

Coding for a Network Coded Fountain

Shenghao Yang* and Raymond W. Yeung*†

*Institute of Network Coding, †Department of Information Engineering
The Chinese University of Hong Kong, Hong Kong SAR, China
Email: shyang@inc.cuhk.edu.hk, whyeung@ie.cuhk.edu.hk

Abstract—Batched sparse (BATS) codes are proposed for transmitting a collection of packets through communication networks employing linear network coding. BATS codes generalize fountain codes and preserve the properties such as ratelessness and low encoding/decoding complexity. Moreover, the buffer size and the computation capability of the intermediate network nodes required to apply BATS codes are independent of the number of packets for transmission. It is verified theoretically for certain cases and demonstrated numerically for the general cases that BATS codes achieve rates very close to the capacity of linear operator channels.

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. Fountain codes, including LT codes [1] and Raptor codes [2], provide a good solution for routing networks, where the intermediate nodes apply store-and-forward. In a routing network, the transmission from the source node to a destination node is modelled by an erasure channel. 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 is just slightly larger than the number of the input packets K , without any knowledge of which packets are received. The encoding/decoding complexity of a Raptor code is $\mathcal{O}(TK)$, where T is the size of a packet.

Store-and-forward, however, is not an optimal operation of the intermediate nodes. For example, the routing capacity of the network in Fig. 1 is 0.64 packet per use. If we allow coding (recoding) at the intermediate node and treat the network as a concatenation of two erasure channels, we see that the capacity of the network is 0.8 packet per use. Treating a packet as a vector of symbols in a finite field, linear network coding [3] allows intermediate nodes to transmit linear combinations of the packets it has received. Random linear network coding [4] achieves the multicast capacity in terms of packets per use [5]. For example, the capacity of the network in Fig. 1 can be achieved using the following random linear network coding scheme [6]. The source node s transmits random linear combinations of the input packets and node a transmits random linear combinations of the packets it has received. The destination node t can decode the input packets when it receives enough coded packets with linearly independent coding vectors. See also [7, Section 19.7] for a discussion.

The random linear network coding scheme for the above example has high complexity. For transmitting a file of K packets, the total encoding complexity is $\mathcal{O}(TK^2)$ and the

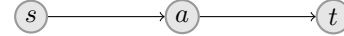


Fig. 1. s is the source node, t is the destination node, and a is the intermediate node that does not demand the file. Both links are capable of transmitting one packet per use and have a packet loss rate 0.2.

decoding complexity is $\mathcal{O}(K^3 + TK^2)$. Besides the encoding/decoding complexity, the complexity of intermediate operations is important though ignored by many works. For an intermediate node in a routing network, no matter how many packets the file contains, the buffer size and the number of the operations for processing a packet remain constant. However, for the network coding scheme, the intermediate node needs to buffer almost K packets and the coding complexity of the intermediate nodes is $\mathcal{O}(TK)$.

Applying fountain codes to networks employing linear network coding cannot reduce the complexity. First, simply using fountain codes in the source node cannot reduce the coding complexity and the buffer size of the intermediate nodes. Second, the fast belief propagation decoding algorithm of fountain codes usually fails after re-coding of the fountain coded packets. Designing “sparse” network coding to approximate fountain codes may be possible for special communication scenarios [8], but in general it is difficult to guarantee that the degree of the received packets follows a specific distribution using distributed encoding.

One practical method to simplify the complexity of network coding is to group input packets into *chunks* (also called generations or classes), a subset of the K packets. Encoding, recoding and decoding are all performed within one chunk. It reduces the encoding and decoding complexity to $\mathcal{O}(TKL)$ and $\mathcal{O}(KL^2 + TKL)$, respectively, where chunks are disjoint and have size L . Maymounkov et al. [9] showed that random scheduling of chunks achieves the min-cut bound when the chunk size goes to infinity. Two groups have independently shown by simulation that using overlapped chunks can improve the throughput [10], [11] of random scheduling for practical chunk sizes. Intuitively, the advantage of overlapped chunks is to use the decoded chunks to help the decoding of the other chunks. Overlapped chunks have some properties similar to fountain codes, but the existing designs of overlapping are still based on heuristics. Moreover, these file distribution schemes [8]–[12] require the intermediate nodes to buffer the whole file.

