Tang *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:106
DOI 10.1186/s13634-015-0297-8

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Expander chunked codes

Bin Tang[1*], Shenghao Yang[2], Baoliu Ye[1], Yitong Yin[1] and Sanglu Lu[1]

Abstract

Chunked codes are efficient random linear network coding (RLNC) schemes with low computational cost, where the input packets are encoded into small chunks (i.e., subsets of the coded packets). During the network transmission, RLNC is performed within each chunk. In this paper, we first introduce a simple transfer matrix model to characterize the transmission of chunks and derive some basic properties of the model to facilitate the performance analysis. We then focus on the design of overlapped chunked codes, a class of chunked codes whose chunks are non-disjoint subsets of input packets, which are of special interest since they can be encoded with negligible computational cost and in a causal fashion. We propose expander chunked (EC) codes, the first class of overlapped chunked codes that have an analyzable performance, where the construction of the chunks makes use of regular graphs. Numerical and simulation results show that in some practical settings, EC codes can achieve rates within 91 to 97 % of the optimum and outperform the state-of-the-art overlapped chunked codes significantly.

Keywords: Random linear network coding, Chunked codes, Iterative decoding, Random regular graph

1 Introduction

Random linear network coding (RLNC) has great potential for data dissemination over communication networks [1–4]. RLNC can be implemented in a distributed fashion due to its random nature and is shown to be asymptotically capacity-achieving for networks with packet loss in a wide range of scenarios [5–7]. In this paper, we propose a low-complexity RLNC scheme called expander chunked (EC) codes and analyze the achievable rates of EC codes.

1.1 Background

For ordinary RLNC studied in literature [3–7], all participating nodes forward coded packets formed by random linear combinations of all the packets received so far. Major issues in applying ordinary RLNC include the computational cost and the coefficient vector overhead. Consider the dissemination of k input packets, each consisting of L symbols from a finite field. For encoding, RLNC requires $\mathcal{O}(kL)$ finite field operations to generate a coded packet, and for decoding, a destination node takes $\mathcal{O}(k^2 + kL)$ finite field operations per packet if Gaussian elimination is employed. Moreover, to recover the transfer matrices of network coding at the destination node, a coefficient vector of k symbols is usually included in each of the transmitted packets [3]. Since the packet length L has an upper bound in real-world communication networks,[1] using large values of k reduces the transmission efficiency. When there are hundreds of input packets, the computational cost and the coefficient vector overhead would make RLNC difficult for real-world implementation.

To resolve these issues, *chunked (network) codes* have been proposed [8], where the input packets are encoded into multiple small *chunks* (also called generations, classes, etc.), each of which is a subset of the coded packets. When using chunked codes, an intermediate network node can only combine the packets of the same chunk. The encoding and decoding complexities per packet of chunked codes are usually $\mathcal{O}(mL)$ and $\mathcal{O}(mL + m^2)$, respectively, where m is the *chunk size*, i.e., the number of packets in each chunk. The coefficient vector overhead also reduces to m symbols per packet since only the transfer matrices of the chunks are required at the destination nodes. Even so, the chunk size should be a small value (e.g., 16 or 32) for the purpose of practical implementation, as demonstrated in [9].

Existing chunked codes are in two categories: *overlapped chunked codes* and *coded chunked codes*. In overlapped chunked codes, the chunks are subsets of the input packets with possibly non-empty intersections. The

*Correspondence: tb@nju.edu.cn
[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
Full list of author information is available at the end of the article

first several designs of chunked codes all belong to this category. However, the existing designs of overlapped chunks are mostly based on heuristics, and no rigorous performance analysis is available for the existing designs [10–12]. In coded chunked codes, chunks are generated by combining multiple input packets. By generalizing fountain codes and LDPC codes, nearly throughput optimal chunked codes have been designed, including BATS code [13, 14], Gamma code [15, 16], and L-chunked (LC) code [17]. Overlapped chunks can be viewed as a degraded class of coded chunks where chunks are generated using certain repetition codes.

Overlapped chunked codes, however, can have lower encoding complexity and latency than general coded chunked codes. First, as no new packets are necessarily generated during the encoding, the encoding complexity is dominated by generating the indices for the packets in each chunk, which does not incur any finite field operation or depend on the packet length L. In contrast, coded chunked codes incur a computational cost that is linear of L to generate a coded packet. For instance, BATS codes require on average $\bar{\Psi}mL$ finite field operations for encoding a chunk, where $\bar{\Psi} \gtrapprox 3m$. Therefore, compared to general coded chunked codes, the computational cost of overlapped chunked codes is usually negligible.

Second, overlapped chunks can be encoded in a *causal fashion*. Suppose that the input packets arrive at the encoder gradually. The first chunk can be generated after collecting m input packets, and for every m input packets collected in the following, at least one new chunk can be formed. Therefore, the generation as well as the transmission of chunks can be performed in parallel with the collection of the input packets, reducing the total transmission latency. In contrast, how to achieve causal encoding for general coded chunked codes is not clear: BATS codes and Gamma codes usually require a large fraction of the input packets for encoding chunks.

These advantages motivate us to study overlapped chunked codes, which are especially suitable for delay sensitive applications and networks where the source node has limited computation and storage power, e.g., wireless sensors and satellites.

1.2 Our contribution

We propose *expander chunked (EC) codes*, the first class of overlapped chunked codes that has analyzable performance. In an EC code, the overlapping between chunks is generated using a regular graph: each chunk corresponds to a node in the graph and two adjacent chunks share an input packet. EC codes can be encoded causally and share the same belief propagation (BP) decoding of general overlapped chunked codes.

We analyze the BP decoding performance of EC codes generated based on random regular graphs. By exploring the locally tree-like property of random regular graphs and then conducting a tree-based analysis similar to that of LT/LDPC code, we obtain a lower bound on the achievable rate depending only on the chunk size, the degree of the regular graph, and the rank distribution of the transfer matrices.

The achievable rates of EC codes are evaluated and compared with other chunked codes in two scenarios. We first compare the achievable rates of EC codes with representative coded chunked codes for randomly sampled rank distributions of the transfer matrices, where the purpose is to understand the general performance of EC codes. We find that the performance of EC codes highly depends on the rank distributions: when the expected rank is relatively large, the average achievable rate (over the rank distributions sampled) of EC codes is close to 90 % of the representative coded chunked codes, as well as a theoretical upper bound. But for relatively small expected ranks, the achievable rate of EC codes varies significantly for different rank distributions.

To further see the real-world potential of EC codes, we then evaluate the performance for a near-optimal chunk transmission scheme over line-topology (line) networks [18]. As most practical routing schemes are single-path based, line networks have attracted a lot of interest [19–21]. Also, the chunked code scheme for line networks can be extended to general network scenarios, including general unicast networks [14, 18], two-way relay networks [22], and wireless broadcast networks [23]. For a wide range of the packet loss rates, with proper optimization of the transmission scheme, EC codes achieve rates very close to those of the coded chunked codes and about 91 % $\sim$ 97 % of the theoretical upper bounds. Besides, we show by simulation that EC codes perform much better than the existing overlapped chunked codes in line networks.

Table 1 gives a brief comparison between EC codes, BATS codes, and LC codes.

As another contribution, a simple transfer matrix model is proposed to characterize the transmission of chunks over networks with packet loss. Compared with a similar model proposed in [14], which is more suitable for BATS codes, our model incorporates some more practical features of network operations for general chunked codes, making the design of efficient network transmission protocols easier. Therefore, our model is of independent interest for chunked codes. We derive some properties of this transfer matrix model for the performance analysis, which can apply to general chunked codes.

1.3 Related work

The simplest way to form a chunked code is to use disjoint subsets of the input packets as chunks [8], which has been used in some applications of RLNC [9, 24, 25]. To

Table 1 Comparison among EC/BATS/LC codes where achievable rates are evaluated over line networks with chunk transmission scheme given in [18]

Design	Causal encoding	Encoding complexity	Decoding complexity	Achievable rates
EC	Support	$\mathcal{O}(n)$	$\mathcal{O}(nL)$	91 % ∼ 97 % of opt.
BATS	Unknown	$\mathcal{O}(nL)$	$\mathcal{O}(nL)$	>99 % of opt.
LC	Unknown	$\mathcal{O}(nL)$	$\mathcal{O}(nL)$	>98 % of opt.

decode a chunk, the transfer matrix of the chunk must have full rank of m; otherwise, none of the packets in the chunk could be recovered with high probability. However, it is not always a simple task to guarantee the success of decoding a chunk at the destination node. One approach is to use feedback-based chunk transmission mechanism [24]. While some efficient feedback protocols for specific applications have been developed [25, 26], in general, such feedback incurs an inevitable delay and also consumes network resources, resulting in degraded system performance. Besides, for some scenarios such as satellite and deep-space communications, feedbacks are not even available. Another approach is to employ a random scheduling-based chunk transmission scheme [27], where every network node always randomly selects a chunk for transmission. But this scheme has poor performance for small chunk sizes [10, 11].

