End-to-End Error-Correcting Codes on Networks with Worst-Case Bit Errors

Qiwen Wang and Sidharth Jaggi

Abstract

In highly dynamic wireless networks, communications face several challenges. In the first place, noise levels between nodes might be difficult to predict *a priori*. Besides, a malicious omniscient attacker, with knowledge of the network topology and observation of all transmissions, can choose where to inject errors to result in the worst corruption at the receiver. Considering that transmissions are usually in bits and hardware in wireless networks usually uses modulation schemes in powers of two, *e.g.* BPSK, QPSK, 16-QAM, 64-QAM etc., to address the above problem, we study coding for networks experiencing worst-case bit errors, and with network codes over binary extension fields. We demonstrate that in this setup prior network error-correcting schemes can be arbitrarily far from achieving the optimal network throughput. A new *transform metric* for errors under the considered model is proposed. Using this metric, we replicate many of the classical results from coding theory. Specifically, new *Hamming-type*, *Plotkin-type*, and *Elias-Bassalygo-type* upper bounds on the network capacity are derived. A commensurate lower bound is shown based on *Gilbert-Varshamov(GV)-type* codes for error-correction. The GV codes used to attain the lower bound can be non-coherent, that is, they require neither prior knowledge of the network topology nor network coding kernels. We also propose a computationally-efficient concatenation scheme. The rate achieved by our concatenated codes is characterized by a *Zyablov-type* lower bound. We provide a *generalized minimum-distance decoding* algorithm which decodes up to half the minimum distance of the concatenated codes. The end-to-end nature of our design enables our codes to be overlaid on the classical distributed random linear network codes. Furthermore, the potentially intensive computation at internal nodes for the link-by-link error-correction is un-necessary based on our design.[1]

I. INTRODUCTION

A source wishes to transmit information to a receiver over a network with noisy links. Such a communication problem faces several challenges.

[1] This paper was presented in part at 2011 IEEE Information Theory Workshop (ITW) [1].

The primary challenge we consider is that in highly dynamic environments such as wireless networks, noise levels on each link might vary significantly across time, and hence be hard to estimate well. For example, in a factory where machines are connected and communicated via the Internet of Things (IoT) built on Wi-Fi, a truck passing through may corrupt parts of the transmission on some links, where the exact locations of corruptions is hard to predict accurately. Another scenario with an omniscient adversary in a network can also cause similar challenge of variable noise levels in links. Specifically, the omniscient adversary knows the network topology, can observe all transmissions, and inject bits into the network that depend on transmitted messages, subject only to a global jamming power constraint. This issue of variable link noise levels exacerbates at least two other challenges that had been considered settled by prior work.

Firstly, since noise exists in the network, network coding might be dangerous. This is because all nodes mix information, so even a small number of bit-flips in transmitted packets may end up corrupting all the information flowing in the network, causing decoding errors. Prior designs for network error-correcting codes exist (*e.g.* [2], [3]) but as we shall see they are ineffective against bit errors in a highly dynamic noise setting. In particular, one line of work considers either packets (*e.g.* [3], [4], [5]) or symbols over a large field (*e.g.* [6], [7], [8], [9], [10], [11]) in the network as either correct or corrupted. Hence, if a packet/symbol encounters even a single bit-flip, these schemes treat it as corresponding to the entire packet/symbol being corrupted. As a result, these schemes may achieve rates that are too pessimistic – the fundamental problem is that the codes are defined over large alphabets (including packet-level), and hence are poor at dealing with bit-level errors. However, because digital transmissions are usually conducted in bits, it is important to consider and correct bit-level errors. Another line of work (*e.g.* [2]) overlays network coding on link-by-link error correction, but requires accurate foreknowledge of the noise levels on each link to have good performance, which is raised above as a primary challenge in dynamic communication environments.

Secondly, in dynamic settings, the coding operations of nodes in the network may be unknown *a priori*. Under the bit error model we consider, the *transform-estimation* strategy of Ho *et al.* [12] does not work, since any headers pre-specified can also end up being corrupted.

This work attempts to settle these challenges. We consider simultaneously the *reliability* and *universality* issues for random linear network coding. Specifically, we design end-to-end distributed schemes that allow reliable network communications in the presence of worst-case network noise, wherein the erroneous bits can be arbitrarily distributed in different network packets with only a constraint on the total number of bit errors. The constraint is that no more than a certain fraction of all bits transmitted in the whole network can get flipped. With internal network nodes just carrying out linear network coding, error-correction is only accomplished by the receiver(s), who are also able to estimate the linear transform imposed on the source's data by the network.[2] Because we consider bit errors, moreover, network hardware usually use modulation schemes in powers of two (*e.g.* BPSK, QPSK, 16-QAM, 64-QAM etc.), we consider linear network codes over a binary extension field in this work. Our results can in general translate to q-ary symbol-level errors where q is a prime number and network codes over extension fields of q.

As noted above, our codes are robust to a wide variety of network conditions – whether bit errors are evenly distributed among all packets, or adversarially concentrated among just a few packets, our codes can detect and correct errors up to a network-wide bound on the total number of errors. Naïve implementations of prior codes (for instance, of link-by-link error-correcting codes [2]) that try to correct for worst-case network conditions may result in much lower rates (see the example in Section III-C). Thus the naturally occurring diversity of network conditions works in our favour rather than against us.

Also, even though our codes correct bit errors rather than errors over larger symbol fields, the end-to-end nature of our design enables our codes to be overlaid on classical linear network codes over finite fields, for instance, the random linear network codes of Ho *et al* [12]. Further, we free internal nodes from having to implement potentially computationally intensive link-by-link error-correction. This property might be useful in networks such as sensor networks, where the internal nodes do not have sufficient computational power to perform link-layer error correction. One application might be, again, IoT where machines/devices are monitored by sensors, where measurements of some quantity (temperature, pressure, *etc.*) needed to be communicated and sent to controllers.

The main tool used to prove our results is a *transform metric* that might be of independent interest (see Section IV for details). It is structurally similar to the rank-metric used by Silva

[2] As is common in coding theory, the upper and lower bounds on error-correction we prove also directly lead to corresponding bounds on error-detection – for brevity we omit discussing error-detection in this work.

et al. [4], but has important differences that give our codes the power of universal robustness against binary noise, as opposed to the packet-based noise considered in [3], [4], [5] *etc.*

A. Previous Work on Network Error Correction

In general, there are two lines of prior work in the literature on network error correction. One approach [4], [5], [6] considers correcting corruptions in packet-level or symbols from a large field; the other [2] overlays network coding on link-by-link error correction. In 2002, Borade [13] proved an information-theoretic outer bound on the rate region for networks with independent and synchronous noisy channels. Simultaneously, Cai and Yeung [6] considered packet-wise worst-case errors and derived generalized Hamming upper bounds and Gilbert-Varshamov lower bounds for networks. In 2003, the algebraic network codes of Kötter and Médard [14] and the random linear network codes by Ho *et al.* [12] are resilient to node/edge failures that do not change the mincut, which can be considered as packet erasures. In 2006, Song, Yeung and Cai [2] proposed a Shannon-type separation theorem for network coding and channel coding in networks consisting of independent channels with random noise, where the channels are not restricted to synchronous ones. In [5], Jaggi *et al.* proposed network codes against adversaries with different attacking power (different numbers of links that the adversaries can eavesdrop/jam). Those schemes are distributed, rate-optimal, and request polynomial design and implementation time. In [15], the authors proposed a layered scheme for improving throughput over wireless mesh networks which has a similar flavor as our work, where the routers (internal nodes) let erroneous bits through without compromising end-to-end reliability. Silva, Kötter and Kschischang [3], [4], [16], [17], [18] used rank-metric codes and subspace codes for networks with packet errors. Following the subspace codes by Kötter and Kschischang, in [19] the authors investigated the coding theory in projective space, and derived bounds corresponding to those by Johnson, Delsarte and Gilvert-Varshamov in the classical coding theory. The works by Yang *et al.* [9], [10] investigated different weight measures for network error correction (with packet errors) and derived counterparts of the Hamming bound, the Singleton bound and the Gilbert-Varshamov bound in the classical coding theory. Although the settings of our work (bit-level errors) are different from [9], [10], the spirits are similar – refining the bounds in the classical coding theory with novel distance metrics designed for network error correction. In [20], the authors proposed an extension of subspace codes [3] capable of correcting certain combinations of dimension errors and symbol errors/erasures in noncoherent networks. A more recent work [21] shares similar spirit as this work – relating rank metrics to Hamming weights and investigating counterparts of classical coding theory in linear network coding. However, the error is modeled at the receiver (sink) instead of resulting from a linear transformation induced by the network as in this work.

