A Lower Bound on the Success Probability of Binary Random Linear Network Codes aided by Noise Decoding

Ioannis Chatzigeorgiou, *Senior Member, IEEE*

***Abstract*—The decoding of binary random linear codes for packet erasures relies on receivers collecting a sufficient number of error-free coded packets for the reconstruction of a transmitted message. Syndrome decoding (SD), which conforms to the broad definition of guessing random additive noise decoding (GRAND), endeavors to repair partially-correct received coded packets – which would otherwise be discarded by the packet erasure decoder – and boost the probability of a receiver successfully recovering the transmitted message. This paper introduces a low-complexity variant of SD, dubbed single-error correcting SD (SEC-SD), and conducts an in-depth analysis of the success probability of packet erasure decoding aided by SEC-SD. The analysis leads to a practical lower bound on the success probability of packet erasure decoding complemented by SD, and provides guidance on how SEC-SD can be modified to potentially achieve a higher success probability than SD at a lower computational cost.**

***Index Terms*—Network coding, binary codes, linear codes, decoding, error analysis.**

I. Introduction

RANDOM linear network coding (RLNC) [1] and fountain coding [2] fall in the general category of random linear packet erasure coding, whereby a message composed of k source packets is encoded into n coded packets, for $n > k$, and transmitted over a packet erasure channel, that is, a channel that introduces errors to packets with a certain probability. Packet erasure decoding at a destination node will be successful at recovering the k source packets if k linearly independent coded packets are received without errors. Received packets that contain errors are referred to as *partial packets* and are discarded by the packet erasure decoder. Proposed solutions often rely on random linear packet erasure coding, e.g., protocols based on network coding or fountain coding that enable the extension of the coverage of terrestrial infrastructures with the support of the satellite segment [3]–[5], or cooperative coded caching schemes based on fountain coding that facilitate the delivery of popular content to connected vehicles [6].

A. Utilizing the Algebraic Properties of RLNC

Partial packet recovery (PPR) is a family of methods that attempt to repair partial packets. When PPR is combined with packet erasure decoding, both the correctly received coded packets and the repaired coded packets can be utilized in the reconstruction of the k source packets. If each coded packet generated by RLNC consists of b bits, and the n transmitted coded packets are stacked to form an $n \times b$ matrix, then each column represents a codeword of n bits. Therefore, the output of RLNC can be viewed not only as the row-wise concatenation of n coded packets of b bits, but also as the column-wise concatenation of b codewords of n bits. Packetized rateless algebraic consistency (PRAC) [7] is a PPR method that exploits the algebraic properties of RLNC to identify codewords that contain errors and iteratively search for valid codewords to replace erroneous codewords. Given the row/column correspondence of coded packets and codewords in RLNC, correcting errors in codewords is equivalent to repairing segments in coded packets. Variants of PRAC that aim to reduce the required decoding time include data aware PRAC (DAPRAC) [8] and segmented PRAC (S-PRAC) [9].

B. Guessing Additive Random Noise

Whereas PRAC methods leverage the algebraic properties of RLNC, syndrome decoding (SD) for RLNC [10] capitalizes on the observation that, in typical wireless channel conditions, up to 95% of the content of partial packets at layers higher than the physical layer is often error-free [11]. Thus, SD attempts to correct sparse errors that channel decoding at the physical layer either failed to correct or introduced because of error propagation. In contrast to PRAC-based schemes, which look for the most likely transmitted codeword given a received erroneous codeword, SD looks for the most likely error pattern that altered a codeword. SD conforms to the broad definition of guessing random additive noise decoding (GRAND), which was initially proposed for the decoding of any channel code at the physical layer [12], although its principles can be applied to upper layers too. GRAND leverages the fact that, in low noise conditions, the search space for all possible error patterns that could have corrupted a codeword is smaller than the search space for all possible codewords of a code. Details about recently proposed GRAND variants, primarily for the physical layer, and their implementation can be found on [13].

Besides SD, which considers coding over the finite field $\mathbb{F}_2$ and transmission over a binary symmetric channel, two other schemes that align with the principles of GRAND have been proposed to aid random linear network decoding. The scheme described in [14] extends SD to handle burst errors, in which case error patterns that altered adjacent codewords are correlated. An alternative to RLNC assisted by SD was proposed in [15], which also assumes transmission over binary symmetric channels, but relies on a syndrome definition that is seemingly different from that used in SD, and considers coding operations

Ioannis Chatzigeorgiou is with the School of Computing and Communications, Lancaster University, LA1 4WA, United Kingdom (email: i.chatzigeorgiou@lancaster.ac.uk).

over any extension of $\mathbb{F}_2$, i.e., any finite field $\mathbb{F}_{2^\rho}$ for $\rho \geqslant 1$. The proposed technique yields a decoding advantage, in terms of error correcting capability, over SD when RLNC is over $\mathbb{F}_2$, but its decoding complexity becomes prohibitive as the packet length b or the number of coded packets n increase. To reduce decoding complexity, the authors of [15] place restrictions on the coding process, which limit the number of error patterns that the decoder needs to consider, but impose the use of finite fields larger than $\mathbb{F}_2$.

C. Objective and Contributions

The objective of this paper is to study random linear network decoding over $\mathbb{F}_2$ aided by SD [10] and derive a closed-form expression that bounds from below the success probability of this joint scheme. We stress that this paper neither develops novel decoding methods nor looks into finite fields larger than $\mathbb{F}_2$. The reason for focusing on RLNC and SD over $\mathbb{F}_2$ is not to remain faithful to the original scheme proposed in [10], but to demonstrate that characterization of the success probability of this joint scheme is not trivial, even when $\mathbb{F}_2$ is considered. We envisage that the proposed framework will motivate further research into the analysis of random linear network decoding complemented by GRAND-inspired methods over large finite fields and different channel models.

In the course of attaining the main objective of this paper, we make the following contributions:

- We prove that the noise decoding methods presented in [10] and [15] attempt to solve the exact same estimation problem.
- We obtain an exact expression for the success probability of random linear network decoding assisted by SD, when SD considers a specific constrained set of possible error patterns. The derived expression serves as a lower bound on the success probability of random linear network decoding aided by SD, when SD has access to the full – unconstrained – set of error patterns, as in [10].
- We demonstrate through simulations that choosing the constrained over the unconstrained set of error patterns for SD has the potential to improve the error correction capability of SD, if the stopping criterion is modifed.

D. Paper Organization

The remainder of this paper has been organized as follows: Section II describes random linear network coding at a source node and decoding at a destination node. Section III contrasts key ideas behind PPR based on noise decoding, and explains how it can be combined with random linear network decoding to improve the reliability of the communication link between the source node and a destination node. Sections IV, V and VI are concerned with the analysis of the contribution of SD to the overall success probability of joint random linear network decoding and SD. Theoretical results from this probability analysis are compared with probability measurements, obtained through simulations, in Section VII. Concluding remarks and research directions for future work are presented in Section VIII.

E. Nomenclature

We use lowercase letters in italic type (e.g. b) to denote scalars, lowercase letters in bold type (e.g., $\mathbf{m}$) to represent column vectors, and uppercase letters in bold type (e.g., $\boldsymbol{\Upsilon}$) to refer to matrices. Let $\boldsymbol{\Upsilon} \in \mathbb{F}_2^{n\times k}$, where $\mathbb{F}_2^{n\times k}$ denotes the set of all $n \times k$ matrices over $\mathbb{F}_2$. The L_1 norm $\|\boldsymbol{\Upsilon}\|_1$ provides the Hamming weight of $\boldsymbol{\Upsilon}$, which enumerates the number of non-zero elements in $\boldsymbol{\Upsilon}$. We have adopted the notation $\boldsymbol{\Upsilon}_{i:i',j:j'}$ to refer to a submatrix of $\boldsymbol{\Upsilon}$ composed of those entries of $\boldsymbol{\Upsilon}$ that occupy rows i to i' and columns j to j'. A range of rows or a single row of $\boldsymbol{\Upsilon}$ is denoted by $\boldsymbol{\Upsilon}_{i:i',*}$ and $\boldsymbol{\Upsilon}_{i,*}$, respectively. Similarly, $\boldsymbol{\Upsilon}_{*,j:j'}$ and $\boldsymbol{\Upsilon}_{*,j}$ represent submatrices of $\boldsymbol{\Upsilon}$ formed by a range of columns or a single column, respectively. For example, if we are tasked to calculate the Hamming weight of the second row of $\boldsymbol{\Upsilon} \in \mathbb{F}_2^{2\times 4}$, defined as:

$$\boldsymbol{\Upsilon} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix},$$

we obtain $\boldsymbol{\Upsilon}_{2,*} = (0\ 1\ 1\ 1)$ and $\|\boldsymbol{\Upsilon}_{2,*}\|_1 = 3$. This nomenclature has been introduced to facilitate the description of packet erasure coding and error estimation that are discussed in Section II and Section III, respectively.

II. Packet Erasure Coding and Decoding based on Binary Random Linear Network Codes

A source node that employs binary RLNC performs the following operation

$$\mathbf{X} = \mathbf{G}\mathbf{U} \tag{1}$$

to encode a source message, expressed as $\mathbf{U} \in \mathbb{F}_2^{k\times b}$, into a coded message, represented by $\mathbf{X} \in \mathbb{F}_2^{n\times b}$, using the *generator* matrix $\mathbf{G} \in \mathbb{F}_2^{n\times k}$. Essentially, the source message has been segmented into k source packets, each composed of b bits. The source packets have then be stacked together to form the $k \times b$ matrix $\mathbf{U}$, that is, $\mathbf{U}_{j,*}$ represents the j-th source packet, for $1 \leqslant j \leqslant k$. Similarly, the rows of $\mathbf{X}$, denoted by $\mathbf{X}_{i,*}$ for $1 \leqslant i \leqslant n$, represent coded packets. The n coded packets are broadcast to one or more destination nodes. We assume that *systematic* coding is employed, which implies that the first k of the n coded packets are identical to the k source packets, while the remaining $n-k$ coded packets are linear combinations of the source packets. The generator matrix $\mathbf{G}$ can thus be expressed as:

$$\mathbf{G} = \begin{pmatrix} \mathbf{I}_k \\ \mathbf{C} \end{pmatrix}, \tag{2}$$

where $\mathbf{I}_k$ is the $k \times k$ identity matrix. Each element of the $(n-k) \times k$ matrix $\mathbf{C}$ is chosen uniformly and at random from $\mathbb{F}_2$. Systematic coding ensures that $\mathbf{G}$ has full rank, i.e., $\mathrm{rank}(\mathbf{G}) = k$, which suggests that $\mathbf{X}$ will be successfully decoded into $\mathbf{U}$ by a destination node when communication is error-free. The work presented in this paper can be extended to non-systematic coding as long as $\mathrm{rank}(\mathbf{G}) = k$ and, therefore, elementary column operations on $\mathbf{G}$ can transform $\mathbf{G}$ into standard form as in (2).

Equation (1) can also be viewed from a column-wise perspective, whereby an input word $\mathbf{U}_{*,l}$ of length k is encoded into a codeword $\mathbf{X}_{*,l}$ of length n using a (n,k) binary random

linear code described by the $n \times k$ generator matrix $\mathbf{G}$, i.e., $\mathbf{X}_{*,l} = \mathbf{G}\,\mathbf{U}_{*,l}$ for $1 \leqslant l \leqslant b$. The (n, k) code is *rateless*, that is, for a given k, the value of n and, by extension, the code rate k/n are not fixed. For $n \geqslant k$, additional rows can be randomly generated and appended to $\mathbf{G}$, leading to an increase in the number of transmitted coded packets or, equivalently, an increase in the length of the generated codewords. In conclusion, matrix $\mathbf{X}$ can be regarded as the outcome of row-wise randomized packet erasure coding or column-wise random linear coding, which are different perspectives on the encoding of matrix $\mathbf{U}$ using a single (n, k) random linear code.

Each coded packet $\mathbf{X}_{i,*}$, for $1 \leqslant i \leqslant n$, is transmitted to a destination node over a binary symmetric channel (BSC) characterized by crossover probability ε. The received coded packet at the destination node is denoted by $\mathbf{Y}_{i,*}$. In line with the methods and terminology used in the literature of PPR, e.g., [8]–[10], each of the n received coded packets is classified as *valid* (i.e., error-free) or *partial* (i.e., erroneous). If $n_{\mathrm{R}} \leqslant n$ received coded packets contain no errors, their indices form the ordered sequence $(v_i)_{i=1}^{n_{\mathrm{R}}} = (v_1, \ldots, v_{n_{\mathrm{R}}})$, where $v_i < v_{i'}$ for $i < i'$.

The rows in $\mathbf{Y}$ that correspond to valid packets can be extracted from $\mathbf{Y}$ and compose a new matrix $\mathbf{Y}^{(\mathrm{v})}$ with the help of sequence $(v_i)_{i=1}^{n_{\mathrm{R}}}$. The $n_{\mathrm{R}} \times b$ matrix $\mathbf{Y}^{(\mathrm{v})}$ can be obtained from the $n \times b$ matrix $\mathbf{Y}$ using:

$$\mathbf{Y}^{(\mathrm{v})}_{i,*} = \mathbf{Y}_{v_i,*} = \mathbf{X}_{v_i,*} \quad \text{for} \quad i = 1, \ldots, n_{\mathrm{R}}, \tag{3}$$

since the valid packets at the destination node are identical to the corresponding coded packets at the source node, which suggests that $\mathbf{Y}^{(\mathrm{v})} = \mathbf{X}^{(\mathrm{v})}$. The rows in $\mathbf{G}$ that contributed to the generation of the n_{R} valid packets can be isolated in a similar manner using $\mathbf{G}^{(\mathrm{v})}_{i,*} = \mathbf{G}_{v_i,*}$ for $i = 1, \ldots, n_{\mathrm{R}}$. Therefore, relationship (1) at the destination node reduces to $\mathbf{X}^{(\mathrm{v})} = \mathbf{G}^{(\mathrm{v})}\,\mathbf{U}$, where the source message conveyed by $\mathbf{U}$ can be recovered if $\mathrm{rank}(\mathbf{G}^{(\mathrm{v})}) = k$. Evidently, decoding requires knowledge of $\mathbf{G}$ at the destination node. To achieve this with minimal overhead, the seed that initializes the pseudo-random number generator, which produces the random elements of $\mathbf{G}$ at the source node, could be embedded in the header of each transmitted coded packet [16]. As a result, the pseudo-random number generator at the destination node would be in sync with its counterpart at the source node [7].

In the event that $\mathrm{rank}(\mathbf{G}^{(\mathrm{v})}) < k$, fewer than k among the n_{R} valid packets are linearly independent, a unique solution for $\mathbf{U}$ cannot be obtained and, thus, packet erasure decoding is deemed unsuccessful. As explained in Section I, PPR does not discard partial packets, but attempts to repair them. Consequently, PPR has the potential to improve the chances that k among the repaired and valid packets are linearly independent, and increase the probability that $\mathrm{rank}(\mathbf{G}^{(\mathrm{v})}) = k$. The following section presents two PPR methods, which estimate the impact of errors on transmitted packets that have been generated by systematic RLNC.