Instead of using disjoint chunks of input packets, chunks with overlaps, i.e., different chunks share some input packets in common, have been proposed by several groups independently [10–12]. It is shown via simulations that overlapped chunked codes have much better performance than disjoint chunks [10, 11]. The random annex codes proposed by Li et al. [12] demonstrate better performance in simulation than the overlapped chunked codes in [10, 11], but only heuristic analysis of the design is provided.

BATS code [13, 14] is the first class of chunked codes that uses coded chunks. Each chunk in a BATS code is generated as linear combinations of a random subset of the input packets. BATS codes can be regarded as a matrix generalization of fountain codes [28, 29] and preserve the ratelessness of fountain codes.

Another kind of coded chunked codes consists of chunks that satisfy some parity-check constraints, similar to those of LDPC codes. The first class of such codes is Gamma codes [15, 16, 30], where the parity-check constraints are applied on the whole chunk [15] or on the individual packets in chunks [30]. Another class of such codes is L-chunked codes [17] which consider more general parity-check constraints and show better performance. Note that the original Gamma code [15] paper is published in parallel with the conference version of this paper [31], while the refined Gamma codes [30] and L-chunked codes are published later than that of our conference version.

Various chunked code-based transmission schemes have been designed and implemented recently [18, 22, 32], which are consistent with our transfer matrix model.

2 Overlapped chunked codes

In this section, we give a general formulation of overlapped chunked codes, including causal encoding and belief propagation (BP) decoding. We also provide a transfer matrix model for general chunked codes.

2.1 Encoding of chunks

Consider transmitting a set of k input packets $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k$ from a source node to a destination node over a network with packet loss. Each input packet composes of L symbols from the finite field $\mathbb{F}_q$ and is regarded as a column vector in $\mathbb{F}_q^L$ henceforth.

Definition 1 (Chunked codes)**.** A *chunk* is a set of packets each of which is a linear combination of the input packets, and a *chunked code* is a collection of chunks. A chunked code is said to be *overlapped* if its chunks are subsets of the input packets with possibly non-empty overlapping.

In this paper, we focus on the design of overlapped chunked codes. Evidently, an overlapped chunked code can be generated by repeating some input packets. Same as most related works, we assume that all the chunks in a chunked code have the same cardinality m, which is called the *chunk size*. As the chunk size is related to the encoding/decoding computational complexities and the coefficient vector overhead, for the sake of the applicability in common networks, we regard the chunk size m as a fixed constant which does not change with the number of input packets.

An overlapped chunked code can be more concisely represented by a collection of index sets of size m. For any integer n, let $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n$ be subsets of $\{1, \ldots, k\}$ with size m. Let $\mathbf{B}_j = \{\mathbf{b}_i : i \in \mathcal{I}_j\}$. We call either $\mathcal{I}_j$ or $\mathbf{B}_j$ a chunk and the subscript j the chunk ID. An overlapped chunked code of n chunks can be given by either $\{\mathcal{I}_j : j = 1, \ldots, n\}$ or $\{\mathbf{B}_j : j = 1, \ldots, n\}$.

Since each chunk is a subset of the input packets, it is not necessary to duplicate the existing input packets for chunk encoding. During the encoding, only the address in the

memory of each packet in a chunk needs to be recorded. Furthermore, every overlapped chunked code can be encoded causally, which is explained in the following.

Definition 2 (Causal encoding)**.** We say that a chunked code can be encoded causally if for any positive integer $i \leq k$, there exist at least $\lfloor \frac{i}{m} \rfloor$ chunks in the chunked code such that

- these chunks are formed by the first i input packets, and
- each of the first $m \times \lfloor \frac{i}{m} \rfloor$ input packets is used for generating these chunks at least once.

It is worth mentioning that, when $m = 1$, systematic encoding of a linear code is a special case of causal encoding. For any overlapped chunked code where each input packet is included by at least one chunk, we can always apply some proper permutation of the indices such that, for any $j \leq n$, the indices of the packets among the first j chunks are $1, 2, \ldots, k_j$, where $k_j \leq mj$. In this sense, every overlapped chunked code can be encoded causally. One example is given when introducing our EC codes in Section 3. Now, consider a scenario where the input packets arrive at the source node sequentially (e.g., the source node is a sensor which keeps on collecting data and encapsulating data into packets). Then, for any m input packets collected consecutively, the source node can generate one new chunk for transmission. Hence, the source node does not necessarily collect all the input packets before encoding and the chunks can be transmitted in parallel with the collection of succeeding input packets. Therefore, by applying an overlapped chunked code, the end-to-end transmission latency could be significantly reduced.

2.2 Transmission of chunks

Each transmitted packet in the network is of the form $(j, \mathbf{c}, \mathbf{b})$, where j specifies a chunk ID, $\mathbf{c} \in \mathbb{F}_q^m$ is the coefficient vector, and $\mathbf{b} = \mathbf{B}_j\mathbf{c}$, a linear combination of packets in $\mathbf{B}_j$, is the payload. Here, with some abuse of notation, $\mathbf{B}_j$ is also treated as a matrix formed by juxtaposing the packets in $\mathbf{B}_j$. For convenience, we refer to a packet with chunk ID j as a j packet.

Now, we describe a chunk transmission model through a network employing linear network coding, which is consistent with the recent design and implementation of chunked code-based network protocols [18, 22, 32]. Consider the jth chunk of packets $\mathbf{b}_{j_1}, \mathbf{b}_{j_2}, \ldots, \mathbf{b}_{j_m}$. The source node first attaches a coefficient vector to each packet and generates $\tilde{\mathbf{b}}_{j_i} = (\mathbf{e}_i, \mathbf{b}_{j_i})$, $i = 1, \ldots, m$, where $\mathbf{e}_i$ is the ith column of the $m \times m$ identity matrix. The source node then generates M_j random linear combinations of $\tilde{\mathbf{b}}_{j_i}$ and transmits these linear combinations after attaching the chunk ID, where M_j is an integer-valued random variable.

At an intermediate network node, suppose that h j packets have been received, denoted by $(j, \mathbf{c}^i, \mathbf{b}^i)$, $i = 1, \ldots, h$. The network node can transmit j packet $(j, \mathbf{c}, \mathbf{b})$ generated by

$$\mathbf{c} = \sum_{i=1}^{h} \phi_i \mathbf{c}^i, \text{ and } \mathbf{b} = \sum_{i=1}^{h} \phi_i \mathbf{b}^i, \tag{1}$$

where ϕ_i, $i = 1, 2, \ldots, h$, are chosen from $\mathbb{F}_q$. A network node does not transmit combinations of packets of different IDs. Note that in (1), we only need to combine the j packets with linearly independent coefficient vectors. For the scheduling issue, i.e., how to choose a chunk $\mathbf{B}_j$ by each intermediate node for each transmission, please refer to some recent proposed network protocols [18, 22, 32].

At the destination node, let $\mathbf{T}_j$ be the matrix formed by the coefficient vectors of all the j packets received, and let $\mathbf{Y}_j$ be the matrix formed by the payloads of all the j packets received. We have

$$\mathbf{Y}_j = \mathbf{B}_j\mathbf{T}_j, \tag{2}$$

where $\mathbf{T}_j$ is called the *transfer matrix* of $\mathbf{B}_j$. Without affecting the decoding performance, we can remove some received j packets so that the remaining set of j packets have linearly independent coefficient vectors. So, we assume that $\mathbf{T}_j$ has a full-column rank. According to the transmission scheme we described, we can further write

$$\mathbf{T}_j = \mathbf{S}_j\mathbf{H}_j$$

where $\mathbf{S}_j$ is an $m \times M_j$ random matrix corresponding to the linear combinations performed by the source node, and $\mathbf{H}_j$ is a random matrix with M_j rows corresponding to the linear operations performed by intermediate nodes as well as the random packet losses over the network links. Here, for a given value of M_j, $\mathbf{S}_j$ is a *totally random* matrix, i.e., every entry of $\mathbf{S}_j$ is chosen from $\mathbb{F}_q$ uniformly and independently at random. Also, we assume that $\mathbf{H}_j$ and $\mathbf{S}_j$ are independent conditionings on M_j and $\mathrm{rk}(\mathbf{S}_j)$, which holds for all the recent chunked code-based network protocols [18, 22, 32].