II. Model

A. Network model

We model our network by a directed acyclic multigraph[3], denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of edges. A single source node $s \in \mathcal{V}$ and a set of sinks $\mathcal{D} \subseteq \mathcal{V}$ are pre-specified in $\mathcal{V}$. We denote $|\mathcal{E}|$ and $|\mathcal{D}|$, respectively the number of edges and sinks in the network, by E and D. A directed edge e leading from node u to node v can be represented by the vector (u, v), where u is called the *tail of* e and v is called the *head of* e. In this case e is called an *outgoing edge of* u and an *incoming edge of* v.

The capacity of each edge is one packet – a length-n vector over a finite field $\mathbb{F}_{2^m}$ – here n and m are design parameters to be specified later. Multiple edges between two nodes are allowed – this allows us to model links with different capacities.[4] As defined in [22], the *network (multicast) capacity*, denoted C, is the minimum over all sinks $t \in \mathcal{D}$ of the mincut of $\mathcal{G}$ from the source s to the sink t. Without loss of generality, we assume there are C edges outgoing from s and incoming edges to t for all sinks $t \in \mathcal{D}$. [5]

B. Code model

The source node s wants to *multicast* a message S to each sink $t \in \mathcal{D}$. To simplify notation, we first consider the scenario with just a single sink, then discuss generalization to the multi-sink case in Section VI-A. The notational conventions are as follows. Matrices are denoted by boldface symbols. A zero matrix is denoted by $\mathbf{0}$ when its dimension is unambiguous. Sets are

[3] Our model also allows non-interfering broadcast links in a wireless network to be modeled via a directed hypergraph – for ease of notation we restrict ourselves to just graphs.

[4] By appropriate buffering and splitting edges into multiple edges, any network can be approximated into such a network with unit capacity edges.

[5] In cases where the number of outgoing edges from s (or the number of incoming edges to t) is not C, we can add a *source super-node* (or *sink super-node*) with C noiseless edges connecting to the original source (or sink) of the network. The change in the number of edges and probability of error on each edge are small compared to those of the original network, so our analysis essentially still applies.

denoted by calligraphic symbols, such as $\mathcal{X}$. The cardinality of a set $\mathcal{X}$ is denoted by $|\mathcal{X}|$. All logarithms in this work are to the base 2, and we use $H(p)$ to denote the *binary entropy function* $-p \log p - (1-p)\log(1-p)$.

Random linear network coding: All internal nodes in the network perform *random linear network coding* [12] over a finite field $\mathbb{F}_{2^m}$. Specifically, each internal node takes uniformly random linear combinations of each incoming packet to generate outgoing packets. That is, let e' and e index incoming and outgoing edges from a node v. The *linear coding coefficient from* e' *to* e is denoted by $f_{e',e} \in \mathbb{F}_q$. Let $\mathbf{Y}_e$ denote the packet (length-n vector over $\mathbb{F}_{2^m}$) transmitted on the edge e. Then $\mathbf{Y}_e = \sum f_{e',e}\mathbf{Y}_{e'}$, where the summation is over all edges e' incoming to the node v, and all arithmetic is performed over the finite field $\mathbb{F}_{2^m}$.

Mapping between $\mathbb{F}_2$ and $\mathbb{F}_{2^m}$: As mentioned above, the network codes are operated over $\mathbb{F}_{2^m}$. However, the noise we consider in this work happens in bit level. Hence, before introducing the noise model, we introduce a mapping between $\mathbb{F}_{2^m}$ and $\mathbb{F}_2$ to link the network codes and bit-level transmission and errors.

There are conventional representations of elements from$\mathbb{F}_{2^m}$ by bits in vector/matrix forms. For example, one conventional vector representation of elements from $\mathbb{F}_{2^m}$ uses m bits, *e.g.* $\mathbb{F}_8 = \{0, 1, 2, \dots, 7\} = \{000, 001, 010, \dots, 111\}$. However, it is not straightforward to represent multiplications with this vector form, except for a multiplication table I. There is another conventional matrix presentation of finite field element which preserves the linear multiplication, *e.g.* see Section 2.5 in [23]. However, this representation needs m^2 instead of m bits to describe an element, leading to more bits transmitted hence reduces throughput.

We use the mappings given in Lemma 1 from [24], which maps multiplication ax over $\mathbb{F}_{2^m}$ to multiplication $\mathbf{A}\vec{x}$ over $\mathbb{F}_2$, where $\mathbf{A}$ is an $m \times m$ binary matrix and $\vec{x}$ is an $m \times 1$ binary vector. Specifically, the mapping maps the second element in the multiplication to a binary vector by the conventional way, with the entries being the coefficients of its polynomial form. For the first element in the multiplication, the ith column of matrix $\mathbf{A}$ is the vector form of a times the element corresponding to $\mathbf{e}_i$ – the unit vector with a 1 at the ith entry. For example, the elements in $\mathbb{F}_8$ are represented as in Fig. 1. One can check that the matrix-vector multiplications between matrices in Fig. 1 and vectors in Table I results in the same vectors as in the multiplication table. For brevity we refer to [24] for the proof. The intuition is that we choose the unit vectors as a basis for $\mathbb{F}_{2^m}$ as a vector space over $\mathbb{F}_2$. Multiplication by an element from $\mathbb{F}_{2^m}$ is then

TABLE I: Multiplication table for $\mathbb{F}_8$.

	000	001	010	011	100	101	110	111
000	000	000	000	000	000	000	000	000
001	000	001	010	011	100	101	110	111
010	000	010	100	110	011	001	111	101
011	000	011	110	101	111	100	001	010
100	000	100	011	111	110	010	101	001
101	000	101	001	100	010	111	011	110
110	000	110	111	001	101	011	010	100
111	000	111	101	010	001	110	100	011

equivalent to a linear transformation (a matrix) in that vector space.

$$0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad 1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad 2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$3 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \qquad 4 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad 5 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$6 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \qquad 7 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Fig. 1: Matrix representation of the elements in $\mathbb{F}_8$.

We use the mapping above to transform packets sent in the network and network codes over $\mathbb{F}_{2^m}$ to $\mathbb{F}_2$.

More specifically, a bijection is defined from each symbol (from $\mathbb{F}_{2^m}$) of each packet transmitted on each edge, to a corresponding length-m bit-vector. For ease of notation henceforth, for each edge e and each $i \in \{1, \dots, n\}$, we use $\mathbf{Y}_e$ and $\mathbf{Y}_e(i)$ solely to denote respectively the length-nm and length-m binary vectors resulting from the bijection operating on packets and their ith symbols, rather than the original analogues over $\mathbb{F}_{2^m}$ traversing that edge e. Separately, each linear coding coefficient $f_{e',e}$ at each node is mapped to a specific $m \times m$ binary matrix $F_{e',e}$. The linear mixing at each node is then taken over the binary field – each length-m binary vector $\mathbf{Y}_{e'}(i)$ (corresponding to the binary mapping of the ith symbol of the packet $\mathbf{Y}_{e'}$ over the field $\mathbb{F}_{2^m}$) equals $\sum F_{e',e}\mathbf{Y}_{e'}(i)$. In what follows, depending on the context, we use the mapping to switch between the scalar (over $\mathbb{F}_{2^m}$) and matrix (over $\mathbb{F}_2$) forms of the network codes' linear coding coefficients, and to switch between the scalar (over $\mathbb{F}_{2^m}$) and vector (over $\mathbb{F}_2$) forms of each symbol in each packet.

