TRAINING OVERHEAD FOR DECODING RANDOM LINEAR NETWORK CODES

Maximilian Riemensberger[b], Yalin Evren Sagduyu[a], Michael L. Honig[a], and Wolfgang Utschick[b]

[a]Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, USA
[b]Associate Institute for Signal Processing, Munich University of Technology, Munich, Germany
riemensberger@tum.de, y-sagduyu@northwestern.edu, mh@eecs.northwestern.edu, utschick@tum.de

ABSTRACT

We consider multicast communications from a single source to multiple destinations over a network of erasure channels. Linear network coding maximizes the achievable (min-cut) rate, and a distributed code assignment can be realized by choosing codes randomly at the intermediate nodes. It is typically assumed that the coding information (combining coefficients) at each node is included in the packet overhead, and forwarded to the destination. Instead, we assume that the network coding matrix is communicated to the destinations by appending training bits to the data bits at the source. End-to-end channel coding can then be applied to the training and data either separately, or jointly, by coding across both training and information bits. Ideally, the training overhead should balance the reliability of communicating the network matrix with the reliability of data detection. We maximize data throughput as a function of the training overhead, and show how it depends on the network size, erasure probability, number of independent messages, and field size. The combination network is used to illustrate our results, and shows under what conditions throughput is limited by training overhead.

I. Introduction

Network coding has been shown to have significant advantages relative to more conventional store-and-forward-based routing solutions [1]. For multicast communications linear network coding achieves the min-cut capacity, and eliminates the need for combinatorial-based routing optimization [2]. Namely, packets consist of symbols from a finite field, and are linearly combined at each node along source-destination paths. Different low-complexity network coding algorithms have been proposed in the context of both deterministic [3] and random network coding [4]. In situations where the network topology and channels are subject to change, such as in wireless networks, the robustness of network coding becomes a significant issue [4].

We consider random linear network coding over networks with erasure channels. Namely, each internal node transmits a random linear combination of the input packets across different links. To decode, each destination node must know the coding matrix, which maps the source symbols to received symbols. We assume that the destinations have no *a priori* knowledge of the coding matrices, and the coding coefficients across nodes and packets are chosen independently. To communicate the network coding matrix to the destinations, it is typically assumed that the combining coefficients at each internal node are forwarded in the packet header across each link. The destination node can then reconstruct the network coding matrix, given knowledge of the network configuration (i.e., routing paths). Of course, the associated packet overhead reduces the achievable rate, assuming that the overhead and data share the same bandwidth.

Changes in the network topology and channel errors may introduce errors in the packet overhead, which would likely prevent successful packet decoding. As the network size grows, so does the coding complexity needed to guarantee reliable reception of the combining coefficients (e.g., by coding packets over intermediate links). Furthermore, for wireless networks with unreliable links, residual channel errors may still accumulate at the destinations. An issue, which then arises, is how much error protection should be used to communicate the network coding matrix.

Here we assume that the network coding matrix is communicated to the destinations by including a training sequence as part of the data packet *at the source node*. The intermediate nodes do not add additional overhead, but simply forward linear combinations of the input packets. The destination can decode the coding matrix along with the data. This can be done separately or jointly, depending on whether the channel coding is applied separately to the training, or jointly across both training and data. With training the destination node does not need to know anything about the network topology. The source need only ensure that sufficient training is provided to communicate the network coding matrix reliably.[1]

For both separate and joint coding/decoding of training and data bits, we analyze the effect of the training overhead on reliability (probability of decoding error) and

This work was supported in part by the U.S. Army Research Office under grant W911-NF-06-1-0339.

[1]The destination must know the number of training bits and their placement within the packet.

978-1-4244-2677-5/08/$25.00 ©

throughput. Namely, we quantify the length of the training sequence needed to achieve a target throughput, taking into account the packet overhead, and assuming a retransmission protocol for error control [5]. In general, this depends on the network size, erasure probability, packet length, and the field size. We also observe that joint coding over training and data automatically balances the reliability of data detection with the reliability of decoding the network coding matrix.