The proposed chunk transmission model does not depend on a particular chunked code and hence can be used for the analysis of other chunked codes. A similar model has been used for BATS codes [14]. Our model, however, explicitly incorporates a parameter M_j indicating the number of packets transmitted of a chunk, which has a clear operation meaning in chunked code-based network protocols. Intuitively, when the network has a higher packet loss rate, we intend to use a larger value of M_j to gain the benefit of network coding. Readers can find more discussion about this parameter in [18].

Now, we present a key result about the transfer matrices, which shows that the column space of each transfer matrix

with a fixed dimension is uniformly distributed over all the subspaces with the same dimension.

Lemma 1. *For any two subspaces* W, U *of* $\mathbb{F}_q^m$ *with the same dimension,*

$$\Pr\{\langle \mathbf{T}_j\rangle = \mathsf{W}\} = \Pr\{\langle \mathbf{T}_j\rangle = \mathsf{U}\},$$

where $\langle \mathbf{T}_j\rangle$ *denotes the column space of matrix* $\mathbf{T}_j$.

Proof. For matrix $\mathbf{A}$ and subspace U, define

$$\mathbf{A}\mathsf{U} = \{\mathbf{Az} : \mathbf{z} \in \mathsf{U}\}.$$

It can be checked that $\mathbf{A}\langle \mathbf{Z}\rangle = \langle \mathbf{AZ}\rangle$. Since U and W have the same dimension, there exists a full-rank $m \times m$ matrix $\mathbf{A}$ such that

$$\mathsf{U} = \mathbf{A}\mathsf{W}.$$

Thus,

$$\begin{aligned}\Pr\{\langle \mathbf{T}_j\rangle = \mathsf{W}\} &= \Pr\{\mathbf{A}\langle \mathbf{T}_j\rangle = \mathsf{U}\}\\ &= \Pr\{\langle \mathbf{AT}_j\rangle = \mathsf{U}\}\\ &= \Pr\{\langle \mathbf{AS}_j\mathbf{H}_j\rangle = \mathsf{U}\}, \qquad (3)\end{aligned}$$

where the first step follows by the invertibility of $\mathbf{A}$.

For any s and r such that $s \geq r$, denote the event $M_j = s$ and $\mathrm{rk}(\mathbf{S}_j) = r$ by $\mathcal{E}_{s,r}$, and define $\mathcal{S}_{s,r}$ to be the set of all $m \times s$ matrices with rank r. For any $S \in \mathcal{S}_{s,r}$, define $\mathcal{H}_S = \{H : \langle SH\rangle = \mathsf{U}\}$. Since $\mathbf{S}_j$ is totally random given $M_j = s$ and $\mathbf{A}$ is invertible, for any $S \in \mathcal{S}_{s,r}$,

$$\Pr\{\mathbf{S} = S \mid \mathcal{E}_{s,r}\} = \Pr\{\mathbf{AS} = S \mid \mathcal{E}_{s,r}\}.$$

Using the assumption that $\mathbf{S}_j$ and $\mathbf{H}_j$ are independent conditionings on M_j and $\mathrm{rk}(\mathbf{S}_j)$, we have

$$\begin{aligned}&\Pr\{\langle \mathbf{AS}_j\mathbf{H}_j\rangle = \mathsf{U}\}\\ &= \sum_{s,r:s\geq r} \Pr\{\langle \mathbf{AS}_j\mathbf{H}_j\rangle = \mathsf{U} \mid \mathcal{E}_{s,r}\}\Pr\{\mathcal{E}_{s,r}\}\\ &= \sum_{s,r:s\geq r}\sum_{S\in\mathcal{S}_{s,r}}\sum_{H\in\mathcal{H}_S} \Pr\{\mathbf{AS}_j = S, \mathbf{H}_j = H \mid \mathcal{E}_{s,r}\}\Pr\{\mathcal{E}_{s,r}\}\\ &= \sum_{s,r:s\geq r}\sum_{S\in\mathcal{S}_{s,r}}\sum_{H\in\mathcal{H}_S} \Pr\{\mathbf{AS}_j = S \mid \mathcal{E}_{s,r}\}\Pr\{\mathbf{H}_j = H \mid \mathcal{E}_{s,r}\}\Pr\{\mathcal{E}_{s,r}\}\\ &= \sum_{s,r:s\geq r}\sum_{S\in\mathcal{S}_{s,r}}\sum_{H\in\mathcal{H}_S} \Pr\{\mathbf{S}_j = S \mid \mathcal{E}_{s,r}\}\Pr\{\mathbf{H}_j = H \mid \mathcal{E}_{s,r}\}\Pr\{\mathcal{E}_{s,r}\}\\ &= \sum_{s,r:s\geq r}\sum_{S\in\mathcal{S}_{s,r}}\sum_{H\in\mathcal{H}_S} \Pr\{\mathbf{S}_j = S, \mathbf{H}_j = H \mid \mathcal{E}_{s,r}\}\Pr\{\mathcal{E}_{s,r}\}\\ &= \sum_{s,r:s\geq r} \Pr\{\langle \mathbf{S}_j\mathbf{H}_j\rangle = \mathsf{U} \mid \mathcal{E}_{s,r}\}\Pr\{\mathcal{E}_{s,r}\}\\ &= \Pr\{\langle \mathbf{S}_j\mathbf{H}_j\rangle = \mathsf{U}\}\\ &= \Pr\{\langle \mathbf{T}_j\rangle = \mathsf{U}\}.\end{aligned}$$

The proof is completed by combining the above equality with (3). □

2.3 BP decoding

The destination node tries to decode the input packets by solving the local linear systems $\mathbf{Y}_j = \mathbf{B}_j\mathbf{T}_j$, $j = 1, 2, \ldots, n$. These local linear systems for chunks jointly give a global linear system of equations on the k input packets, but solving the global linear system without considering the chunk structure usually has high computational cost. Therefore, we consider the following BP decoding of overlapped chunked codes.

The BP decoding includes multiple iterations. Initially, each chunk $\mathbf{B}_j$ is associated with a linear system given in (2). We say chunk $\mathbf{B}_j$ is *decodable* if its linear system is uniquely solvable. Each iteration consists of two phases:

- in the first phase, decode every decodable chunk that has not been decoded by solving its associated linear system using, e.g., Gaussian elimination, and
- in the second phase, for each input packet $\mathbf{b}$ in $\mathbf{B}_j$ that is decoded in the last phase and each chunk $\mathbf{B}_i \neq \mathbf{B}_j$ that includes $\mathbf{b}$, substitute the value of $\mathbf{b}$ into the linear system of $\mathbf{B}_i$, reducing the number of unknown input packets in this linear system.

The BP decoding stops when all the chunks have been decoded or all the chunks that have not been decoded are not decodable.

Now, we analyze the time cost of the BP decoding algorithm measured in finite field operations. Solving the linear system of a chunk can be done by first inverting the coefficient matrix, which costs $\mathcal{O}(m^3)$ and then using the inverse to recover all the unknown input packets, which costs $\mathcal{O}(m^2L)$. As there are n chunks, all the first phases cost $\mathcal{O}((m^3 + m^2L)n)$ in total. The substitution of an input packet into a linear system costs $\mathcal{O}(mL)$ and can happen at most mn times, so all the second phases cost $\mathcal{O}(m^2Ln)$ in total. Therefore, the BP decoding algorithm costs $\mathcal{O}((m^3 + m^2L)n)$ finite field operations.

Assume that $\mathrm{rk}(\mathbf{T}_j)$ follows the probability distribution $t = (t_0, t_1, \ldots, t_m)$, i.e., $\Pr\{\mathrm{rk}(\mathbf{T}_j) = i\} = t_i$ for $i = 0, 1, \ldots, m$. We have the following theorem, which is the footstone for the analysis of the above BP decoding algorithm.

Theorem 2. *Let* $\mathbf{D}$ *be a fixed matrix with* m *rows and* $rk(\mathbf{D}) = w$. *Then,*

$$\Pr\{rk([\,\mathbf{T}_j\mathbf{D}]\,) = m\} = \sum_{i=m-w}^{m} \frac{q^{(m-i)(m-w)}\begin{bmatrix} w \\ m-i\end{bmatrix}}{\begin{bmatrix} m \\ i\end{bmatrix}} t_i \triangleq \beta_w,$$

where $\begin{bmatrix} w \\ i\end{bmatrix} = \prod_{j=0}^{i-1}\frac{q^w - q^j}{q^i - q^j}$ *is the Gaussian binomial coefficient. This implies that, given that* w *of the* m *packets in a chunk* $\mathbf{B}_j$ *have been decoded due to the decoding of other chunks, the probability that chunk* $\mathbf{B}_j$ *becomes decodable is* β_w.