Noise: We consider worst-case noise in this work, wherein an arbitrary number of bit-flips can happen in any transmitted packet, subject to the constraint that no more that a fraction of p bits over all transmitted packets are flipped. From the above discussion about mapping between $\mathbb{F}_{2^m}$ and $\mathbb{F}_2$, each transmitted symbol is converted into a length-m binary vector. Hence, each packet of n symbols from $\mathbb{F}_{2^m}$ are transmitted in nm bits. To present the linear transformation, a packet, when presented in binary form, is an $m \times n$ binary matrix. Hence, we present the bit errors in the whole network as a binary matrix. Specifically, the *noise matrix* $\mathbf{Z}$ is an $Em \times n$ binary matrix with at most $pEmn$ nonzero entries which can be arbitrarily distributed. In particular, the $m(i-1)+1$ through the mi rows of $\mathbf{Z}$ represent the bit-flips to the packet $\mathbf{Y}_{e_i}$ transmitted on the ith link. If the $((k-1)m+j)$th bit of the length-mn binary vector is flipped, i.e. the jth bit of the kth symbol over $\mathbb{F}_{2^m}$ in $\mathbf{Y}_{e_i}$ is flipped, correspondingly, the $(m(i-1)+j,k)$ bit in $\mathbf{Z}$ equals 1, else it equals 0. Thus the noise matrix $\mathbf{Z}$ represents the noise pattern of the network. An example of how $\mathbf{Z}$ models the bit-flips on the links is shown in Fig. 2. To model the noise as part of the linear transform imposed by the network, we add an artificial super-node s' connected to all the edges in the network, injecting noise into each packet transmitted on each edge in the network according to entries of the noise matrix $\mathbf{Z}$.

Source: The source has a set of 2^{RCmn} messages $S \in \{1,\ldots,2^{RCmn}\}$ it wishes to communicate to each sink, where R is the *rate* of the source. Corresponding to each message S it generates a *codeword* $\mathbf{X}(S)$ using some *encoder* (to make notation easier we usually do not explicitly reference the parameter S and instead refer simply to $\mathbf{X}$). This $\mathbf{X}$ is represented by either a $C \times n$ matrix over $\mathbb{F}_{2^m}$, or alternatively a $Cm \times n$ matrix over $\mathbb{F}_2$ (similar as the noise matrix $\mathbf{Z}$). Each row of the matrix in $\mathbb{F}_{2^m}$ or each m rows in the matrix in $\mathbb{F}_2$ corresponds to a packet transmitted over a distinct edge leaving the source node.

Receiver(s): Each sink t receives a batch of C packets. Similarly to the source, it organizes the received packets into a matrix $\mathbf{Y}$, which can be equivalently viewed as a $C \times n$ matrix over $\mathbb{F}_{2^m}$ or a $Cm \times n$ binary matrix. Each sink t decodes the message $\hat{S}$ from the received matrix

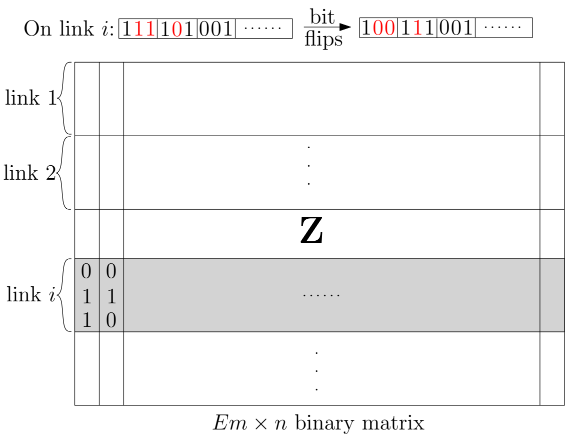

Fig. 2: An example of the noise matrix Z. The $m \times n$ sub-matrix consisting the $[(i-1)m+1]$th row up to the imth row represents the bit-flips on link i in the network.

$\mathbf{Y}$ with some *decoder*.

Transfer matrix and Impulse response matrix: Having defined the linear coding coefficients of internal nodes, the packets transmitted on the incoming edges of each sink t can inductively be calculated as linear combinations of the packets on the outgoing edges of s. We denote the $C \times C$ *transfer matrix* from outgoing edges of s to incoming edges of t by $\mathbf{T}$, over the finite field $\mathbb{F}_{2^m}$. Alternatively, using the mapping from $\mathbb{F}_{2^m}$ to $\mathbb{F}_2$ described above, $\mathbf{T}$ may be viewed alternatively as a $Cm \times Cm$ binary matrix.

Definition 1. *Let $\mathcal{X} \subseteq \{\mathbb{F}_2^{Cm\times n}\}$ be a codebook for the worst-case binary-error network channel, the transformed codebook $\mathbf{T}\mathcal{X}$ is obtained by multiplying every codeword in $\mathcal{X}$ by $\mathbf{T}$.*

We similarly define $\hat{\mathbf{T}}$ to be the *impulse response matrix*, which is the transfer matrix from the imaginary source node s' we discussed when introducing the noise – who injects errors into all edges – to the sink t. Note that $\mathbf{T}$ is a sub-matrix of $\hat{\mathbf{T}}$, composed specifically of the C columns corresponding to the C outgoing edges of s. In this work, we require that every $C \times C$ sub-matrix of $\hat{\mathbf{T}}$ "sitting" on a min-cut, *i.e.* transfer matrix from any min-cut to the sink, is invertible. As noted in, for instance, [12], [14] this happens with high probability for random linear network codes. Alternatively, deterministic designs of network error-correcting codes [6] also have this property.

C. Worst-case binary-error network channel

Using the above definitions the network can thus be abstracted by the equation (1) below as a *worst-case binary-error network channel.*

$$\mathbf{Y} = \mathbf{T}\mathbf{X} + \hat{\mathbf{T}}\mathbf{Z}. \tag{1}$$

Similar equations have been considered before (for instance in [3], [6], [5]) – the key difference in this work is that we are interested in $\mathbf{Z}$ matrices which are fundamentally best defined over the binary field, and hence, when needed, transform the other matrices in equation (1) also into binary matrices.

Performance of code: The source encoder and the decoder(s) at sink(s) together comprise *worst-case binary-error-correcting network channel codes*. A *good* worst-case binary-error-correction network channel code has the property that, for all messages S, and noise patterns $\mathbf{Z}$ with at most $pEmn$ bit-flips, the estimated message at the decoder $\hat{S} = S$. A rate R is said to be *achievable* for the worst-case binary-error channel if, for all sufficiently large n, there exists a good code with rate R. (As we shall see in Section III, the higher order terms of the achievable rate R is independent of the parameter m, *i.e.* the alphabet size.)

In the network channel (1), denote the set of noise matrices by $\mathcal{Z} = \{\mathbf{Z} \in \mathbb{F}_2^{Em\times n} : W_{\text{Hamming}}(\mathbf{Z}) \leq pEmn\}$, and denote the codebook by $\mathcal{X}$ where the codewords are matrices chosen from $\mathbf{X} \in \mathbb{F}_2^{Cm\times n}$. Let $\Delta_{\hat{\mathbf{T}}\mathcal{Z}} = \{\hat{\mathbf{T}}\mathbf{Z}_1 - \hat{\mathbf{T}}\mathbf{Z}_2 : \mathbf{Z}_1, \mathbf{Z}_2 \in \mathcal{Z}\}$ and $\Delta_{\mathbf{T}\mathcal{X}} = \{\mathbf{T}\mathbf{X}_1 - \mathbf{T}\mathbf{X}_2 : \mathbf{X}_1, \mathbf{X}_2 \in \mathcal{X}\}$, the codebook $\mathcal{X}$ can correct any noise pattern in $\mathcal{Z}$ if and only if $\Delta_{\mathbf{T}\mathcal{X}} \cap \Delta_{\hat{\mathbf{T}}\mathcal{Z}} = \mathbf{0}^{Cm\times n}$, where $\mathbf{0}^{Cm\times n}$ denotes the zero matrix of dimension $Cm \times n$.