We illustrate our results with the combination network [6], which is a three-layer network with one source and multiple destinations. Network coding is performed only at the source, which transmits multiple packets simultaneously to multiple destinations. For this network routing cannot achieve the max-flow min-cut rates, and it was shown in [6] that the network coding gain can be arbitrarily large, ignoring the overhead needed to provide the network coding matrices to the destinations. We determine the minimum packet overhead needed to detect the network coding matrix, and show that this overhead limits the throughput as the network size increases.

Finally, we remark that the training approach presented is quite general, and is not confined only to the erasure network model considered here. Namely, it can applied to networks with different channel models, which generate errors or erasures. The erasure model is considered here due to its practicality, and analytical tractability.

The paper is organized as follows. Section II presents the system model and describes the training-based detection of the network coding matrix. In Section III, we introduce the separate and joint coding/decoding schemes for the coding matrix and data packets. Section IV formulates the overhead optimization problem using throughput as the objective criterion, and Section V presents results for combination networks. Finally, we draw conclusions and present thoughts for future work in Section VI.

II. System Model

We consider an acyclic delay-free network with independent links. A single source node wishes to deliver the same messages to a set of destinations $\mathcal{D}$. Random linear network coding over a finite field $\mathbb{F}_q$ with q elements is assumed, where $q = 2^L$ and $L \in \mathbb{N}$ is a natural number. While packets are relayed through the network, some symbols in the packets may be erased. We define an extended set[2] $\mathbb{F}_{q,e} = \mathbb{F}_q \cup \{e\}$ that includes the erasure symbol "e" to mark the corresponding positions in each packet.

The source messages are random row vectors $\mathbf{s}^{\mathrm{T}} \in \mathbb{F}_q^{1\times D}$ of D symbols, or equivalently DL bits, where $\{\cdot\}^{\mathrm{T}}$ denotes matrix transpose. The elements of all messages are independent and uniformly distributed random variables. These messages are mapped to packets (row vectors) $\mathbf{x}^{\mathrm{T}} \in \mathbb{F}_q^{1\times K}$ of K symbols, or equivalently, $B = KL$ bits. At each packet interval, the source transmits $N \le M$ independent messages $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_N]^{\mathrm{T}} \in \mathbb{F}_q^{N\times D}$, encoded as packets $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}} \in \mathbb{F}_q^{N\times K}$, over the network, where M is the minimum cut capacity of the network.

A. Random Network Coding over Erasure Channels

With random linear network coding each matrix of source packets $\mathbf{X}$ experiences a linear transformation $\mathbf{G}^d \in \mathbb{F}_q^{M_d\times N}$ from the source to destination $d \in \mathcal{D}$, where $\mathbf{G}^d$ is the random coding matrix and M_d is the number of incoming links at d. We assume that the entries of the coding matrices are uniformly distributed and independent (both within each matrix and across destinations). In practice, the network topology may introduce dependence among the coding matrices, which we ignore here. This facilitates tractability, and also serves to provide a worst-case estimate of the training overhead and associated throughput.

In the absence of channel erasures, the received packets at destination d are therefore the rows of $\mathbf{V}^d = \mathbf{G}^d\mathbf{X} \in \mathbb{F}_q^{M_d\times K}$. With erasures we denote the matrix of received packets as

$$\mathbf{Y}^d = \mathcal{H}^d(\mathbf{V}^d) = \mathcal{H}^d(\mathbf{G}^d\mathbf{X}). \tag{1}$$

where $\mathbf{Y}^d \in \mathbb{F}_{q,e}^{M_d\times K}$, and $\mathcal{H}^d$ denotes the operation of a discrete memoryless symbol erasure channel. We will assume that the symbol erasures are independent and identically distributed across all entries of the output matrix $\mathbf{Y}^d$, and denote ϵ_d as the corresponding probability of symbol erasure. This serves as an end-to-end channel model between the source and destinations and does not account for statistical dependencies, which may be introduced by the network topology. (Alternatively, we can assume an interleaver and deinterleavers at the source and destinations, respectively.)