Proof. The proof is based on the uniformity property of transfer matrices given in Lemma 1 together with counting. For a subspace U, denote its dimension by $\dim(\mathsf{U})$. Since

$$\mathrm{rk}([\mathbf{T}_j\mathbf{D}]) = \mathrm{rk}(\mathbf{T}_j) + \mathrm{rk}(\mathbf{D}) - \dim(\langle\mathbf{T}_j\rangle \cap \langle\mathbf{D}\rangle),$$

we have that, for any $m - w \le i \le m$,

$$\begin{aligned}&\Pr\{\mathrm{rk}([\mathbf{T}_j\mathbf{D}]) = m|\mathrm{rk}(\mathbf{T}_j) = i\}\\ &= \Pr\{\dim(\langle\mathbf{T}_j\rangle \cap \langle\mathbf{D}\rangle) = i + w - m|\dim(\langle\mathbf{T}_j\rangle) = i\}.\end{aligned}$$

As there are $\begin{bmatrix} m \\ i \end{bmatrix}$ i-dimensional subspaces of $\mathbb{F}_q^m$, and $q^{(m-i)(m-w)}\begin{bmatrix} w \\ m-i \end{bmatrix}$ i-dimensional subspaces of $\mathbb{F}_q^m$ such that $\dim(\langle\mathbf{T}_j\rangle \cap \langle\mathbf{D}\rangle) = r + w - i$ (Refs. [33, 34]), by Lemma 1, we have

$$\begin{aligned}&\Pr\{\dim(\langle\mathbf{T}_j\rangle \cap \langle\mathbf{D}\rangle) = i + w - m|\dim(\langle\mathbf{T}_j\rangle) = i\}\\ &= \frac{q^{(m-i)(m-w)}\begin{bmatrix} w \\ m-i \end{bmatrix}}{\begin{bmatrix} m \\ i \end{bmatrix}}.\end{aligned}$$

Therefore,

$$\begin{aligned}&\Pr\{\mathrm{rk}([\mathbf{T}_j\mathbf{D}]) = m\}\\ &= \sum_{i=0}^{m} \Pr\{\mathrm{rk}([\mathbf{T}_j\mathbf{D}]) = m|\mathrm{rk}(\mathbf{T}_j) = i\} \Pr\{\mathrm{rk}(\mathbf{T}_j) = i\}\\ &= \sum_{i=m-w}^{m} \frac{q^{(m-i)(m-w)}\begin{bmatrix} w \\ m-i \end{bmatrix}}{\begin{bmatrix} m \\ i \end{bmatrix}} t_i.\end{aligned}$$

□

Let $\zeta_i^w = \frac{q^{(m-i)(m-w)}\begin{bmatrix} w \\ m-i \end{bmatrix}}{\begin{bmatrix} m \\ i \end{bmatrix}}$ be the probability that a chunk $\mathbf{B}_j$ is decodable given that $\mathrm{rk}(\mathbf{T}_j) = i$ and w out of the m packets in $\mathbf{B}_j$ have been decoded due to the decoding of other chunks. We can easily show that ζ_w^i is an increasing function of the finite field size q. Here, we give some values of ζ_w^i with different i and w for $q = 2$ and $q = 256$ in Table 2. From this table, we can see that the value of ζ_i^w could be much larger when $q = 256$. In particular, for all the instances such that $i + w = m$, the probability that $\mathbf{B}_j$ is decodable is close to 1 when $q = 256$, while it is less than 0.4 when $q = 2$. Therefore, the chunked codes could perform better when larger finite field is used (also see Section 4). On the other hand, using larger finite field would incur more computation cost as well as more coefficient vector overhead. How to choose the proper parameters such as the chunk size and the size of finite field depends on the application scenario, which is beyond the scope of the paper.

Table 2 Some values of ζ_i^w for different i and w

w	2		4		6	
i	$q = 2$	$q = 256$	$q = 2$	$q = 256$	$q = 2$	$q = 256$
26	-	-	-	-	02933	0.9961
27	-	-	-	-	0.5687	1.0000
28	-	-	0.3076	0.9961	0.7823	1.0000
29	-	-	0.6152	1.0000	0.8940	1.0000
30	0.3750	0.9961	0.8203	1.0000	0.9536	1.0000
31	0.7500	1.0000	0.9375	1.0000	0.9844	1.0000

2.4 Achievable rate

Definition 3 (Achievable rate)**.** We say that a rate R is *achievable* by chunked codes using BP decoding if for any constant $\epsilon > 0$, there exists a chunked code with $k \ge (R - \epsilon)mn$ input packets and n chunks for all sufficiently large n such that with probability at least $1 - \epsilon$, when the BP decoding stops, at least $(R - \epsilon)mn$ input packets are recovered.

Remark 1. It is not necessary that the chunked code recovers all the input packets. When all the input packets are required to be recovered by the destination node, we can either retransmit the input packets that are not recovered or use the precode technique as in Raptor codes [29].

Our objective is to design an efficient class of overlapped chunked codes according to the given rank distribution. A natural upper bound on the achievable rates of chunked codes is established as follows.

Proposition 3. *The achievable rate of chunked codes for transfer matrices with rank distribution $t = (t_0, t_1, \ldots, t_m)$ is upper bounded by $\bar{t}/m$, where*

$$\bar{t} = \mathrm{E}[rk(\mathbf{T}_j)] = \sum_{i=1}^{m} i t_i.$$

Proof. Assume that $\lambda = \bar{t}/m + \delta, \delta > 0$ is achievable by chunked codes. Fix $\epsilon = \delta/2$, by the definition of achievable rates, there exists a chunked code with n chunks for all sufficiently large n such that at least $(\lambda - \epsilon)mn$ input packets are recovered with probability at least $1 - \epsilon$.

Note that in the decoding of a chunked code, only received packets of a chunk with linearly independent coefficient vectors are useful. Therefore, the number of decodable input packets is upper bounded by $\sum_{j=1}^{n} \mathrm{rk}(\mathbf{T}_j)$. Then, we have the decoding error probability

$$\begin{aligned}P_{\mathrm{err}} &\ge \Pr\left\{\sum_{j=1}^{n} \mathrm{rk}(\mathbf{T}_j) < (\lambda - \epsilon)mn\right\}\\ &= \Pr\left\{\sum_{j=1}^{n} \mathrm{rk}(\mathbf{T}_j) < (\bar{t} + m\delta/2)n\right\}\\ &\ge 1 - e^{-\frac{m^2\delta^2 n}{12\bar{t}}},\end{aligned}$$

where the last inequality follows from the Chernoff bound. For a sufficiently large n, we have $P_{\text{err}} > \epsilon$, a contradiction! □

3 Expander chunked codes

In this section, we introduce a family of overlapped chunked codes, named *EC codes*.[2]

3.1 Code description

An EC code has three parameters: the number of chunks n, chunk size m, and degree d ($3 \le d \le m$). Without loss of generality, in the following, we assume that dn is even.[3] Let $k = n(m - d/2)$. An EC code is generated by a d-regular graph $G(V, E)$, called the *generator (graph)*, where $V = \{1, 2, \ldots, n\}$ is the node set and E is the edge set. We will discuss the design of G later in this paper. The chunks in the EC code are constructed by the following steps.

1. Label each edge $e \in E$ with a distinct integer in $\{1, \ldots, k\}$, and denote the integer by i_e. Label the rest $k - nd/2 = (m - d)n$ integers in $\{1, \ldots, k\}$ evenly to the n nodes in V, and denote the set of integers labelled to node v by $\mathcal{I}'_v$.
2. Form n chunks $\{\mathcal{I}_v, 1 \le v \le n\}$, where

$$\mathcal{I}_v = \mathcal{I}'_v \cup \{i_e : e \text{ is incident to node} v\}.$$

Due to the one-to-one correspondence between nodes in G and the chunks, we equate a node with its corresponding chunk henceforth in the discussion. We call $\mathcal{I}_v$ chunk v and i_e an overlapping packet of chunk v.

As discussed in the previous section, EC code can be encoded causally. Specifically, the first step of the construction can be done as follows, where each index in $\{1, 2, \ldots, k\}$ is used in an increasing order. First, label node 1 with the first $m-d$ indices and label the d edges incident to node 1 in an arbitrary order with the next d indices. Then, label node 2 with the next $m - d$ indices and label each of the edges incident to node 2 but unlabelled with a next index, and so on. Clearly, for any chunk v, the largest index in $\mathcal{I}_v$ is less than or equal to mv. See Fig. 1 for an illustration of this assignment of indices such that the chunks are suitable for causal encoding.

