

# Transversal GRAND for Network Coded Data

Ioannis Chatzigeorgiou

School of Computing and Communications

Lancaster University, United Kingdom

Email: i.chatzigeorgiou@lancaster.ac.uk

**Abstract**—This paper considers a transmitter, which uses random linear coding (RLC) to encode data packets. The generated coded packets are broadcast to one or more receivers. A receiver can recover the data packets if it gathers a sufficient number of coded packets. We assume that the receiver does not abandon its efforts to recover the data packets if RLC decoding has been unsuccessful; instead, it employs syndrome decoding in an effort to repair erroneously received coded packets before it attempts RLC decoding again. A key assumption of most decoding techniques, including syndrome decoding, is that errors are independently and identically distributed within the received coded packets. Motivated by the ‘guessing random additive noise decoding’ (GRAND) framework, we develop transversal GRAND: an algorithm that exploits statistical dependence in the occurrence of errors, complements RLC decoding and achieves a gain over syndrome decoding, in terms of the probability that the receiver will recover the original data packets.

**Index Terms**—Network coding, fountain coding, random linear coding, GRAND, syndrome decoding, error burst.

## I. INTRODUCTION

Random linear coding (RLC) [1], which encompasses fountain coding and network coding, is a forward error correction scheme that has the potential to increase throughput and improve resilience to network attacks. It was originally developed for layers higher than the physical layer, e.g., the network layer and the application layer. RLC encoding generates coded packets that are random linear combinations of input data packets. RLC decoding discards received coded packets that have been corrupted by errors and attempts to reconstruct the original data packets from correctly received coded packets.

When RLC decoding is unsuccessful, discarded corrupted packets can be utilized. Syndrome decoding [2], which focuses on binary transmission, aims to repair corrupted packets in polynomial time. Although syndrome decoding can improve the chances of the RLC decoder recovering the data packets, it does not account for correlations between errors. To address this deficiency, we look into the recently proposed ‘guessing random additive noise decoding’ (GRAND) scheme.

GRAND is a code-agnostic decoding scheme that endeavors to identify and nullify the effect of noise on transmitted signals. The theoretical foundations and properties of GRAND were laid in [3] and [4]. A practical implementation of the GRAND framework, referred to as GRAND Markov order (GRAND-MO), was presented in [5]. GRAND-MO is a hard detection decoder of binary codewords that have been impaired by errors correlated over time.

GRAND-MO focuses on physical-layer coding, according to which  $N$ -bit codewords are generated by  $K$ -bit input words,

where  $N \geq K$ , and are transmitted over a noisy channel. As a result, correlated errors are distributed over the  $N$  bits of each codeword. On the other hand, an RLC encoder organizes input bits into  $K$  packets and generates  $N$  coded packets, which are random linear combinations of the  $K$  packets. In essence, a bit in position  $b$  of a coded packet is obtained by the randomly weighted modulo-2 addition of the bits that occupy position  $b$  in all of the  $K$  input packets. Therefore, the bits in position  $b$  of the  $N$  coded packets create an  $N$ -bit codeword. Equivalently, if the coded packets are stacked up and form the rows of a matrix, the codewords will occupy the columns of this matrix. Correlated errors introduced by the channel will be present in each coded packet. This means that errors across adjacent codewords will be correlated but errors within each codeword will be uncorrelated. The objective of this paper is to build on the concept of GRAND and propose *transversal GRAND*, an algorithm that exploits correlations between errors across multiple codewords – and is thus suitable for assisting RLC decoding – in contrast to GRAND-MO, which leverages correlations between errors within codewords.

The remainder of this paper has been organized as follows: Section II briefly describes RLC decoding, which discards packets received in error. Section III explains how syndrome decoding can repair packets in error and assist RLC decoding. Transversal GRAND is then introduced as a means to exploit the statistical structure of errors and improve the chances of repairing packets. Performance comparisons between syndrome decoding and transversal GRAND are presented in Section IV and key findings are summarized in Section V.

## II. RANDOM LINEAR CODING AND DECODING

We consider a source node that intends to broadcast messages to one or more destination nodes. Before initiating transmission, the source node segments each message into  $K$  packets. Each source packet has been modeled as a sequence of  $B$  bits. The  $K$  source packets of  $B$  bits can be expressed as a matrix  $\mathbf{U} \in \mathbb{F}_2^{K \times B}$ , where  $\mathbb{F}_2^{K \times B}$  denotes the set of all  $K \times B$  matrices over  $\mathbb{F}_2 = \{0, 1\}$ . RLC [1] is used to encode the  $K$  source packets into  $N \geq K$  coded packets. The  $N$  coded packets can also be expressed in matrix form as  $\mathbf{X} \in \mathbb{F}_2^{N \times B}$ . The relationship between matrices  $\mathbf{X}$  and  $\mathbf{U}$  is:

$$\mathbf{X} = \mathbf{G}\mathbf{U}, \quad (1)$$

where  $\mathbf{G}$  is known as the *generator matrix*. We employ *systematic* RLC encoding [6], according to which the first  $K$  of the  $N$  transmitted packets are identical to the  $K$  source

packets, while the remaining  $N - K$  coded packets are random linear combinations of the source packets. In systematic RLC, the generator matrix  $\mathbf{G}$  can be expressed in standard form:

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_K \\ \mathbf{P} \end{bmatrix}, \quad (2)$$

where  $\mathbf{I}_K$  is the  $K \times K$  identity matrix. Each element of the  $(N - K) \times K$  matrix  $\mathbf{P}$  is chosen uniformly and at random from  $\mathbb{F}_2$ . As explained in [2], a seed could be used to initialize the pseudo-random number generator that outputs the elements of  $\mathbf{P}$ . Given that the value of the seed can be conveyed in the headers of the coded packets, we assume that the destination nodes have knowledge of  $\mathbf{G}$ .

Let  $\mathbf{Y}$  be an erroneous copy of  $\mathbf{X}$  that has been received by a destination node, defined as:

$$\mathbf{Y} = \mathbf{X} \oplus \mathbf{E}, \quad (3)$$

where  $\oplus$  represents modulo-2 addition. The *error matrix*  $\mathbf{E}$  contains ones in positions where errors have occurred and zeros in the remaining positions. The destination node classifies the received coded packets into error-free and erroneous coded packets, e.g., using cyclic redundancy checks (CRCs). Let  $\mathcal{R}$  be the set of the row indices of  $\mathbf{Y}$  that correspond to correctly received coded packets, and  $N_{\mathcal{R}}$  be the number of coded packets that contain no errors, i.e.,  $|\mathcal{R}| = N_{\mathcal{R}} \leq N$ . The destination node constructs matrix  $\mathbf{Y}_{\mathcal{R}}$  from the  $N_{\mathcal{R}}$  correctly received coded packets and matrix  $\mathbf{G}_{\mathcal{R}}$  from those  $N_{\mathcal{R}}$  rows of  $\mathbf{G}$  with indices in  $\mathcal{R}$ . On the other hand, the  $N - N_{\mathcal{R}}$  erroneously received coded packets have indices in the set  $\bar{\mathcal{R}} = \{1, \dots, N\} \setminus \mathcal{R}$  and make the rows of matrix  $\mathbf{Y}_{\bar{\mathcal{R}}}$ . Using the indices in  $\bar{\mathcal{R}}$ , matrices  $\mathbf{X}_{\bar{\mathcal{R}}}$ ,  $\mathbf{G}_{\bar{\mathcal{R}}}$  and  $\mathbf{E}_{\bar{\mathcal{R}}}$  are obtained. The relationships between these matrices are summarized below:

$$\mathbf{Y} = \mathbf{X} \oplus \mathbf{E} \Leftrightarrow \begin{cases} \mathbf{Y}_{\mathcal{R}} = \mathbf{X}_{\mathcal{R}} = \mathbf{G}_{\mathcal{R}}\mathbf{U}, & (4a) \\ \mathbf{Y}_{\bar{\mathcal{R}}} = \mathbf{X}_{\bar{\mathcal{R}}} \oplus \mathbf{E}_{\bar{\mathcal{R}}}. & (4b) \end{cases}$$

RLC decoding utilizes only the  $N_{\mathcal{R}}$  correctly received coded packets represented by  $\mathbf{Y}_{\mathcal{R}}$ . The  $N - N_{\mathcal{R}}$  erroneous coded packets contained in  $\mathbf{Y}_{\bar{\mathcal{R}}}$  are discarded. The source message, represented by matrix  $\mathbf{U}$ , can be obtained from  $\mathbf{X}_{\mathcal{R}} = \mathbf{G}_{\mathcal{R}}\mathbf{U}$  if the rank of  $\mathbf{G}_{\mathcal{R}}$  is  $K$ . In that case, the relationship  $\mathbf{X}_{\mathcal{R}} = \mathbf{G}_{\mathcal{R}}\mathbf{U}$  can be seen as a system of  $N_{\mathcal{R}} \geq K$  linear equations, which can be reduced to a system of  $K$  linearly independent equations with  $K$  unknowns, i.e., source packets. This  $K \times K$  system can be solved, e.g., using Gaussian elimination, and return a unique solution for  $\mathbf{U}$ . Otherwise, if  $\text{rank}(\mathbf{G}_{\mathcal{R}}) < K$ , the linear system has fewer linearly independent equations than unknowns, hence a unique solution cannot be obtained.