III. PPR based on Error Estimation

GRAND was proposed as a universal decoding method for the physical layer of any (n, k) code, when n takes small to moderate values and the code rate k/n is high [12]. In this case, the entropy of the noise is smaller than the entropy of the codewords, therefore identifying the error pattern that corrupted a codeword is simpler than exhaustively searching through all possible codewords. As GRAND is not code-specific, it has rendered decoding of random linear codes feasible. Section II established that RLNC is equivalent to column-wise random linear coding. This suggests that GRAND and, in general, error estimation methods are not only suitable for the physical layer, but can also be used at layers higher than the physical layer to correct bit errors in partial packets generated by RLNC.

Since partial and valid packets form the rows of matrix $\mathbf{Y}$ at the destination node, as mentioned in Section II, we can write:

$$\mathbf{Y} = \mathbf{X} + \mathbf{E}, \tag{4}$$

where $\mathbf{E}$ is the *error* matrix. The objective of the two PPR methods described in [10] and [15], and discussed in this section, is the estimation of $\mathbf{E}$, which can help repair partial packets that form rows of $\mathbf{Y}$ in (4). The repaired packets will increase the number of valid packets, increase the number of rows in $\mathbf{G}^{(\mathrm{v})}$, raise the probability that $\mathrm{rank}(\mathbf{G}^{(\mathrm{v})}) = k$ and improve the likelihood that the source message will be successfully reconstructed.

In operational conditions, physical-layer techniques (e.g., forward error correction) at the destination node counteract most of the effects of the channel noise, thus significantly reducing the number of errors passed on to layers higher than the physical layer, which is where random linear network decoding and PPR are employed. This implies that the error matrix $\mathbf{E}$ in (4) is expected to be *sparse*, that is, most of the entries of $\mathbf{E}$ are zero. The uniqueness of the solution produced by the error estimation process in PPR relies on $\mathbf{E}$ being sufficiently sparse [10].

A. Column-wise Error Matrix Estimation

A destination node employing syndrome decoding (SD) [10] exploits knowledge of the $n \times k$ generator matrix $\mathbf{G}$ to build the $n \times (n - k)$ *parity-check* matrix $\mathbf{H}$ as follows:

$$\mathbf{H} = \left(\; -\mathbf{C} \mid \mathbf{I}_{n-k} \; \right)^{\top}, \tag{5}$$

so that:

$$\mathbf{H}^{\top}\mathbf{G} = \mathbf{0}_{(n-k)\times k}\,, \tag{6}$$

where $-\mathbf{C} = \mathbf{C}$ when arithmetic operations are in $\mathbb{F}_2$, and $\mathbf{0}_{(n-k)\times k}$ is the $(n - k) \times k$ zero matrix. Multiplication of $\mathbf{H}^{\top}$ by $\mathbf{Y}$ at the destination node produces the $(n - k) \times b$ *syndrome* matrix $\mathbf{S}$, i.e., $\mathbf{S} = \mathbf{H}^{\top}\mathbf{Y}$. Using (1), (4) and (6), we find that the relationship between the syndrome matrix $\mathbf{S}$ and the error matrix $\mathbf{E}$ is:

$$\mathbf{S} = \mathbf{H}^{\top}\mathbf{Y} = \mathbf{H}^{\top}(\mathbf{X} + \mathbf{E}) = \mathbf{H}^{\top}(\mathbf{G}\mathbf{U} + \mathbf{E}) = \mathbf{H}^{\top}\mathbf{E}. \tag{7}$$

Let sequence $(p_i)_{i=1}^{n-n_{\mathrm{R}}}$ contain the indices of the partial packets, which were discarded by packet erasure decoding, as seen in Section II. Row $\mathbf{E}_{p_i,*}$, for $i = 1, \ldots, n - n_{\mathrm{R}}$, should contain at least one non-zero element, since it has altered the transmitted coded packet $\mathbf{X}_{p_i,*}$ and led to the reception of

partial packet $\mathbf{Y}_{p_i,*}$. The remaining n_R rows of $\mathbf{E}$ should contain zeros only. In a similar manner to (3), the non-zero rows in the error matrix $\mathbf{E}$ can be extracted from $\mathbf{E}$ and form the $(n-n_\mathrm{R})\times b$ matrix $\mathbf{E}^{(\mathrm{p})}$ using:

$$\mathbf{E}^{(\mathrm{p})}_{i,*} = \mathbf{E}_{p_i,*} \quad \text{for} \quad i = 1,\ldots,n-n_\mathrm{R}. \tag{8}$$

The all-zero rows of $\mathbf{E}$ and the corresponding columns of $\mathbf{H}^\top$ do not impact the result of the product $\mathbf{H}^\top\mathbf{E}$ and can thus be removed. In other words, $\mathbf{H}^\top\mathbf{E} = (\mathbf{H}^{(\mathrm{p})})^\top\mathbf{E}^{(\mathrm{p})}$, where $\mathbf{H}^{(\mathrm{p})}_{i,*} = \mathbf{H}_{p_i,*}$ for $i = 1,\ldots,n-n_\mathrm{R}$, which implies that (7) can be rewritten as:

$$\mathbf{S} = [\mathbf{H}^{(\mathrm{p})}]^\top\ \mathbf{E}^{(\mathrm{p})}. \tag{9}$$

Mohammadi *et al.* [10] decomposed the problem of estimating the $(n-n_\mathrm{R})\times b$ matrix $\mathbf{E}^{(\mathrm{p})}$ into b independent problems, each aiming at estimating a column of $\mathbf{E}^{(\mathrm{p})}$:

$$\mathbf{S}_{*,j} = [\mathbf{H}^{(\mathrm{p})}]^\top\ \mathbf{E}^{(\mathrm{p})}_{*,j} \quad \text{for} \quad j = 1,\ldots,b. \tag{10}$$

Given that $\mathbf{E}$ is a sparse matrix, the solution to (10) can be formulated as:

$$\hat{\mathbf{E}}^{(\mathrm{p})}_{*,j} = \arg\min_{\mathbf{m}} \|\mathbf{m}\|_1 \tag{11a}$$

$$\text{subject to } [\mathbf{H}^{(\mathrm{p})}]^\top\mathbf{m} = \mathbf{S}_{*,j}, \tag{12a}$$

where $\hat{\mathbf{E}}^{(\mathrm{p})}_{*,j}$ is the estimate of $\mathbf{E}^{(\mathrm{p})}_{*,j}$, and $\mathbf{m}\in\mathbb{F}_2^{n-n_\mathrm{R}}$ is a column vector that satisfies constraint (12a) and has the minimum possible Hamming weight. SD considers (11a) and initiates an exhaustive search for a solution. The Hamming weight of $\mathbf{m}$ is gradually incremented until a realization of $\mathbf{m}$ that has the lowest Hamming weight and satisfies (12a) is found. Then, column $\hat{\mathbf{E}}^{(\mathrm{p})}_{*,j}$ is set equal to the identified vector $\mathbf{m}$. This process is repeated for $j = 1,\ldots,b$ until all columns of $\hat{\mathbf{E}}^{(\mathrm{p})}$ have been obtained.

B. Full Error Matrix Estimation

A PPR method, which also requires the derivation of a syndrome matrix for the estimation of $\mathbf{E}$ and the subsequent correction of rows in $\mathbf{Y}$ using (4), was proposed in [15]. The proposed method leverages the fact that the last $n-k$ transmitted packets, also known as *non-systematic* packets, are random linear combinations of the first k transmitted packets, called *systematic* packets. According to (1) and (2), the relationship between the systematic packets $\mathbf{X}_{1,*},\ldots,\mathbf{X}_{k,*}$ and the non-systematic packets $\mathbf{X}_{k+1,*},\ldots,\mathbf{X}_{n,*}$ is:

$$\mathbf{X}_{k+1:n,*} = \mathbf{C}\mathbf{X}_{1:k,*}. \tag{13}$$

Upon formation of matrix $\mathbf{Y}$ from the n received packets, as described in Section II, the destination node calculates the following $(n-k)\times b$ syndrome matrix:

$$\mathbf{\Delta} = \mathbf{Y}_{k+1:n,*} + \mathbf{C}\mathbf{Y}_{1:k,*}. \tag{14}$$

The syndrome matrix $\mathbf{\Delta}$ can be rewritten as a function of the error matrix $\mathbf{E}$, if (14) is combined with (4) and (13):

$$\begin{aligned}\mathbf{\Delta} &= (\mathbf{X}_{k+1:n,*} + \mathbf{E}_{k+1:n,*}) + \mathbf{C}(\mathbf{X}_{1:k,*} + \mathbf{E}_{1:k,*})\\ &= \mathbf{E}_{k+1:n,*} + \mathbf{C}\mathbf{E}_{1:k,*},\end{aligned} \tag{15}$$

since $\mathbf{X}_{k+1:n,*} + \mathbf{X}_{k+1:n,*} = \mathbf{0}_{(n-k)\times b}$. An estimate of the error matrix $\mathbf{E}$, denoted by $\hat{\mathbf{E}}$, can be obtained using $\mathbf{\Delta}$, $\mathbf{C}$ as well as $(v_{i'})_{i'=1}^{n_\mathrm{R}}$ and $(p_i)_{i=1}^{n-n_\mathrm{R}}$, which contain the indices of the valid and partial packets, respectively, as follows [15]:

$$\hat{\mathbf{E}} = \arg\min_{\mathbf{M}} \sum_{p=p_1}^{p_{n-n_\mathrm{R}}} \|\mathbf{M}_{p,*}\|_1 \tag{16a}$$

$$\text{subject to } \mathbf{M}_{i,*} \neq \mathbf{0}_{1\times b} \ \text{ for } \ i = p_1,\ldots,p_{n-n_\mathrm{R}} \tag{17a}$$

$$\mathbf{M}_{i',*} = \mathbf{0}_{1\times b} \ \text{ for } \ i' = v_1,\ldots,v_{n_\mathrm{R}} \tag{18a}$$

$$\mathbf{M}_{k+1:n,*} + \mathbf{C}\mathbf{M}_{1:k,*} = \mathbf{\Delta}. \tag{19a}$$

The estimated error matrix will be a matrix $\mathbf{M}\in\mathbb{F}_2^{n\times b}$ that has the lowest Hamming weight, as per (16a). Condition (17a) ensures that rows of $\mathbf{M}$ with indices corresponding to partial packets should not contain only zeros, as this would imply that the respective received packets should have been valid. By contrast, rows of $\mathbf{M}$ that contain only zeros should have indices that correspond to valid packets, as imposed by (18a). Matrix $\mathbf{M}$ is accepted as a solution if it satisfies (19a).

C. Comparison of the Estimation Methods

Closer inspection of the syndrome matrix $\mathbf{S}$, used in [10] and defined in (7), and the syndrome matrix $\mathbf{\Delta}$, used in [15] and defined in (14), reveals that the two matrices are identical:

$$\begin{aligned}\mathbf{S} &= \mathbf{H}^\top\mathbf{E}\\ &= \left(\mathbf{C} \mid \mathbf{I}_{n-k}\right)\left(\frac{\mathbf{E}_{1:k,*}}{\mathbf{E}_{k+1:n,*}}\right)\\ &= \mathbf{C}\mathbf{E}_{1:k,*} + \mathbf{I}_{n-k}\mathbf{E}_{k+1:n,*}\\ &= \mathbf{\Delta}.\end{aligned} \tag{20}$$

Thus, the optimization problem in (16) can be rewritten as:

$$\hat{\mathbf{E}} = \arg\min_{\mathbf{M}} \sum_{p=p_1}^{p_{n-n_\mathrm{R}}} \|\mathbf{M}_{p,*}\|_1 \tag{21a}$$

$$\text{subject to } \mathbf{M}_{i,*} \neq \mathbf{0}_{1\times b} \ \text{ for } \ i = p_1,\ldots,p_{n-n_\mathrm{R}} \tag{22a}$$

$$\mathbf{M}_{i',*} = \mathbf{0}_{1\times b} \ \text{ for } \ i' = v_1,\ldots,v_{n_\mathrm{R}} \tag{23a}$$

$$\mathbf{H}^\top\mathbf{M} = \mathbf{S}, \tag{24a}$$

where constraint (19a) has been replaced by the equivalent constraint (24a).

We conclude that the PPR methods developed in [10] and [15] for RLNC consider the same problem, that is, the estimation of the error matrix $\mathbf{E}\in\mathbb{F}_2^{n\times b}$ using $\mathbf{S} = \mathbf{H}^\top\mathbf{E}$, given $\mathbf{S}\in\mathbb{F}_2^{(n-k)\times b}$ and $\mathbf{H}\in\mathbb{F}_2^{n\times(n-k)}$. The proposers of SD [10] acknowledged the computational complexity of a solving method that treats $\mathbf{E}$ as a single block and opted for (11), which estimates $\mathbf{E}$ column by column. Since (11) focuses only on the $n-n_\mathrm{R}$ non-zero rows of $\mathbf{E}$, the search space for each column of $\mathbf{E}$ contains 2^{n-n_R} possible column realizations. As each of the b columns of $\mathbf{E}$ is estimated separately, the search space for the whole matrix $\mathbf{E}$ has size $b2^{n-n_\mathrm{R}}$, which increases *linearly* with the number of bits b in a packet. On the other hand, the authors of [15] favored (16), which is equivalent to (21), to estimate the whole matrix $\mathbf{E}$ as a single block. In that case, the search space for each non-zero row of $\mathbf{E}$ has

size $2^b - 1$. However, each possible realization of a single row of $\mathbf{E}$ needs to be considered along with every possible realization of all other rows of $\mathbf{E}$, leading to a search space of size $(2^b - 1)^{n-n_\mathrm{R}}$, which grows *exponentially* with b.

We established that SD [10] is significantly less computationally expensive than the method proposed in [15], as it trades estimation accuracy for computational efficiency. The column-wise estimation process of the error matrix in SD does not eliminate the possibility of incorrectly returning a matrix that contains only zeros in rows that correspond to partial packets. The probability of SD returning an error matrix that contains all-zero rows at invalid positions reduces with an increasing packet size b or an increasing crossover probability ε. This paper derives a lower bound on the probability that a destination node, which employs RLNC decoding aided by SD, will decode the received packets and recover the source message. The bound will reveal the minimum achievable increase in the decoding probability when error matrix estimation complements RLNC decoding.