With these assumptions the capacity of the channel corresponding to input $[\mathbf{V}^d]_{m,k}$ and output $[\mathbf{Y}^d]_{m,k}$ ((m,k)th entry of $\mathbf{V}^d$ and $\mathbf{Y}^d$, respectively) is

$$c_d = 1 - \epsilon_d \tag{2}$$

symbols per channel use (i.e., the capacity of the erasure channel). The multicast capacity (source to destination) is given by

$$C_d = M \min_{d\in\mathcal{D}} c_d \tag{3}$$

symbols per channel use, where M is the min-cut capacity. This assumes that the erasure probabilities are known to the source and all destinations. No knowledge of the network

[2]This set is not a field, since addition and multiplication with erasures are not defined.

topology is assumed at the destinations. The destinations need only know the number of packets N transmitted by the source. The source only needs to know the min-cut capacity M and the set of destinations $\mathcal{D}$.

B. Relation of Training to Linear Coding

We now describe how training can be used to detect the network coding matrices at the destinations. We first assume that the source packets $\boldsymbol{X}$ consist only of training symbols, which are known at the destinations. Referring to (1), our objective is to detect the *code matrix* $\mathbf{G}^d$ given the *training matrix* $\boldsymbol{X}$. Hence the training matrix acts as a code, which is applied to the (unknown) network code matrix. The destinations then detect $\mathbf{G}^d$ by decoding $\boldsymbol{X}$. Training-based detection of the coding matrix is therefore achieved by decoding the linear channel code represented by $\boldsymbol{X}$.

The performance of the code matrix detection depends on the particular training sequence transmitted. Namely, we wish to transmit a "good" code as the training sequence, which has a low-complexity decoding algorithm. For symbol erasure channels, linear maximum distance separable (MDS) codes are optimal in terms of decoding error probability [7]. The number of erasures that can be recovered is bounded by the difference between the codeword length and the sourceword length. The existence of such codes is guaranteed if the codeword length does not exceed the field size q [7].

Each destination requires only one decoder if the end-to-end channel code and the training are the same type of linear code. Next we combine training and data in each packet and observe that a single code can be used for joint coding and detection of both the network coding (system) matrix and the data. Each packet is then a single codeword with enough redundancy to provide for reliable system and data detection in the presence of channel errors.

III. Encoding and Decoding

For the network coding model described in the preceding section there are three sources of errors:

(i) Erasures of data symbols;
(ii) Insufficient training symbols (perhaps caused by erasures); and
(iii) The rank of the coding matrix for at least one destination is smaller than N.

An erasure correcting code can protect against the first two sources of error, but not the third. Namely, the probability that the coding matrix is rank-deficient decreases as the field size q increases. Therefore, the nodes applying coding in the network need to agree on a common field size that is large enough for successful transmission.

A. Encoding

The N source messages corresponding to the coded packets $\mathbf{X}$ are denoted as $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_N]^{\mathrm{T}}$. To communicate the coding matrix, we concatenate each source message $\mathbf{s}_n^{\mathrm{T}}$ with the nth unit vector $\boldsymbol{e}_n^{\mathrm{T}}$, i.e., we transmit $[\boldsymbol{e}_n^{\mathrm{T}} \ \mathbf{s}_n^{\mathrm{T}}]$ for each n. Therefore, $\mathbf{S}$ is extended to $[\boldsymbol{I}_N \ \mathbf{S}]$, where the $N \times N$ identity matrix $\boldsymbol{I}_N$ is the uncoded overhead. In the absence of erasures the uncoded received matrix of packets is $[\mathbf{G}^d \ \mathbf{G}^d\mathbf{S}]$, so that the coding matrix is obtained directly.

For reliability in the presence of erasures, we must code both the training and the data. For the **individual** training and channel coding scheme, we use two MDS codes, a $(T, N)_q$ code for training and a $(K-T, D)_q$ code for data protection with respective codeword lengths T and $K-T$, and respective sourceword lengths N and D, all measured in the number of q-ary symbols.

Hence instead of passing along the combining coefficients as additional overhead on each link, as suggested in [4], all of the necessary overhead is included at the source. The encoding procedure for this scheme is described by

$$\mathbf{X} = [\boldsymbol{F}_{11} \ \mathbf{S}\boldsymbol{F}_{22}] = [\boldsymbol{I}_N \ \mathbf{S}]\boldsymbol{F}, \tag{4}$$

where $\boldsymbol{F}_{11}$ and $\boldsymbol{F}_{22}$ are the generator matrices of the channel codes for training and data, respectively, and the overall generator matrix is given by

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_{11} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{F}_{22} \end{bmatrix}. \tag{5}$$