We propose *batched sparse (BATS) codes*, which extend the idea of fountain codes to the realm of networks and

also take the advantages of network coding. BATS codes are fully compatible with linear network coding by employing a new design freedom called *batch*. A batch is a set of M packets generated by using a same subset of input packets. The encoding complexity of a BATS code is $\mathcal{O}(TKM)$ and the corresponding decoding complexity is $\mathcal{O}(KM^2 + TKM)$. Moreover, when applying BATS codes, an intermediate node uses $\mathcal{O}(TM)$ time to recode a packet and buffers $\mathcal{O}(M)$ packets.

As an end-to-end coding scheme working at the source and destination nodes, BATS codes are suitable for a large range of networks as long as the end-to-end operation on the packets of a batch is a linear transformation, which can be different for different batches. BATS codes are robust against dynamical network topology and packet loss since the end-to-end operation remains linear. Moreover, BATS codes work with random linear network coding with small finite fields. Most existing works on random linear network coding requires a large field size to guarantee a full rank for the transfer matrix. For BATS codes, however, the transfer matrices of the batches are allowed to have arbitrary rank deficiency.

The linear transformation on batches can be modelled by a linear operator channel (LOC), a channel model studied for linear network coding [13]. We verify theoretically for certain cases and demonstrate numerically for the general cases that BATS codes achieve rates very close to the capacity of memoryless LOCs. The batch size M determines the tradeoff between the complexity and the maximum achievable rate. When $M = 1$, BATS codes degenerate to LT codes, which have the lowest complexity but without the benefit of network coding. When $M = K$, BATS codes has the same complexity of random linear network coding, and at the same time the potential of network coding can be fully realized.

II. BATCHED SPARSE (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 the following discussion, we equate a set of packets to the matrix formed by juxtaposing the packets in this set. For example, the set of the input packets is denoted by the matrix

$$\mathbf{B} = [b_1, b_2, \dots, b_K],$$

where b_i is the i th input packets. When treating as a set, we also write $b_i \in \mathbf{B}$, $\mathbf{B}' \subset \mathbf{B}$, etc. We use $\text{rk}(\mathbf{A})$ to denote the matrix rank of \mathbf{A} .

A. Encoding of Batches

A *batch* is a set of M coded packets generated from a subset of these input packets. For $i = 1, 2, \dots$, the i th batch \mathbf{X}_i is generated using a subset $\mathbf{B}_i \subset \mathbf{B}$ of the input packets as

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

where \mathbf{G}_i is called the *generator matrix* of the i th batch. We call the packets in \mathbf{B}_i the contributors of the i th batch. The formation of \mathbf{B}_i depends on a *degree distribution* $\Psi =$

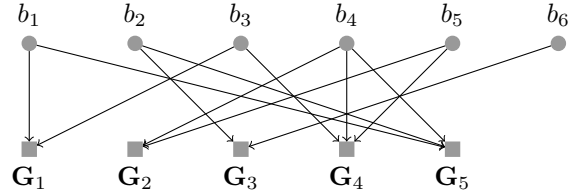


Fig. 2. Tanner graph for encoding and transmitting of the first five batches.

$(\Psi_0, \Psi_1, \dots, \Psi_K)$: First sample the distribution Ψ which returns a degree d_i with probability Ψ_{d_i} ; Then uniformly at random choose d_i input packets forming \mathbf{B}_i . The design of Ψ is discussed later in Section III.

