Tunable Sparse Network Coding

Soheil Feizi1, Daniel E. Lucani2, Muriel Médard1

1Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
2Instituto de Telecomunicações, DEEC, Faculdade de Engenharia da Universidade do Porto, Porto 4200-465, Portugal
Email:sfeizi@mit.edu; dlucani@fe.up.pt; medard@mit.edu

Abstract—A fundamental understanding of the relationship between delay performance and complexity in network coding is instrumental towards its application in practical systems. The main argument against delay-optimal random linear network coding (RLNC) is its decoding complexity, which is $O(n^3)$ for $n$ original packets. Fountain codes, such as LT and Raptor codes, reduce the decoding load on the receiver but at the cost of introducing additional, non-negligible delay. The source of this increased delay is the inherent sparsity of the code, which significantly reduces the impact of a new coded packet, i.e., the probability of a packet to be linearly independent from the knowledge at the receiver, as we approach the end of the transmission (when few degrees of freedom are missing). Thus, the additional overhead is mainly due to the transmission of the last data packets. Our key observation is that switching gears to denser codes as the transmission process progresses could considerably reduce the delay overhead while maintaining the complexity advantages of a sparse code. We propose tunable sparse network coding as a dynamic coding mechanism with a code structure with two regions: a sparse region, with various levels of sparsity, and a dense region, where packets are generated as per RLNC. We characterize the problem in multicast sessions on general networks and illustrate trade-offs in especial cases of interest. We also present a novel mechanism to perform efficient Gaussian elimination for sparse matrices that guarantees linear decoding complexity with high probability.

I. INTRODUCTION

Network coding constitutes a disruptive concept for the operation of communication networks [1] that has evolved from an information theoretic result for achieving capacity in multicast to become an enabler of a wide variety of applications, including communications in ad-hoc networks, data gathering in sensor networks, and data storage. One of the key features of network coding is that it encourages the mixing (coding) of data packets at intermediate nodes, rather than limiting intermediate nodes to store and forward packets. Under this premise, it is no longer required for the system to keep track of which packets have been received: receivers need only accumulate enough coded packets in order to recover the information. Ref. [2] and [3] show that linear codes over a network are sufficient to achieve multicast capacity, while Ref. [4] proved that randomly generated linear codes achieve capacity with high probability.

One of the main arguments against the use of RLNC is that its decoding complexity is $O(n^3)$, where $n$ constitutes the number of original packets. Sparse end-to-end erasure correcting codes, such as LT [5] and Raptor codes [6], reduce the decoding load on the receiver but at the cost of introducing additional, non-negligible delay. More importantly, they lack RLNC’s capability to use re-encode at intermediate nodes. Sparse end-to-end erasure correcting codes tend to choose sparsity in a static manner. Although this sparsity helps to reduce the complexity, it also introduces additional delay overheads. In particular, this inherent (and static) sparsity reduces the impact of a coded packet as the transmission session progresses. In fact, the additional overhead of these codes is mainly due to the transmission of the latter packets.

We revisit the idea of using sparse coding to reduce the decoding complexity, but using a fundamentally different approach in which i) we tune the level of sparsity as the transmission process evolves, and ii) we transition to RLNC for the last transmissions. This tuning process can reduce the delay overhead by using denser codes towards the end of the transmission, while maintaining complexity advantages of a sparse code. We call this scheme Tunable Sparse Network Coding. Unlike RLNC, in tunable sparse network coding the coding coefficients are chosen sparsely, i.e., mostly zero. As the session progresses, these coefficients become denser, i.e., statistical properties of the coding coefficients vary in time.

There are three key ideas in tunable sparse network coding:

- **Sparse coding is more beneficial at the beginning of a transmission session.** Since destination nodes have only received a few degrees of freedom (independent linear combination of the original packets), any sparsely coded packet will constitute a new degree of freedom (or innovative packet) with high probability.
- **Dense coding is required towards the end of the sessions for fast completion** because dense coded packets are innovative with high probability, while sparse coded packets are innovative with low probability.
- **Sparse coding translates in reduced complexity.** A decoder can exploit an encoding matrix with a large number of zero coefficients for improved complexity.