Specifically, one necessary condition for $\mathcal{X}$ to be able to correct $\mathcal{Z}$ is that $|\mathbf{T}\mathcal{X}| \cdot |\hat{\mathbf{T}}\mathcal{Z}| \leq 2^{Cmn}$, which leading to the Hamming-type upper bound on the code rate. On the other hand, if $|\mathbf{T}\mathcal{X}| \cdot |\Delta_{\hat{\mathbf{T}}\mathcal{Z}}| \leq 2^{Cmn}$, it is sufficient to say that there exists a codebook $\mathcal{X}$ which can correct $\mathcal{Z}$ – this leads to the Gilbert-Varshamov(GV)-type lower bound on the code rate. The main challenge here is to bound the sizes of $\hat{\mathbf{T}}\mathcal{Z}$ and $\Delta_{\hat{\mathbf{T}}\mathcal{Z}}$, which are linear transformations of Hamming balls of binary matrices. Hence, we introduce a *transform metric* in Section IV, using which we derive upper and lower bounds (including Hamming- and GV-type bounds) on the transmission rate for the network channel, which are summarized In Section III below.

Although the network channel (1) results from modeling of binary errors in networks which conduct random linear network coding, the abstracted out “channels with linear transformations” and corresponding bounds on code rates might be of interest to other applications as well.

III. Main Results

In this section, we state the main results of this paper, and discuss comparisons with some previous results. Our converses and achievable rates can be viewed as counterparts of the classical coding theory, by using a special metric (introduced in Section IV) for the worst-case binary-error network channel. All the proofs are deferred to Section V and VI.

A. Converses

Theorem 1 (Hamming-Type Bound)**.** *For all p less than $\frac{C}{E}$, an upper bound on the achievable rate of any code over the worst-case binary-error channel is $1 - H\left(\frac{E}{C}p\right) + o\left(\frac{\log(Cmn+1)}{Cmn}\right)$.*

Theorem 2 (Plotkin-Type Bound)**.**

1) For networks with $E \geq 2C$,

i. for all p less than $\left(1-\frac{C}{E}\right)\frac{C}{E}$, an upper bound on the achievable rate of any code over the worst-case binary-error network channel is $1 - \frac{E^2}{CE-C^2}p$;

ii. for all p greater than $\left(1-\frac{C}{E}\right)\frac{C}{E}$, the asymptotic rate achieved by any code over the worst-case binary-error network channel is 0.

2) For networks with $E < 2C$,

i. for all p less than $1/4$, an upper bound on the achievable rate of any code over the worst-case binary-error network channel is $1-4p$;

ii. for all p greater than $1/4$, the asymptotic rate achieved by any code over the worst-case binary-error network channel is 0.

Theorem 3 (Elias-Bassalygo-Type Bound)**.** *For all p less than $\frac{C}{E}\left(1-\frac{C}{E}\right)$, an upper bound on the achievable rate of any code over the worst-case binary-error network channel is $1 - H\left(\frac{E}{2C}(1-\sqrt{1-4p})\right) + o\left(\frac{\log(Cmn+1)}{Cmn}\right)$.*

B. Achievability

Theorem 4 (Gilbert-Varshamov-Type Bound)**.**

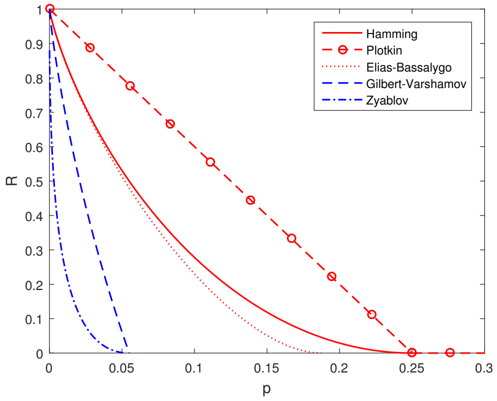

Fig. 3: The plot of main results in the senario for a diamond network where $E = 8$ and $C = 4$.

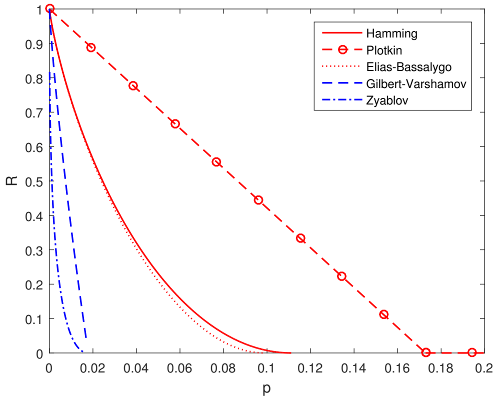

Fig. 4: The plot of main results in the senario for the butterfly network where $E = 9$ and $C = 2$.

1) *Coherent GV-type network codes achieve a rate of at least* $1 - \frac{E}{C}H(2p) - o\left(\frac{\log(2pEmn+1)}{Cmn}\right)$.
2) *Non-coherent GV-type network codes achieve a rate of at least* $1 - \frac{E}{C}H(2p) - o\left(\frac{\log(2pEmn+1)+CEm}{Cmn}\right)$.

Theorem 5 (Zyablov-Type Bound)**.** *Concatenation network codes achieve a rate of at least*

$$\max_{0<r<1-\frac{E}{C}H(2p)} r \cdot \left(1 - \frac{2p}{H^{-1}\left(\frac{C}{E}(1-r)\right)}\right).$$

In Fig. 3 and Fig. 4 we plot our results for two different networks.

Remark: If we set $C = E = m = 1$, *i.e.,* the classical point-to-point worst-case binary-error channel, all our bounds in Theorems 1-5 reduce to classical Hamming bound, Plotkin bound, Elias-Bassalygo bound, Gilbert-Varshamov bound, and Zyablov bound.

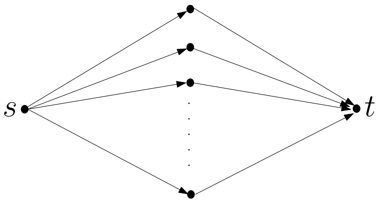

Fig. 5: A network with C parallel paths from the source to the destination. Each internal node performs random linear network coding.

C. Motivating Example

We demonstrate via an example that in networks with worst-case bit-errors, prior schemes have inferior performance compared to our scheme. In Figure 5, the network has C paths with a total of $2C$ links that might experience worst-case binary-errors ($C \geq 2$).

Benchmark 1: If link-by-link error-correction[6] is applied as in [2], *every* link is then required to be able to correct $2Cpmn$ *worst-case* bit-flip, since all the bit-errors may be concentrated in any single link. Using GV codes ([25], [26]) a rate of $1 - H(4Cp)$ is achievable on each link, and hence the overall rate scales as $C(1 - H(4Cp))$. As C increases without bound, the throughput thus actually goes to zero. The primary reason is that every link has to prepare for the worst-case number of bit-flips aggregated over the entire network. However in large networks, the total number of bit-flips in the worst-case might be too much for any single link to be able to tolerate.

Benchmark 2: Consider now a more sophisticated scheme, combining link-by-link error correction with end-to-end error-correction as in [3]. Suppose each link can correct $\frac{2Cpmn}{k}$ worst-case bit-flips, where k is a parameter to be determined such that the rate is optimized. Then at most k links will fail. Overlaying an end-to-end network error-correcting code as in [3] with link-by-link error-correcting codes such as GV codes (effectively leading to a concatenation-type scheme) leads to an overall rate of $(C - 2k)\left(1 - H(\frac{4Cp}{k})\right)$. For large C, this is better than benchmark 1 since interior nodes no longer attempt to correct *all* worst-case errors and hence can operate at higher rates – the end-to-end code corrects the errors on those links that do experience errors. Nonetheless, as we observe below, our scheme still outperforms this scheme,

[6] Since interior nodes might perform network coding, naïve implementations of end-to-end error-correcting codes are not straightforward – indeed – that is the primary goal of our constructions.

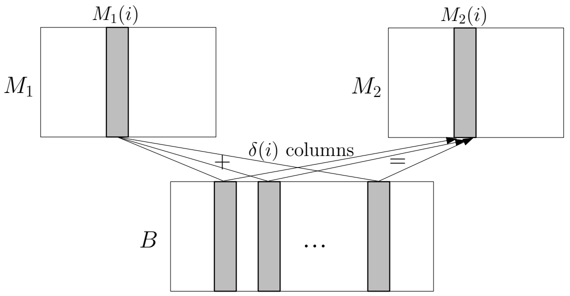

Fig. 6: Transform metric: the minimal number of columns of $\mathbf{B}$ that need to be added to $\mathbf{M}_1(i)$ to obtain $\mathbf{M}_2(i)$ is $\delta(i)$.

since concatenation-type schemes in general have lower rates than single-layer schemes.