The dimension of \mathbf{G}_i is $d_i \times M$. There are two options for designing \mathbf{G}_i . i) \mathbf{G}_i are pre-designed. ii) \mathbf{G}_i are generated on the fly. In this paper, we analyze BATS codes with random generator matrices, i.e., all the components of \mathbf{G}_i are independently chosen, uniformly at random by the encoder. Random generation matrix is not only good for analysis, but also implementable. E.g., \mathbf{G}_i , $i = 1, 2, \dots$, can be generated by a pseudorandom generator and can be recovered in the destinations by the same pseudorandom generator.

The encoding of BATS codes can be described by Tanner graphs. A Tanner graph has K *variable nodes*, where the variable node i corresponds to the i th input packet b_i , and n *check nodes*, where the check node j corresponds to the j th batch \mathbf{X}_j . Check node j is connected to variable node i if b_i is a contributor of batch j . Fig. 2 illustrates an example of Tanner graph for encoding.

B. Transmission of Batches

To transmit a batch, the source node transmits the packets in the batch. No feedback is required to stop the transmission of each batch. BATS codes are rateless codes, i.e., the number of batches that can be transmitted is not fixed. When applying linear network coding, an intermediate node encodes the received packets of a batch into new packets using linear combinations and transmit these new packets on its outgoing links. These new packets are considered to be in the same batch. The rule is that the packets in different batches are not mixed inside the network. BATS codes are robust against dynamical network topology and packet loss since the end-to-end operation remains linear.

To apply BATS codes, we further need to consider how to schedule the transmission of batches in the source node and the intermediate nodes and how to manage the buffers at the intermediate nodes. The design of these network operations varies for different applications. For example, for the file transmission in a directed acyclic network, when the intermediate network nodes do not require the file, sequential scheduling of batches at the source node and the intermediate nodes can minimize the buffer requirement at the intermediate nodes. In contrast, for the file distribution in a peer-to-peer network, since all network nodes request the file, random scheduling of batches can reduce the protocol overhead.

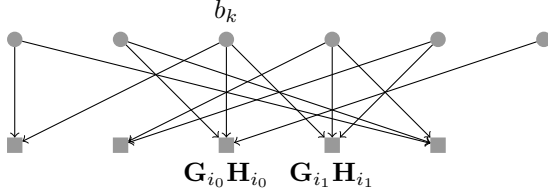


Fig. 3. A decoding graph

Given the end-to-end transformations applied to the batches, the design of BATS codes does not depend on the details of the network operations. So we will not discuss details of the network operations on the batches in this paper. Nevertheless, we demonstrate how a BATS code works in the three-node network in Section IV-B, where some general guidelines on the design of the intermediate operation are given. Note that though the three-node network is simple, it arises in some important applications, e.g. wireless relay networks.

Let \mathbf{Y}_i be the received packets at a destination node that belong to the i th batch. We have

$$\mathbf{Y}_i = \mathbf{X}_i \mathbf{H}_i = \mathbf{B}_i \mathbf{G}_i \mathbf{H}_i, \quad (1)$$

where \mathbf{H}_i is the transfer matrix of the i th batch determined by linear network coding [3], [14]. The number of rows of \mathbf{H}_i is M , while the number of columns varies for different batches. We assume that $\mathbf{G}_i \mathbf{H}_i$ is known by the destination node through the coding vectors in the packet headers. When the packet length T is sufficiently large, this overhead is negligible.

The operation of the network on the batches in (1) can be modelled as a *linear operator channel (LOC)*, which has been studied for linear network coding [13]. In this sense, a BATS code is a channel code for LOCs. If \mathbf{H}_i , $i = 1, 2, \dots$, are independent instances of a generic random matrix H , then the LOC is a memoryless channel with transfer matrix H . We will analyze the performance of BATS codes for memoryless LOCs. Unless otherwise specified, the LOCs referred to in the rest of this paper are memoryless.

C. Belief Propagation (BP) Decoding

A destination tries to decode the input packets using \mathbf{Y}_i and the knowledge of $\mathbf{G}_i \mathbf{H}_i$ for $i = 1, 2, \dots, n$. The decoding process is better described using the bipartite graph in Fig. 3, which is the same with the encoding graph except that associated with each check node i is the matrix $\mathbf{G}_i \mathbf{H}_i$.