This paper proposes a framework for studying our network coding mechanism over packet erasure networks to analyze the trade-off between the delay performance (i.e., mean number of transmissions) and the decoding complexity of multicast sessions. Finally, we introduce a simple, yet powerful algorithm to organize a sparse matrix in order to provide a linear decoding complexity with high probability.

II. MODEL AND PRELIMINARIES

Consider a general multicast network with one source and several receivers. We model the network as a directed graph, with independent edge erasure probabilities. This assumption may be relaxed in future studies, but it is a fair assumption in many networks. Suppose the source node has $n$ packets to transmit to a subset of the nodes in the network, i.e., a multicast session. Suppose $X$ is an $n \times 1$ vector of source packets whose $i^{th}$ component, $X^i$, represents the $i^{th}$ source packet in the finite field $\mathbb{F}_q$. Suppose the minimum min-cut between
each receiver and source is $m$. Therefore, the network can deliver $m$ packets at each transmission round. A transmission session is complete when all destination nodes receive $n$ source packets. Throughout this paper, we assume $m << n$. Hence, multiple transmission rounds of $m$ coded packets are required in a session. Each transmission round is indexed by $t$.

Let us leverage some of the ideas from the algebraic framework for linear network coding from Ref. [3] to model our problem. For the case of no erasures, the receiver $i$ gets $A_i X$, where $A_i$ is a $m \times n$ full-rank matrix (since the min-cut rate is $m$) after each round. $A_i$ is called the network coding matrix associated with the receiver $i$ and can be decomposed into two matrices $B_i$ and $G_i$ (i.e., $A_i = G_i B_i$). $B_i$ is a $m \times n$ precoding matrix which is applied at the source node. This matrix is the same for all receivers. $G_i$ is a $m \times n$ matrix representing the effect of performing network coding at intermediate nodes from the source node to the receiver $i$.

Now, consider the case of having packet erasures over the network links. Since some packets are lost as they traverse the network, the min-cut rate of this network is smaller than the error-free min-cut rate of the network (i.e., $m$). For receiver $i$, define a $m \times m$ diagonal matrix $P_i$, whose $j^{th}$ component on the diagonal (i.e., $P_{ij}$) is zero with probability $p_j$, and one with probability $1 - p_j$. Probabilities $p_j$ depend on edge erasure probabilities. Hence, at the end of each transmission round, the destination node $i$, receives $P_i A_i X$. In other words, some of coded packets are erased over the network. Considering multiple transmission rounds, $Y(t) = P_i A_i(t) X$ represents the received coded packets at receiver $i$ in round $t$.

### III. Tunable Sparse Network Coding

In this section, we introduce a framework for studying a time-varying coding scheme, with particular emphasis on the proposed case of tunable sparse network coding. A coded packet is uniformly coded when its coding coefficients are uniformly chosen from a finite field $\mathbb{F}_q$. On the other hand, a coded packet is $k$-sparse if it contains exactly $k$ non-zero coding coefficients. Non-zero coefficients are chosen uniformly from $\mathbb{F}_q - \{0\}$. Finally, a coded packet is $k$-sparse with high probability (w.h.p.) when, with probability $1 - k/n$, each coding coefficient is zero, otherwise, it is chosen uniformly from $\mathbb{F}_q - \{0\}$. For simplicity, consider that $q$ is sufficiently large, although our techniques extend naturally to small $q$.

Define $\rho(A_i(t))$, the density of a matrix $A_i(t)$, as the ratio of its non-zero components. A matrix is dense if this ratio is high, e.g., RLNC’s $A_i(t)$ is dense w.h.p. If the ratio is low, the matrix is sparse. In a tunable sparse network coding scheme, the density of $A_i(t)$ is tuned depending on the transmission round index $t$. The key idea is that, at the beginning of the transmission session, receivers have only a few coded packets making a sparsely coded packet innovative to the receivers w.h.p. However, towards the end of the transmission session, sparsely coded packets are innovative with low probability because destination nodes have received most of the required degrees of freedom (doF). Hence, $\rho(A_i(t))$ should be increased, i.e., densely coded packets should be transmitted.