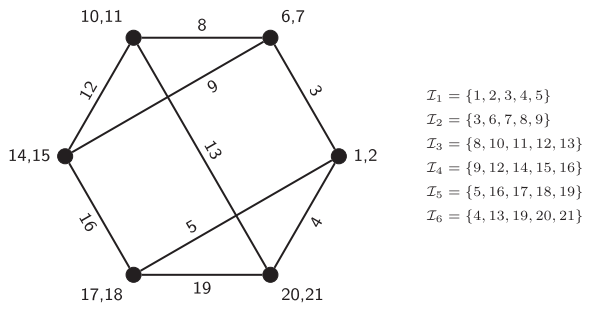

Fig. 1 An EC code with $n = 6$, $m = 5$ and $d = 3$. The generator graph of the code is a 3-regular graph with 6 nodes

3.2 Achievable rates

The performance of EC code with a particular generator graph is difficult to analyze. Instead, we analyze the performance of an EC code with a *random d-regular graph* as the generator. There are various probability models for random d-regular graphs. We adopt the uniform model, *i.e.*, G is uniformly chosen from all d-regular graphs with node set V. One can obtain the similar result for the permutation model, the perfect matching model [35], etc.

The details of the performance analysis are provided in the next subsection; here, we first characterize the achievable rates of EC codes under BP decoding. To state the main result, we need to introduce some notations. For any $3 \le d \le m$, define a function $\alpha_d(y)$ over the interval $[0, 1]$ as

$$\alpha_d(y) = \sum_{w=0}^{d-1} \binom{d-1}{w} y^w (1-y)^{d-1-w} \beta_w, \tag{4}$$

where β_w is defined in Theorem 2. Note that

$$\alpha_d(0) = \beta_0 = t_m > 0, \tag{5}$$

and

$$\alpha_d(y) \le 1, \quad y \in [0, 1]. \tag{6}$$

We can further check that $\alpha_d(y)$ is monotonically increasing in y. With function $\alpha_d(y)$ and its functional powers, we introduce a sequence

$$\alpha_d(0), \alpha_d^2(0), \alpha_d^3(0), \ldots, \tag{7}$$

where $\alpha_d^{i+1}(0) = \alpha_d\left(\alpha_d^i(0)\right)$ for all $i > 0$. This sequence is well defined since the range of α_d is in $[0, 1]$. Further, since $\alpha_d(0) > 0$ and $\alpha_d(y)$ is monotonically increasing, we can check inductively that the sequence in (7) is also monotonically increasing. Since the sequence is bounded above, it must converge. Denote

$$\alpha_d^* = \lim_{i \to \infty} \alpha_d^i(0).$$

We further define

$$\tau_d = \alpha_{d+1}(\alpha_d^*),$$

and

$$\lambda_d = 1 - (1 - \alpha_d^*)^2.$$

Theorem 4. *EC codes with the degree d and chunk size m can achieve a rate at least* $\tau_d(1 - d/m) + \lambda_d d/(2m)$.

Note that, for any fixed degree d, the achievable rate given in Theorem 4 is easy to calculate numerically. Thus, we can easily find a proper degree d to maximize the achievable rate.

3.3 Performance analysis

We provide an analysis of the BP decoding of the EC code with a random d-regular graph as the generator and prove Theorem 4.

Definition 4. For any generator graph $G = (V, E)$, the l neighborhood of a node $v \in V$, denoted by $G_l(v)$, is the subgraph of G induced by all the nodes u with distance at most l to v.

After $l + 1$ iterations of the BP decoding, whether all the input packets in chunk v are recovered is determined by $G_l(v)$. Hence, we study the BP decoding performance $G_l(v)$.

Definition 5. For any generator graph $G = (V, E)$, a node $v \in V$ is said to be l decodable if all the input packets in chunk v can be decoded when the decoding process is applied on $G_l(v)$.

In the following, we set

$$l = \left\lfloor \frac{1}{3} \log_{d-1} n \right\rfloor.$$

We first show that a random regular graph has the *locally tree-like* property, *i.e.*, almost all the nodes in G have their l neighborhoods being trees.

Lemma 5. *For a random d-regular graph G with n nodes, let T be the number of nodes with their l neighborhoods being trees. Then, for any constant* $\epsilon > 0$,

$$\Pr\{T > (1-\epsilon)n\} \geq 1 - \mathcal{O}\left(n^{-1/3}/\epsilon\right).$$

Proof. Let X_r be the number of cycles of length r in G. One important fact is that a node whose l neighborhood is not a tree must belong to a cycle with length less than or equal to $2l + 1$. Therefore,

$$n - T \leq \sum_{r=3}^{2l+1} r X_r. \tag{8}$$

Since $(d-1)^{2l+1} = o(n)$, it was shown in [36] that, for any $3 \leq r \leq 2l + 1$,

$$\begin{aligned} \mathrm{E}[X_r] &= \frac{(d-1)^r}{2r}\left(1 + \mathcal{O}\left(\frac{r(r+d)}{n}\right)\right) \\ &= \frac{(d-1)^r}{2r}(1 + o(1)). \end{aligned} \tag{9}$$

Taking expectation on both sides of (8) and substituting (9) gives

$$\begin{aligned} \mathrm{E}[n - T] &\leq \sum_{r=3}^{2l+1} r\mathrm{E}[X_r] \\ &= \sum_{r=3}^{2l+1} \frac{(d-1)^r}{2}(1 + o(1)) \\ &= \mathcal{O}\left((d-1)^{2l+1}\right) \\ &= \mathcal{O}\left(n^{2/3}\right). \end{aligned}$$

Finally, by Markov's inequality, we get

$$\begin{aligned} \Pr\{T \leq (1-\epsilon)n\} &= \Pr\{n - T \geq \epsilon n\} \\ &\leq \frac{\mathrm{E}[n-T]}{\epsilon n} \\ &\leq \mathcal{O}\left(n^{-1/3}/\epsilon\right). \end{aligned}$$

□

Now, we show the probability that a node v is l decodable given that $G_l(v)$ is a tree. Note that the tree-based analysis of EC codes can be viewed as a variation of the and-or-tree analysis used for LT and LDPC codes.

Lemma 6. *Let* $v \in V$ *be a node such that* $G_l(v)$ *is a tree. Then, for any constant* $\epsilon > 0$ *and sufficiently large n,*

- *the probability that chunk v is l decodable is at least* $(1-\epsilon)\tau_d$, *and*
- *the probability that an overlapping packet in chunk v can be recovered by BP decoding on* $G_l(v)$ *is at least* $(1-\epsilon)\lambda_d$.

Proof. We first prove the first part. Consider the tree $G_l(v)$ rooted at v. Clearly, the root v has d children nodes and all other internal nodes have $d - 1$ children nodes. Let h_i be the probability that a node u at level i (here, we assume that the node v is at level l and the leaves are at level 0) is decodable when the decoding process of u is restricted within the subtree of $G_l(v)$ rooted at u. In the following, we calculate h_i in a bottom-up fashion.

For a leaf node u, since it cannot get any help from other chunks in $G_l(v)$,

$$h_0 = \Pr\{\mathrm{rk}(\mathbf{T}_u) = m\} = t_m = \beta_0.$$

For any node u at level i, $1 \leq i \leq l-1$, suppose that w out of the $d-1$ children nodes v' of node u are decodable when the decoding process of v' is restricted within the subtree of $G_l(v)$ rooted at v'. Note that each of these children nodes (regarded as chunks) overlaps with chunk u at a distinct packet. Therefore, when decoding u, these w overlapping packets provide additional w linearly independent coding vectors beyond $\mathbf{T}_u$. According to Theorem 2, the

probability that u is decodable is β_w. Since the local decoding processes of all the children nodes of node u are mutually independent, we have

$$\begin{aligned} h_i &= \sum_{w=0}^{d-1} \binom{d-1}{w} h_{i-1}^w (1-h_{i-1})^{d-1-w} \beta_w \\ &= \alpha_d(h_{i-1}). \end{aligned}$$

By induction, we have

$$h_i = \alpha_d^{i+1}(0),\ i = 0, 1, \ldots, l-1.$$

Similarly, since the node v in the level l has d children nodes,

$$\begin{aligned} h_l &= \sum_{w=0}^{d} \binom{d}{w} h_{l-1}^w (1-h_{l-1})^{d-w} \beta_w \\ &= \alpha_{d+1}(h_{l-1}) \\ &= \alpha_{d+1}(\alpha_d^l(0)). \end{aligned}$$

When $n \to \infty$, which implies $l \to \infty$, $\alpha_d^l(0) \to \alpha_d^*$. Therefore, $h_l \to \tau_d$ as $\alpha_{d+1}(y)$ is continuous. Hence, for any constant $\epsilon > 0$,

$$h_l > (1-\epsilon)\tau_d$$

for n sufficiently large.