D. Problem Decomposition

In order for SD to estimate a column of the error matrix, all realizations of the $1 \times (n - n_\mathrm{R})$ vector $\mathbf{m}$ that have a specific weight are considered in (11a). If none of them satisfy (12a), the weight is incremented by one and the process is repeated. For example, if the all-zero vector of length $n - n_\mathrm{R}$ does not meet (12a) for column j of the syndrome matrix, SD tests all $n - n_\mathrm{R}$ vectors $\mathbf{m}$ that have weight $\|\mathbf{m}\|_1 = 1$. The non-zero entries in $\mathbf{m}$ are gradually increased and all $\binom{n-n_\mathrm{R}}{\|\mathbf{m}\|_1}$ possible vector realizations are tested for each weight value $\|\mathbf{m}\|_1$. The entries of the first vector $\mathbf{m}$ that satisfies (12a) are copied over to column j of $\hat{\mathbf{E}}^{(\mathrm{p})}$ as per (11a). To obtain a lower bound on the success probability of RLNC decoding aided by SD, we introduce a variant of SD that imposes the following additional constraint on (11a) and (12a):

$$\|\mathbf{m}\|_1 \leqslant 1. \tag{25}$$

We shall refer to this variant of SD as *single-error correcting SD* (SEC-SD). In SEC-SD, only realizations of $\mathbf{m}$ that have weight 0 and 1 are queried. If no suitable vector $\mathbf{m}$ is identified, the search is abandoned and estimation of the error matrix is deemed unsuccessful. Whereas the search space for the error matrix has size $b2^{n-n_\mathrm{R}}$ when SD is used, the search space reduces to $b(n - n_\mathrm{R} + 1)$ when SEC-SD is employed. Fig. 1 exemplifies the difference between SD and SEC-SD.

Although SEC-SD has been introduced as a theoretical tool to study joint random linear network decoding and syndrome decoding, it could also be used in practice. For example, if 0.5 is the probability that a packet of $b = 64$ bits will be received in error, and if errors are uniformly distributed in each packet, the crossover probability will be $\varepsilon = 1-(1-0.5)^{1/64} \approx 0.01$. If $n = 25$ packets are transmitted, the 25×64 error matrix will contain $25 \times 64 \times 0.01 = 16$ uniformly distributed non-zero entries, on average. SEC-SD will be able to handle error matrices that contain up to $b = 64$ non-zero entries, provided that each column contains at most one non-zero entry.

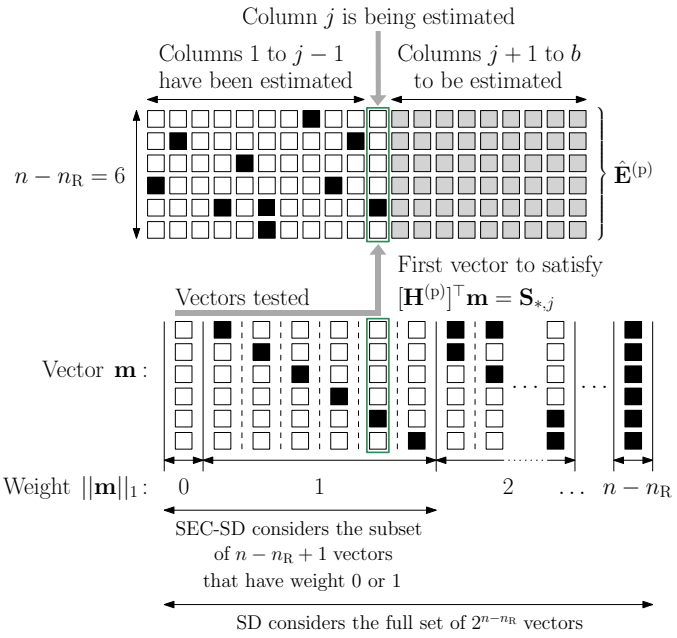

Fig. 1. Application of SD for the identification of the lowest-weight vector $\mathbf{m}$ that satisfies (12a) and thus meets the requirements for use as column j of matrix $\hat{\mathbf{E}}^{(\mathrm{p})}$. Entries set to 1 are depicted by '■', whereas entries set to 0 are represented by '□'. The figure also depicts the smaller set of vectors that SEC-SD considers for a column of $\hat{\mathbf{E}}^{(\mathrm{p})}$.

With the help of SEC-SD, the problem of obtaining a lower bound on the probability of RLNC decoding aided by SD can be decomposed into the following three parts:

- Analysis of the probability that the BSC will alter at most one bit in each of the b codewords generated by the (n, k) random linear code or, equivalently, that each column of the $n \times b$ error matrix contains at most one non-zero entry. However, the non-zero entries in each column are conditioned by the reception of n_R valid packets and $n - n_\mathrm{R}$ partial packets. In other words, the non-zero entries should be distributed in the columns of $\mathbf{E}$ in such a way that n_R rows of $\mathbf{E}$ contain only zeros, while the remaining $n-n_\mathrm{R}$ rows contain at least one non-zero entry. This task is carried out in Section IV.
- Derivation of the probability that SEC-SD will return an accurate estimate of the error matrix, provided that each column of the error matrix contains at most one non-zero entry. This task is challenging because we cannot consider the decoding of a fixed rate (n, k) systematic random linear code. Instead, we need to consider the decoding of a variable rate $(n-n_\mathrm{R}, k)$ random linear code, which has been obtained by randomly puncturing the output of an (n, k) systematic random linear code, for values of n_R between 0 and $n-1$. This problem is tackled in Section V.
- Evaluation of the probability of RLNC decoding aided by SEC-SD, which bounds from below the probability of RLNC decoding aided by SD. This work is presented in Section VI.

IV. Error Matrix Requirements for SEC

As explained in Section II, RLNC is a packet-wise process that produces n coded packets of length b bits, which compose

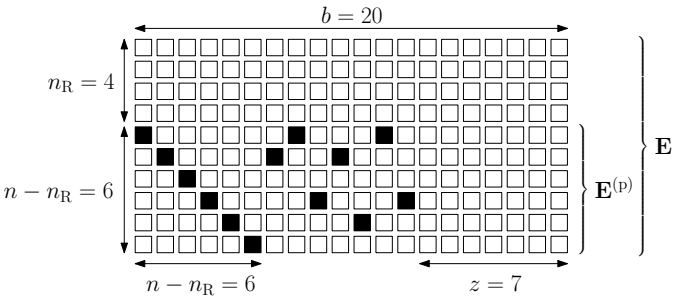

Fig. 2. Example of the $n \times b$ matrix $\mathbf{E}$ and the $(n - n_{\rm R}) \times b$ submatrix $\mathbf{E}^{(\rm p)}$ for $n = 10$, $n_{\rm R} = 4$ and $b = 20$. Entries set to 1 are depicted by '■', whereas zero-valued entries are represented by '□'. In this example, the last $z = 7$ columns contain only zeros.

the *rows* of the $n \times b$ matrix $\mathbf{X}$. This process is equivalent to performing bit-wise random linear coding to generate b codewords of length n bits, which form the *columns* of $\mathbf{X}$.

To derive the probability that SD will correct single errors in codewords generated by an (n, k) random linear code, we first need to evaluate the probability that the $n \times b$ error matrix $\mathbf{E}$ is composed of columns that contain a single 1 or only zeros. We should also take into account the fact that knowledge of the indices of the $n_{\rm R}$ valid packets and the $n - n_{\rm R}$ partial packets is available at a destination node, which implies that $\mathbf{E}$ consists of $n_{\rm R}$ all-zero rows and $n-n_{\rm R}$ non-zero rows. The aforementioned conditions can be used to define the following set of matrices:

$$\begin{aligned}\mathcal{S} = \{\boldsymbol{\Upsilon} \in \mathbb{F}_2^{n\times b} \colon\ & \|\boldsymbol{\Upsilon}_{*,j}\|_1 \leqslant 1, \text{ for } j = 1, \ldots, b, \\ & \|\boldsymbol{\Upsilon}_{v_{i'},*}\|_1 = 0, \text{ for } i' = 1, \ldots, n_{\rm R}, \\ & \|\boldsymbol{\Upsilon}_{p_i,*}\|_1 \geqslant 1, \text{ for } i = 1, \ldots, n - n_{\rm R}\},\end{aligned}$$

where $v_{i'}$ and p_i are entries of the sequences $(v_{i'})_{i'=1}^{n_{\rm R}}$ and $(p_i)_{i=1}^{n-n_{\rm R}}$, which contain the indices of the valid and partial packets, respectively. The error matrix $\mathbf{E}$ meets the requirements for single error correction (SEC) if $\mathbf{E} \in \mathcal{S}$.

Recall that $\mathbf{E}^{(\rm p)}$ is the $(n - n_{\rm R}) \times b$ submatrix of the error matrix $\mathbf{E}$ that is formed of the $n - n_{\rm R}$ non-zero rows of $\mathbf{E}$. If we remove the condition for $n_{\rm R}$ all-zero rows from $\mathcal{S}$, we obtain the set:

$$\begin{aligned}\mathcal{S}^{(\rm p)} = \{\boldsymbol{\Upsilon} \in \mathbb{F}_2^{(n-n_{\rm R})\times b} \colon\ & \|\boldsymbol{\Upsilon}_{*,j}\|_1 \leqslant 1, \text{for } j = 1, \ldots, b, \\ & \|\boldsymbol{\Upsilon}_{i,*}\|_1 \geqslant 1, \text{for } i = 1, \ldots, n - n_{\rm R}\}.\end{aligned}$$

We infer that, if $\mathbf{E}^{(\rm p)} \in \mathcal{S}^{(\rm p)}$, then $\mathbf{E} \in \mathcal{S}$. Fig. 2 shows an example of an $n \times b$ error matrix $\mathbf{E}$ that fulfills the requirements for SEC. It also illustrates that the absence of all-zero rows from the $(n - n_{\rm R}) \times b$ submatrix $\mathbf{E}^{(\rm p)}$ imposes the existence of $n - n_{\rm R}$ columns in $\mathbf{E}^{(\rm p)}$ where the non-zero entry in each of these columns occupies a different row, as is the case with the first $n - n_{\rm R}$ columns of $\mathbf{E}^{(\rm p)}$ in this example. To complete $\mathbf{E}^{(\rm p)}$, we set the entries in z columns to 0 and inserted a single 1 in a random position in each of the remaining columns, as shown in Fig. 2. In general, $0 \leqslant z \leqslant b - n + n_{\rm R}$. Although Fig. 2 provides a specific example of $\mathbf{E}$ so that $\mathbf{E}^{(\rm p)} \in \mathcal{S}^{(\rm p)}$ and thus $\mathbf{E} \in \mathcal{S}$, it helps visualize the general structure of $\mathbf{E}$ and guide the derivation of an expression for the probability that $\mathbf{E} \in \mathcal{S}$ for any value of $n_{\rm R}$.

Let $n' = n - n_{\rm R}$ and $b' = b - z$, and let us initially focus on the $n' \times b'$ submatrix of $\mathbf{E}^{(\rm p)}$ that contains no all-zero columns. The following lemma enumerates all possible realizations of this $n' \times b'$ submatrix.

Lemma 1: The number of $n' \times b'$ matrices, where each row contains *at least* one 1, each column holds *exactly* one 1, and all other entries are set to 0, is given by:

$$f(n', b') = \sum_{i=0}^{n'} \binom{n'}{i} (-1)^{n'-i}\, i^{b'}. \tag{26}$$

Proof: The condition for at least one non-zero entry in each row and exactly one non-zero entry in each column of the $n' \times b'$ matrix implies that $b' \geqslant n'$. Furthermore, n' of the b' columns will be *orthogonal unit column vectors*, that is, their non-zero entries will occupy different positions, as previously explained. The remaining $b' - n'$ columns will be randomly selected copies of unit column vectors.

For $b' = n'$, the n' available orthogonal unit column vectors will be distributed across the n' columns of the $n' \times n'$ matrix. The first column will be occupied by one of the n' available orthogonal column vectors, the second column will be set equal to one of the remaining $n' - 1$ orthogonal unit column vectors, etc. In general, the i-th column will be occupied by one of the remaining $n'-i+1$ orthogonal unit column vectors, for $i = 1, \ldots, n'$, resulting in $f(n', n') = (n')!$ possible realizations of the $n' \times n'$ matrix.

For $b' = n' + 1$, assume that the first $i - 1$ columns and the last $n' - i$ columns of the $n' \times (n' + 1)$ matrix are occupied by orthogonal unit column vectors, while the i-th column is a copy of one of the first $i - 1$ columns. For a fixed value of i, the number of $n' \times (n' + 1)$ matrix realizations are:

$$\begin{aligned}&\underbrace{n' \cdot (n' - 1) \cdot \ldots \cdot (n' - i + 2)}_{\substack{\text{number of permutations} \\ \text{for the first } i-1 \text{ columns}}} \cdot (i - 1) \cdot \underbrace{(n' - i + 1) \cdot \ldots \cdot 1}_{\substack{\text{number of permutations} \\ \text{for the last } n'-i \text{ columns}}} \\ &= (n')!(i - 1),\end{aligned}$$

since the i-th column can be a copy of any of the previous $i-1$ columns, while the remaining n' columns will be mapped to one of the $(n')!$ permutations of the n' available orthogonal unit column vectors. If every possible value of i is considered, the expression for the total number number of $n' \times (n' + 1)$ matrix realizations assumes the form:

$$f(n', n' + 1) = (n')! \sum_{i=2}^{n'+1} (i - 1) = (n')! \sum_{i=1}^{n'} i. \tag{27}$$

Following this line of reasoning for any $b' \geqslant n'$, we obtain:

$$f(n', b') = (n')! \sum_{i_1=1}^{n'} i_1 \sum_{i_2=1}^{i_1} i_2 \ldots \sum_{i_{b'-n'}=1}^{i_{b'-n'-1}} i_{b'-n'}. \tag{28}$$

The nested sum in (28) is equal to [17, Eq. (6)]:

$$\sum_{i_1=1}^{n'} i_1 \sum_{i_2=1}^{i_1} i_2 \ldots \sum_{i_{b'-n'}=1}^{i_{b'-n'-1}} i_{b'-n'} = \left\{ {b' \atop n'} \right\}, \tag{29}$$

where $\left\{ {b' \atop n'} \right\}$ is known as the Stirling number of the second kind and determines the number of ways to partition a set of b'

elements into n' non-empty subsets. A closed-form expression for the Stirling number of the second kind is [18, Eq. (13.13)]:

$$\left\{ \begin{matrix} b' \\ n' \end{matrix} \right\} = \frac{1}{(n')!} \sum_{i=0}^{n'} \binom{n'}{i} (-1)^{n'-i} \, i^{b'}. \tag{30}$$

Substituting $\left\{ \begin{matrix} b' \\ n' \end{matrix} \right\}$ with the right-hand side of (30) into (29) and then into (28) gives (26). ∎

The result of Lemma 1 leads us to the following corollary:

Corollary 1: The probability that the $(n - n_{\mathrm{R}}) \times b$ matrix $\mathbf{E}^{(\mathrm{p})}$ is a member of set $\mathcal{S}^{(\mathrm{p})}$, for any value of $n - n_{\mathrm{R}}$, when z of its columns contain only zero entries, is:

$$\mathbb{P}\left[\mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, n - n_{\mathrm{R}}, z\right] = \\ = \binom{b}{z} (1-\varepsilon)^{(n-n_{\mathrm{R}}-1)b+z} \, \varepsilon^{b-z} \, f(n - n_{\mathrm{R}}, b - z). \tag{31}$$

Proof: Enumeration of all matrix realizations, conditioned on being members of $\mathcal{S}^{(\mathrm{p})}$ for any value of z, considers all valid realizations of the $b - z$ non-zero columns, provided by $f(n - n_{\mathrm{R}}, b - z)$ in (26), and all possible ways of choosing the remaining z of the b columns to be all-zero columns. The $b - z$ non-zero columns of every matrix realization contain a total of $b - z$ non-zero entries, each occurring with probability ε. The remaining $(n - n_{\mathrm{R}} - 1)b + z$ entries of the $(n - n_{\mathrm{R}}) \times b$ matrix hold zeros, each occurring with probability $1 - \varepsilon$. ∎

Based on Corollary 1, we can now derive the probability that the error matrix $\mathbf{E}$ meets the requirements for SEC.