Our schemes: The rate achieved by the Gilbert-Varshamov scheme (as demonstrated in Section VI-A) is at least $C(1-2H(2p))$. The computationally-efficient concatenated scheme (details in Section VI-B) achieves rate of at least $\max_{0<r<1-2H(2p)} r \cdot \left(1 - \frac{2p}{H^{-1}\left(\frac{1}{2}(1-r)\right)}\right)$. As can be verified, for small p both our schemes achieve rates higher than either of the benchmark schemes.

IV. Transform Metric

We first define a "natural" distance function between binary matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ related as $\mathbf{M}_1 = \mathbf{M}_2 + \mathbf{B}\mathbf{Z}$ for a binary *basis matrix* $\mathbf{B}$ and a binary matrix $\mathbf{Z}$.

Let $\mathbf{M}_1$ and $\mathbf{M}_2$ be arbitrary $a \times b$ binary matrices. Let $\mathbf{B}$ be a given $a \times c$ matrix with full row rank. Let $\mathbf{M}_1(i)$ and $\mathbf{M}_2(i)$ denote respectively the ith columns of $\mathbf{M}_1$ and $\mathbf{M}_2$. We define $d_{\mathbf{B}}(\mathbf{M}_1, \mathbf{M}_2)$, the *transform distance between* $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *in terms of* $\mathbf{B}$, as follows.

Definition 2. *Let $\delta(i)$ denote the minimal number of columns of $\mathbf{B}$ that need to be added to $\mathbf{M}_1(i)$ to obtain $\mathbf{M}_2(i)$. Then the transform distance $d_{\mathbf{B}}(\mathbf{M}_1, \mathbf{M}_2)$ equals $\sum_{i=1}^{b} \delta(i)$. Correspondingly, the transform metric weight of $\mathbf{M}_1$ is denoted as $W_{\mathbf{B}}(\mathbf{M}_1) = d_{\mathbf{B}}(\mathbf{M}_1, \mathbf{0})$.*

The definition of this transform distance is visualized in Fig. 6. The reason we look into this matrix-based metric is because we want to capture how bit-flips in $\mathbf{Z}$ perturb $\mathbf{TX}$ to the actually received $\mathbf{Y}$ (recall in (1) that $\mathbf{Y} = \mathbf{TX} + \hat{\mathbf{T}}\mathbf{Z}$). In this case, Hamming distance certainly does not work, because a very sparse error matrix $\mathbf{Z}$ may lead to large Hamming distance between $\mathbf{TX}$ and $\mathbf{Y}$. In other words, Hamming distance is not able to quantify the noise level of the network. Notice that the 1's in $\mathbf{Z}(i)$ choose the corresponding columns of $\hat{\mathbf{T}}$ and add to $\mathbf{TX}(i)$.

Claim 6. *The function $d_{\mathbf{B}}(\mathbf{M}_1, \mathbf{M}_2)$ is a distance measure.*

Remark 1: It can be proved directly that $d_{\mathbf{B}}(\mathbf{M}_1, \mathbf{M}_2)$ is a metric by checking the conditions, *i.e.* non-negativity, identity of indiscernibles ($d_{\mathbf{B}}(\mathbf{M}_1, \mathbf{M}_2) \Leftrightarrow \mathbf{M}_1 = \mathbf{M}_2$), symmetry and triangle inequality. A more conceptual proof relates to coset decoding and Cayley graph.

Firstly, consider the case when $n = 1$, *i.e.*, when the matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ reduce to column vectors. Let $\mathbf{B}$ be a binary $a \times c$ matrix with full row rank, which hence implies $a \leq c$. We can regard $\mathbf{B}$ as a parity-check matrix of a binary code $\mathcal{C}$ of length c and dimension $c-a$. the function $d_{\mathbf{B}}(\mathbf{M}_1, \mathbf{M}_2)$ is closely related to coset decoding of $\mathcal{C}$. A column vector $\mathbf{S}$ of length a can be interpreted as the syndrome vector. In coset decoding, we want to find the minimum number of columns in $\mathbf{B}$ which sum to the vector $\mathbf{S}$. This is called the syndrome weight in [27] or the coset leader weight in [28]. Hence, $\delta(i)$ defined in Definition 2 is the same as the syndrome weight of $\mathbf{M}_1 - \mathbf{M}_2$ with respect to the parity-check matrix $\mathbf{B}$. The fact that the syndrome weight induces a metric on $\mathbb{F}_2^a$ can be seen by considering the Cayley graph on $\mathbb{F}_2^a$, generated by the columns of $\mathbf{B}$, *i.e.*, the vertices of the Cayley graph are identified with the vectors in $\mathbb{F}_2^a$, and two vertices are adjacent if and only if their difference is one of the columns in $\mathbf{B}$. Then, $d_{\mathbf{B}}(\mathbf{M}_1, \mathbf{M}_2)$ is the length of a shortest path from $\mathbf{M}_1$ to $\mathbf{M}_2$ in the graph. This is the graph distance and is hence a metric satisfying the triangle inequality. For $n \leq 2$, $d_{\mathbf{B}}(\mathbf{M}_1, \mathbf{M}_2)$ is the sum of n syndrome weights, and therefore is also a metric.[7]

Remark 2: The transform metric reduces to some "commonly used" metrics for specially chosen basis matrix $\mathbf{B}$. For example, the binary Hamming distance is recovered if the matrix $\mathbf{B}$ is the identity matrix. The 2^m-ary Hamming distance is recovered if the matrix $\mathbf{B}$ is the parity-check matrix for the $(2^m - 1, 2^m - m - 1)$-binary Hamming code. Hence, the transform metric might be of independent interest to other applications with different matrices.

V. Converses

This section includes proofs of Theorems 1, 2 and 3, which provide lower bounds on information rates that can be communicated by any code through networks with worst-case bit-flips. In Section V-A, we derive the Hamming-type upper bound. Our bounding technique is similar

[7] This relation between our transform metric and the syndrome weight/coset leader weight in coset decoding is pointed out by Prof. Kenneth W. Shum.

as that in classical coding theory [29] – the main technique lies in deriving lower bounds for the volumes of spheres with our network channel model and corresponding transform metric. In Section V-B, the Plotkin-type upper bound is derived, which provides a constraint on the fraction of error p for achieving positive asymptotic rates. Finally, the Elias-Bassalygo-type upper bound is derived in Section V-C, which is tighter than the Hamming-type bound in the regime where p is less than $\frac{C}{E}\left(1-\frac{C}{E}\right)$.

A. *Proof of Theorem 1 (Hamming-Type Bound)*

Suppose $\mathbf{X}$ is transmitted, by the definitions of the worst-case bit-error network channel in equation (1), the received $\mathbf{Y}$ lies in the radius-$pEmn$ ball (in the transform metric) defined as $\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{TX}, pEmn) = \{\mathbf{Y} | d_{\hat{\mathbf{T}}}(\mathbf{TX}, \mathbf{Y}) \leq pEmn\}$. For the message corresponding to $\mathbf{X}$ to be uniquely decodable, it is necessary that the balls $\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{TX}, pEmn)$ be non-intersecting for each $\mathbf{X}$ chosen to be in the codebook. Hence to get an upper bound on the number of codewords that can be chosen, we need to derive a lower bound of the volume of $\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{TX}, pEmn)$. Recall that $\mathbf{Y}$ equals $\mathbf{TX} + \hat{\mathbf{T}}\mathbf{Z}$. Hence we need to bound from below the number of distinct values of $\hat{\mathbf{T}}\mathbf{Z}$ for $\mathbf{Z}$ with at most $pEmn$ 1's.