Next, we prove the second part. Let u be an arbitrary children of node v. According to the above analysis, we know that node u is decodable with probability $h_{l-1} = \alpha_d^l(0)$. Meanwhile, under the condition that chunk u is not decodable, we can consider a new tree obtained by deleting the subtree rooted at u from $G_l(v)$. Similarly, we can show that node v can be decoded on the new tree with probability $\alpha_d(h_{l-1}) = \alpha_d^{l+1}(0)$. Therefore, the common packet of chunk u and chunk v can be decoded with probability at least $1-(1-\alpha_d^l(0))(1-\alpha_d^{l+1}(0))$, which approaches λ_d when n goes to infinity. The proof is accomplished. □

Lemma 5 and Lemma 6 together give a bound on the expected number of packets that can be recovered by BP decoding. Finally, we complete the proof of Theorem 4 by showing that the number of recovered packets is sharply concentrated to its expectation.

Proof of Theorem 4. Let Z be the number of input packets recovered when the decoding process of every chunk is restricted within its l neighborhoods, and let T be the number of nodes whose l neighborhood is a tree. According to Lemma 6 and noting that each chunk has $m-d$ non-overlapping packets and each of the d overlapping packets only appear in two chunks, we have that for sufficiently large n,

$$\mathrm{E}[Z|T] \geq (1-\epsilon/4)(\tau_d(m-d) + \lambda_d d/2)T. \tag{10}$$

Now, consider an exposure martingale on G as follows. Let

$$Z_0 = \mathbf{E}[Z|T], \tag{11}$$

and for $i = 1, 2, \ldots, n$, let

$$Z_i = \mathbf{E}[Z|\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_i, T],$$

where $\mathbf{T}_i$ denotes the transfer matrix of chunk $\mathbf{B}_i$. The sequence $Z_0, Z_1, \ldots, Z_n$ gives a standard Doob martingale [37]. Recall that the decoding process of each node v is restricted within the l neighborhood $G_l(v)$. Therefore, the exposure of $\mathbf{T}_v$ affects the expected number of recovered packets by at most the number of nodes in $G_l(v)$ times the chunk size. More precisely, for each $1 \leq i \leq n$,

$$|Z_i - Z_{i-1}| \leq m|G_l(v)| = \Theta\left((d-1)^l\right) = \Theta\left(n^{1/3}\right).$$

Applying the Azuma-Hoeffding inequality [37], we have

$$\begin{aligned} &\Pr\left\{Z_n \leq Z_0 - \frac{\epsilon}{4}(\tau_d(m-d) + \lambda_d d/2)T\right\} \\ &\quad\leq \exp\left(-\frac{\left(\frac{\epsilon}{4}(\tau_d(m-d) + \lambda_d d/2)T\right)^2}{2n\left(\Omega\left(n^{1/3}\right)\right)^2}\right) \\ &\quad= \exp\left(-\Omega\left(\epsilon^2 n^{1/3}\right)\right). \end{aligned} \tag{12}$$

Combining (10), (11), and (12) and noting that $Z_n = Z$, we get

$$\Pr\left\{Z \leq \left(1-\frac{\epsilon}{2}\right)(\tau_d(m-d) + \lambda_d d/2)T\right\} \leq \exp\left(-\Theta\left(\epsilon^2 n^{1/3}\right)\right). \tag{13}$$

Finally, since $T \geq (1-\epsilon/2)n$ almost surely according to Lemma 5, and Z is a natural lower bound on the number of packets that can be decoded by the BP decoding algorithm, we complete the proof of Theorem 4. □

3.4 Generator graph design

The above performance analysis implies that most d-regular graphs have the locally tree-like structure and hence the corresponding EC codes have the desired BP decoding performance. Therefore, the generator graph G can be designed randomly. That is, we randomly generated a d-regular graph as the generator graph, which can be done in expected $\mathcal{O}(n)$ time by the McKay-Wormald algorithm [38]. We will use this approach in our performance evaluation.

Since a randomly generated d-regular graph lacks a structure, we may need the whole adjacency matrix to preserve the graph. Note that the adjacency matrix is sparse and hence can be compressed. Alternatively, we may just

save the seed of the pseudorandom generator used for generating the d-regular graph.

Structured d-regular graphs can further simplify the generation and/or preservation of the EC code. When $d = 8$, Margulis' method [39] gives a structured 8-regular graph. However, currently, we do not have an efficient algorithm for generating structured regular graphs with any parameters d and n. Construction of structured regular graphs is of independent interest in mathematics and computer sciences, and many researches have been conducted on developing new approaches [40].

4 Performance evaluation

In this section, we evaluate the performance of EC codes with comparison against the state-of-the-art overlapped chunked codes (H2T codes [11] and random annex codes (RAC) [12]) and coded chunked codes (BATS codes [14] and L-chunked (LC) codes [17]). In all the evaluations, unless specified, we use $m = 32$ and $q = 256$, which gives a good balance between the achievable rates and the encoding/decoding cost.

4.1 Random transfer rank distributions

The performance of EC codes, as well as of BATS codes and LC codes, depends on the rank distribution $t = (t_0, t_1, \ldots, t_m)$. So, we first evaluate the performance of EC codes for general rank distributions, which may provide some guidance on the application of EC codes.

Recall that the achievable rate of chunked codes is upper bounded by $\bar{t}/m$ (see Proposition 3). For each fixed value $\bar{t}/m = 0.5, 0.6, 0.7, 0.8$, we sample a number of rank distributions[4] and derive the corresponding achievable rates of EC codes, BATS codes, and LC codes numerically. For BATS and LC codes, the achievable rate is obtained by solving the corresponding degree distribution optimization problem. For EC codes, the achievable rate is given by Theorem 4 with an optimized d. In particular, in order to see how the finite field size q affects the performance of EC codes, the achievable rate of EC codes for $q = 2$ is also evaluated.

The results are summarized in Table 3. From the table, we see that EC codes with $q = 256$ can achieve higher rates than with $q = 2$, which is consistent with the analysis given in Section 2. Also, when $\bar{t}/m = 0.5$, the average achievable rate of EC codes is much lower than the upper bound 0.5. (Actually, EC codes perform worse when $\bar{t}/m$ is lower.) The reason is roughly as follows: each input packet in an EC code is duplicated at most once, so the total number of packets in an EC code nm is no more than $2k$, where k is the number of input packets. When $\bar{t}/m = 0.5$, the effective number of received packets (removing the packets in each chunk that have linearly dependent coefficient vectors) is about $nm/2 \leq k$. We see that EC codes in this case may not have enough redundancy for recovering a significant fraction of the input packets.

When the value of $\bar{t}/m$ becomes larger, the achievable rate of EC codes consistently becomes more close to $\bar{t}/m$. When $\bar{t}/m = 0.8$, for example, the average achievable rate of EC codes is nearly 90 % of $\bar{t}/m$. It is not surprising to see that both BATS codes and LC codes outperform EC codes due to the much more complicated encoding process and degree distribution optimization in the former codes.

By comparing the maximum and minimum achievable rates, we notice that the performance of EC codes varies significantly for different rank distributions, especially when $\bar{t}/m$ is relatively small. When $\bar{t}/m = 0.5$, for some rank distributions, EC codes achieve more than 80 % of $\bar{t}/m$; while for some other rank distributions, EC codes can only achieve less than half of the rate of BATS/LC codes.

In many potential applications of chunked codes, the rank distributions of the transfer matrices have certain features, instead of occurring purely randomly. For instance, the number of packets in a chunk received by the destination node is a summation of multiple binomial random variables, which can be roughly approximated by a Poisson random variable. Also, in an optimized transmission scheme, if the average packet loss rate over the network is higher, the number M_j of packets transmitted for each chunk usually also becomes larger, so that the average rank $\bar{t}$ has a relatively large value [18]. In practice, EC codes can benefit from these features of rank distributions and achieve much higher rates than a rank distribution randomly generated. Therefore, in the remainder of this section, we focus on the performance of EC codes in a practical scenario.