Corollary 2: The probability that the $n \times b$ error matrix $\mathbf{E}$ is a member of set $\mathcal{S}$, when n_{R} of its rows contain only zero entries, is:

$$\mathbb{P}\left[\mathbf{E} \in \mathcal{S}, n_{\mathrm{R}}\right] = \\ = \binom{n}{n_{\mathrm{R}}} (1-\varepsilon)^{n_{\mathrm{R}} b} \sum_{z=0}^{b-n+n_{\mathrm{R}}} \mathbb{P}\left[\mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, n - n_{\mathrm{R}}, z\right]. \tag{32}$$

Proof: Derivation of the probability that $\mathbf{E} \in \mathcal{S}$, for any value of n_{R}, exploits the fact that n_{R} of the n rows contain zeros with probability $(1-\varepsilon)^{n_{\mathrm{R}} b}$, while the remaining $n - n_{\mathrm{R}}$ rows form the submatrix $\mathbf{E}^{(\mathrm{p})}$. Hence, $\mathbb{P}\left[\mathbf{E} \in \mathcal{S}, n_{\mathrm{R}}\right]$ depends on the probability that $\mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}$ for all realizations of $\mathbf{E}^{(\mathrm{p})}$, that is, for every possible number of all-zero columns in $\mathbf{E}^{(\mathrm{p})}$, i.e., $z = 0, \ldots, b - n + n_{\mathrm{R}}$. ∎

V. Successful Decoding using SEC-SD

Section IV established that any column of the $(n - n_{\mathrm{R}}) \times b$ submatrix $\mathbf{E}^{(\mathrm{p})}$ should contain zeros or a single non-zero entry to meet the requirements for SEC. Let $\mathbf{e}_i$ denote a unit vector of $n - n_{\mathrm{R}}$ entries, where the i-th entry is set to 1 and all other entries contain zeros. If $\mathbf{e}_i \neq \mathbf{e}_\ell$ for $i \neq \ell$, then $\mathbf{e}_i$ and $\mathbf{e}_\ell$ are called orthogonal unit vectors, as also mentioned in Section IV. Using this notation, we expect:

$$\forall j \in \{1, \ldots, b\}, \mathbf{E}^{(\mathrm{p})}_{*,j} = \mathbf{0}_{(n-n_{\mathrm{R}})\times 1} \text{ or} \\ \exists i \in \{1, \ldots, n - n_{\mathrm{R}}\} : \mathbf{E}^{(\mathrm{p})}_{*,j} = \mathbf{e}_i. \tag{33}$$

Section IV also explained that $\mathbf{E}^{(\mathrm{p})}$ should contain $n - n_{\mathrm{R}}$ columns, where the non-zero entry in each of these columns occupies a different row, that is, $\mathbf{E}^{(\mathrm{p})}$ should contain $n - n_{\mathrm{R}}$ orthogonal columns:

$$\forall i \in \{1, \ldots, n - n_{\mathrm{R}}\}, \exists j \in \{1, \ldots, b\} : \mathbf{E}^{(\mathrm{p})}_{*,j} = \mathbf{e}_i. \tag{34}$$

Analysis of SEC-SD, which is the focus of this section, will invoke both (33) and (34).

If the SEC requirements are satisfied, we can deduce from (10) and (33) that column j of the $(n-k) \times b$ syndrome matrix $\mathbf{S}$, for $j = 1, \ldots, b$, will be equal to:

$$\mathbf{S}_{*,j} = \begin{cases} \mathbf{0}_{(n-k)\times 1}, & \text{if } \mathbf{E}^{(\mathrm{p})}_{*,j} = \mathbf{0}_{(n-n_{\mathrm{R}})\times 1} \quad (35a) \\ \left[\mathbf{H}^{(\mathrm{p})}_{i,*}\right]^\top, & \text{if } \mathbf{E}^{(\mathrm{p})}_{*,j} = \mathbf{e}_i. \quad (35b) \end{cases}$$

According to (35a), if the j-th codeword is received free from errors and, thus, the j-th column of $\mathbf{E}^{(\mathrm{p})}$ contains only zeros, then all entries in the j-th column of $\mathbf{S}$ will be set to zero. On the other hand, if only the i-th entry of the j-th column of $\mathbf{E}^{(\mathrm{p})}$ is equal to 1, the i-th column of $[\mathbf{H}^{(\mathrm{p})}]^\top$ will be singled out and its contents will be copied over to the j-th column of $\mathbf{S}$, as per (35b). Note that the i-th column of $[\mathbf{H}^{(\mathrm{p})}]^\top$ is the transpose of the i-th row of $\mathbf{H}^{(\mathrm{p})}$, i.e., $[\mathbf{H}^{(\mathrm{p})}]^\top_{*,i} = [\mathbf{H}^{(\mathrm{p})}_{i,*}]^\top$.

In line with (11) and (25), SEC-SD attempts to estimate $\mathbf{E}^{(\mathrm{p})}_{*,j}$ and produce $\hat{\mathbf{E}}^{(\mathrm{p})}_{*,j}$. If $\mathbf{S}_{*,j}$ is an all-zero column vector, the decoder will assume that $\hat{\mathbf{E}}^{(\mathrm{p})}_{*,j} = \mathbf{0}_{(n-k)\times 1}$. If $\mathbf{S}_{*,j}$ contains non-zero entries, then $\mathbf{S}_{*,j}$ will be compared to each column of $[\mathbf{H}^{(\mathrm{p})}]^\top$, and the indices of all columns that are identical copies of $\mathbf{S}_{*,j}$ will be stored in a set, denoted by $\mathcal{I}$. If $\mathcal{I}$ is not empty, an element i' will be randomly selected from $\mathcal{I}$, and the decoder will output $\hat{\mathbf{E}}^{(\mathrm{p})}_{*,j} = \mathbf{e}_{i'}$. If $\mathcal{I}$ is empty, decoding will be deemed unsuccessful and the process will terminate. The estimation process can be summarized as follows:

$$\hat{\mathbf{E}}^{(\mathrm{p})}_{*,j} = \begin{cases} \mathbf{0}_{(n-n_{\mathrm{R}})\times 1}, & \text{if } \mathbf{S}_{*,j} = \mathbf{0}_{(n-k)\times 1} \quad (36a) \\ \mathbf{e}_{i'} \text{ for } i' \in \mathcal{I}, & \text{if } \mathcal{I} \neq \emptyset \quad (36b) \\ \mathbf{0}_{(n-n_{\mathrm{R}})\times 1}, & \text{if } \mathcal{I} = \emptyset \text{ (failure).} \quad (36c) \end{cases}$$

Note in (36c) that, after the j-th column of $\hat{\mathbf{E}}^{(\mathrm{p})}$ is set to the all-zero column vector for convenience, a failure is declared and no attempt is made to estimate the remaining columns of the error matrix. The reason for specifically assigning the all-zero column vector to the j-th column of $\hat{\mathbf{E}}^{(\mathrm{p})}$ in (36c), given that decoding has failed, will be clarified in Section VII.

As can be inferred from (35) and (36), the probability that SEC-SD is successful, that is, the probability that $\hat{\mathbf{E}}^{(\mathrm{p})} = \mathbf{E}^{(\mathrm{p})}$ and by extension $\hat{\mathbf{E}} = \mathbf{E}$, given that the SEC requirements are fulfilled, depends on $[\mathbf{H}^{(\mathrm{p})}]^\top$.

A. Parity-check Matrices for SEC-SD

Whereas the rank of $\mathbf{G}^{(\mathrm{v})}$ is important in RLNC decoding, the *spark* of $[\mathbf{H}^{(\mathrm{p})}]^\top$ plays a key role in SD, as will become apparent in this section. The term 'spark' of a matrix was first coined by Donoho and Elad in [19], and is defined as follows:

Definition 1 (Adapted from [20, p. 23]): The spark of a matrix $\boldsymbol{\Upsilon}$ with entries from $\mathbb{F}_2$ is the smallest number of columns in $\boldsymbol{\Upsilon}$ that are linearly dependent:

$$\mathrm{spark}(\boldsymbol{\Upsilon}) \doteq \min\{\|\mathbf{m}\|_1\} \text{ subject to } \boldsymbol{\Upsilon}\mathbf{m} = \mathbf{0} \text{ for } \mathbf{m} \neq \mathbf{0}.$$

By definition, $\text{spark}(\boldsymbol{\Upsilon}) \geqslant 1$, where equality holds if and only if $\boldsymbol{\Upsilon}$ contains an all-zero column.

The spark of the $(n-k)\times(n-n_{\text{R}})$ matrix $[\mathbf{H}^{(\text{p})}]^{\top}$ coincides with the notion of the *minimum distance* of an $(n-n_{\text{R}},k)$ linear code, which has been obtained by puncturing the output of an (n,k) linear code that has parity-check matrix $\mathbf{H}$. The impact of the spark of $[\mathbf{H}^{(\text{p})}]^{\top}$ in decoding success is well-known in the coding theory community, but is briefly explained below for completeness:

- $\text{spark}([\mathbf{H}^{(\text{p})}]^{\top}) = 1$: At least one column of $[\mathbf{H}^{(\text{p})}]^{\top}$ contains only zeros. Let ℓ be the index of that column. According to (34), $\mathbf{E}^{(\text{p})}$ contains all possible $n-n_{\text{R}}$ orthogonal unit column vectors, therefore a value for j exists such that $\mathbf{E}^{(\text{p})}_{*,j} = \mathbf{e}_{\ell}$, where $j \in \{1,\ldots,b\}$. The decoder obtains $\mathbf{S}_{*,j} = [\mathbf{H}^{(\text{p})}_{\ell,*}]^{\top} = \mathbf{0}_{(n-n_{\text{R}})\times 1}$ from (35b), and outputs $\hat{\mathbf{E}}^{(\text{p})}_{*,j} = \mathbf{0}_{(n-n_{\text{R}})\times 1} \neq \mathbf{E}^{(\text{p})}_{*,j}$ according to (36a). In conclusion, a value for j always exists in $\{1,\ldots,b\}$ that leads to $\hat{\mathbf{E}}^{(\text{p})}_{*,j} \neq \mathbf{E}^{(\text{p})}_{*,j}$, which suggests that $\hat{\mathbf{E}}^{(\text{p})} \neq \mathbf{E}^{(\text{p})}$ is always true.
- $\text{spark}([\mathbf{H}^{(\text{p})}]^{\top}) = 2$: At least two columns of $[\mathbf{H}^{(\text{p})}]^{\top}$ are identical, hence the size of $\mathcal{I}$, denoted by $|\mathcal{I}|$, is $|\mathcal{I}| \geqslant 2$ for some values of $j \in \{1,\ldots,b\}$. For those values of j, the value of i' in (36b) cannot be determined with certainty to ensure that $\hat{\mathbf{E}}^{(\text{p})}_{*,j} = \mathbf{E}^{(\text{p})}_{*,j}$, but can be guessed with probability $1/|\mathcal{I}|$.
- $\text{spark}([\mathbf{H}^{(\text{p})}]^{\top}) \geqslant 3$: The columns of $[\mathbf{H}^{(\text{p})}]^{\top}$ are different from each other, i.e., they are unique, but not necessarily linearly independent. Thus, $|\mathcal{I}| = 1$ for any value of $j \in \{1,\ldots,b\}$, so i' in (36b) is always assigned the value for which $\hat{\mathbf{E}}^{(\text{p})}_{*,j} = \mathbf{E}^{(\text{p})}_{*,j}$.

We deduce that the spark of $[\mathbf{H}^{(\text{p})}]^{\top}$ should be at least 2 for SEC-SD to have a non-zero probability of obtaining an accurate estimate of the error matrix, provided that the error matrix fulfills the requirements for SEC.

Recall that systematic RLNC is employed, which allows us to express the $(n-k)\times n$ transpose of the parity-check matrix as $\mathbf{H}^{\top} = (\mathbf{C} \mid \mathbf{I}_{n-k})$, as mentioned in Section III-A. Given that n_{R} denotes the total number of valid packets, let $k_{\text{R}} \leqslant k$ be the number of valid systematic packets. Section III-A also explained that $[\mathbf{H}^{(\text{p})}]^{\top}$ is formed by $n-n_{\text{R}}$ columns of $\mathbf{H}^{\top}$. We can infer that the first $k-k_{\text{R}}$ columns of $[\mathbf{H}^{(\text{p})}]^{\top}$ will be drawn from $\mathbf{C}$ and be random column vectors. The remaining $(n-n_{\text{R}})-(k-k_{\text{R}})$ columns of $[\mathbf{H}^{(\text{p})}]^{\top}$ will be drawn from $\mathbf{I}_{n-k}$ and be *ordered* orthogonal unit column vectors, that is, if the (i,j)-th entry and the $(i',j+1)$-th entry of $[\mathbf{H}^{(\text{p})}]^{\top}$ are both 1, then $i' > i$ for $j = k-k_{\text{R}}+1,\ldots,n-n_{\text{R}}-1$.