A check node i is called decodable if $\mathbf{G}_i \mathbf{H}_i$ has rank d_i . Assume that check node i_0 is decodable. Then \mathbf{B}_{i_0} is recovered by solving the linear system of equations $\mathbf{Y}_{i_0} = \mathbf{B}_{i_0} \mathbf{G}_{i_0} \mathbf{H}_{i_0}$, which must have a unique solution since $\text{rk}(\mathbf{G}_{i_0} \mathbf{H}_{i_0}) = d_{i_0}$. After decoding the i_0 th batch, we recover the d_{i_0} input packets in \mathbf{B}_{i_0} . Then substitute the values of these input packets in the undecoded batches. Consider that b_k is in \mathbf{B}_{i_0} . If variable node k has only one edge that connects with check node i_0 , just remove variable node k . If variable node k also connects check node $i_1 \neq i_0$, then besides removing the node, also

remove the row in $\mathbf{G}_{i_1} \mathbf{H}_{i_1}$ corresponding to variable node k . In the decoding graph, this is equivalent to first removing check node i_0 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.

In Section III, we give a sufficient condition of the degree distribution such that the BP decoding of BATS codes succeeds with high probability.

D. Precoding of BATS Codes

The same technique of Raptor codes is applied here to reduce the encoding/decoding complexity of BATS codes. The input packets are first encoded using a traditional erasure code (precoding), and then encoded by a BATS code. We require that the belief propagation decoding of BATS codes recovers a constant fraction of its input packets. The traditional erasure code is capable of recovering the original input packets in face of a fixed fraction of erasures.

E. Complexity

The complexity of encoding a batch with degree d is $\mathcal{O}(TMd)$. For a encoding graph with n check nodes, i.e., n batches, the encoding complexity is $\mathcal{O}(TM \sum_{i=1}^n d_i) = \mathcal{O}(TMm)$, where $m = \sum_{i=1}^n d_i$ is the number of edges in the decoding graph, which converges to $\mathcal{O}(TMn \mathbb{E}[\Psi])$ when n is large where $\mathbb{E}[\Psi] = \sum_d d \Psi_d$.

Let $k_i = \text{rk}(\mathbf{H}_i)$ and k'_i be the rank of $\mathbf{G}_i \mathbf{H}_i$ when check node i is decodable. It is clear that $k'_i \leq k_i \leq M$. The computation involved in the decoding includes two parts: the first part is the decoding of decodable check nodes, which has complexity $\mathcal{O}(\sum_i k_i'^3 + T \sum_i k_i'^2)$; the second part is updating the decoding graph, which has complexity $\mathcal{O}(T \sum_i (d_i - k'_i) M)$. 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 $\mathcal{O}(M^3 n + TMn \mathbb{E}[\Psi])$. Usually, T and $\mathbb{E}[\Psi]$ is larger than M and the second term is dominant.

To reduce the encoding/decoding complexity, we hope to have small $\mathbb{E}[\Psi]$. We show that it is possible to have $\mathbb{E}[\Psi] = \mathcal{O}(M)$. In the design of BATS codes, M is a parameter independent of K . The rate of the code is $\frac{K}{nM}$ packets per transmission. When the rate of the code converges to a constant value, we see that the encoding and decoding complexity are $\mathcal{O}(TKM)$ and $\mathcal{O}(KM^2 + TKM)$, respectively.

III. DEGREE DISTRIBUTIONS

In this section we look at how to design degree distributions such that i) the BP decoding succeeds with high probability, ii) the encoding/decoding complexity is low, and iii) the coding rate is large. The clue is in the decoding process of BATS codes.