If the decoder does not discard the erroneous coded packets but attempts to repair them, the probability of recovering the source message can potentially be increased. The following section describes syndrome decoding [2], which aims to repair coded packets that contain errors while operating under the assumption that transmission is over a binary symmetric channel (BSC). We then propose transversal GRAND, which takes into consideration the statistical properties of the channel and increases the chances of recovering the source message.

### III. GUESSING RANDOM ADDITIVE NOISE

Guessing random additive noise is the process of calculating an estimate of  $\mathbf{E}_{\bar{\mathcal{R}}}$ , denoted by  $\hat{\mathbf{E}}_{\bar{\mathcal{R}}}$ . An estimate of the transmitted coded packets can then be derived from (4b):

$$\hat{\mathbf{X}}_{\bar{\mathcal{R}}} = \mathbf{Y}_{\bar{\mathcal{R}}} \oplus \hat{\mathbf{E}}_{\bar{\mathcal{R}}}. \quad (5)$$

To determine which of the estimated rows of  $\hat{\mathbf{X}}_{\bar{\mathcal{R}}}$  correspond to successfully repaired coded packets, CRC verification is carried out. Let  $\nu$  denote the number of rows in  $\hat{\mathbf{X}}_{\bar{\mathcal{R}}}$  that passed CRC verification. The indices of the  $\nu$  repaired coded packets are removed from set  $\bar{\mathcal{R}}$  and are added to set  $\mathcal{R}$ , while the corresponding rows of  $\hat{\mathbf{X}}_{\bar{\mathcal{R}}}$  are moved to  $\mathbf{Y}_{\mathcal{R}}$  in (4a). The cardinalities of sets  $\bar{\mathcal{R}}$  and  $\mathcal{R}$  change to  $N - N_{\mathcal{R}} - \nu$  and  $N_{\mathcal{R}} + \nu$ , respectively, while the dimensions of  $\hat{\mathbf{X}}_{\bar{\mathcal{R}}}$  and  $\mathbf{Y}_{\mathcal{R}}$  change to  $(N - N_{\mathcal{R}} - \nu) \times B$  and  $(N_{\mathcal{R}} + \nu) \times B$ , respectively. The indices of the  $\nu$  repaired packets are also used to identify the rows of the generator matrix  $\mathbf{G}$  that should be appended to  $\mathbf{G}_{\mathcal{R}}$  in (4a). If the  $\nu$  repaired packets increase the rank of the enlarged  $(N_{\mathcal{R}} + \nu) \times K$  matrix  $\mathbf{G}_{\mathcal{R}}$  to  $K$ , the process of guessing random additive noise will have been successful in assisting the RLC decoder to recover the  $K$  source packets.

This section briefly describes syndrome decoding [2], which is an established method for computing  $\hat{\mathbf{E}}_{\bar{\mathcal{R}}}$ . We explain the shortcomings of syndrome decoding when the errors in each coded packet are correlated, and we propose transversal GRAND, which builds on the concept of GRAND and exploits error correlations that can lead to error bursts.

#### A. Syndrome Decoding

Given that the  $N \times K$  generator matrix  $\mathbf{G}$  is known to all destination nodes, the  $N \times (N - K)$  *parity-check matrix*  $\mathbf{H}$  can be derived as follows:

$$\mathbf{H} = \left[ -\mathbf{P} \mid \mathbf{I}_{N-K} \right]^T, \quad (6)$$

so that:

$$\mathbf{H}^T \mathbf{G} = \mathbf{0}. \quad (7)$$

When operations are performed in  $\mathbb{F}_2$ , negation has no effect on a matrix, i.e.,  $-\mathbf{P} = \mathbf{P}$ . Note that  $\mathbf{H}$  can be obtained from  $\mathbf{G}$  even when RLC is non-systematic, as long as  $\mathbf{G}$  has rank  $K$ ; in that case, we can apply column-wise Gaussian elimination to transform  $\mathbf{G}$  into standard form, as in (2), identify  $\mathbf{P}$  and obtain  $\mathbf{H}$  using (6).