As noted before in Section II-B, with high probability for random linear network coding, every $C \times C$ sub-matrix of $\hat{\mathbf{T}}$ sitting on the min-cut is invertible. Specifically, $\mathbf{T}$ is the sub-matrix of $\hat{\mathbf{T}}$ corresponding to the outgoing edges of s, and is invertible with high probability. If $pEmn \leq Cmn$, we claim that all noise patterns attacking on the same min-cut (*e.g.* the outgoing edges of s) result in different $\hat{\mathbf{T}}\mathbf{Z}$. At first, one can check that with the mapping from $\mathbb{F}_{2^m}$ to $\mathbb{F}_2$ in Section II-B, every invertible matrix in $\mathbb{F}_{2^m}$ maps to an invertible matrix in $\mathbb{F}_2$. Hence, every different $\mathbf{Z}$ with 1's only in the corresponding min-cut results in different $\hat{\mathbf{T}}\mathbf{Z}$.

Hence the number of distinct values for $\hat{\mathbf{T}}\mathbf{Z}$ is at least $\binom{Cmn}{pEmn}$, which by Stirling's approximation [30] gives us that

$$|\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{TX}, pEmn)| \geq 2^{CmnH\left(\frac{E}{C}p\right) - \log(Cmn+1)}. \tag{2}$$

The total number of $Cm \times n$ binary matrices is 2^{Cmn}. Thus an upper bound on the size of any codebook for the worst-case binary-error channel is

$$\frac{2^{Cmn}}{2^{CmnH\left(\frac{E}{C}p\right) - \log(Cmn+1)}} = 2^{\left(1 - H\left(\frac{E}{C}p\right) + \frac{\log(Cmn+1)}{Cmn}\right)Cmn},$$

which, asymptotically in n, gives the Hamming-type upper bound on the rate of any code as $1 - H\left(\frac{E}{C}p\right) + o\left(\frac{\log(Cmn+1)}{Cmn}\right)$.

B. Proof of Theorem 2 (Plotkin-Type Bound)

In this section, we derive a Plotkin-type upper bound [31] on the achievable rate over worst-case bit-flip networks. Lemma 7 below shows that when the minimum distance of a codebook is beyond certain values, the codebook size diminishes. Lemma 8 derives upper bounds on the codebook size with restricted minimum distance.

Lemma 7. *Let $\mathcal{X} \subseteq \{\mathbb{F}_2^{Cm\times n}\}$ be a codebook for the worst-case binary-error network channel, the transformed codebook $\mathbf{T}\mathcal{X}$ of which has minimum transform distance d.*

1) *For networks with $E \geq 2C$, if $d > 2\left(1-\frac{C}{E}\right)Cmn$, then $|\mathcal{X}| \leq \frac{d}{d-2\left(1-\frac{C}{E}\right)Cmn}$.*
2) *For networks with $E < 2C$, if $d > Emn/2$, then $|\mathcal{X}| \leq \frac{2d}{2d-Emn}$.*

Proof: Let $\mathbf{X}_1, \dots, \mathbf{X}_M$ be codewords in $\mathcal{X}$, where M denotes the codebook size. Because $\mathbf{T}\mathcal{X}$ has minimum transform distance d, we have $d_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}_i, \mathbf{T}\mathbf{X}_j) \geq d$ for all $i \neq j$. Hence the sum of all distances between codewords in $\mathbf{T}\mathcal{X}$ can be bounded by

$$\sum\nolimits_{1\leq i\leq j\leq M} d_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}_i, \mathbf{T}\mathbf{X}_j) \geq \binom{M}{2} d. \tag{3}$$

On the other hand, considering the columns of the codewords, let $\delta_{ij}(k)$ denote the minimum number of columns from $\hat{\mathbf{T}}$ that need to be added to $\mathbf{T}\mathbf{X}_i(k)$ to obtain $\mathbf{T}\mathbf{X}_j(k)$, we have

$$\sum\nolimits_{1\leq i\leq j\leq M} d_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}_i, \mathbf{T}\mathbf{X}_j) = \sum\nolimits_{k=1}^{n} \sum\nolimits_{1\leq i\leq j\leq M} \delta_{ij}(k). \tag{4}$$

For any column k, to bound the sum of transform metric distances of $\{\mathbf{T}\mathbf{X}_1(k), \dots, \mathbf{T}\mathbf{X}_M(k)\}$ write $\mathbf{T}X_i(k) = \hat{\mathbf{T}}V_i$ for some binary vector V_i for all $1 \leq 1 \leq M$. Recall that $\hat{\mathbf{T}}$ has full column rank with high probability for random linear network coding, we can require the Hamming weight of V_i to be no more than Cm. Hence, $\delta_{ij}(k)$ can be bounded from above by the Hamming distance between V_i and V_j, *i.e.,*

$$\sum\nolimits_{1\leq i\leq j\leq M} \delta_{ij}(k) \leq \sum\nolimits_{1\leq i\leq j\leq M} d_{\text{Hamming}}(V_i, V_j). \tag{5}$$

We arrange the vectors $V_1, V_2, \dots, V_M$ to an $M \times Em$ binary matrix, where the ith row of the matrix correspond to vector V_i. Suppose for column l of the matrix, there are s_l 1's and $M - s_l$

0's. Then

$$\sum\nolimits_{1\leq i\leq j\leq M} d_{\text{Hamming}}(V_i, V_j) = \sum\nolimits_{l=1}^{Em} s_l(M-s_l). \tag{6}$$

We divide into two cases regarding network parameters C and E.

Case 1 ($E \geq 2C$): In this case, we cannot set $s_l = M/2$. Otherwise the V_i's have average Hamming weight $Em/2$, which is larger than the constraint Cm. Recall that we have the constraint that $W_{\text{Hamming}}(V_i) \leq Cm$. Hence the total number of 1's in the matrix is bounded from above by $\sum_{l=1}^{Em} s_l \leq MCm$. Hence,

$$\begin{aligned}\sum\nolimits_{l=1}^{Em} s_l(M-s_l) &= M\sum\nolimits_{l=1}^{Em} s_l - \sum\nolimits_{l=1}^{Em} {s_l}^2\\ &\leq M\sum\nolimits_{l=1}^{Em} s_l - \left(\sum\nolimits_{l=1}^{Em} s_l\right)^2/Em\\ &\leq M^2Cm - \frac{(MCm)^2}{Em}\\ &= M^2\left(1-\frac{C}{E}\right)Cm \end{aligned} \tag{7}$$

Combining equations (4), (5), (6) and (7), we have

$$\sum\nolimits_{1\leq i\leq j\leq M} d_{\hat{\mathbf{T}}}(\mathbf{TX}_i, \mathbf{TX}_j) \leq nM^2\left(1-\frac{C}{E}\right)Cm. \tag{8}$$

From equations (3) and (8), we have

$$M(M-1)d/2 \leq nM^2\left(1-\frac{C}{E}\right)Cm.$$

Hence when $d > 2\left(1-\frac{C}{E}\right)Cmn$, we have $|\mathcal{X}| = M \leq \frac{d}{d-2\left(1-\frac{C}{E}\right)Cmn}$.

Case 2 ($E < 2C$): In this case, let $s_l = M/2$ to maximize equation (6). Hence,

$$\begin{aligned}\sum\nolimits_{l=1}^{Em} s_l(M-s_l) &\leq \sum\nolimits_{l=1}^{Em} M^2/4\\ &= EmM^2/4. \end{aligned} \tag{9}$$

Combining equations (4), (5), (6) and (9), we have

$$\sum\nolimits_{1\leq i\leq j\leq M} d_{\hat{\mathbf{T}}}(\mathbf{TX}_i, \mathbf{TX}_j) \leq nEmM^2/4. \tag{10}$$

From equations (3) and (10), we have

$$M(M-1)d/2 \leq nEmM^2/4.$$

Hence when $d > Emn/2$, we have $|\mathcal{X}| = M \leq \frac{2d}{2d-Emn}$. □

Lemma 8. *Let $\mathcal{X} \subseteq \{\mathbb{F}_2^{Cm\times n}\}$ be a codebook for the worst-case binary-error network channel, the transformed codebook $\mathbf{T}\mathcal{X}$ of which has minimum transform distance d.*