We analyze the decoding of a BATS code with random generator matrices over a memoryless LOC with transfer matrix H . Associated with each check node is a degree and a rank. The probability that a check node has degree d is Ψ_d

and for a check node with degree d , the probability that it has rank r is $h_{d,r} = \Pr\{\text{rk}(G_d H) = r\}$, where G_d is a $d \times M$ random matrix with uniform i.i.d components. The generator matrix of a batch with degree d is just an instance of G_d . $h_{d,r}$ can be computed using only the rank distribution of H (see [15] for the expression). For convenience, we also call the pair (d, r) the degree of a check node. Let $\Psi_{d,r} = \Psi_d h_{d,r}$ be the probability that a check node has degree (d, r) . A decoding graph with K variable nodes and n check nodes is denoted by $\text{BATS}(K, n, \{\Psi_{d,r}\})$. The design coding rate of the BATS code is $\theta = K/n$.

We use the result of density evolution to show the asymptotic decoding performance of a sequence of decoding graph $\text{BATS}(K, n, \{\Psi_{d,r}\})$ with constant θ . We apply Wormald's theorem [16] to approximate the density evolution by differential equations. The details of the analysis are omitted and can be found in [15]. Assume that the maximum D such that Ψ_D is nonzero is not related to K . Let

$$\Omega(x) = \sum_{r=1}^M h_{r,r}^* \sum_{d=r+1}^D d \Psi_d I_{d-r,r}(x) + \sum_{r=1}^M h_{r,r} r \Psi_r,$$

where $h_{r,r}^* = \frac{1-q^{-1}}{1-q^{-r-1}} h_{r+1,r}$ and

$$I_{a,b}(x) = \sum_{j=a}^{a+b-1} \binom{a+b-1}{j} x^j (1-x)^{a+b-1-j}$$

is called *regularized incomplete beta function*. Define

$$\tilde{\rho}_1(\tau) = \frac{(1-\tau/C_0)}{\mathbb{E}[\Psi]} (\Omega(\tau/C_0) + \theta \ln(1-\tau/C_0)),$$

where $C_0 = \theta/\mathbb{E}[\Psi]$. We obtain the following sufficient condition of the degree distribution such that the BP decoding of BATS codes succeeds with high probability when K is sufficiently large.

Theorem 1: Consider a sequence of decoding graph $\text{BATS}(K, n, \{\Psi_{d,r}\})$ with constant θ . For any $\epsilon > 0$, consider a degree distribution with $\tilde{\rho}_1(\tau) \geq \epsilon$ for $\tau \in [0, C_0(1-\eta)]$. There exist constant K_0 , c and c' such that when $K \geq K_0$, with probability at least $1 - cn^{7/24} \exp(-c'n^{1/8})$, the decoding terminates with at most ηK input packets erased.

Theorem 1 enables us to consider the following optimization problem to find an asymptotically optimal degree distribution that maximizes the coding rate:

$$\begin{aligned} \max \quad & \theta \\ \text{s.t.} \quad & \Omega(x) + \theta \ln(1-x) \geq 0, \quad 0 < x \leq 1-\eta \\ & \Psi_d \geq 0, \quad d = 1, \dots, D \\ & \sum_d \Psi_d = 1. \end{aligned} \quad (2)$$

The only channel information required in the optimization problem is the rank distribution of H . We can further show that using $D > \lceil M/\eta \rceil - 1$ does not give better optimal value in the above optimization problem. Thus we set $D = \lceil M/\eta \rceil - 1$, which complies our assumption that D is not related to K .

IV. ACHIEVABLE RATES

The coding rate of a BATS codes is given by the average number of packets that can be transmitted using one batch. The rate can also be normalized by the batch size.

A. Asymptotically Achievable Rates

The BP decoding algorithm, if succeeds, recovers at least $(1-\eta)K$ packets. Thus, the maximum achievable rate of BATS codes is at least $\hat{\theta}(1-\eta)$, where $\hat{\theta}$ is the optimal value of the optimization problem (2). In terms of packets per use, the capacity of a LOC with the transfer matrix H is $\mathbb{E}[\text{rk}(H)]$ [13]. As channel codes for LOCs, the maximum achievable rate of BATS codes is upper bounded by the capacity of LOCs. So $\hat{\theta}(1-\eta) \leq \mathbb{E}[\text{rk}(H)]$. The maximum achievable rate of BATS codes is lower bounded by the following theorem (proved in [15]).