Multiplication of  $\mathbf{H}^T$  by the received matrix  $\mathbf{Y}$  produces the  $(N - K) \times B$  *syndrome matrix*  $\mathbf{S}$ , i.e.,  $\mathbf{S} = \mathbf{H}^T \mathbf{Y}$ . Using (7), we find that the relationship between the syndrome matrix  $\mathbf{S}$  and the error matrix  $\mathbf{E}$  is:

$$\mathbf{S} = \mathbf{H}^T \mathbf{Y} = \mathbf{H}^T (\mathbf{X} \oplus \mathbf{E}) = \mathbf{H}^T (\mathbf{G}\mathbf{U} \oplus \mathbf{E}) = \mathbf{H}^T \mathbf{E}. \quad (8)$$

As our focus is on the  $N - N_{\mathcal{R}}$  erroneous coded packets, we use the set  $\bar{\mathcal{R}}$  to isolate the  $N - N_{\mathcal{R}}$  of the  $N$  rows of  $\mathbf{H}$  and  $\mathbf{E}$  in (8), and construct  $\mathbf{H}_{\bar{\mathcal{R}}}$  and  $\mathbf{E}_{\bar{\mathcal{R}}}$ , respectively. Consequently, expression (8) changes to:

$$\mathbf{S} = (\mathbf{H}_{\bar{\mathcal{R}}})^T \mathbf{E}_{\bar{\mathcal{R}}}. \quad (9)$$

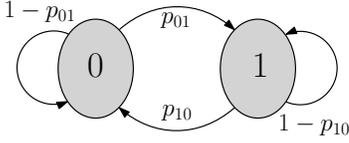


Fig. 1. The two-state Markov model, where 0 and 1 represent the ‘good’ state and the ‘bad’ state, respectively.

If column  $b$  of  $\mathbf{S}$  and  $\mathbf{E}_{\bar{\mathcal{R}}}$  is denoted by  $[\mathbf{S}]_{*,b}$  and  $[\mathbf{E}_{\bar{\mathcal{R}}}]_{*,b}$ , respectively, expression (9) can be re-written as  $B$  independent systems of  $N - K$  linear equations with  $N - N_{\mathcal{R}}$  unknowns per equation:

$$[\mathbf{S}]_{*,b} = (\mathbf{H}_{\bar{\mathcal{R}}})^{\top} [\mathbf{E}_{\bar{\mathcal{R}}}]_{*,b} \quad \text{for } b = 1, \dots, B. \quad (10)$$

Mohammadi *et al.* [2] observed that erroneously received coded packets usually contain large error-free segments, thus  $\mathbf{E}_{\bar{\mathcal{R}}}$  is a *sparse* matrix, that is, most of the elements in  $\mathbf{E}_{\bar{\mathcal{R}}}$  are zero-valued. Based on this observation, the solution to (10) can be formulated as:

$$[\hat{\mathbf{E}}_{\bar{\mathcal{R}}}]_{*,b} = \arg \min_{\mathbf{w}^{\top}} \|\mathbf{w}\|_0 \quad (11a)$$

$$\text{subject to } (\mathbf{H}_{\bar{\mathcal{R}}})^{\top} \mathbf{w}^{\top} = [\mathbf{S}]_{*,b} \quad (11b)$$

where  $\mathbf{w} \in \mathbb{F}_2^{N-N_{\mathcal{R}}}$  is a row vector that should satisfy constraint (11b) and have the minimum possible number of non-zero elements. The norm  $\|\mathbf{w}\|_0$ , which counts the non-zero elements in  $\mathbf{w}$ , is defined as  $\|\mathbf{w}\|_0 = |w_1|^0 + \dots + |w_{N-N_{\mathcal{R}}}|^0$  assuming that  $0^0 = 0$  [7]. Syndrome decoding considers (11a) and initiates an exhaustive search for a solution; the sparsity of the row vector  $\mathbf{w}$  is gradually reduced, i.e., the number of ones in  $\mathbf{w}$  increases, and the search concludes when the sparsest vector  $\mathbf{w}$  that satisfies (11b) has been identified.

Although syndrome decoding has the potential to assist RLC decoding and increase the probability of recovering the source packets [2], it does not exploit the statistical properties of the channel. It selects columns for  $\hat{\mathbf{E}}_{\bar{\mathcal{R}}}$  that are as sparse as possible or, equivalently, have the lowest possible Hamming weight, but it does not account for the possibility that the columns of  $\hat{\mathbf{E}}_{\bar{\mathcal{R}}}$  might be correlated. Thus, syndrome decoding is suitable for memoryless channels, like the BSC, although channels often have memory in practice. The two-state Markov model [8] is a widely used channel model for the characterization of the correlation between consecutive errors.

### B. Two-state Markov Model for Error Bursts

As in [5], we use a two-state Markov chain to model error bursts. According to this model, a bit is received in error if a transition to the ‘bad’ state of the chain, denoted by 1, occurs. Otherwise, a transition to the ‘good’ state, denoted by 0, implies that the bit has been received correctly [8]. Transitions between the two states occur with probabilities  $p_{01}$ ,  $1 - p_{01}$ ,  $p_{10}$  and  $1 - p_{10}$ , as shown in Fig. 1. The steady-state probability of being in state 1 represents the bit error probability and is given by  $\varepsilon = p_{01}/(p_{01} + p_{10})$  [9], [10]. The expected number of consecutive errors provides the average length of an error burst, and is equal to  $\Lambda = 1/p_{10}$  [10].