Each extended source message $[\boldsymbol{e}_n^{\mathrm{T}} \ \ \mathbf{s}_n^{\mathrm{T}}]$ is therefore mapped to the transmitted packet $\mathbf{x}_n^{\mathrm{T}}$, where the first T symbols are the coded training bits (equivalently, the network coding overhead), and the remaining $K-T$ symbols are the coded source data. The redundancy introduced for training (i.e., training length T) strongly influences the reliability of data detection, and can be optimized. Namely, if T is too short the coding matrix is unreliable, whereas if T is too large, then not enough symbols are reserved for the data.[3]

Referring to (4), there is actually no need to constrain the generator matrix $\boldsymbol{F}$ to have the block diagonal structure (5). Instead, we can apply **joint** training and data protection with the same encoding procedure

$$\mathbf{X} = [\boldsymbol{I}_N \ \mathbf{S}]\boldsymbol{F}, \tag{6}$$

[3]This is analogous to the optimization of training overhead for MIMO channel estimation considered in [8]. A key difference is that here we wish to *detect* the system matrix with acceptable probability of error, as opposed to *estimating* a complex-valued channel matrix. Also, the joint coding scheme described here is not directly applicable to the MIMO estimation problem.

where $\boldsymbol{F}$ is a *full* generator matrix. For the analysis that follows, we will assume that $\boldsymbol{F}$ is a $(K, N+D)_q$ MDS code.

Contrary to the individual scheme, the overhead and data parts are not distinguished in the transmitted packets $\mathbf{x}_n^{\mathrm{T}}$ but both parts are jointly encoded into the entire packet. Since only one channel code is used for both training and data, there is no explicit trade-off between the reliability of the training bits versus the data bits. Namely, the channel code $\boldsymbol{F}$ implicitly balances the training and data protection.

In general, joint encoding of training and data should achieve a lower probability of error than separate encoding for any choice of training length T. (An error event occurs when a received packet with erasures is incorrectly decoded, given that both data and overhead are unknown.) The reason is that coding the overhead and data jointly with one long code is better than coding them separately with two shorter codes.

B. Decoding

For the encoding procedures (4) and (6), the output at destination d is given by

$$\mathbf{Y}^d = \mathcal{H}^d(\mathbf{G}^d\mathbf{X}) = \mathcal{H}^d(\mathbf{G}^d[\boldsymbol{I}_N\ \mathbf{S}]\boldsymbol{F}) = \mathcal{H}^d(\mathbf{U}^d\boldsymbol{F}), \quad (7)$$

where the auxiliary random matrix $\mathbf{U}^d \in \mathbb{F}_q^{M_d \times (N+D)}$ is partitioned as

$$\mathbf{U}^d = [\mathbf{U}_1^d\ \mathbf{U}_2^d] = [\mathbf{G}^d\ \mathbf{G}^d\mathbf{S}]. \quad (8)$$

The left part $\mathbf{U}_1^d = \mathbf{G}^d$ has N columns and the right part $\mathbf{U}_2^d = \mathbf{G}^d\mathbf{S}$ has D columns. With separate coding of training and data the output (7), $\boldsymbol{F}$ is block diagonal, and $\mathbf{Y}^d$ can be further decomposed as

$$\mathbf{Y}^d = \mathcal{H}^d([\mathbf{G}^d\boldsymbol{F}_{11}\ \mathbf{G}^d\mathbf{S}\boldsymbol{F}_{22}]), \quad (9)$$

where the left and right parts correspond to the training and data, respectively.

Because the channel code $\boldsymbol{F}$ and the random network code $\mathbf{G}^d$ are linear, we can decode these sequentially. Namely, we first decode the channel code $\boldsymbol{F}$ to retrieve the auxiliary matrix $\mathbf{U}^d$ from the output $\mathbf{Y}^d$, and subsequently decode the actual source messages $\mathbf{S}$ from the decoded auxiliary packets (e.g., by inverting $\mathbf{G}^d$). The decoding procedure is given by the following algorithm:[4]

1: If less than N rows of $\boldsymbol{Y}^d$ have an erasure pattern that can be corrected, then return a decoding error and request a retransmission.

2: Recover the decodable rows of the auxiliary matrix $\boldsymbol{U}^d$ from the respective rows of $\boldsymbol{Y}_d$ by decoding the joint channel code, or the two individual channel codes.