Depending on the function $\rho(A_i(t))$ and its control method, we can classify various tunable sparse network coding schemes. If the density control of $A_i(t)$ is performed without having feedback from receivers, we say the coding scheme is open-loop. Otherwise, it is a closed-loop scheme. By having different time dependencies of elements of $A_i(t)$, a variety of coding schemes, such as window-based coding, sliding window-based codings, and generation-based encoding, can be characterized and studied with our framework.

This work focuses on an open loop approach with prior knowledge of the network’s erasure probabilities. In particular, we characterize the delay and complexity performance from the perspective of a receiver and for the special case where the error-free min-cut rate is one. However, the analysis extends seamlessly to (i) multiple receivers with an error-free min-cut rate $m$ greater than one, and (ii) close-loop schemes, because our analysis relies on characterizing receivers’ performance given the number and type of received coded packets.

### A. From a Receiver’s Perspective

A tunable sparse network coding scheme with $k$-sparsely coded packets for $m = 1$ can be described as follows: the source node first transmits $n$ uncoded packets, where some of these packets are lost due to erasures. These uncoded packets are followed by sparse packets. We shall study two different approaches: i) transmission of $n_k$ $k$-sparse coded packets, i.e., with exactly $k$ non-zero coefficients chosen at random, where each non-zero coefficient is chosen uniformly from $\mathbb{F}_q - \{0\}$, which we call deterministic density, and ii) transmission of $n_k$ $k$-sparse coded w.h.p., which we call randomized density. Finally, $n_{\text{anti}}$ uniformly coded packets are transmitted.

We analyze the effect of $n_k$ on the delay performance and the decoding complexity for the general case of several levels of sparsity. Without loss of generality, we restrict our delay performance achievability result in Theorem 3 to the case of one type of sparsely coded packets, i.e., only one $k \neq 1$ for which $n_k > 0$. This result extends seamlessly to the case of different levels of sparsely coded packets.

### B. Delay Performance

This subsection characterizes the probability that a newly received, sparse coded packet is innovative under specific density distributions. This problem is key to modeling the general problem of a multicast session over a general network, where at each transmission time $t$ up to $m$ sparse packets will arrive to a receiver. In the general network case, the choice of precoding and the coding within the network as well as the link erasures will determine the sparsity of the received coded packets for that time $t$. Considering a scheme with multiple levels of sparsity, a transmission of $g(d_k|n_p)$ sparse packets for
Lemma 1. For a receiver in a multicast session with $n_p < n$ previously received degrees of freedom,

$$g_d(d_k|n_p) = \frac{n_p + d_k}{\sum_{i=n_p}^{n-1} \frac{1}{1 - (\prod_{j=0}^{k-1} \frac{i-j}{n-j})}} \quad \text{if } k < (1 - \alpha)n, \quad (1)$$

$$gr(d_k|n_p) = \frac{n_p + d_k}{\sum_{i=n_p}^{n-1} \frac{1}{1 - (1 - k/n(1 - i/n))^n}}. \quad (2)$$

Proof: First we compute $g_d(d_k|n_p)$ for the case of deterministic density and then point out the necessary changes for $gr(d_k|n_p)$. Suppose at time $t$, we have $i$ dof at the receiver. For the case of deterministic density, a $k$-sparse coded packet is not innovative with probability $\left(\prod_{j=1}^{k-1} \frac{i-j}{n-j}\right)$, which represents the probability of all the $k$ non-zero coefficients being chosen amongst packets that do not increase the knowledge space of the receiver. We model the reception of a new packet as the Markov chain shown in Figure 1. The expected number of required sparsely coded packets to have $d_k$ new dof, conditioned on having $n_p$ innovative packets at time $t$ is Eq. (1).

For the case of randomized density, a $k$-sparsely coded packet is not innovative with probability $\left(1 - k/n(1 - i/n)^n\right)$ and again model this as a Markov chain to show Eq. (2).

The following corollary provides approximate expressions for Eq. (1) and (2) for $n$ is large enough and $d_k << n - n_p$.

Corollary 2. For large $n$ and $d_k << n - n_p$. If $(n_p/n)^k << 1$ then $g_d(d_k|n_p) \approx d_k(1 + (1 - n_p/n)^k)$. If $(1 - (k/n)(1 - n_p/n))^n << 1$ then $gr(d_k|n_p) \approx d_k(1 + (1 - (k/n)(1 - n_p/n))^n)$.