### C. Transversal GRAND

In syndrome decoding, column vectors that have the same number of non-zero elements, i.e., the same Hamming weight, are treated as being equally likely and can thus be queried in any order by (11). Furthermore, all column vectors of a given Hamming weight  $w$  are queried before column vectors of weight  $w + 1$ . The proposed method, which we call transversal GRAND, queries column vectors in descending order of likelihood, as dictated by the transition probabilities of the Markov model and not by the Hamming weights of the vectors.

The transversal GRAND algorithm is illustrated in Fig. 2 by means of an example, in which column  $b$  of  $\hat{\mathbf{E}}_{\bar{\mathcal{R}}}$ , for  $b = 1, \dots, B$ , is estimated based on column  $b - 1$  of  $\hat{\mathbf{E}}_{\bar{\mathcal{R}}}$  and the transition probabilities  $p_{01}$  and  $p_{10}$  of the Markov model. State 0 is the initial state of the Markov chain, therefore the all-zero column is input to the algorithm for the estimation of the first column (i.e.,  $b = 1$ ) of  $\hat{\mathbf{E}}_{\bar{\mathcal{R}}}$ . The length of the input and output column vectors is  $L = N - N_{\mathcal{R}}$ .

At first, the algorithm identifies the positions of the elements of the input vector that are equal to 0 or 1. Let  $L_0$  and  $L_1$  be the number of zeros and ones, respectively. The elements of the input vector do not need to be arranged based on their values, as long as the algorithm keeps track of the positions of each value. However, we opted to arrange the elements such that the  $L_0$  zeros are grouped at the top of the vector and the  $L_1$  ones are grouped at the bottom of the vector, in order to facilitate the description of the algorithm. As shown in Fig. 2, the  $L_0 = 2$  zeros of the input vector are moved from positions 2 and 5 to positions 1 and 2, while the  $L_1 = 3$  ones are moved to the three bottom positions. Essentially, the input vector contains the current states of  $L$  independent Markov chains;  $L_0$  of them are in state 0 (the ‘good’ state) and the remaining  $L_1$  chains are in state 1 (the ‘bad’ state).

According to the Markov model, the values of  $\ell_i$  of the  $L_i$  elements will change from  $i$  to  $j$  with probability  $p_{ij}$ , for  $i, j \in \{0, 1\}$ ,  $i \neq j$  and  $\ell_i = 0, \dots, L_i$ . The values of the remaining  $L_i - \ell_i$  elements will stay the same with probability  $1 - p_{ij}$ . The overall probability of occurrence of the aforementioned transitions is:

$$p_{01}^{\ell_0} (1 - p_{01})^{L_0 - \ell_0} p_{10}^{\ell_1} (1 - p_{10})^{L_1 - \ell_1}.$$

The probability of occurrence is calculated for  $\ell_0 = 0, \dots, L_0$  and  $\ell_1 = 0, \dots, L_1$  resulting in  $(L_0 + 1)(L_1 + 1)$  probability values, which are sorted in descending order. As can be seen in Fig. 2, for  $L_0 = 2$  and  $L_1 = 3$ , twelve probability values are computed and ordered. For  $p_{01} = 0.2$  and  $p_{10} = 0.7$ , the input vector transitioning into the all-zero vector is the most likely event to occur, i.e., the event of the two zeros of the input vector remaining unchanged and the three ones changing to 0 with overall probability 0.2195.

For each value of the probability of occurrence, which has been obtained for a specific value of  $\ell_0$  and  $\ell_1$ , the algorithm generates a sequence of  $\binom{L_0}{\ell_0} \binom{L_1}{\ell_1}$  equally likely vectors. For