3: If the left part $\mathbf{U}_1^d$ of the decoded matrix $\boldsymbol{U}^d = [\mathbf{U}_1^d\ \ \mathbf{U}_2^d]$ has rank smaller than N, then return a decoding error and request retransmission.

4: Solve the system of linear equations $\boldsymbol{U}_1^d\boldsymbol{S} = \boldsymbol{U}_2^d$ for $\boldsymbol{S}$.

For the joint coding scheme, the first step returns a decoding error if the number of erasures in more than $M^d - N$ rows of $\mathbf{Y}^d$ exceeds $K-N-D$. For the individual coding scheme, the first step returns a decoding error if in more than $M^d - N$ rows of $\boldsymbol{Y}^d$ either the number of erasures within the first T symbols is greater than $T-N$ or the number of erasures in the remaining $K-T$ symbols is greater than $K-T-D$. The last two steps decode the network code by solving a set of linear equations. Successful decoding is possible only if the decoded rows of $\boldsymbol{U}_1^d$ are linearly independent.

C. Decoding Error Probability at a Single Destination

In this section, we derive the probability of successful decoding at a single destination d. There are two different causes of decoding errors:

(i) The network erases too many symbols;

(ii) The randomly selected combining coefficients give a rank-deficient coding matrix.

For the joint training and decoding scheme with an MDS erasure code the probability that destination d can decode a particular row of $\mathbf{Y}^d$ is given by

$$Q_d^{(\mathrm{j})} = \sum_{\alpha=0}^{K-D-N} \binom{K}{\alpha} \epsilon_d^{\alpha}(1-\epsilon_d)^{K-\alpha}. \quad (10)$$

For the individual training and decoding scheme, two MDS codes need to be decoded for each row. Therefore the probability that destination d can decode a certain row of $\mathbf{Y}^d$ is given by

$$Q_d^{(\mathrm{i})} = \sum_{\alpha=0}^{T-N}\sum_{\beta=0}^{K-T-D} \binom{T}{\alpha}\binom{K-T}{\beta} \epsilon_d^{\alpha+\beta}(1-\epsilon_d)^{K-\alpha-\beta}. \quad (11)$$

For both schemes the probability that destination d can decode all source messages can then be written as

$$P_d = \sum_{m=N}^{M_d} \binom{M_d}{m} Q_d^m (1-Q_d)^{M_d-m}\Phi(m), \quad (12)$$

where Q_d is either $Q_d^{(\mathrm{i})}$ for the individual scheme or $Q_d^{(\mathrm{j})}$ for the joint scheme, and $\Phi(m)$ is the probability that

[4]This decoding procedure is suboptimal; however, the improvement in performance achieved by maximum likelihood decoding of $\mathbf{S}$ should typically be small.

the detected network coding matrix $\mathbf{U}_1^d$ has full rank. From counting arguments in [9] the probability that m successfully decoded rows of the system matrix $\mathbf{G}^d$ have rank N is given by

$$\Phi(m) = \prod_{i=0}^{N-1} (1 - q^{i-m}). \tag{13}$$

IV. Network Coding Overhead

In this section, we evaluate the necessary overhead for detecting the network coding matrix. The source has three parameters to choose, namely, the number of data symbols per packet D and the number of independent packets per transmission N. Those can be further optimized by adapting the number of bits per symbol L (equivalently, the field size $q = 2^L$). In particular, L must be large enough to ensure that a network code exists for multicasting packets to all destinations in $\mathcal{D}$ [10], and that an MDS code exists with packet length B (bits) [7].

For comparison purposes, we introduce a reference scheme in which each destination knows its coding matrix *a priori*. Hence instead of appending the training overhead $\boldsymbol{I}_N$, we assume that N additional data symbols per packet are transmitted. The coding matrix in this reference scheme may still be rank-deficient. That event would occur with the same probability as in the other schemes, so that the successful decoding probability P_d at destination d is given by (12), where Q_d is given in (10) evaluated at $N = 0$.