For compactness, let $n' = n-n_{\text{R}}$ and $k' = k-k_{\text{R}}$. Furthermore, let us consider the more general case of matrices of r' rows. If we enumerate all matrices in $\mathbb{F}_2^{r'\times n'}$ that have spark 2 or greater, and are formed by the concatenation of k' random columns and $n'-k'$ ordered orthogonal unit columns, we can then focus on the special case of $r' = n-k$ and derive the probability of $[\mathbf{H}^{(\text{p})}]^{\top}$ being one of those matrices. To facilitate this enumeration, we introduce two additional parameters to characterize a matrix that meets the aforementioned constraints:

- $\mathbf{a} = (a_1 \cdots a_{n'})$ describes the partitioning of the n' columns of the matrix in groups based on the multiplicity of each column, where a_i signifies the number of different columns that appear i times in the matrix, and can thus form a_i disjoint groups of size i. If all n' columns of the matrix are unique, then $a_1 = n'$ and $a_2 = \ldots = a_{n'} = 0$. Vector $\mathbf{a}$ should be an element of

$$\mathcal{A} = \Big\{(a_1 \ \cdots \ a_{n'}) \in \mathbb{Z}_+^{n'} : \sum_{i=1}^{n'} i a_i = n', \quad n'-k' \leqslant \sum_{i=1}^{n'} a_i \leqslant 2^{r'}-1\Big\},$$

  where $\mathbb{Z}_+$ represents the set of non-negative integers. The first condition ensures that the number of groups weighted by their sizes matches the total number of columns. The second condition establishes that the number of groups should be equal to or greater than the number of orthogonal unit columns but cannot be larger than the number of all possible column realizations, excluding the all-zero column. The all-zero column has been excluded to avoid enumerating spark-1 matrices.
- $\mathbf{q} = (q_1 \cdots q_{n'})$ expresses the contribution of the orthogonal unit columns to the partitioning of all columns of the matrix. It specifies that q_i among the a_i groups of size i have been seeded by q_i orthogonal unit columns. In other words, each of the q_i groups is composed by one of the $n'-k'$ orthogonal unit columns and $i-1$ identical columns from the first k' columns of the matrix. Vector $\mathbf{q}$ belongs to a set that depends on vector $\mathbf{a}$. This set is defined as:

$$\mathcal{Q}(\mathbf{a}) = \Big\{(q_1 \ \cdots \ q_{n'}) \in \mathbb{Z}_+^{n'} : \sum_{i=1}^{n'} q_i = n'-k', \quad 0 \leqslant q_i \leqslant a_i\Big\}.$$

  The first condition ensures that the number of orthogonal unit columns matches the number of groups seeded by them, provided that the number of groups for every valid i falls within a range, specified by the second condition.

To clarify the derivation of vectors $\mathbf{a}$ and $\mathbf{q}$, let us consider the matrix below, where $r' = 3$, $k' = 7$ and $n' = 9$:

$$\Bigg(\underbrace{\begin{matrix}1&0&1&1&0&0&1\\0&1&0&1&1&0&0\\0&1&0&1&1&1&1\end{matrix}}_{k'=7}\ \underbrace{\begin{matrix}1&0\\0&0\\0&1\end{matrix}}_{n'-k'=2}\Bigg).$$

The matrix does not contain an all-zero column but some columns are identical, hence the spark of the matrix is 2. Two of the columns are unique ($a_1 = 2$), two columns are repeated twice ($a_2 = 2$) and one column is repeated three times ($a_3 = 1$), thus $\mathbf{a} = (2\ 2\ 1\ 0\ \cdots\ 0)$ describes how the $n' = 9$ columns of the matrix can be partitioned into $a_1+a_2+a_3 = 5$ groups. The last two columns of the matrix are orthogonal unit columns, which contributed to the formation of a group of size 2 ($q_2 = 1$) and the only group of size 3 ($q_3 = 1$), i.e., $\mathbf{q} = (0\ 1\ 1\ 0\ \cdots\ 0)$.

Lemma 2: The number of matrices in $\mathbb{F}_2^{r' \times n'}$ that have spark 2 or greater and are formed by the concatenation of k' random columns and $n'-k'$ preset ordered orthogonal unit columns, for a partition $\mathbf{a} \in \mathcal{A}$ of the n' columns, is given by:

$$g(r',k',n',\mathbf{a}) = \\ = \sum_{\forall \mathbf{q} \in \mathcal{Q}(\mathbf{a})} \binom{n'-k'}{q_1,\ldots,q_{n'}} \frac{k'!}{(n'!)^{a_{n'}-q_{n'}} \prod_{t=1}^{n'-1}(t!)^{a_t-q_t+q_{t+1}}} \cdot \\ \cdot \prod_{i=1}^{n'} \binom{2^{r'}-1-(n'-k')-\sum_{j=1}^{i-1}(a_j-q_j)}{a_i-q_i}. \quad (37)$$

Proof: Appendix A presents a sketch of the proof. ∎

Expression (37) can be applied to the specifc case of the $(n-k)\times(n-n_{\mathrm{R}})$ matrix $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ and contribute to an expression for the probability that all columns of $\mathbf{E}^{(\mathrm{p})}$ will be correctly estimated when the spark of $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ is at least 2.

B. Error Matrix Estimation in SEC-SD

The number of matrix realizations for $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ with spark at least 2 for a partition $\mathbf{a} \in \mathcal{A}$, where the first $k-k_{\mathrm{R}}$ columns of every matrix realization form a submatrix that is selected uniformly and at random from $\mathbb{F}_2^{(n-k)\times(k-k_{\mathrm{R}})}$, can be obtained from (37) for $r'=n-k$, $k'=k-k_{\mathrm{R}}$ and $n'=n-n_{\mathrm{R}}$. Division of $g(n-k,k-k_{\mathrm{R}},n-n_{\mathrm{R}},\mathbf{a})$ by the total number of matrix realizations, given by $2^{(n-k)(k-k_{\mathrm{R}})}$, yields the probability that $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ has spark 2 or higher for a particular $\mathbf{a} \in \mathcal{A}$. SEC-SD will be successful if every column of $\hat{\mathbf{E}}^{(\mathrm{p})}$ matches the corresponding column of $\mathbf{E}^{(\mathrm{p})}$, provided that $\mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}$, and if $\mathrm{spark}([\mathbf{H}^{(\mathrm{p})}]^{\top}) \geqslant 2$ for every possible partition $\mathbf{a} \in \mathcal{A}$, as illustrated by the following lemma.

Lemma 3: The probability of SEC-SD correctly estimating $\mathbf{E}^{(\mathrm{p})}$ and repairing $n-n_{\mathrm{R}}$ partial packets when $k-k_{\mathrm{R}}$ of them are partial systematic packets, provided that $\mathbf{E}^{(\mathrm{p})}$ satisfies the requirements for SEC and contains z all-zero columns, is:

$$\mathbb{P}\left[\hat{\mathbf{E}}^{(\mathrm{p})} = \mathbf{E}^{(\mathrm{p})} \mid \mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, k-k_{\mathrm{R}}, n-n_{\mathrm{R}}, z\right] = \\ = \sum_{\forall \mathbf{a} \in \mathcal{A}} \left[\frac{g(n-k,k-k_{\mathrm{R}},n-n_{\mathrm{R}},\mathbf{a})}{2^{(n-k)(k-k_{\mathrm{R}})}} \cdot \right. \\ \left. \cdot \prod_{i=1}^{n-n_{\mathrm{R}}} \left(\frac{1}{i}\right)^{ia_i} \left(\frac{\sum_{j=1}^{n-n_{\mathrm{R}}} a_j}{n-n_{\mathrm{R}}}\right)^{b-z-n+n_{\mathrm{R}}} \right]. \quad (38)$$

Proof: The proof has been deferred to Appendix B. ∎

The final step for this section is the derivation of the success probability of SEC-SD when n_{R} packets have been received without errors, averaged over all possible values of k_{R} and z, which immediately follows from Lemma 3 as shown below.

Corollary 3: The probability that SEC-SD will successfully guess the error matrix $\mathbf{E}$, when $\mathbf{E}$ meets the SEC requirements and n_{R} received packets are valid, can be expressed as:

$$\mathbb{P}\left[\hat{\mathbf{E}} = \mathbf{E},\, \mathbf{E} \in \mathcal{S},\, n_{\mathrm{R}}\right] = \\ = \sum_{k_{\mathrm{R}}=k_{\min}}^{k_{\max}} \left\{ \binom{k}{k_{\mathrm{R}}} \binom{n-k}{n_{\mathrm{R}}-k_{\mathrm{R}}} (1-\varepsilon)^{n_{\mathrm{R}}b} \cdot \right. \\ \cdot \sum_{z=0}^{b-n+n_{\mathrm{R}}} \left(\mathbb{P}\left[\hat{\mathbf{E}}^{(\mathrm{p})} = \mathbf{E}^{(\mathrm{p})} \mid \mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, k-k_{\mathrm{R}}, n-n_{\mathrm{R}}, z\right] \cdot \right. \\ \left. \left. \cdot \mathbb{P}\left[\mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, n-n_{\mathrm{R}}, z\right] \right) \right\}, \quad (39)$$

where $k_{\min} = \max(0, n_{\mathrm{R}}-n+k)$ and $k_{\max} = \min(n_{\mathrm{R}}, k)$.

Proof: Using the same reasoning as in Corollary 2 for the relationship between $\mathbf{E}$ and submatrix $\mathbf{E}^{(\mathrm{p})}$, we can write:

$$\mathbb{P}\left[\hat{\mathbf{E}} = \mathbf{E},\, \mathbf{E} \in \mathcal{S},\, n_{\mathrm{R}}\right] = \binom{n}{n_{\mathrm{R}}} (1-\varepsilon)^{n_{\mathrm{R}}b} \cdot \\ \cdot \sum_{z=0}^{b-n+n_{\mathrm{R}}} \mathbb{P}\left[\hat{\mathbf{E}}^{(\mathrm{p})} = \mathbf{E}^{(\mathrm{p})},\, \mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})},\, n-n_{\mathrm{R}},\, z\right]. \quad (40)$$

The probability in (40) can be decomposed as follows:

$$\mathbb{P}\left[\hat{\mathbf{E}}^{(\mathrm{p})} = \mathbf{E}^{(\mathrm{p})},\, \mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})},\, n-n_{\mathrm{R}},\, z\right] = \\ = \sum_{k_{\mathrm{R}}=k_{\min}}^{k_{\max}} \left(\mathbb{P}\left[\hat{\mathbf{E}}^{(\mathrm{p})} = \mathbf{E}^{(\mathrm{p})} \mid \mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, k-k_{\mathrm{R}}, n-n_{\mathrm{R}}, z\right] \cdot \right. \\ \left. \cdot \mathbb{P}\left[k-k_{\mathrm{R}} \mid n-n_{\mathrm{R}}\right] \mathbb{P}\left[\mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, n-n_{\mathrm{R}}, z\right] \right), \quad (41)$$

where the first of the three product terms in (41) has been derived in (38) and is conditioned on the number of partial systematic packets, $k-k_{\mathrm{R}}$. If $n-n_{\mathrm{R}}$ of the n transmitted packets are partially received, the probability that $k-k_{\mathrm{R}}$ of them are partial systematic packets, and the remaining ones are partial coded packets, is given by the second term in (41), which can be expanded into:

$$\mathbb{P}\left[k-k_{\mathrm{R}} \mid n-n_{\mathrm{R}}\right] = \binom{k}{k_{\mathrm{R}}} \binom{n-k}{n_{\mathrm{R}}-k_{\mathrm{R}}} \Big/ \binom{n}{n_{\mathrm{R}}}. \quad (42)$$

The first product term in (41) is also conditioned on $\mathbf{E}^{(\mathrm{p})}$ fulfilling the SEC requirements when it contains $n-n_{\mathrm{R}}$ non-zero rows and z all-zero columns. The probability that $\mathbf{E}^{(\mathrm{p})}$ satisfies the aforementioned structural expectations is given by the third product term in (41), which has been derived in (31). Substituting (31), (38) and (42) into (41) gives (39). The sum in (39) is calculated over all valid values of k_{R}, where the smallest value, $k_{\min}$, should be a non-negative integer and the largest value, $k_{\max}$, should not exceed n_{R} when $n_{\mathrm{R}} < k$. ∎

VI. Analysis of RLNC Decoding aided by SEC-SD

The aim of this paper is to derive an exact expression for the probability that RLNC decoding aided by SEC-SD, denoted by P_{SEC}, will recover the k source packets that compose the source message. Such an expression can then be used to bound from below the probability P of obtaining the source message when RLNC decoding is combined with SD, i.e.,

$$P \geqslant P_{\mathrm{SEC}}. \quad (43)$$

Recall that SD, in contrast to SEC-SD, does not terminate if no vectors of weight 0 or 1 that satisfy (12a) exist, but continues to search for vectors of increasingly larger weight until a solution is found. Derivation of an expression for P_{SEC} can be facilitated by Lemma 4, the proof of which follows a similar line of reasoning as that of the Singleton bound [21].

Lemma 4: If $\mathrm{rank}(\mathbf{G}^{(\mathrm{v})}) = k$ and hence RLNC decoding is successful, then SEC-SD would also be successful if the error matrix met the SEC requirements.

Proof: Assume that systematic RLNC encodes k packets into n packets using $\mathbf{G}$, as in (2), errors occur during transmission and $\mathrm{rank}(\mathbf{G}^{(\mathrm{v})}) = k$ at a destination node. If k_{R} systematic packets are valid and the corresponding k_{R} pivot columns of $\mathbf{G}^{(\mathrm{v})}$ are discarded, the remaining $k - k_{\mathrm{R}}$ columns should contain an $(n_{\mathrm{R}} - k_{\mathrm{R}}) \times (k - k_{\mathrm{R}})$ submatrix that has rank $k - k_{\mathrm{R}}$. Given (5), this $(n_{\mathrm{R}} - k_{\mathrm{R}}) \times (k - k_{\mathrm{R}})$ submatrix is also located within the first $k - k_{\mathrm{R}}$ columns of $[\mathbf{H}^{(\mathrm{p})}]^{\top}$, which is the transpose of the parity-check matrix of the equivalent $(n-n_{\mathrm{R}}, k)$ linear code (see Section V-A). Thus, the first $k - k_{\mathrm{R}}$ columns of $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ are linearly independent and, if considered with the remaining $(n - k) - (n_{\mathrm{R}} - k_{\mathrm{R}})$ orthogonal unit columns of $[\mathbf{H}^{(\mathrm{p})}]^{\top}$, we deduce that the rank of $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ is $n - n_{\mathrm{R}}$ and the spark of $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ is $n - n_{\mathrm{R}} + 1$. For $n - n_{\mathrm{R}} \geqslant 2$, $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ consists of two or more columns that are all unique and different from the all-zero column, since $\mathrm{spark}([\mathbf{H}^{(\mathrm{p})}]^{\top}) \geqslant 3$. For $n - n_{\mathrm{R}} = 1$, $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ reduces to a column, which is different from the all-zero column, since $\mathrm{spark}([\mathbf{H}^{(\mathrm{p})}]^{\top}) = 2$. In general, for $n - n_{\mathrm{R}} > 0$, the columns of $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ are unique and non-zero, therefore the $(n - n_{\mathrm{R}}, k)$ linear code has SEC capability. Consequently, if the SEC requirements are satisfied, SEC-SD will be successful in determining the error matrix. ∎

The probability P_{SEC} can be expressed as the sum of the likelihoods of the following three complementary events:

1) The error matrix is non-zero and fulfills the SEC requirements. Lemma 4 established that an RLNC decoding success can be interpreted as a SEC-SD success. Otherwise, if RLNC decoding is unsuccessful, the task of recovering the source message falls on SEC-SD. Thus, we can say that the decoding outcome depends solely on SEC-SD, hence the RLNC decoding stage can be ignored.
2) The error matrix is non-zero but does not fulfill the SEC requirements. In that case, the decoding outcome depends solely on RLNC decoding, as SEC-SD will fail if RLNC decoding is unsuccessful.
3) All entries of the error matrix are zero because none of the received packets contain bit errors, so neither RLNC decoding nor SEC-SD are required.