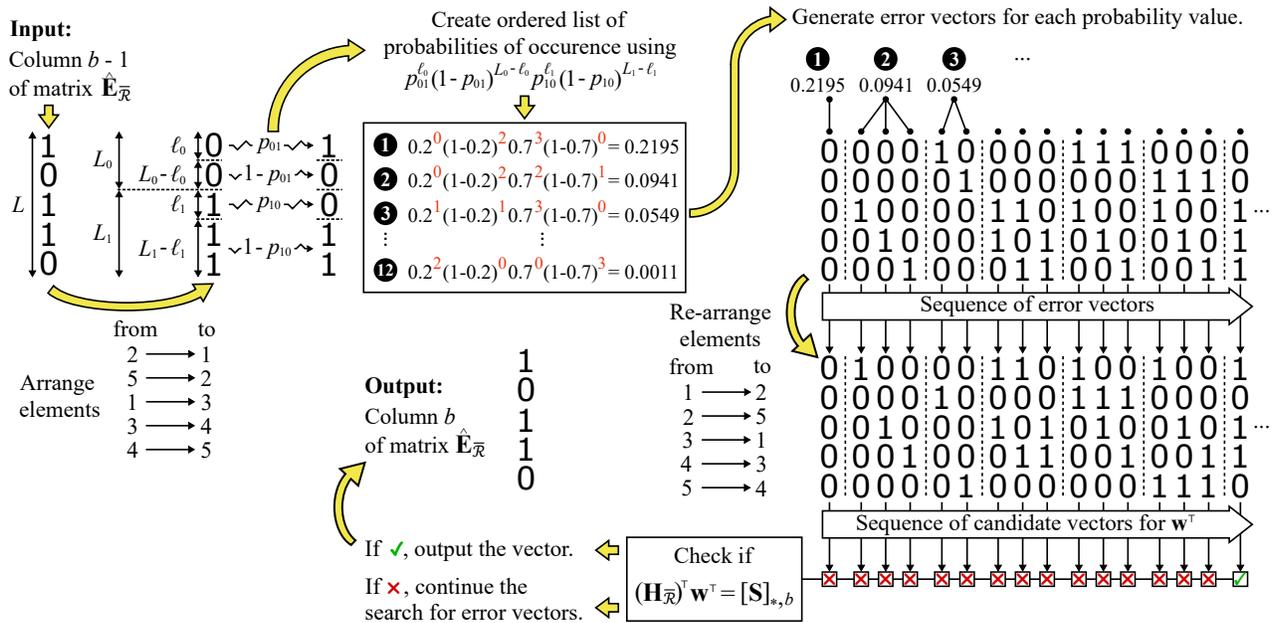


Fig. 2. Pictorial representation of transversal GRAND by means of an example, in which all column vectors have length  $L = 5$ , the input vector contains  $L_0 = 2$  zeros and  $L_1 = 3$  ones, and the transition probabilities are  $p_{01} = 0.2$  and  $p_{10} = 0.7$ .

example, the ordered list of probabilities of occurrence in Fig. 2 shows that the probability of  $\ell_0 = 0$  zeros changing to 1 and  $\ell_1 = 2$  ones changing to 0 is 0.0941. The input vector, which contains  $L_0 = 2$  zeros and  $L_1 = 3$  ones, could thus transition to one of  $\binom{2}{0} \binom{3}{2} = 3$  vectors, seen in the top-right quarter of Fig. 2, with probability 0.0941. As expected, the  $(L_0 + 1)(L_1 + 1)$  probability values are mapped to a total of

$$\sum_{\ell_0=0}^{L_0} \sum_{\ell_1=0}^{L_1} \binom{L_0}{\ell_0} \binom{L_1}{\ell_1} = \left[ \sum_{\ell_0=0}^{L_0} \binom{L_0}{\ell_0} \right] \left[ \sum_{\ell_1=0}^{L_1} \binom{L_1}{\ell_1} \right] = 2^{L_0+L_1} = 2^L$$

vectors. Observe in Fig. 2 that transversal GRAND, in contrast to syndrome decoding, does not assign by default the same probability of occurrence to vectors that have the same Hamming weight. Furthermore, the vectors that are generated by transversal GRAND are not necessarily of increasing Hamming weight. For example, the generated weight-1 vectors in Fig. 2 are followed by nine of the ten possible weight-2 vectors because one of the weight-3 vectors, that is,  $[00111]^T$ , is more likely to occur than the remaining weight-2 vector.

If the original positions of the elements of the input vector changed in the first stage of the algorithm, the elements of the generated vectors should also be re-arranged, so that they occupy their original positions. For each vector of re-arranged elements, denoted by  $\mathbf{w}^T$ , the algorithm checks whether  $\mathbf{w}^T$  is a solution to the linear equation  $(\mathbf{H}_{\mathcal{R}})^T \mathbf{w}^T = [\mathbf{S}]_{*,b}$ . As also explained in Fig. 2, if  $\mathbf{w}^T$  is indeed a solution, then it is selected to be column  $b$  of  $\hat{\mathbf{E}}_{\mathcal{R}}$ . Otherwise, the algorithm continues to generate and query vectors in descending order of likelihood, until a solution is found. The algorithm terminates when solutions for all  $B$  linear equations, shown in (10), have been estimated and  $\hat{\mathbf{E}}_{\mathcal{R}}$  has been obtained.

The following section compares syndrome decoding and transversal GRAND for different channel conditions. Note that when  $p_{01} + p_{10} = 1$ , the Markov chain reduces to the BSC model, and transversal GRAND becomes equivalent to syndrome decoding.