The performance criterion of interest is network throughput Λ, which is defined as the average amount of data that can be successfully communicated to all destinations in a single network use. [5] It depends on the network data rate

$$R = N\frac{D}{K} = N\frac{LD}{B}, \tag{14}$$

i.e., the number of data messages the source transmits in one network use, and also on the average delay, i.e., the expected number of network uses (including retransmissions) needed to ensure that all destinations receive all packets correctly.

We assume that an automatic repeat request (ARQ) scheme is used to ensure reliable communications. Packets are retransmitted until they are successfully received at all destinations. The ARQ schemes considered require one bit (ACK/NACK) feedback per transmitted packet to inform the source whether or not the transmitted packets have been successfully decoded.

[5] Alternatively, we could consider data rate subject to a maximal tolerated erasure probability.

A throughput rate, which approaches the max-flow min-cut bound can be achieved using the joint queue management and coding scheme in [5], in which case

$$\Lambda = R \min_{d\in\mathcal{D}} P_d. \tag{15}$$

We will use this throughput as our performance objective. Note that for the multicast application considered, ARQ schemes may require additional overhead, which specifies the destinations and possibly the coding methods for each retransmission. We do not consider this overhead here.

The resulting throughput can be optimized with respect to the number of data symbols per packet D as well as the number of independent source packets N. Note that both R and P_d depend on D and N through (14) and (12), so that this is a nonlinear integer program, which must be solved numerically. In addition, optimizing the training length T for the individual scheme is another integer program.

For comparison with the reference scheme we define the relative throughput loss due to overhead as

$$\Delta = 1 - \frac{\Lambda}{\Lambda^{(\mathrm{r})}}, \tag{16}$$

where $\Lambda^{(\mathrm{r})}$ is the throughput accounting for overhead. Alternatively, we can define the total packet overhead due to both channel and network coding as

$$O_{\mathrm{tot}} = \frac{K - D}{K}. \tag{17}$$

We also wish to determine how the overhead is allocated across network and channel coding. For the individual scheme the overhead for network coding is simply

$$O_{\mathrm{NC}}^{(i)} = \frac{T}{K}, \tag{18}$$

since the first T symbols of a packet are training and the remaining $K-T$ symbols are encoded data. In contrast, no distinct part of the packet is associated with the overhead in the joint coding scheme. However, since N training bits are appended to each information message, we can define the network coding overhead for the joint scheme as

$$O_{\mathrm{NC}}^{(j)} = \frac{N}{N + D}. \tag{19}$$

V. Throughput and Overhead Trade-Offs

We use the combination network model to evaluate the achievable throughput and packet overhead necessary for network coding. The $\binom{A}{M}$ combination network is a three-layer network with one source node, A intermediate nodes and $\binom{A}{M}$ destination nodes. The source has one outgoing link to each intermediate node. Each destination node has M incoming links from a unique set of intermediate nodes.

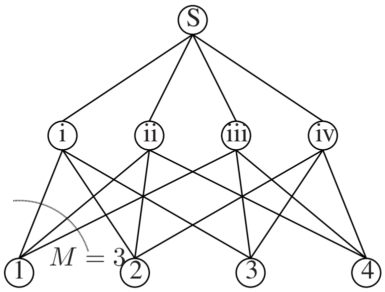

Figure 1. Combination network $\binom{4}{3}$

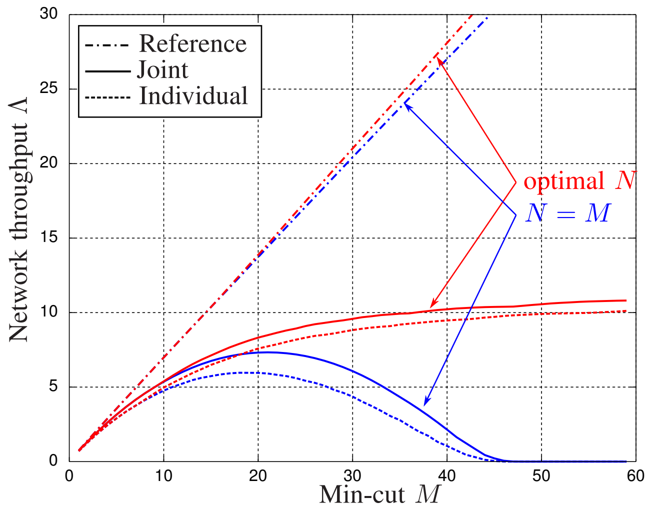

Figure 2. Network throughput Λ as a function of the Min-cut M optimized over D (blue curves) and N, D (red curves).