Table 3 Achievable rates of EC/BATS/LC codes with 10 random rank distributions

	$\bar{t}/m = 0.5$			$\bar{t}/m = 0.6$			$\bar{t}/m = 0.7$			$\bar{t}/m = 0.8$		
	Average	Min.	Max.	Average	Min.	Max.	Average	Min.	Max.	Average	Min.	Max.
EC ($q = 2$)	0.103	0.015	0.172	0.500	0.487	0.511	0.569	0.547	0.589	0.677	0.645	0.691
EC	0.294	0.184	0.411	0.523	0.508	0.532	0.591	0.569	0.619	0.719	0.694	0.740
BATS	0.497	0.495	0.498	0.598	0.598	0.598	0.698	0.696	0.699	0.798	0.798	0.759
LC	0.478	0.470	0.486	0.581	0.570	0.592	0.687	0.673	0.697	0.786	0.778	0.792

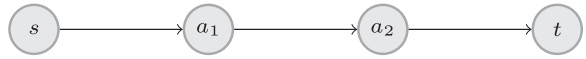

Fig. 2 Line network with length three. Node *s*: source node, node *t*: destination node

4.2 Line networks

We consider a line network formed by tandem homogeneous links, each of which has the same packet loss probability ϵ. Figure 2 illustrates a line network of length three. Line networks are generic building blocks of more complicated communication networks and have attracted a lot of research interests [19–21]. The chunk transmission schemes of line networks can be extended to general unicast networks and some multicast networks [14, 18], preserving the optimality. In order to compare with the line network capacity directly, we instead evaluate the *achievable network transmission rate*, i.e., the number of packets that are transmitted on average by one use of the network reliably.

We use the near-optimal chunk transmission scheme described in [18] over the line network. In this scheme, the chunks are transmitted in a sequential manner, and every node v, except for the destination node, transmits $M_j^{(v)}$ packets of each chunk $\mathbf{B}_j$, where $M_j^{(v)}$ is an integer-valued random variable. For the source node s, $M_j^{(s)}$ is just the variable M_j defined in Section 2.2. For all the network nodes and chunks, $M_j^{(s)}$ has the same mean value $\bar{M}$. For a fixed $\bar{M}$, the distribution of $M_j^{(v)}$ is optimized hop-by-hop according to the number of j packets received/possessed by node v. The value of $\bar{M}$ is chosen such that $\bar{t}/\bar{M}$ is maximized, which is an upper bound on the network transmission rate that can be achieved by any chunked code under this transmission scheme.

We evaluate the performance of EC, BATS, and LC codes in line networks with different network lengths and packet loss probabilities. The results as well as some important parameters are summarized in Tables 4, 5 and 6. From these tables, we can see that when the network length or packet loss probability is larger, the optimized $\bar{M}$ is also larger, keeping $\bar{t}/\bar{M}$ at a high value, close to the network capacity (note that if the computational cost and/or buffer size of intermediate nodes is restricted to be $\mathcal{O}(1)$, the network capacity is smaller than $1-\epsilon$ and decreases when the network length grows [20, 21]). Moreover, EC codes can achieve a network transmission rate that is about 91 % $\sim$ 97 % of the bound $\bar{t}/\bar{M}$ and is about 80 % $\sim$ 94.5 % of the network capacity $1-\epsilon$. This demonstrates the great real-world potential of EC codes.

Table 4 Achievable network transmission rates of chunked codes in line networks with $\epsilon = 0.1$

Network length	EC	LC	BATS	$\bar{t}/\bar{M}$	$\bar{M}$
2	0.851	0.874	0.878	0.879	32
3	0.825	0.852	0.866	0.866	33
4	0.817	0.853	0.857	0.857	33
5	0.809	0.847	0.850	0.850	33
6	0.795	0.840	0.844	0.845	34

Table 5 Achievable network transmission rates of chunked codes in line networks with $\epsilon = 0.2$

Network length	EC	LC	BATS	$\bar{t}/\bar{M}$	$\bar{M}$
2	0.743	0.764	0.772	0.773	35
3	0.718	0.752	0.756	0.757	36
4	0.702	0.741	0.745	0.746	36.5
5	0.691	0.732	0.737	0.738	37
6	0.682	0.727	0.731	0.731	37.5

4.3 Comparison with overlapped chunked codes

We then compare EC codes with two overlapped chunked codes: the chunked code with a head-to-tail type of overlapping (H2T) [11] and random annex codes (RAC) [12]. Since we do not have the analytical results to calculate the achievable rates of these two codes, we conduct simulations in line networks with $\epsilon = 0.2$ and lengths four and ten for the performance comparison. For each code and each length, we perform 10,000 runs of the simulation. In all the runs, the number of chunks in each code is set to be 500, which thus fixes the same transmission cost. The parameters involved in H2T and RAC are chosen optimally in the sense that the average number of decodable input packets is maximized. Note that given the parameters of a chunked code, the number of input packets is then determined, which varies over different classes of chunked codes. The empirical cumulative distribution functions (eCDFs) of the number of decodable input packets for each code in line networks with lengths four and ten are plotted in Figs. 3 and 4, respectively. From these

Table 6 Achievable network transmission rates of chunked codes in line networks with $\epsilon = 0.4$

Network length	EC	LC	BATS	$\bar{t}/\bar{M}$	$\bar{M}$
2	0.533	0.559	0.569	0.570	44
3	0.523	0.543	0.553	0.554	46
4	0.504	0.539	0.542	0.543	48
5	0.493	0.523	0.534	0.534	49
6	0.484	0.523	0.527	0.528	50

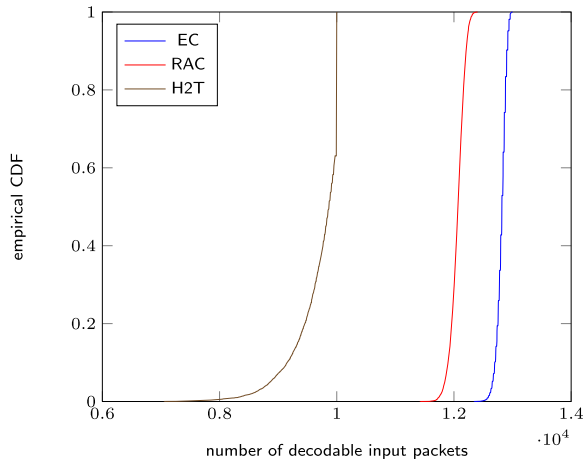

Fig. 3 The eCDF of the number of decodable input packets in a line network with length four. Here, $\bar{M} = 36.5$

figures, we can see that EC codes outperform both H2T and RAC significantly in both line networks.

5 Conclusions

In this paper, we studied the performance of overlapped chunked codes with constant chunk sizes. We proposed and analyzed EC codes, a novel class of random regular graph-based chunked codes, which outperform state-of-the-art overlapped chunked codes. Compared with coded chunked codes, EC codes can achieve a rate very close to that of BATS codes and L-chunked codes in line networks with a proper optimization of the transmission scheme, but EC codes can support causal encoding and have lower encoding complexity.

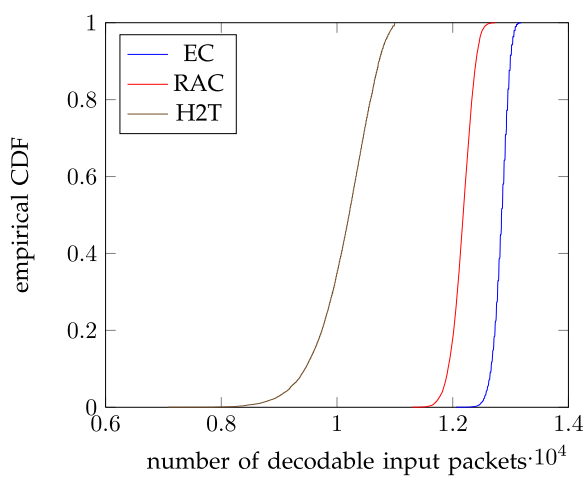

Fig. 4 The eCDF of the number of decodable input packets in a line network with length ten. Here, $\bar{M} = 39$

Endnotes

[1]For example, network protocols usually have a maximum transmission unit (MTU) ranging from hundred to thousand bytes.

[2]EC codes were motivated by the expander graphs, and the expansion property was applied in the first analysis of EC codes to obtain a lower bound on the achievable rates [31]. In this paper, we provide a better bound on the achievable rate without an explicit application of the expansion property, but the name of the code is preserved.

[3]If both d and n are odd, then an EC code with n chunks and degree d can be generated by attaching an arbitrary chunk to an EC code with $n-1$ chunks and degree d using the described method, which does not affect the asymptotic performance of EC codes.