1) For networks with $E \geq 2C$, if $d \leq 2\left(1-\frac{C}{E}\right)Cmn$, then

$$|\mathcal{X}| \leq d \cdot 2^{Cmn - \frac{E}{2(E-C)}d + \frac{E}{2(E-C)}}.$$

2) For networks with $E < 2C$, if $d \leq Emn/2$, then

$$|\mathcal{X}| \leq 2d \cdot 2^{Cmn - \frac{2C}{E}(d-1)}.$$

Proof: **Case 1 ($E \geq 2C$):** When $d \leq 2\left(1-\frac{C}{E}\right)Cmn$, we have $n \geq \frac{E}{2C(E-C)m}d$. Let $l = n - \frac{E}{2C(E-C)m}(d-1)$, for each matrix $\mathbf{G} \in \mathbb{F}_2^{Cm\times l}$, let $\mathcal{X}_G$ be a subcode of $\mathcal{X}$ consisting of all codewords which have $\mathbf{G}$ as the $Cm \times l$ submatrix in the first l columns, then puncture the first l columns. Formally,

$$\mathcal{X}_{\mathbf{G}} = \{\mathbf{X}^{[l+1,n]} | \mathbf{X}(i) = \mathbf{G}(i) \text{ for } 1 \leq i \leq l\},$$

where $\mathbf{X}^{[l+1,n]} \in \mathbb{F}_2^{Cm\times(n-l)}$ is the submatrix of $\mathbf{X}$ consisting of the $n-l$ columns $\mathbf{X}(l+1), \mathbf{X}(l+2), \ldots, \mathbf{X}(n)$. For each $\mathbf{G}$, the subcode $\mathcal{X}_{\mathbf{G}}$ is a codebook with block length $n-l = \frac{E}{2C(E-C)m}(d-1)$. The original codebook $\mathcal{X}$ has minimum transform metric distance d, so does the subcode $\mathcal{X}_{\mathbf{G}}$. Hence $d > 2(1-\frac{C}{E})Cm\left(\frac{E}{2C(E-C)m}(d-1)\right) = d-1$, and by Lemma 7.1 we have $|\mathcal{X}_{\mathbf{G}}| \leq d$. The original codebook size can relate to sizes of subcodes as $|\mathcal{X}| = \sum_{\mathbf{G}\in\mathbb{F}_2^{Cm\times l}} |\mathcal{X}_{\mathbf{G}}|$. Hence we have $|\mathcal{X}| \leq d \cdot 2^{Cml} = d \cdot 2^{Cm\left(n - \frac{E}{2C(E-C)m}d + \frac{E}{2C(E-C)m}\right)}$.

Case 2 ($E < 2C$):

When $d \leq Emn/2$, we have $n \geq 2d/Em$. Let $l = n - \frac{2}{Em}(d-1)$, and puncture the codebook the same way as in Case 1 by puncturing the first l columns. In this case, each subcode $\mathcal{X}_G$ has block length $n-l = \frac{2}{Em}(d-1)$. Because $d > Em/2 \cdot \frac{2}{Em}(d-1) = d-1$, by Lemma 7.2 we have $|\mathcal{X}_G| \leq 2d$. Hence we have $|\mathcal{X}| = \sum_{G\in\mathbb{F}_2^{Cm\times l}} |\mathcal{X}_G| \leq 2d \cdot 2^{Cml} = 2d \cdot 2^{Cmn - \frac{2C}{E}(d-1)}$.

□

With Lemma 7 and Lemma 8, our Plotkin-type upper bound follows naturally. For successful decoding, a codebook needs to have minimum transform metric distance at least $d = 2pEmn+1$.

Case 1 (**$E \geq 2C$**)**:** If $p > \left(1-\frac{C}{E}\right)\frac{C}{E}$, the minimum distance $d = 2pEmn+1 > 2\left(1-\frac{C}{E}\right)Cmn$ and by Lemma 7.1, the codebook size is of order $\mathcal{O}(n)$. Hence the rate goes to 0 as $n \to \infty$.

When $p \leq \left(1-\frac{C}{E}\right)\frac{C}{E}$, the minimum distance $d = 2pEmn+1 \leq 2\left(1-\frac{C}{E}\right)Cmn$ and by Lemma 8.1, the size of any codebook is bounded from above by $|\mathcal{X}| \leq d\cdot 2^{Cm\left(n-\frac{E}{2C(E-C)m}d+\frac{E}{2C(E-C)m}\right)}$. Hence, asymptotically $R = \lim_{n\to\infty}\frac{1}{Cmn}\log|\mathcal{X}| \leq 1-\frac{E^2}{CE-C^2}p$.

Case 2 (**$E < 2C$**)**:** If $p > 1/4$, the minimum distance $d = 2pEmn+1 > Emn/2$ and by Lemma 7.2, the codebook size is of order $\mathcal{O}(n)$. Hence the rate goes to 0 as $n \to \infty$.

When $p \leq 1/4$, the minimum distance $d = 2pEmn+1 \leq Emn/2$ and by Lemma 8.2, the size of any codebook is bounded from above by $|\mathcal{X}| \leq 2d\cdot 2^{Cml} = 2d\cdot 2^{Cmn-\frac{2C}{E}(d-1)}$. Hence, asymptotically $R = \lim_{n\to\infty}\frac{1}{Cmn}\log|\mathcal{X}| \leq 1-4p$.

C. Proof of Theorem 3 (Elias-Bassalygo-Type Bound)

In this section, we derive an Elias-Bassalygo-type bound [32] with the transform metric. Firstly, in Lemma 9 below we derive a Johnson-type bound.

Lemma 9 (Johnson-Type Bound)**.** *Let $J_{\hat{\mathbf{T}}}(Cm\times n, d, e)$ be the maximum number of codewords in a ball of transform metric radius e for any transformed codebook by matrix $\mathbf{T}$ with minimum transform metric distance d. If $\frac{e}{Emn} < \frac{1}{2}\left(1-\sqrt{1-\frac{2d}{Emn}}\right)$, then*

$$J_{\hat{\mathbf{T}}}(Cm\times n, d, e) \leq \frac{dEmn}{2}.$$

Proof: For a transformed codebook $\mathbf{T}\mathcal{X}$ (Definition 1) with minimum transform metric distance d, and a matrix $\mathbf{O} \in \mathbb{F}_2^{Cm\times n}$ being the center of the ball $\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{O}, e)$. Let $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_M$ be the codewords in $\mathcal{X}$ where M denotes the codebook size. Let $\mathbf{X}_i' = \mathbf{X}_i - \mathbf{T}^{-1}\mathbf{O}$ for all $1 \leq i \leq M$, *i.e.,*, shift the ball $\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{O}, e)$ and the codebook to be centered at the zero matrix $\mathbf{0}^{Cm\times n}$, the shifted codewords $\mathbf{X}_i'$'s satisfy two conditions: 1) for $1 \leq i \leq M$, $W_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}_i') \leq e$; 2) for $i \neq j$, $d_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}_i', \mathbf{T}\mathbf{X}_j') \geq d$. The following proof is quite similar as that of Lemma 7, except that there is the additional weight constraint $W_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}_i') \leq e$.

Let $\bar{e}$ denote the average weight (in transform metric) of the shifted codebook, with similar steps as in Lemma 7, we have that

$$M(M-1)d/2 \leq \sum\nolimits_{1 \leq i \leq j \leq M} d_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}'_i, \mathbf{T}\mathbf{X}'_j) \leq M^2 \bar{e} - \frac{M^2 \bar{e}^2}{Emn}.$$

Rearranging we obtain $\left(d - 2\bar{e} + \frac{2\bar{e}^2}{Emn}\right) M \leq d$. For $e \leq \frac{Emn}{2}\left(1 - \sqrt{1 - \frac{2d}{Emn}}\right)$, note that the average weight is bounded from above by the radius, that is, $\bar{e} \leq e$, we also have $\bar{e} \leq \frac{Emn}{2}\left(1 - \sqrt{1 - \frac{2d}{Emn}}\right)$. Hence,

$$\begin{aligned} M &\leq \frac{dEmn}{dEmn - 2\bar{e}Emn + 2\bar{e}^2} \\ &= \frac{dEmn/2}{(Emn/2 - \bar{e})^2 - (Emn/2 - d)Emn/2} \end{aligned}$$

The denominator is positive, and it must be at least 1 because it is an integer. Hence we have $J_{\hat{\mathbf{T}}}(Cm \times n, d, e) \leq \frac{dEmn}{2}$ if $\frac{e}{Emn} < \frac{1}{2}\left(1 - \sqrt{1 - \frac{2d}{Emn}}\right)$. □

Using the Johnson-type bound, the Elias-Bassalygo-type bound is derived as follows.