Breaking down the decoding process into events E1, E2 and E3 paves the way for the following theorem:

Theorem 1: The probability that RLNC decoding assisted by SEC-SD will recover a message composed of k source packets of length b bits, when $n > k$ packets have been transmitted over a BSC with crossover probability ε, can be obtained from

$$
\begin{aligned}
P_{\mathrm{SEC}} &= \\
&= \sum_{n_{\mathrm{R}}=0}^{n-1} \Bigg\{ \sum_{z=0}^{b-n+n_{\mathrm{R}}} \vartheta(n_{\mathrm{R}}, z) \sum_{k_{\mathrm{R}}=k_{\min}}^{k_{\max}} \binom{k}{k_{\mathrm{R}}} \binom{n-k}{n_{\mathrm{R}}-k_{\mathrm{R}}} \cdot \\
&\quad \cdot \sum_{\forall \mathbf{a} \in \mathcal{A}} \Bigg[\frac{g(n-k, k-k_{\mathrm{R}}, n-n_{\mathrm{R}}, \mathbf{a})}{2^{(n-k)(k-k_{\mathrm{R}})}} \cdot \\
&\quad \cdot \prod_{i=1}^{n-n_{\mathrm{R}}} \left(\frac{1}{i}\right)^{i a_i} \left(\frac{\sum_{j=1}^{n-n_{\mathrm{R}}} a_j}{n-n_{\mathrm{R}}} \right)^{b-z-n+n_{\mathrm{R}}} \Bigg] \Bigg\} + \\
&\quad + \sum_{n_{\mathrm{R}}=k}^{n-1} \Bigg\{ \Bigg[(1-\varepsilon)^{n_{\mathrm{R}} b} \Big(1-(1-\varepsilon)^{b}\Big)^{n-n_{\mathrm{R}}} - \sum_{z=0}^{b-n+n_{\mathrm{R}}} \vartheta(n_{\mathrm{R}}, z) \Bigg] \cdot \\
&\quad \cdot \sum_{k_{\mathrm{R}}=k_{\min}}^{k} \binom{k}{k_{\mathrm{R}}} \binom{n-k}{n_{\mathrm{R}}-k_{\mathrm{R}}} \prod_{i=0}^{k-k_{\mathrm{R}}-1} \left(1 - 2^{-n_{\mathrm{R}}+k_{\mathrm{R}}+i}\right) \Bigg\} + \\
&\quad + (1-\varepsilon)^{nb},
\end{aligned} \tag{44}
$$

where

$$
\vartheta(n_{\mathrm{R}}, z) = \binom{b}{z} (1-\varepsilon)^{(n-1)b+z} \, \varepsilon^{b-z} \, f(n-n_{\mathrm{R}}, b-z),
$$

$k_{\min} = \max(0, n_{\mathrm{R}} - n + k)$, $k_{\max} = \min(n_{\mathrm{R}}, k)$, and functions $f(\cdot)$ and $g(\cdot)$ have been defined in (26) and (37), respectively.

Proof: Probability P_{SEC} can be expressed as the sum of three terms:

$$
P_{\mathrm{SEC}} = P_{\mathrm{E1}} + P_{\mathrm{E2}} + P_{\mathrm{E3}}, \tag{45}
$$

where each term corresponds to one of the aforementioned events, namely E1, E2 and E3.

When event E1 occurs for a particular value of n_{R}, the error matrix satisfies the SEC conditions, i.e., $\mathbf{E} \in \mathcal{S}$. SEC-SD will be successful if the estimated error matrix matches the actual error matrix, i.e., $\hat{\mathbf{E}} = \mathbf{E}$. The expression for P_{E1}, which considers all valid values of n_{R}, assumes the form:

$$
P_{\mathrm{E1}} = \sum_{n_{\mathrm{R}}=0}^{n-1} \mathbb{P}\left[\hat{\mathbf{E}} = \mathbf{E}, \, \mathbf{E} \in \mathcal{S}, \, n_{\mathrm{R}}\right], \tag{46}
$$

where $\mathbb{P}\big[\hat{\mathbf{E}} = \mathbf{E}, \, \mathbf{E} \in \mathcal{S}, \, n_{\mathrm{R}}\big]$ is given in (39).

Event E2 occurs when the error matrix does not satisfy the SEC conditions, that is, $\mathbf{E} \notin \mathcal{S}$. In that case, the RLNC decoder will use the n_{R} valid packets to recover the k source packets, and will succeed if $\mathrm{rank}(\mathbf{G}^{(\mathrm{v})}) = k$, which is possible when $n_{\mathrm{R}} \geqslant k$. The probability that occurrence of event E2 will lead to a decoding success for $k \leqslant n_{\mathrm{R}} \leqslant n-1$ is given by:

$$
P_{\mathrm{E2}} = \sum_{n_{\mathrm{R}}=k}^{n-1} \mathbb{P}\left[\mathbf{E} \notin \mathcal{S}, n_{\mathrm{R}}\right] \mathbb{P}\left[\mathrm{rank}(\mathbf{G}^{(\mathrm{v})}) = k \mid n_{\mathrm{R}}\right]. \tag{47}
$$

To obtain $\mathbb{P}\left[\mathbf{E} \notin \mathcal{S}, n_{\mathrm{R}}\right]$, we can simply subtract $\mathbb{P}\left[\mathbf{E} \in \mathcal{S}, n_{\mathrm{R}}\right]$ from the probability of receiving n_{R} valid packets and $n - n_{\mathrm{R}}$ partial packets:

$$
\begin{aligned}
\mathbb{P}\left[\mathbf{E} \notin \mathcal{S}, n_{\mathrm{R}}\right] &= \binom{n}{n_{\mathrm{R}}} (1-\varepsilon)^{n_{\mathrm{R}} b} (1-(1-\varepsilon)^{b})^{n-n_{\mathrm{R}}} - \\
&\quad - \mathbb{P}\left[\mathbf{E} \in \mathcal{S}, n_{\mathrm{R}}\right],
\end{aligned} \tag{48}
$$

where $\mathbb{P}\left[\mathbf{E} \in \mathcal{S}, n_{\mathrm{R}}\right]$ is given in (32). The second product term in (47), which represents the probability of successful RLNC decoding for a given value of $n_{\mathrm{R}} \geqslant k$ when binary systematic RLNC is used, has been derived in [22]:

$$\mathbb{P}\Big[\mathrm{rank}(\mathbf{G}^{(\mathrm{v})}) = k \mid n_{\mathrm{R}}\Big] = \sum_{k_{\mathrm{R}}=k_{\min}}^{k} \frac{\binom{k}{k_{\mathrm{R}}}\binom{n-k}{n_{\mathrm{R}}-k_{\mathrm{R}}}}{\binom{n}{n_{\mathrm{R}}}} \cdot \cdot \prod_{i=0}^{k-k_{\mathrm{R}}-1} \left(1 - 2^{-n_{\mathrm{R}}+k_{\mathrm{R}}+i}\right). \quad (49)$$

The probability of event E3, that is, the probability that all b bits in each of the n packets will be received correctly, is $P_{\mathrm{E3}} = (1-\varepsilon)^{nb}$. Substitution of P_{E1}, P_{E2} and P_{E3} into (45) results in (44). ∎

The next section compares simulation results with theoretical values obtained using (44), which is an exact expression for the success probability of RLNC decoding aided by SEC-SD, and serves as a lower bound on the success probability of RLNC decoding assisted by SD, as per (43).

VII. Results and Discussion

In addition to the proof of Theorem 1 in Section VI, Fig. 3 provides a visual confirmation of the validity of (44), which exactly matches measurements of the success probability of RLNC decoding aided by SEC-SD that have been obtained through simulations[1]. For completeness, theoretical values [22] and simulation results for the success probability of RLNC decoding have also been included in the plot. For the sake of compactness, RLNC decoding has been shortened to RD, and RLNC decoding aided by SEC-SD has been abbreviated to RD/SEC-SD. Fig. 3 shows how the probability of recovering k source packets is affected by the number of transmitted packets n, the packet length b and the crossover probability ε. The nine subfigures in Fig. 3 have been organized into a 3×3 grid-like arrangement, where the value of k is fixed for subfigures in the same column, and the value of b is fixed for subfigures in the same row. In each subfigure, two values for ε are considered, while n takes values in the range $k \leqslant n \leqslant 1.2k$, where $k \in \{50, 100, 200\}$ and $b \in \{16, 24, 32\}$.

Observe in Fig. 3 that SEC-SD only marginally improves the success probability of RD/SEC-SD for values of n either very close to k or much higher than k. In the former case, the difference $n-k$ is a very small number, hence the $(n-k) \times (n-n_{\mathrm{R}})$ matrix $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ consists of only a few rows. As a result, the random columns of $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ are chosen from a small pool of 2^{n-k} column vectors, making the presence of all-zero columns and identical columns in $[\mathbf{H}^{(\mathrm{p})}]^{\top}$ very likely. Consequently, $\mathrm{spark}([\mathbf{H}^{(\mathrm{p})}]^{\top}) < 3$ with high probability, thus the chances of SEC-SD correctly guessing the error matrix are hindered. On the other hand, when $n \geqslant k/(1-\varepsilon)^b$, k linearly independent packets are found among the valid packets with high probability, which suggests that RLNC decoding is often successful without assistance from SEC-SD. However, for values of n between $k+1$ and $k/(1-\varepsilon)^b$, the contribution of SEC-SD to the success probability of RD/SEC-SD is more substantial and helps RD/SEC-SD achieve a notable coding gain over RD.

Fig. 4 compares the success probabilities of RD and RD/SEC-SD, and contrasts them with the success probability of RD/SD, for $k = 10$, $b = 16$ and $n \in \{12, 14, 16\}$. As described in Section III-D, SEC-SD abandons its attempt to estimate all of the b columns of the error matrix when at least one of its columns has weight greater than 1. On the other hand, SD considers column vectors of increasingly large weight until (12a) is met for every column of the error matrix. As the crossover probability ε increases and the sparsity of the error matrix reduces, SEC-SD is less likely to return an accurate estimate of the error matrix than SD. Fig. 4 confirms that the gap between the success probabilities of RD/SEC-SD and RD/SD widens as the value of ε increases. Nevertheless, Fig. 4 also confirms that the exact expression for the success probability of RD/SEC-SD, given in (44), is a useful lower bound on the success probability of RD/SD.

The success probability of RD combined with a variant of SEC-SD, dubbed 'do not quit' SEC-SD and abbreviated as SEC-SD(dnq), is also depicted in Fig. 4. SEC-SD(dnq) looks for column vectors of weight 0 or 1 that satisfy (12a) but, in contrast to SEC-SD, it does not quit if no solution is found for a column of the estimated error matrix $\hat{\mathbf{E}}$. Instead, the all-zero column vector is assigned to that column, as per (36c), and SEC-SD(dnq) moves on to the estimation of the next column of $\hat{\mathbf{E}}$. When estimation of a column of $\hat{\mathbf{E}}$ is unsuccessful and, thus, $\hat{\mathbf{E}} = \mathbf{E}$ cannot be achieved, SEC-SD quits and declares a decoding failure, whereas SEC-SD(dnq) continues to run until all columns of $\hat{\mathbf{E}}$ have been assigned a column vector of weight 0 or 1, despite the fact that $\hat{\mathbf{E}} \neq \mathbf{E}$ is unavoidable. When $\hat{\mathbf{E}} \neq \mathbf{E}$ in SEC-SD(dnq), ν non-zero rows of $\hat{\mathbf{E}}$ could still match the corresponding rows of $\mathbf{E}$, for $0 \leqslant \nu < n - n_{\mathrm{R}}$, in which case ν of the $n-n_{\mathrm{R}}$ partial packets would be repaired. As a result, the number of valid packets would grow from n_{R} to $n_{\mathrm{R}}+\nu$ and the probability of decoding the k source packets would increase, compared to SEC-SD, as shown in Fig. 4.

Of interest is the potential of RD/SEC-SD(dnq) to achieve a marginally higher success probability than RD/SD, as illustrated in Fig. 4, even though the former decoding scheme utilizes a smaller search space for the estimation of the error matrix than the latter decoding method, as explained in Section III-C. This observation can be justified intuitively if we take a closer look at the estimation process of a column of the error matrix $\mathbf{E}$ that has weight greater than 1, i.e., at least two of its elements are non-zero. SEC-SD(dnq), being unable to find a weight-1 column vector that meets (12a), will assign the all-zero column vector to the corresponding column of $\hat{\mathbf{E}}$. The all-zero column will have no impact on partial packets, as it will neither correct any channel errors nor result in any decoding errors. SD, on the other hand, will continue to test column vectors of weight greater than 1 against (12a) until a column vector that satisfies (12a) is identified. If the estimated column in $\hat{\mathbf{E}}$ is different from the corresponding column in $\mathbf{E}$, the incorrectly guessed non-zero elements in that column will

[1]Software simulations were implemented in MATLAB. The probability of successful decoding for a tuple (k, n, ε, b) has been averaged over 3×10^5 channel realizations. For each realization, submatrix $\mathbf{C}$ was randomly generated, as described in Section II. The MATLAB-based function 'Partitions of an integer' [23] was used to build set $\mathcal{A}$ in (44).

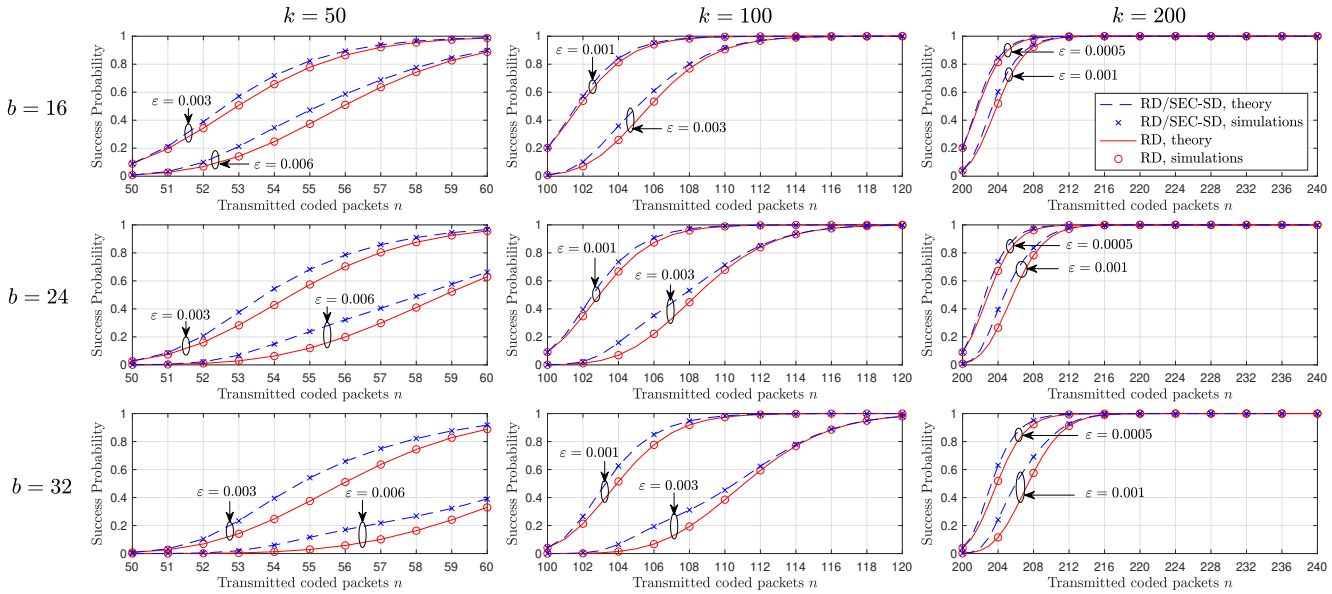

Fig. 3. Theoretical values, which have been calculated using (44) and depicted as dashed curves (−−), for the probability that RLNC decoding aided by SEC-SD (RD/SEC-SD) will be successful, are compared to simulation results, depicted as crosses (×). The number of source packets k is fixed for subfigures in the same column, where $k \in \{50, 100, 200\}$. Similarly, the packet length b in bits is fixed for subfigures in the same row, where $b \in \{16, 24, 32\}$. Each subfigure considers two values for the crossover probability ε. For reference, theoretical values (—) and simulation results (○) for the probability of successful RLNC decoding (RD) have been included in all subfigures, as explained in the legend.