Theorem 2: Let $\hat{\theta}$ be the optimal value of the optimization in (2). Then

$$\hat{\theta} \geq \max_{r=1,2,\dots,M} r h_{r,r}.$$

Even though the lower bound given by the theorem is loose in general, it shows that BATS codes achieve rates arbitrarily close to the capacity for the following special case. We call an LOC with transfer matrix H *full-rank* if $h_1 = h_2 = \dots = h_{M-1} = 0$, where $h_i = \Pr\{\text{rk}(H) = i\}$. For a full-rank LOC, $\hat{\theta} \geq M h_{M,M} \rightarrow M h_M = \mathbb{E}[\text{rk}(H)]$ when the field size $q \rightarrow \infty$. Since η can be taken arbitrarily small, BATS codes achieve rates arbitrarily close to the capacity of full-rank LOCs over sufficiently large finite fields.

To see the achievable rates for the general cases, we numerically solve the optimization problem (2) by taking discrete values for x . Let $\tilde{\theta}$ be the optimal value of this relaxed version of (2). Set $M = 5$, $q = 16$ and $\eta = 0.01$. A rank distribution $\{h_0, h_1, \dots, h_M\}$ is generated as follows: First, $h_0 = 0$ and for $i > 1$, h_i is independently and uniformly chosen between zero and one; Then, normalize the rank distribution such that $\sum_i h_i = 1$. We compute $\tilde{\theta}$ for 24345 rank distributions independently generated and compare $(1-\eta)\tilde{\theta}$ with $\sum_{r=1}^M r h_r$ by computing $\lambda = (\sum_{r=1}^M r h_r - (1-\eta)\tilde{\theta}) / \sum_{r=1}^M r h_r$. The results show that for more than 99% rank distributions, λ is smaller than 0.05, and the largest λ is 0.1145. This means that BATS codes achieve rates very close to the capacity even for LOCs over small finite fields. When using larger fields, the gap between the maximum achievable rate and the capacity becomes smaller. E.g., after changing the field size to $q = 64$, for more than 99% rank distributions, λ is smaller than 0.026, and the largest λ reduces to 0.0876.

B. Finite Length Performance

We use the network in Fig. 1 to illustrate the performance of BATS codes. The source node s applies BATS code encoding. In each time slot, s sends a packet to a . Assume transmission is instantaneous and node a receives the packet, if not erased, at the same time slot. No matter whether particular packets are received or not, node a transmits at each time slot a linear combination of the packets it has received so far. After M time

i	26	27	34	35	43
Ψ_i	0.0163	0.245	0.082	0.0937	0.0325
i	44	55	56	71	72
Ψ_i	0.1046	0.0371	0.0736	0.0345	0.057
i	94	95	131	203	204
Ψ_i	0.0412	0.0373	0.0694	0.0655	0.0104

TABLE I
DOMINANT PROBABILITY MASSES FOR $M = 32$.

slots, node s switches to another batch and node a clears its buffer for the last batch. These operations of the network minimize the transmission delay and are asymptotically optimal when M goes to infinity.

The operation at node a for a batch is given by a random matrix T , an $M \times M$ upper unitriangular matrix with all the upper triangular, off-diagonal entries independently and uniformly distributed¹. Let D be an $M \times M$ random diagonal matrix with independent components. A diagonal component of D is zero with probability 0.2 and is one with probability 0.8. D models the erasures on a link. The transfer matrix of the network is $H = D_1 T D_2$, where D_1 , T and D_2 are independent; D_1 and D_2 follow the same distribution of D .