The min-cut capacity of the combination network is M. Figure 1 shows an example of a $\binom{4}{3}$ combination network. In this network the source node is the only node, which performs network coding. It chooses A random linear combinations of its N independent packets and transmits one coded packet to each intermediate node. Each of these nodes forwards its received linear combination to all connected destination nodes.

The *bit* erasure probability is denoted as $\epsilon^{(\mathrm{b})}$. This is assumed to be the same on all links and across all bits. For the combination network, the end-to-end symbol erasure probability is

$$\epsilon = 1 - (1 - \epsilon^{(\mathrm{b})})^{2L} \tag{20}$$

and is the same for all destinations. For the results that follow we assume $A = 60$ nodes in layer 2, $M = 1, \ldots, 59$ connections per destination, bit erasure probability $\epsilon^{(\mathrm{b})} = 10^{-2}$, field size $q = 2^8$, and packet length $B = 480$ bits. The choices of erasure probability and packet length correspond to a wireless scenario. The field size is chosen to guarantee that MDS codes exist.

Figure 2 shows the throughput given by (15) as a function of M. The throughput of the reference scheme grows approximately linearly with M, since no network coding

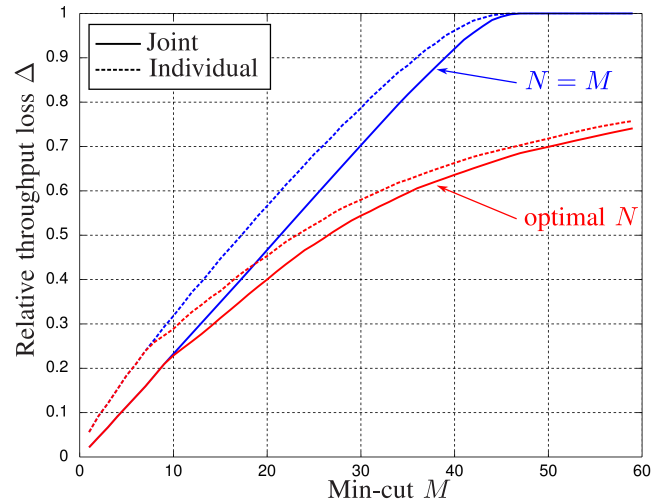

Figure 3. Throughput loss Δ vs. Min-cut M with opt. parameters.

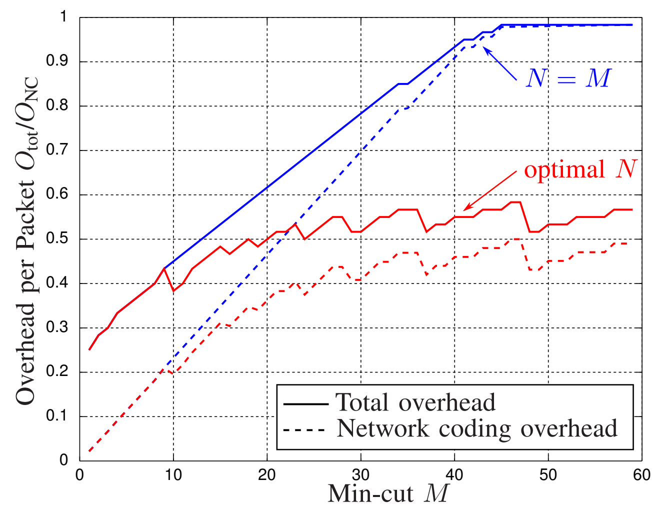

Figure 4. Total (O_{tot}) and network coding (O_{NC}) overhead vs. min-cut M for joint training and coding with optimized parameters.

overhead is necessary. However, if the coding matrices are unknown to the destinations, then the throughput is bounded for both $N = M$ and the optimized value of N. The results show that for fixed $N = M$ the throughput goes to zero as M increases, since the overhead occupies almost the entire packet. For large M the optimal value of N is strictly smaller than M for both the reference and training-based schemes. This is illustrated in the figure, which shows that the throughput, optimized with respect to N, is better than that with fixed $N = M$. (This follows from (12), where N changes the probability Φ given in (13), and the number of terms in the sum (12).) The figure also shows that, as expected, joint coding over training and data performs better than separate coding, although the difference is small.