Proof: Under conditions of Corollary 2, we have that

$$g_d(d_k|n_p) \approx \frac{n_p + d_k}{\sum_{n/n}^{n-1} \frac{n}{1 - x^k}} \approx \frac{n_p + d_k/n}{1 + x^k} \approx d_k(1 + (1 - n_p/n)^k).$$

We use similar approximations for $gr(d_k|n_p)$.

Define a $(n_k, n_{uni})$ tunable sparse network coding scheme as a scheme that transmits $n$ systematic packets, $n_{uni}$ uniformly coded packets, and $n_k$ $k$-sparse coded packets. This scheme is achievable if the receiver is expected to recover all $n$ packets. The following result characterizes achievable for an erasure probability $\alpha$ as seen by the receiver. For this result, define $f(d_k)$ as the number of sparse coded packets to have $d_k$ innovative ones in average. $f(d_k) - d_k$ represents the performance loss due to using sparse network coding instead of RLNC with large $q$.

Theorem 3. For a network with a single receiver and $m = 1$, a $(n_k, n_{uni})$ tunable sparse network coding scheme is achievable where $n_k = \frac{f(d_k)}{d_k}$ and $d_k + (1 - \alpha)n_{uni} = n_m$. For a deterministic density of $k < (1 - \alpha)n$, $f(d_k) = g_d(d_k)(1 - \alpha)n$. For a randomized density, $f(d_k) = gr(d_k)(1 - \alpha)n$.

Proof: Since the expected value of $n_p$ is $(1 - \alpha)n$ we compute having $f(d_k) = E[g_d(d_k|D_1)]$ and $f(d_k) = E[gr(d_k|D_1)]$ using Lemma 1. We conclude the proof following a typical argument for large $n$ since the distribution of $n_p$ is given by a binomial distribution with success probability $1 - \alpha$.

C. Complexity Analysis

We present a simple algorithm that reorders the matrix to avoid operations in the forward pass w.h.p. for the processing of sparse packets. This has the crucial effect of avoiding additional density to be created in sparse packets, which keeps the number of operations in the backward pass for the sparse packets as $O(d_k)$. Define $P_1$ as the $i$-th original packet, $C_j(Z)$ as the set of coded packets of $j$-sparse packets drawn from $Z$ a subset of the original packets, and $S_1$ as the set of packets in a 1-sparse combination. Define $Sum(F, Z)$ as the set of original packet of subset $Z$ used only once in the set of coded packets $F$. Finally, we define $\Omega$ as the set of all $n$ original packets.

Fig. 2 (a) illustrates the case of a transmission using tunable sparse network coding. For simplicity, we have assumed that received coded packets were either uncoded, i.e., from the systematic region, or sparse, although for this example $n$ is kept small for illustration purposes. Fig. 2 (a) shows that packets $P_1$ and $P_2$ were received uncoded and we use them to subtract their contribution from coded packets with a non-zero coefficient for $P_1$ and $P_2$, i.e., all non-zero elements of the first two columns. The algorithm counts the number of times a packet corresponding to columns on the lower right-hand side, as defined by the dashed lines, is involved in a linear combination. Involvement of the packet in combinations below the dashed line are not counted. Rows and columns shall be reorganized so that packets with a single occurrence will be in the upper left corner, as defined by the dashed lines in Fig. 2 (b). At this point we repeat the process until no single-occurrence packets are identified or until all sparse coded packets have been processed. In Fig. 2 (b), it is clear that the latter is attained because packet $P_3$ is involved in only one linear combination below the dashed line. The algorithm is stated in the following.

Algorithm 1

STEP 0: $R(0) = \{C_k(\Omega) : k > 1\}$ $C(0) = \{P_j : P_j \notin S_1\}$

$i = 0$. Reorder such that first rows are 1-sparse.

STEP i: If $Sum(R(i), C(i)) \neq \emptyset$

$R(i+1) = R(i) - \{C_k(Sum(R(i), C(i)) : k > 1\}$

$C(i+1) = C(i) - Sum(R(i), C(i))$

$i = i + 1$. Go to next step

Else Stop. Proceed with Standard Gaussian Elimination

Fig. 2. Example of proposed decoding algorithm (Algorithm 1)
Theorem 4 shows that the complexity of decoding $d_k$ $k$-sparsely coded packets with a randomized density, given that $n_p$ innovative packets have already been ordered as per Algorithm 1 is $O(d_k)$.