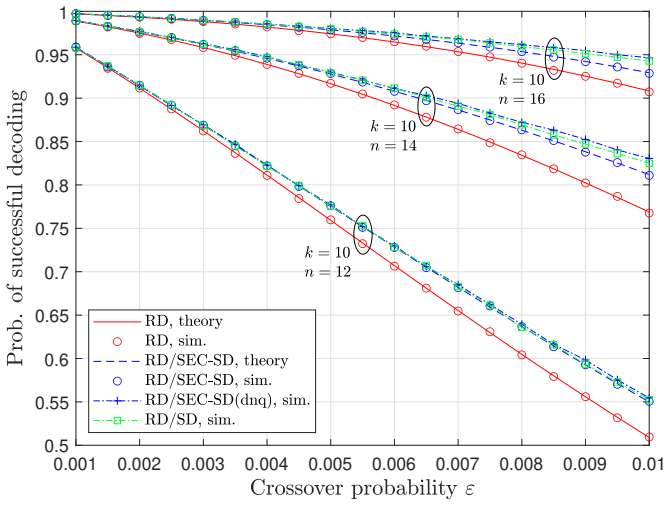

Fig. 4. Probability of successful decoding, when $k = 10$ source packets of $b = 16$ bits are encoded into n packets, transmitted over a BSC with crossover probability ε and decoded using RD, RD/SEC-SD, RD/SEC-SD(dnq) or RD/SD, where $n \in \{12, 14, 16\}$ and $0.001 \leqslant \varepsilon \leqslant 0.01$.

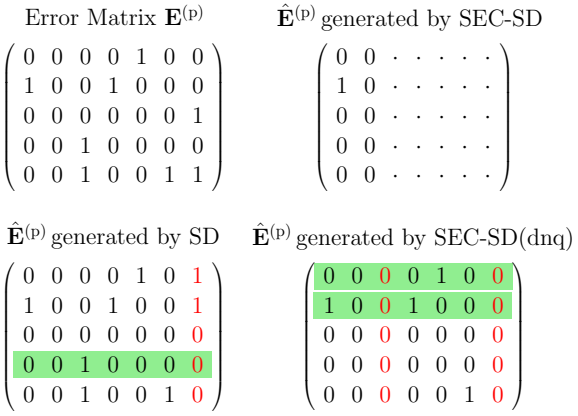

Fig. 5. Example of a case where SEC-SD(dnq) repaired more partial packets than SD, when the number of partial packets is $n - n_{\mathrm{R}} = 5$ and each packet contains $b = 7$ bits. The 5×7 error matrix $\mathbf{E}^{(\mathrm{p})}$ to be estimated is shown at the top-left of the figure. SEC-SD estimated the 1st and 2nd columns of the error matrix but terminated execution when the 3rd column was reached, because the solution is a weight-2 vector but SEC-SD only considers weight-0 and weight-1 vectors. SD correctly estimated the first six columns of the error matrix and identified a vector for the 7th column, which does not match the 7th column of $\mathbf{E}^{(\mathrm{p})}$. SEC-SD(dnq), in contrast to SEC-SD, did not terminate execution when the estimation process could not assign a weight-0 or weight-1 vector to a column, but assigned the all-zero vector to that column. Although SEC-SD(dnq) correctly estimated fewer columns of $\mathbf{E}^{(\mathrm{p})}$ than SD in this example, it successfully guessed more rows (highlighted in green) and, thus, repaired more partial packets, than SD.

introduce decoding errors in partial packets. Fig. 5 shows an example where SD incorrectly estimated a single column of the error matrix and inadvertently introduced errors to all but one of the rows of the error matrix. SD could thus repair a single partial packet, whereas SEC-SD(dnq) could repair two partial packets in the same example depicted in Fig. 5, even though SEC-SD(dnq) was unable to estimate two of the columns of the error matrix and assigned the all-zero column vector to those columns.

Another observation that can be made in Fig. 4 is that the success probabilities of RD/SEC-SD, RD/SEC-SD(dnq) and RD/SD shift closer together as the crossover probability ε drops, for a fixed value of n. This observation is corroborated by Table I, which presents simulation results for $k = 100$, $n = 120$ and $b = 512$, when ε varies between 0.00035 and 0.00045. For $\varepsilon = 0.00045$, the increased packet error

TABLE I
COMPARISON OF PERFORMANCE AND COMPLEXITY MEASURES OF RD/SEC-SD, RD/SEC-SD(DNQ) AND RD/SD FOR $k = 100$ SOURCE PACKETS, $n = 120$ TRANSMITTED CODED PACKETS AND $b = 512$ BITS PER PACKET. THE CROSSOVER PROBABILITY ε VARIES FROM 0.00035 TO 0.00045. THE SUCCESS PROBABILITY OF A DECODING SCHEME HAS BEEN USED AS A PERFORMANCE MEASURE. COMPLEXITY CONSIDERS THE NUMBER OF COLUMN VECTORS THAT HAVE BEEN TESTED, ON AVERAGE, FOR THE ESTIMATION OF ALL COLUMNS OF THE ERROR MATRIX.

Crossover probability ε	Packet error probability $1-(1-\varepsilon)^b$	Success prob. of RD	Success prob. of RD/SEC-SD	Success probability of RD/SEC-SD(dnq)	Success prob. of RD/SD	Avg. number of vectors queried by SEC-SD	Avg. number of vectors queried by SEC-SD(dnq)	Avg. number of vectors queried by SD
0.00035	0.16409	0.44079	0.76007	0.99985	0.99979	810.3	1054.8	1137.3
0.00036	0.16836	0.39557	0.73372	0.99984	0.99976	813.3	1069.3	1157.4
0.00037	0.17261	0.35129	0.70903	0.99984	0.99976	818.8	1085.1	1178.3
0.00038	0.17683	0.31193	0.68384	0.99981	0.99975	824.2	1101.6	1200.5
0.00039	0.18104	0.27246	0.65809	0.99981	0.99975	829.9	1118.7	1224.9
0.00040	0.18522	0.23742	0.63503	0.99981	0.99973	836.0	1136.3	1251.0
0.00041	0.18939	0.20455	0.61062	0.99980	0.99970	840.3	1155.2	1276.4
0.00042	0.19353	0.17451	0.58685	0.99980	0.99970	844.7	1173.8	1305.3
0.00043	0.19765	0.14914	0.56423	0.99979	0.99969	850.9	1194.5	1334.2
0.00044	0.20175	0.12509	0.53969	0.99976	0.99969	853.9	1215.8	1366.7
0.00045	0.20583	0.10754	0.52260	0.99975	0.99961	861.2	1238.4	1399.2

probability impairs the success probability of RD, which sets at 0.10754. SEC-SD can boost the success probability of RD from 0.10754 to 0.52260, which establishes that RD/SD has the potential to achieve an ever higher success probability. Indeed, RD/SD – but also RD/SEC-SD(dnq) – achieve comparable success probabilities that are close to 1, even though estimation of the error matrix by SD requires the testing of more column vectors, on average, than SEC-SD(dnq). As in Fig. 4, Table I confirms that the success probability of RD/SEC-SD approaches the success probabilities of RD/SEC-SD(dnq) and RD/SD for low values of ε. As ε reduces, the sparsity of the error matrix increases, thus the columns of the error matrix are more likely to have weight 0 or 1, which increases the likelihood of all three decoding schemes producing identical estimates. Given that the packet error probability observed at the output of the physical layer in practical scenarios is 0.1 or lower [24], the error matrix is expected to be sparse and, therefore, the success probability of RD/SEC-SD will be a tight lower bound on the success probability of RD/SD, as shown in Fig. 4.

VIII. CONCLUSIONS AND FUTURE DIRECTIONS

This work established that two noise decoding methods, presented in [10] and [15], rely on the same syndrome matrix to estimate the error matrix, that is, the matrix that specifies the location of bit errors in coded packets generated by RLNC and transmitted over a binary symmetric channel. The focus of the paper then shifted to the method proposed in [10], referred to as syndrome decoding (SD), on account of its low complexity, which increases linearly with packet size, and not exponentially as the scheme in [15]. The introduction of an early stopping criterion gave rise to SEC-SD, which is a minimum-complexity variant of SD that traverses a smaller search space than SD and is, hence, suboptimal in terms of error estimation performance. A comprehensive analysis led to the derivation of an exact closed-form expression for the probability that joint random linear network decoding and SEC-SD will be successful, which bounds from below the success probability of random linear network decoding aided by SD. A minor modification in the implementation of SEC-SD uncovered its potential to repair more partial coded packets, on average, than SD while keeping complexity lower than that of SD.

A prospective research direction is the derivation of an upper bound, which could be guided by the method proposed in [15] and the optimization problem in (21). An upper bound, together with the lower bound derived in this paper, would provide fundamental limits on the probability that random linear network decoding complemented by noise decoding over $\mathbb{F}_2$ will be successful. Generalization of this work to any finite field $\mathbb{F}_{2^\rho}$, for $\rho \geqslant 1$, would be equally important but challenging; this is because the sparsity of the error matrix – which greatly facilitated the probability analysis for operations over $\mathbb{F}_2$ – will be reduced when the error matrix at the output of the binary symmetric channel is converted from $\mathbb{F}_2$ to $\mathbb{F}_{2^\rho}$ in order to be used as a map of errors in coded packets over $\mathbb{F}_{2^\rho}$. The theoretical analysis of PRAC [7] would also be of interest, as it would pave the way for a rigorous comparison between noise decoding and PRAC, and would identify channel conditions for which each approach is better suited. Last but not least, existing designs that combine random linear network coding with turbo coding [25] or low-density parity check coding [26] could be revisited to account for the contribution of noise decoding to the overall performance.

APPENDIX A
PROOF OF LEMMA 2

The problem in Lemma 2 can be restated as: "Determine the number of sequences of length n' that can be formed for a given partition $\mathbf{a} \in \mathcal{A}$, if the first k' entries can take any value in the range $\{1, 2, \ldots, 2^{r'} - 1\}$ and the last $n' - k'$ entries are pre-selected integers, different from each other, that have been drawn from the same range."

An example sequence of $n' = 14$ integers for $k' = 11$ and $r' = 3$ is shown in Fig. 6(a). The vector $\mathbf{a} = (2\ 3\ 2\ 0\ \cdots\ 0\}$ describes how the 14 integers can be partitioned into groups. In this example, the last $n' - k' = 3$ entries of the sequence have contributed to the formation of $q_2 = 2$ groups of size 2 and $q_3 = 1$ group of size 3, as depicted in Fig. 6(b), resulting

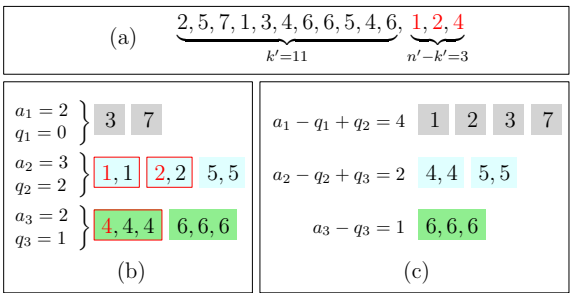

Fig. 6. (a) Sequence of $n' = 14$ integers in $\{1, 2, \ldots, 7\}$. The first $k' = 11$ integers have been randomly selected. The last $n'-k' = 3$ integers (shown in red) have been pre-selected and are different from each other. (b) Partitioning of the sequence into groups is based on the multiplicity of each entry. The sequence is formed by $a_1 = 2$ unique integers, $a_2 = 3$ groups containing two integers, and $a_3 = 2$ groups composed of three integers. The last three entries of the sequence contribute to the creation of $q_2 = 2$ of the two-element groups and $q_3 = 1$ of the three-element groups (framed in red). (c) Regrouping, after the removal of the last three entries of the sequence from the groups shown in (2), leads to the formation of 4 single-element groups, 2 two-element groups and 1 three-element group.

in vector $\mathbf{q} = (0\ 2\ 1\ 0\ \cdots\ 0\}$. In the remainder of this section, we show that the number of sequences of length n' for a given partition $\mathbf{a} \in \mathcal{A}$ can be obtained from the product of three terms summed over every $\mathbf{q} \in \mathcal{Q}(\mathbf{a})$, that is:

$$g(r', k', n', \mathbf{a}) = \sum_{\forall \mathbf{q} \in \mathcal{Q}(\mathbf{a})} \zeta_1 \zeta_2 \zeta_3. \tag{50}$$

For a given vector $\mathbf{q} = (q_1\ \cdots\ q_{n'}\}$, the number of ways that the last $n' - k'$ entries of a sequence can be distributed into q_1 single-element groups, q_2 two-element groups, and so on, where $q_1 + \ldots + q_{n'} = n' - k'$ and $q_i \geqslant 0$ for $i = 1, \ldots, n'$, is given by the multinomial coefficient:

$$\zeta_1 = \binom{n' - k'}{q_1, \ldots, q_{n'}}. \tag{51}$$

Whereas term ζ_1 is concerned with the last $n' - k'$ entries of a sequence, term ζ_2 in (50) focuses on the first k' entries and enumerates the ways they can be re-grouped after the last $n' - k'$ entries have been removed from the original groups. If $\mathbf{a}' = (a'_1\ \cdots\ a'_{n'})$ describes the partitioning of the first k' entries, $a'_1, \ldots, a'_{n'}$ can be expressed in terms of the elements of $\mathbf{a}$ and $\mathbf{q}$ as $a'_i = a_i - q_i + q_{i+1}$ for $i = 1, \ldots, n' - 1$ and $a'_{n'} = a_{n'} - q_{n'}$, as illustrated in Fig. 6(c). The multinomial coefficient can be used again to count the number of ways the first k' entries can be distributed into each group, as follows:

$$\begin{aligned}\zeta_2 &= \binom{k'}{\underbrace{1, \ldots, 1}_{a_1 - q_1 + q_2}, \underbrace{2, \ldots, 2}_{a_2 - q_2 + q_3}, \ \ldots\ , \underbrace{n', \ldots, n'}_{a_{n'} - q_{n'}}} \\ &= \frac{k'!}{(n'!)^{a_{n'} - q_{n'}} \prod_{t=1}^{n'-1} (t!)^{a_t - q_t + q_{t+1}}}. \end{aligned} \tag{52}$$

Essentially, the product $\zeta_1 \zeta_2$ provides the number of ways that the n' entries of a sequence can be partioned into a_i groups of size i, for $i = 1, ..., n'$, when $n' - k'$ of those entries are occupied by pre-selected distinct integers that contributed to the formation of q_i of the a_i groups. All possible ways of filling up the remaining $a_i - q_i$ groups with integers drawn from the range $\{1, 2, \ldots, 2^{r'} - 1\}$ are enumerated by term ζ_3 in (50). Starting with the a_1 unique entries of the sequence, we know that $a_1 - q_1$ of them occupy the first k' positions. The values of these $a_1 - q_1$ entries will be chosen from an interval of $2^{r'} - 1$ integers that excludes the $n' - k'$ pre-selected integers in the last $n' - k'$ positions. In general, $a_i - q_i$ groups of size i will be populated by integers occupying the first k' positions and taking values from an interval of $2^{r'} - 1$ integers, from which the values of the last $n' - k'$ entries and the values of the previously considered $i - 1$ groups have been removed. The total number of combinations, for $i = 1, \ldots, n'$, is:

$$\zeta_3 = \prod_{i=1}^{n'} \binom{2^{r'} - 1 - (n' - k') - \sum_{j=1}^{i-1} (a_j - q_j)}{a_i - q_i}. \tag{53}$$

Substituting ζ_1, ζ_2 and ζ_3 into (50) gives (37).