#### IV. RESULTS AND DISCUSSION

In this section, the bit error probability,  $\varepsilon$ , and the average length of an error burst,  $\Lambda$ , are used as the channel parameters. For this reason, the transition probabilities,  $p_{10}$  and  $p_{01}$ , have been expressed in terms of  $\varepsilon$  and  $\Lambda$  based on the definitions in Section III-B, i.e.,  $p_{10} = 1/\Lambda$  and  $p_{01} = \varepsilon/[\Lambda(1-\varepsilon)]$ , and have been used as inputs to the transversal GRAND algorithm. The objective of this section is to compare the performance of stand-alone RLC decoding, RLC decoding combined with syndrome decoding (RLC with SD) and RLC decoding combined with transversal GRAND (RLC with T-GRAND) in terms of their *decoding probability*, that is, the probability of a destination node recovering the  $K$  source packets when  $N$  coded packets have been transmitted. Measurements of the decoding probability, obtained through simulations for  $K = 10$  and  $N$  varying from 10 to 20, are presented in Fig. 3.

Fig. 3a shows measurements for an increasing probability of the channel flipping bits in transmitted packets, when each packet contains  $B = 64$  bits and errors occur in bursts of length  $\Lambda = 4$  on average. For  $\varepsilon = 0.01$  and a growing number of transmitted packets, the destination node is likely to receive  $K$  error-free linearly independent packets among the received packets and recover the  $K$  source packets using RLC decoding; SD and T-GRAND offer only a marginal improvement in decoding probability. If the value of  $\varepsilon$  is raised to 0.05, the proportion of erroneously received packets increases, the RLC

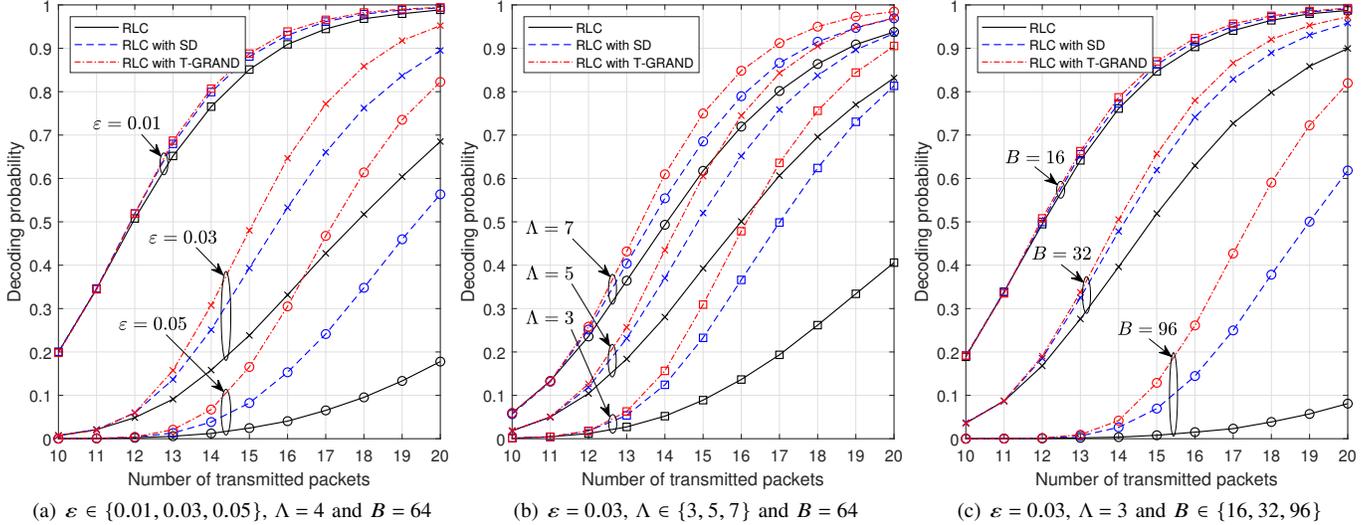


Fig. 3. Comparison of the decoding probabilities of RLC decoding (RLC), RLC decoding combined with syndrome decoding (RLC with SD) and RLC decoding combined with transversal GRAND (RLC with T-GRAND) for  $K = 10$  source packets and  $N$  transmitted coded packets, where  $N = 10, \dots, 20$ . System configurations for different values of bit error probability ( $\varepsilon$ ), average length of error bursts ( $\Lambda$ ) and packet length ( $B$ ) have been considered.

decoder experiences a notable drop in decoding performance, and SD and T-GRAND become instrumental in improving the chances of recovering the source packets. For example, when  $\varepsilon = 0.05$  and  $N = 20$ , SD increases the decoding probability of RLC from 0.18 to 0.56 whereas T-GRAND, which considers correlations in errors when repairing coded packets, boosts the decoding probability to 0.82.