**Theorem 4.** For tunable sparse network coding with randomized density with $d_k$ $k$-sparsely random packets when $n_p$ degrees of freedom have been processed before with Algorithm 1, and, the average number of operations to process the $d_k$ sparse packets using Algorithm 1 with probability $\prod_{i=0}^{d_k-1} \left(1 - \left(1 - \frac{k}{n} \right)^{d_k-1-i} \right)^{n_p-i}$ is $c(n_p, d_k) = k d_k$

**Proof:** First, the probability of a packet being used only once in $j$ independent linear $k$-sparsely coded packets is $P_j = (1 - k/n)^{j-1}$. Second, we process on a row by row basis such that there is a decreasing number of linear combinations from the original $d_k$. For the $r$-th processed row, the probability of having at least one packet being used only once is given by $\mathcal{P}(r) = 1 - \left(1 - \frac{k}{n} \right)^{d_k-1-r}$. If all $d_k$ packets can be processed exclusively with the reordering algorithm, only $O(d_k)$ operations for reordering during the forward pass and $O(d_k)$ operations for the backward elimination pass are required. The probability of this event is $\prod_{i=0}^{d_k-1} \mathcal{P}(i)$.

**IV. PERFORMANCE EVALUATION**

This section illustrates the trade-off between complexity and delay performance via numerical examples and show that the threshold to switch from sparse to dense codes is key in determining this trade-off. For simplicity, we assume that no packets are sent uncoded at the beginning of the transmission. We compare the following schemes:

- **3-sparse Network Coding:** This scheme considers the transmission of 3-sparse packets only. For determining the delay performance, we use Eq. (1) and use Algorithm 1 to determine complexity.

- **RLNC:** This constitutes a classical RLNC scheme with cubic complexity. In particular, we consider that $GE(n) = n^3 / 3 + n^2 / 2 - 5n/6$, which is a usual estimate of operations necessary to perform Gaussian elimination. For RLNC we assume that every incoming packet is innovative, which is a good approximation when using large field sizes.

- **Tunable Sparse Network Coding Scheme:** This scheme relies on both a sparse and a dense region, with a tunable threshold to decide when to switch from one to the other. Packets generated in the sparse region are $k$-sparse, with $k$ being incremented every time $n/100$ coded packets sent.

Fig. 3 shows the performance of the three schemes as more packets are coming into the system. The right and top-most point for each curve constitutes the final pair of values for delay and complexity. Fig. 3 shows our proposed scheme for different threshold levels, namely when $1, 2, 5, 10, 50$ and $75\%$ of the degrees of freedom are missing at the receiver. We observe that i) switching too fast to RLNC is counterproductive from a complexity perspective, and ii) gains in terms of complexity become important when the switch to RLNC happens for the very last packets (See case of $1\%$). Finally, the case of switching for the last $1\%$ helps us confirm our initial intuition, namely that a sparse code’s delay performance is primarily degraded when few degrees of freedom are missing.

**V. CONCLUSIONS**

We revisit the problem of trading off delay performance and decoding complexity but providing a novel framework that uses both sparse and dense coded packets and allows coding at intermediate nodes for multicast sessions in general networks. More generally, we propose that the sparsity level should evolve as the transmission session progresses. To enable this coding flexibility, we proposed an efficient and intuitive algorithm for reordering rows and columns of a matrix in order to perform efficient Gaussian elimination in sparse matrices. This algorithm guarantees linear decoding complexity of sparse packets w. h. p.

Our approach can benefit considerably from the use of feedback in order to change the density of the code and studying efficient feedback mechanisms will be the focus of future work. Future work will also consider efficient and practical pre-coding mechanisms at the source for the case of single and multi source multicast to deal with specific network topologies and practical coding policies at intermediate nodes.

**VI. ACKNOWLEDGEMENTS**

This work was supported in part by the Air Force Office of Scientific Research (AFOSR) under award No. FA9550-08-1-0159 and by FCT project PEst-OE/EEI/LA0008/2011.

**REFERENCES**