The rank distribution of H is approximated by the empirical distribution obtained using 10^5 independent samples of H . Using the (empirical) rank distribution, a degree distribution is obtained by solving (2) by taking discrete values of x . For example, when $\eta = 0.08$, $M = 32$ and $q = 4$, the dominant probability masses are given in Table I. Note that we do not optimize the degree distribution for different K , so the coding rates we obtained can be further improved.

BATS codes are rateless codes, i.e., the coding rate is not fixed. To see the performance of a BATS code, we use the average coding rate defined as follows. Consider that the source node encode K packets using a BATS code, and the decoder stops after recovering $\bar{\eta}K$ packets. Repeat the above simulation J times and let n_j be the number of batches used when the decoder stops in the j th simulation. The average coding rate of the BATS code is defined as $\bar{\eta}KJ/\sum_j n_j$. In the following, we will compare the average coding rates for different parameters, and the average coding rates are maximized over $\bar{\eta}$.

The first thing we want to show is that BATS codes outperform fountain codes. We know that when $M = 1$, BATS codes become Raptor codes and the intermediate operation T becomes forwarding. BATS codes can achieve rates exceeding 0.64 (see the rates in bold letters in Table II), the routing capacity, which serves as an upper bound on the maximum achievable rate for Raptor codes.

Another trend we observe is that using large q also increases the rates. A closer look at the simulations further reveals that the gain by increasing q becomes smaller when q is large. For example, when $K = 32000$, increasing q from 2 to 4 gains

¹A unitriangular matrix has unit entries on the main diagonal. The intermediate operation modelled by a unitriangular matrix becomes forwarding when $M = 1$.

K	$q = 2$	$q = 4$	$q = 8$	$q = 16$
16000	0.5826	0.6145	0.6203	0.6248
32000	0.6087	0.6441	0.6524	0.6574
64000	0.6259	0.6655	0.6762	0.6818

TABLE II
AVERAGE CODING RATES FOR $M = 32$.

5.82% in the rate, but increasing q from 4 to 8 gains 1.29%.

V. CONCLUDING REMARKS

Benefiting from network coding and the properties of fountain codes, BATS codes are ideal for transmitting files through communication networks. Besides 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.

ACKNOWLEDGMENT

The work of S. Yang and R. W. Yeung was partially supported by a grant from the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-02/08).

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] 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.
- [4] 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.
- [5] 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.
- [6] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton Conf. on Comm., Control, and Computing*, Oct. 2003.
- [7] R. W. Yeung, *Information Theory and Network Coding*. Springer, 2008.
- [8] M.-L. Champel, K. Huguenin, A.-M. Kermarrec, and N. L. Scouarnec, "LT network codes," INRIA, Research Report RR-7035, 2009. [Online]. Available: <http://hal.inria.fr/inria-00416671/PDF/RR-7035.pdf>
- [9] P. Maysounkov, N. J. A. Harvey, and D. S. Lun, "Methods for efficient network coding," in *Proc. Allerton Conf. Comm., Control, and Computing*, Sep. 2006.
- [10] D. Silva, W. Zeng, and F. R. Kschischang, "Sparse network coding with overlapping classes," in *Proc. Workshop Network Coding, Theory, and Applications NetCod '09*, 2009, pp. 74–79.
- [11] A. Heidarzadeh and A. H. Banihashemi, "Overlapped chunked network coding," in *Proc. ITW '10*, 2010, pp. 1–5.
- [12] R. Maysounkov and R. Sreenivas, "Relaying a fountain code across multiple nodes," in *Proc. ITW '08*, 2008, pp. 149–153.
- [13] S. Yang, S.-W. Ho, J. Meng, E. hui Yang, and R. W. Yeung, "On linear operator channels over finite fields," 2009. [Online]. Available: <http://arxiv.org/abs/1002.2293v1>
- [14] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.
- [15] S. Yang and R. W. Yeung, "Batched sparse codes: Coding for a network coded fountain," 2011, under preparation.
- [16] 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.