Figure 3 shows the throughput loss with respect to the reference scheme. For fixed $N = M$, the loss increases approximately linearly until it reaches one (i.e., the throughput is zero), whereas for the optimal choice of N it increases more slowly and approaches one asymptotically.

The overhead per packet is shown in Figure 4 for the joint coding scheme. Both total overhead (due to both channel and network coding) and overhead due to only network coding are shown. For the optimal choice of N, the overhead is between between 0.5 and 0.6 for large M, whereas for $N = M$ the overhead approaches a value close to one (specifically, $(K-1)/K$). That is, as M increases with $N = M$, all but one symbol are used for overhead. The fluctuations with M shown in the figure are due to the optimization of Λ over integer values of N. For $N = M$, the total overhead of the individual scheme is always greater than the overhead of the joint scheme. However, the optimal values of N are generally different for different schemes making it difficult to compare the overhead properties. For large values of M, the total overhead is approximately equal for both schemes on the average, although the total overhead is different for most values of M.

VI. Conclusions

Methods for communicating network coding matrices from a source to multiple destinations have been presented based on the use of training sequences. With these methods the destinations do not require knowledge of the network topology. In that sense training schemes are robust with respect to topology changes (e.g., due to mobility). Also, the end-to-end channel code(s) used to code the training and data at the source (as opposed to the codes for each link) provides additional robustness with respect to residual errors, which may accumulate across unreliable links.[6]

With separate coding of the training and data, optimization of the training balances the trade-off between the reliability of communicating the code and the packet overhead. This trade-off is not explicitly present with joint coding across training and data, since the channel code automatically balances the reliability of training and data bits. Joint coding performs somewhat better than separate coding, due to the longer code length, and should not be much more complex to decode. Furthermore, from a system point of view, decoding a single fixed-length code (which depends on the packet length and field size) is likely to be preferable to decoding two shorter variable-length codes.

For a given network topology and packet length the throughput for each training scheme, accounting for the overhead, can be maximized over the network parameters (e.g., number of packets simultaneously transmitted by the sources). Results for the combination network show that as the min-cut M becomes large, the optimized overhead is approximately half of the packet size, and that this limits the overall throughput.

To ensure that sufficient training is appended to the data bits, the source needs to know the min-cut capacity. This might be provided through feedback, e.g., which indicates the rank of the detected coding matrices. Adaptive feedback schemes, which acquire the necessary knowledge about the network and adapt the code rate and the number of independent source packets per transmission, is a topic for future work.

Our system model is based on the assumption that the coding coefficients are independent and randomly chosen for each packet transmission. This is primarily for analytical convenience, and also provides a worst-case estimate of the amount of overhead required. Another possibility for future work is to consider an alternative model, where the coding matrices over successive transmissions are correlated. The overhead could then be reduced by applying joint source-channel coding, which compresses the sequence of code matrices. Finally, other types of feedback schemes could be considered, which further improve throughput at the expense of additional overhead.

[6]In practice, the channel code may be used in combination with a CRC check.

References

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

[2] 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.

[3] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *Information Theory, IEEE Transactions on*, vol. 51, no. 6, pp. 1973–1982, June 2005.

[4] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[5] Y. E. Sagduyu and A. Ephremides, "On Network Coding for Stable Multicast Communication," in *Proc. IEEE MILCOM*, Oct. 2007.

[6] C. K. Ngai and R. W. Yeung, "Network Coding Gain of Combination Networks," in *IEEE ITW*, San Antonio, Texas, Oct. 2004, pp. 283–287.

[7] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North-Holland Publishing Company, 1977.

[8] B. Hassibi and B. M. Hochwald, "How much training is needed in multiple-antenna wireless links?" *IEEE Trans. Inform. Theory*, vol. 49, no. 4, pp. 951–963, Apr. 2003.

[9] T. Migler, K. E. Morrison, and M. Ogle, "How Much Does a Matrix of Rank k Weigh?" *Mathematics Magazine*, vol. 79, no. 4, pp. 262–271, Oct. 2006.

[10] R. Koetter and M. Medard, "An Algebraic Approach to Network Coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.