Appendix B
Proof of Lemma 3

For compactness, we drop the arguments $k - k_{\mathrm{R}}$, $n - n_{\mathrm{R}}$ and z, and write the probability of SEC-SD correctly estimating $\mathbf{E}^{(\mathrm{p})}$, given that $\mathbf{E}^{(\mathrm{p})}$ meets the SEC requirements, as $\mathbb{P}\big[\hat{\mathbf{E}}^{(\mathrm{p})} = \mathbf{E}^{(\mathrm{p})} \mid \mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, \ldots\big]$ in (38). This probability can be expressed as the product of two terms summed over all partitions in $\mathcal{A}$, that is:

$$\begin{aligned} &\mathbb{P}\Big[\hat{\mathbf{E}}^{(\mathrm{p})} = \mathbf{E}^{(\mathrm{p})} \mid \mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, \ldots\Big] = \\ &= \sum_{\forall \mathbf{a} \in \mathcal{A}} \mathbb{P}\Big[\mathrm{spark}([\mathbf{H}^{(\mathrm{p})}]^\top) \geqslant 2 \mid \ldots\Big] \cdot \\ &\cdot \mathbb{P}\Big[\hat{\mathbf{E}}^{(\mathrm{p})} = \mathbf{E}^{(\mathrm{p})} \mid \mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, \mathrm{spark}([\mathbf{H}^{(\mathrm{p})}]^\top) \geqslant 2, \ldots\Big]. \end{aligned} \tag{54}$$

The first product term in (54) represents the probability that the spark of $[\mathbf{H}^{(\mathrm{p})}]^\top$ is 2 or greater. The second term evaluates the probability that $\mathbf{E}^{(\mathrm{p})}$ will be correctly estimated, given that the SEC conditions are met and the spark of $[\mathbf{H}^{(\mathrm{p})}]^\top$ is at least 2.

As also explained in Section V-B, the first term of the product in (54) relies on (37) and is equal to:

$$\mathbb{P}\Big[\mathrm{spark}([\mathbf{H}^{(\mathrm{p})}]^\top) \geqslant 2 \mid \ldots\Big] = \frac{g(n - k, k - k_{\mathrm{R}}, n - n_{\mathrm{R}}, \mathbf{a})}{2^{(n-k)(k-k_{\mathrm{R}})}},$$

which is the fraction of realizations of $[\mathbf{H}^{(\mathrm{p})}]^\top$ that have spark 2 or greater, for a specific partition $\mathbf{a} \in \mathcal{A}$. The second term of the product in (54) can be decomposed into a product of three probabilities:

$$\begin{aligned} &\mathbb{P}\left[\hat{\mathbf{E}}^{(\mathrm{p})} = \mathbf{E}^{(\mathrm{p})} \mid \mathrm{spark}([\mathbf{H}^{(\mathrm{p})}]^\top) \geqslant 2, \mathbf{E}^{(\mathrm{p})} \in \mathcal{S}^{(\mathrm{p})}, \ldots\right] = \\ &= P_1 P_2 P_3, \end{aligned} \tag{55}$$

where P_1, P_2 and P_3 describe the probabilities of individual columns of $\hat{\mathbf{E}}^{(\mathrm{p})}$ matching the corresponding columns of $\mathbf{E}^{(\mathrm{p})}$.

Recall from Section IV and Fig. 2 that $\mathbf{E}^{(\mathrm{p})}$ is composed of $n - n_{\mathrm{R}}$ orthogonal unit columns, $b - z - n + n_{\mathrm{R}}$ random unit columns and z all-zero columns. Multiplication of the $n - n_{\mathrm{R}}$ orthogonal unit columns of $\mathbf{E}^{(\mathrm{p})}$ with $[\mathbf{H}^{(\mathrm{p})}]^\top$ copies the $n - n_{\mathrm{R}}$ columns of $[\mathbf{H}^{(\mathrm{p})}]^\top$ over to $n - n_{\mathrm{R}}$ columns of the syndrome matrix $\mathbf{S}$. This implies that $n - n_{\mathrm{R}}$ of the columns of $\mathbf{S}$ form a submatrix that is described by the same partitioning vector $\mathbf{a}$ as $[\mathbf{H}^{(\mathrm{p})}]^\top$. Let $\mathbf{S}_{*,j}$ be a column of this submatrix

and i be the number of columns of $[\mathbf{H}^{(\mathrm{p})}]^\top$ that match $\mathbf{S}_{*,j}$. In line with (36b), the probability that the syndrome decoder will select the column of $[\mathbf{H}^{(\mathrm{p})}]^\top$ that produces $\hat{\mathbf{E}}^{(\mathrm{p})}_{*,j} = \mathbf{E}^{(\mathrm{p})}_{*,j}$ is thus $1/i$. As ia_i of $n-n_{\mathrm{R}}$ columns of $\mathbf{S}$ form a_i groups of size i, this process will be repeated ia_i times, for $i = 1, \ldots, n-n_{\mathrm{R}}$. The probability of correctly guessing $n-n_{\mathrm{R}}$ columns of $\mathbf{E}^{(\mathrm{p})}$ from those $n-n_{\mathrm{R}}$ columns of $\mathbf{S}$ is:

$$P_1 = \prod_{i=1}^{n-n_{\mathrm{R}}} \left(\frac{1}{i}\right)^{ia_i}.$$

The second product term in (55), denoted by P_2, represents the probability of successfully estimating the $b-z-n+n_{\mathrm{R}}$ random unit columns of $\mathbf{E}^{(\mathrm{p})}$. Let $\mathbf{S}_{*,j}$ be a column of the syndrome matrix that has been generated by a random unit column of $\mathbf{E}^{(\mathrm{p})}$. Since ia_i columns of $[\mathbf{H}^{(\mathrm{p})}]^\top$ form a_i groups of size i, we infer that $\mathbf{S}_{*,j}$ would be a copy of one of those ia_i columns with probability $ia_i/(n-n_{\mathrm{R}})$. Based on $\mathbf{S}_{*,j}$, the syndrome decoder will output $\hat{\mathbf{E}}^{(\mathrm{p})}_{*,j} = \mathbf{E}^{(\mathrm{p})}_{*,j}$ with probability $1/i$, if i columns of $[\mathbf{H}^{(\mathrm{p})}]^\top$ are equal to $\mathbf{S}_{*,j}$. The joint probability of $\mathbf{S}_{*,j}$ matching a column of $[\mathbf{H}^{(\mathrm{p})}]^\top$ that is repeated i times and enabling the syndrome decoder to produce $\hat{\mathbf{E}}^{(\mathrm{p})}_{*,j} = \mathbf{E}^{(\mathrm{p})}_{*,j}$ is thus $(ia_i/(n-n_{\mathrm{R}}))(1/i) = a_i/(n-n_{\mathrm{R}})$. In general, $\mathbf{S}_{*,j}$ could be a copy of a unique column of $[\mathbf{H}^{(\mathrm{p})}]^\top$ or a column that is repeated any number of times up to $n-n_{\mathrm{R}}$. Therefore, the overall probability that the syndrome decoder will correctly guess a random unit column of $\mathbf{E}^{(\mathrm{p})}$ is $(a_1 + \ldots + a_{n-n_{\mathrm{R}}})/(n-n_{\mathrm{R}})$. As every random unit column of $\mathbf{E}^{(\mathrm{p})}$ is generated independently of the other columns, the probability of successfully estimating all of the $b-z-n+n_{\mathrm{R}}$ random unit columns is given by:

$$P_2 = \left(\frac{\sum_{i=1}^{n-n_{\mathrm{R}}} a_i}{n-n_{\mathrm{R}}}\right)^{b-z-n+n_{\mathrm{R}}}.$$

The third term of the product in (55) is concerned with the probability of correctly estimating an all-zero column of $\mathbf{E}^{(\mathrm{p})}$, which is $P_3 = 1$ based on (35a) and (36a). Substituting P_1, P_2 and P_3 into (55) completes the expression of the second product term in (54) and leads to (38).

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] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM*, Vancouver, Canada, Sep. 1998.

[3] J. Cloud, F. D. P. Calmon, W. Zeng, G. Pau, L. M. Zeger, and M. Médard, "Multipath-TCP with network coding for mobile devices in heterogeneous networks," in *Proc. IEEE 78th Veh. Technol. Conf.*, Las Vegas, NV, USA, Sep. 2013.

[4] Y. Cui, L. Wang, X. Wang, H. Wang, and Y. Wang, "FMTCP: A fountain code-based multipath transmission control protocol," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 465–478, Apr. 2015.

[5] G. Giambene, D. K. Luong, T. de Cola, V. A. Le, and M. Muhammad, "Analysis of a packet-level block coding approach for terrestrial-satellite mobile systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8117–8130, Aug. 2019.

[6] J. Chen, H. Wu, P. Yang, F. Lyu, and X. Shen, "Cooperative edge caching with location-based and popular contents for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10 291–10 305, Sep. 2020.

[7] G. Angelopoulos, M. Médard, and A. P. Chandrakasan, "Harnessing partial packets in wireless networks: Throughput and energy benefits," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 694–704, Feb. 2017.

[8] J. A. Cabrera, G. Nguyen, D. E. Lucani, M. V. Pedersen, and F. H. P. Fitzek, "Taking the trash back in: Practical joint channel and network coding for improving IEEE 802.11 networks," in *Proc. European Wireless*, Dresden, Germany, May 2017.

[9] K. D. Irianto, J. A. Cabrera, G. T. Nguyen, H. Salah, and F. H. P. Fitzek, "S-PRAC: Fast partial packet recovery with network coding in very noisy wireless channels," in *Proc. Wireless Days*, Manchester, UK, Apr. 2019.

[10] 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.

[11] J. Xie, W. Wu, and Z. Zhang, "Revisiting partial packet recovery in 802.11 wireless LANs," in *Proc. 9th ACM Int. Conf. on Mob. Systems, Appl. and Services (MobiSys)*, Washington, DC, USA, Jun. 2011.

[12] K. R. Duffy, J. Li, and M. Médard, "Capacity-achieving guessing random additive noise decoding," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4023–4040, Jul. 2019.

[13] GRAND™: Guessing random additive noise decoding. Massachusetts Institute of Technology. Accessed: March 31, 2025. [Online]. Available: https://granddecoder.mit.edu

[14] I. Chatzigeorgiou and D. Savostyanov, "Guessing random additive noise decoding of network coded data transmitted over burst error channels," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 12 842–12 857, Sep. 2024.

[15] R. Su, Q. T. Sun, M. Deng, Z. Zhang, and J. Yuan, "GRAND-assisted random linear network coding in wireless broadcasts," in *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, Athens, Greece, Jul. 2024.

[16] J. Jin, B. Li, and T. Kong, "Is random network coding helpful in WiMAX?" in *Proc. IEEE 27th Conf. on Comp. Commun. (INFOCOM)*, Phoenix, AZ, USA, Apr. 2008.

[17] S. Butler and P. Karasik, "A note on nested sums," *Journal of Integer Sequences*, vol. 13, no. 4, pp. 1–8, 2010.

[18] J. H. van Lint and R. M. Wilson, *A course in combinatorics*, 2nd ed. Cambridge University Press, 2001.

[19] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proc. Natl. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, Mar. 2003.

[20] M. Elad, *Sparse and redundant representations: From theory to applications in signal and image processing*. New York, USA: Springer, 2010.

[21] R. C. Singleton, "Maximum distance q-nary codes," *IEEE Trans. Inf. Theory*, vol. 10, no. 2, pp. 116–118, Apr. 1964.

[22] 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.

[23] J. D'Errico. (2018, March) Partitions of an integer, v.1.21. MATLAB Central File Exchange, Retrieved October 11, 2024. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/12009-partitions-of-an-integer

[24] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution*. John Wiley & Sons, 2011.

[25] C. Hausl and J. Hagenauer, "Iterative network and channel decoding for the two-way relay channel," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Istanbul, Turkey, Jun. 2006.

[26] X. Bao and J. Li, "A unified channel-network coding treatment for user cooperation in wireless ad-hoc networks," in *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, Seattle, WA, USA, Jul. 2006.

Ioannis Chatzigeorgiou (Senior Member, IEEE) received the Dipl.Ing. degree in Electrical Engineering from Democritus University of Thrace, Greece, the M.Sc. degree in Satellite Communication Engineering from the University of Surrey, UK, and the Ph.D. degree from the University of Cambridge, UK. He is currently a Senior Lecturer at Lancaster University, UK. Prior to his appointment, he held postdoctoral positions at the University of Cambridge and the Norwegian University of Science and Technology, supported by the Engineering and Physical Sciences Research Council (EPSRC) and the European Research Consortium for Informatics and Mathematics (ERCIM), respectively. His research interests include signal processing, coding theory and performance analysis of communication systems.