[4]To the best of our knowledge, no efficient algorithms have been developed for uniformly sampling a rank distribution with a given mean value. Here, we use the following method for randomly sampling rank distributions. For a fixed $\bar{t}$, denote $a = \lfloor \bar{t} \rfloor$. We first sample a distribution $(t_0, t_1, \ldots, t_a)$ over the set $\{0, 1, \ldots, a\}$ and a distribution $(t_{a+1}, t_{a+2}, \ldots, t_m)$ over the set $\{a+1, a+2, \ldots, m\}$ using the method in [41], which gives almost uniform sampling of distributions over the corresponding set. Let $\eta = (\sum_{i=a+1}^{m} it_i - \bar{t})/(\sum_{i=a+1}^{m} it_i - \sum_{i=0}^{a} it_i) > 0$. Then, we get a distribution $(\eta t_0, \eta t_1, \ldots, \eta t_a, (1-\eta)t_{a+1}, (1-\eta)t_{a+2}, \ldots, (1-\eta)t_m)$, whose expectation is equal to $\bar{t}$.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This work was partially supported by NSFC Grants (Nos. 61501221, 61170069, 61373014, 91218302, 61321491, and 61471215); Natural Science Foundation of Jiangsu Province Grant (No. BK20150588), Science and Technology Pillar Program (Industry) of Jiangsu Province Grant (No. BE2013116); Collaborative Innovation Center of Novel Software Technology and Industrialization; EU FP7 IRSES MobileCloud Project Grant (No. 612212); and a grant from the University Grants Committee of the Hong Kong Special Administrative Region (Project No. AoE/E-02/08). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

Author details
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China. [2]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China.

Received: 28 June 2015 Accepted: 7 December 2015
Published online: 22 December 2015

References

1. R Ahlswede, N Cai, S-YR Li, RW Yeung, Network information flow. IEEE Trans. Inf. Theory. **46**(4), 1204–1216 (2000)
2. S-YR Li, RW Yeung, N Cai, Linear network coding. IEEE Trans. Inf. Theory. **49**(2), 371–381 (2003)
3. T Ho, R Koetter, M Medard, DR Karger, M Effros, in *Proc. IEEE International Symposium on Information Theory*. The benefits of coding over routing in a randomized setting, (2003), p. 442

4. T Ho, M Medard, R Koetter, DR Karger, M Effros, J Shi, B Leong, A random linear network coding approach to multicast. IEEE Trans. Inf. Theory. **52**(10), 4413–4430 (2006)
5. Y Wu, in *Proc. International Symposium on Information Theory*. A trellis connectivity analysis of random linear network coding with buffering, (2006), pp. 768–772
6. AF Dana, R Gowaikar, R Palanki, B Hassibi, M Effros, Capacity of wireless erasure networks. IEEE Trans. Inf. Theory. **52**(3), 789–804 (2006). doi:10.1109/TIT.2005.864424
7. DS Lun, M Medard, R Koetter, M Effros, On coding for reliable communication over packet networks. Phys. Commun. **1**(1), 3–20 (2008)
8. PA Chou, Y Wu, K Jain, in *Proc. 41st Allerton Conference on Communication, Control, and Computing*. Practical network coding, (2003), pp. 40–49
9. Z Liu, C Wu, B Li, S Zhao, in *Proc. IEEE International Conference on Computer Communications*. UUSee: Large-scale operational on-demand streaming with random network coding, (2010), pp. 1–9
10. D Silva, W Zeng, FR Kschischang, in *Proc. Workshop on Network Coding*. Sparse network coding with overlapping classes, (2009), pp. 74–79
11. A Heidarzadeh, AH Banihashemi, in *Proc. IEEE Information Theory Workshop*. Overlapped chunked network coding, (2010), pp. 1–5
12. Y Li, E Soljanin, P Spasojevic, Effects of the generation size and overlap on throughput and complexity in randomized linear network coding. IEEE Trans. Inf. Theory. **57**(2), 1111–1123 (2011)
13. S Yang, RW Yeung, in *Proc. IEEE International Symposium on Information Theory*. Coding for a network coded fountain, (Saint Petersburg, Russia, 2011), pp. 2647–2651
14. S Yang, RW Yeung, Batched sparse codes. IEEE Trans. Inf. Theory. **60**(9), 5322–5346 (2014)
15. K Mahdaviani, M Ardakani, H Bagheri, C Tellambura, in *Proc. International Symposium on Network Coding*. Gamma codes: a low-overhead linear-complexity network coding solution, (2012), pp. 125–130
16. K Mahdaviani, R Yazdani, M Ardakani, in *Proc. International Symposium on Network Coding*. Overhead-optimized gamma network codes, (2013), poster
17. S Yang, B Tang, in *Proc. IEEE Information Theory Workshop*. From LDPC to chunked network codes, (2014), pp. 406–410
18. B Tang, S Yang, B Ye, S Lu, S Guo, Near-optimal one-sided scheduling for coded segmented network coding. IEEE Trans. Comput. (2015), to appear
19. P Pakzad, C Fragouli, A Shokrollahi, in *Proc. IEEE International Symposium on Information Theory*. Coding schemes for line networks, (2005), pp. 1853–1857. doi:10.1109/ISIT.2005.1523666
20. U Niesen, C Fragouli, D Tuninetti, On capacity of line networks. IEEE Trans. Inf. Theory. **53**(11), 4039–4058 (2007)
21. BN Vellambi, N Torabkhani, F Fekri, Throughput and latency in finite-buffer line networks. IEEE Trans. Inf. Theory. **57**(6), 3622–3643 (2011). doi:10.1109/TIT.2011.2137070
22. Q Huang, K Sun, X Li, D Wu, in *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing*. Just fun: a joint fountain coding and network coding approach to loss-tolerant information spreading, (Philadelphia, PA, USA, 2014), pp. 83–92
23. X Xu, PKM Gandhi, YL Guan, PHJ Chong, *Two-phase cooperative broadcasting based on batched network code*. arXiv preprint arXiv:1504.04464, (2015). http://arxiv.org/abs/1504.04464
24. S Chachulski, M Jennings, S Katti, D Katabi, Trading structure for randomness in wireless opportunistic routing. SIGCOMM Comput. Commun. Rev. **37**(4), 169–180 (2007)
25. Y Lin, B Li, B Liang, in *Proc. IEEE International Conference on Network Protocols*. CodeOR: Opportunistic routing in wireless mesh networks with segmented network coding, (2008), pp. 13–22
26. D Koutsonikolas, C-C Wang, YC Hu, Efficient network-coding-based opportunistic routing through cumulative coded acknowledgments. IEEE/ACM Trans. Netw. **19**(5), 1368–1381 (2011)
27. P Maymounkov, NJA Harvey, DS Lun, in *Proc. 44th Allerton Conference on Communication, Control, and Computing*. Methods for efficient network coding, (2006)
28. M Luby, in *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*. LT codes, (2002), pp. 271–282
29. A Shokrollahi, Raptor codes. IEEE Trans. Inf. Theory. **52**(6), 2551–2567 (2006)
30. K Mahdaviani, R Yazdani, M Ardakani, Linear-complexity overhead-optimized random linear network codes. arXiv preprint arXiv:1311.2123 (2013). http://arxiv.org/abs/1311.2123
31. B Tang, S Yang, Y Yin, B Ye, S Lu, in *Proc. IEEE International Symposium on Information Theory*. Expander graph based overlapped chunked codes, (Cambridge, MA, USA, 2012), pp. 2451–2455
32. S Yang, RW Yeung, HF Cheung, HHF Yin, in *Proc. 52nd Allerton Conference on Communication, Control, and Computing*. BATS: Network coding in action, (2014), pp. 1204–1211
33. GE Andrews, *The Theory of Partitions vol. 2*. (Cambridge University Press, New York, NY, USA, 1998)
34. M Gadouleau, Z Yan, Packing and covering properties of subspace codes for error control in random linear network coding. IEEE Trans. Inf. Theory. **56**(5), 2097–2108 (2010)
35. L Shi, N Wormald, in *Surveys in combinatorics*. Models of random regular graphs (LMS Lecture Note Series 267, 1999), pp. 239–298
36. BD McKay, NC Wormald, B Wysocka, Short cycles in random regular graphs. Electron. J. Comb. **11**(1), 1–12 (2004)
37. M Mitzenmacher, E Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. (Cambridge University Press, New York, NY, USA, 2004)
38. BD McKay, NC Wormald, Uniform generation of random regular graphs of moderate degree. J. Algoritm. **11**(1), 52–67 (1990)
39. G Margulis, Explicit construction of concentrators. Probl. Peredachi Inf. **9**(4), 325–332 (1973)
40. S Hoory, N Linial, A Wigderson, Expander graphs and their applications. Bull. Amer. Math. Soc. **43**(4), 439–561 (2006)
41. NA Smith, RW Tromble, Sampling uniformly from the unit simplex. http://www.cs.cmu.edu/~nasmith/papers/smith+tromble.tr04.pdf