copenhagen, Tokyo Institute of Technology, and Munich University of Technology. He was a Consultant in a project of Jet Propulsion Laboratory, Pasadena, CA, for salvaging the malfunctioning Galileo Spacecraft and a Consultant for NEC, USA.

He is the author of the textbooks *A First Course in Information Theory* (Kluwer Academic/Plenum 2002) and its revision *Information Theory and Network Coding* (Springer 2008). His research interests include information theory and network coding.

Dr. Yeung was a member of the Board of Governors of the IEEE Information Theory Society from 1999 to 2001. He has served on the committees of a number of information theory symposiums and workshops. He was General Chair of the First and the Fourth Workshop on Network, Coding, and Applications (NetCod 2005 and 2008), a Technical Co-Chair for the 2006 IEEE International Symposium on Information Theory, and a Technical Co-Chair for the 2006 IEEE Information Theory Workshop, Chengdu, China. He currently serves as an Editor-at-Large of *Communications in Information and Systems*, an Editor of *Foundation and Trends in Communications and Information Theory* and of *Foundation and Trends in Networking*, and was an Associate Editor for Shannon Theory of this Transactions from 2003 to 2005. He was a recipient of the Croucher Foundation Senior Research Fellowship for 2000/2001, the Best Paper Award (Communication Theory) of the 2004 International Conference on Communications, Circuits and System (with C. K. Ngai), the 2005 IEEE Information Theory Society Paper Award (for his paper "Linear network coding" co-authored with S.-Y. R. Li and N. Cai), and the Friedrich Wilhelm Bessel Research Award of the Alexander von Humboldt Foundation in 2007. He is a Fellow of the IEEE and the Hong Kong Institution of Engineers.

Since January 2010, Dr. Yeung has been serving as Co-Director of the Institute of Network Coding at The Chinese University of Hong Kong.