Fig. 3b depicts the impact of the average length of error bursts on the decoding probability, when  $\varepsilon = 0.03$ . When the value of  $\varepsilon$  is constant, the same number of errors – on average – impair the transmitted packets, but errors aggregate in fewer packets as the average length of error bursts increases. Consequently, destination nodes receive an increasingly larger proportion of error-free packets as  $\Lambda$  grows, and RLC decoding stands a greater chance of success. Although fewer packets are received in error for higher values of  $\Lambda$ , erroneous packets are more severely damaged by longer error bursts. Nevertheless, SD and especially T-GRAND can still improve the decoding probability of RLC. For example, the decoding probability of RLC, RLC with SD and RLC with T-GRAND is 0.72, 0.79 and 0.85, respectively, for  $\Lambda = 7$  and  $N = 16$ . On the other hand, low values of  $\Lambda$  have a negative impact on RLC decoding; for a decreasing length of error bursts and a fixed average number of errors, the proportion of correctly received packets reduces. Packets corrupted by errors may dominate but errors are more sparsely distributed, and both SD and T-GRAND are more effective in repairing corrupted packets. Notice in Fig. 3b that the decoding probability of RLC, RLC with SD and RLC with T-GRAND is 0.41, 0.81 and 0.91, respectively, for  $\Lambda = 3$  and  $N = 20$ .

The effect of the packet length  $B$  on the decoding probability is shown in Fig. 3c. For fixed values of  $\varepsilon$  and  $\Lambda$ , the likelihood that a packet will be corrupted by errors increases with the packet length and, thus, the performance of RLC

decoding deteriorates. When it comes to SD and T-GRAND, the value of  $B$  also signifies the number of times that the algorithms have run in order to estimate the errors in the  $B$  positions of the corrupted packets. The advantage of T-GRAND over SD becomes apparent as the packet length increases. For  $B = 96$  and  $N = 20$ , SD lifts the decoding probability of RLC from 0.08 to 0.62; T-GRAND further increases the decoding probability to 0.82 as is more successful than SD in making consecutive accurate guesses of error occurrences in the course of the  $B = 96$  runs.

## V. CONCLUSIONS

This paper builds on the ‘guessing random additive noise decoding’ (GRAND) framework and develops transversal GRAND, a hard detection decoder that leverages correlations between channel errors and works in tandem with the decoder of a random linear code (RLC). Transversal GRAND generates and queries error vectors in order of likelihood and attempts to correct bits that occupy the same position in all of the erroneously received packets. After all bit positions have been considered, RLC decoding utilizes both repaired and correctly received packets to recover the original information packets. Simulation results demonstrated that transversal GRAND has the potential to markedly improve the decoding performance of RLC when the communication channel introduces error bursts in the transmitted packets.

## REFERENCES

- [1] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.
- [2] M. S. Mohammadi, Q. Zhang, and E. Dutkiewicz, “Reading damaged scripts: Partial packet recovery based on compressive sensing for efficient random linear coded transmission,” *IEEE Trans. Commun.*, vol. 64, no. 8, pp. 3296–3310, Aug. 2016.

- [3] K. R. Duffy, J. Li, , and M. Médard, “Guessing noise, not code-words,” in *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 2018.
- [4] —, “Capacity-achieving guessing random additive noise decoding,” *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4023–4040, Jul. 2019.
- [5] W. Ann, M. Médard, and K. R. Duffy, “Keep the bursts and ditch the interleavers,” in *Proc. IEEE Conf. on Global Telecomm. (GLOBECOM)*, Taipei, Taiwan, Dec. 2020.
- [6] A. L. Jones, I. Chatzigeorgiou, and A. Tassi, “Binary systematic network coding for progressive packet decoding,” in *Proc. IEEE Int. Conf. on Commun. (ICC)*, London, UK, Jun. 2015.
- [7] D. L. Donoho, “Sparse components of images and optimal atomic decompositions,” *Constr. Approx.*, vol. 17, no. 1, pp. 353–382, Jan. 2001.
- [8] E. N. Gilbert, “Capacity of a burst-noise channel,” *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1266, Sep. 1960.
- [9] T. Rancurel, D. Roviras, and F. Castanié, “Expression of the capacity for the Gilbert channel in presence of interleaving,” in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Salt Lake City, USA, May 2001, pp. 2533–2536.
- [10] J. Cloud and M. Médard, “Network coding over SATCOM: Lessons learned,” in *Proc. Int. Conf. on Wireless and Satellite Systems (WiSATS)*, Bradford, UK, Jul. 2015, pp. 272–285.