We first prove that given a codebook $\mathcal{X}$ of size M, for any η there exists a transform metric ball $\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)$ of radius η containing at least $M \cdot \mathrm{Vol}\left(\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)\right)/2^{Cmn}$ codewords.

Picking a transform metric ball $\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)$ of radius η around a random center. For each $\mathbf{X} \in \mathcal{X}$, let $\mathbb{1}_{\mathbf{X}}$ be an indicator variable of the event that $\mathbf{X} \in \mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)$. The expected number of codewords from $\mathcal{X}$ in the ball $\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)$ is given by $E[\mathbb{1}_{\mathbf{X}}] = \Pr(\mathbb{1}_{\mathbf{X}} = 1) = \mathrm{Vol}\left(\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)\right)/2^{Cmn}$. Hence, the expected total number of codewords in $\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)$ is $\sum_{\mathbf{X} \in \mathcal{X}} E[\mathbb{1}_X] = M \cdot \mathrm{Vol}\left(\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)\right)/2^{Cmn}$. There must be at least one ball achieving the expectation, hence there exists a transform metric ball $\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)$ of radius η containing at least $M \cdot \mathrm{Vol}\left(\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)\right)/2^{Cmn}$ codewords.

Now set $\eta = \frac{Emn}{2}\left(1 - \sqrt{1 - \frac{2d}{Emn}}\right) - 1$, by the Johnson-type bound in Lemma 9, there can be no more than $\frac{dEmn}{2}$ codewords in the ball. Hence

$$M \cdot \mathrm{Vol}\left(\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)\right)/2^{Cmn} \leq \frac{dEmn}{2}.$$

To obtain an upper bound on the codebook size M, we need to characterize a lower bound on the volume of the ball $\mathrm{Vol}\left(\mathcal{B}_{\hat{\mathbf{T}}}(\cdot, \eta)\right)$. Recall in (2) in the proof of the Hamming-type bound in Theorem 1 we have already bounded this quantity. Note that the distance $d = 2pEmn + 1$, hence $\eta = \frac{Emn}{2}\left(1 - \sqrt{1 - \frac{2d}{Emn}}\right) - 1 \leq \frac{Emn}{2}(1 - \sqrt{1 - 4p})$.

If $p < \frac{C}{E}\left(1 - \frac{C}{E}\right)$, we have $\eta \leq Cmn$. Similarly as in Section V-A, the volume can be bounded from below by $\mathrm{Vol}\left(\mathcal{B}_{\hat{\mathbf{T}}}(\cdot,\eta)\right) \geq \binom{Cmn}{\eta}$, which by Stirling's approximation is at least $2^{CmnH(\eta/Cmn)-\log(Cmn+1)}$. Substituting η/Cmn by $\frac{E}{2C}\left(1-\sqrt{1-4p+\frac{2}{Emn}}\right)-\frac{1}{Cmn}$, we have that

$$\begin{aligned} M &\leq \frac{dEmn}{2} 2^{Cmn} / \mathrm{Vol}\left(\mathcal{B}_{\hat{\mathbf{T}}}(\cdot,\eta)\right) \\ &\leq \frac{dEmn}{2} 2^{Cmn\left(1-H\left(\frac{E}{2C}\left(1-\sqrt{1-4p+\frac{2}{Emn}}\right)-\frac{1}{Cmn}\right)+\frac{\log(Cmn+1)}{Cmn}\right)}, \end{aligned}$$

which, asymptotically in n, leads to the Elias-Bassalygo-type upper bound on the rate of any code as $1-H\left(\frac{E}{2C}(1-\sqrt{1-4p})\right)+o\left(\frac{\log(Cmn+1)}{Cmn}\right)$.

VI. Achievability

In this section, we present communication schemes and corresponding achievable rates for networks with worst-case bit-errors. In Section VI-A, schemes motivated by the well-known Gilbert-Varshamov (GV) bound from classical coding theory [25], [26] are provided. Specifically, section VI-A1 considers the *coherent* scenario, *i.e.*, when the linear coding coefficients in the network, or at least the transfer matrix $\mathbf{T}$ and the impulse response matrix $\hat{\mathbf{T}}$, are known in advance to the receiver. This setting is primarily used for exposition, as a foundation for the *non-coherent* setting, when no advance information about the topology of the network, the linear coding coefficients used, or $\mathbf{T}$ or $\hat{\mathbf{T}}$ are known in advance to the receiver. In Section VI-A2, it is demonstrated that essentially the same rates are still achievable, albeit with a rate-loss that is asymptotically negligible in the block-length n. In Section VI-B, a concatenated version of previously presented codes are presented, so that the computational complexity of resulting codes scales polynomially in the block-length (albeit still exponentially in network parameters). The rate achieved by the concatenation scheme is characterized by a Zyablov-type lower bound. Finally, in Section VI-B3, a *generalized minimum distance decoding* scheme is provided, which is able to correct up to half of the minimum distance of the concatenated codes.

A. Proof of Theorem 4 (Gilbert-Varshamov-type bounds)

1) Coherent GV-type network codes: We first discuss the case when the network transfer matrix $\mathbf{T}$ and impulse response matrix $\hat{\mathbf{T}}$ are known in advance.

Codebook design: Initialize set $\mathcal{A}$ as the set of all binary $Cm \times n$ matrices. Choose a uniformly random $Cm \times n$ binary matrix $\mathbf{X}_1$ as the first codeword. Eliminate from $\mathcal{A}$ all matrices in the radius-$2pEmn$ ball (in the transform metric) $\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}_1, 2pEmn)$. Then choose a matrix $\mathbf{Y}_2$ uniformly at random in the remaining set and choose $\mathbf{X}_2 = \mathbf{T}^{-1}\mathbf{Y}_2$ as the second codeword. Now, further eliminate all matrices in the radius-$2pEmn$ ball $\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{Y}_2, 2pEmn)$ from $\mathcal{A}$, choose a random $\mathbf{Y}_3$ from the remaining set, and choose the third codeword $\mathbf{X}_3$ as $\mathbf{X}_3 = \mathbf{T}^{-1}\mathbf{Y}_3$. Repeat this procedure until the set $\mathcal{A}$ is empty.

Decoder: The receiver uses a minimum distance decoder with the transform metric, that is, the decoder picks the codeword $\mathbf{X}$ which minimizes the transform metric distance $d_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}, \mathbf{Y})$.

To prove Theorem 4.1, we need an upper bound on $\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}, 2pEmn)$ (rather than a lower bound on $\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}, pEmn)$ as in Section V-A), that is, we need to bound from above the number of distinct $\hat{\mathbf{T}}\mathbf{Z}$ for $\mathbf{Z}$ with at most $2pEmn$ 1's. We know that in $\hat{\mathbf{T}}$, every $Cm \times Cm$ submatrix on a min-cut is full rank with high probability. In fact, because we assume that the number of incoming links to the sink is C, the submatrix of $\hat{\mathbf{T}}$ corresponding to the incoming links to the sink is an identity matrix. However, the exact number of distinct $\hat{\mathbf{T}}\mathbf{Z}$ also depends on the rank of other columns of $\hat{\mathbf{T}}$, hence depends on the network topology. In general, the number of distinct $\hat{\mathbf{T}}\mathbf{Z}$ can be bounded from above by the number of different $\mathbf{Z}$. This equals $\sum_{i=0}^{2pEmn} \binom{Emn}{i}$. The dominant term this summation is when i equals $2pEmn$. Hence the summation can be bounded from above by $(2pEmn+1)\binom{Emn}{2pEmn}$. By Stirling's approximation [30] we have that $|\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}, 2pEmn)| \leq (2pEmn+1)2^{H(2p)Emn}$. Thus a lower bound on the size of the codebook for coherent GV-type codes is $\frac{2^{Cmn}}{(2pEmn+1)2^{H(2p)Emn}} = 2^{\left(1-\frac{E}{C}H(2p)-\frac{\log(2pEmn+1)}{Cmn}\right)Cmn}$,, which, asymptotically in n, gives the rate of coherent GV-type bound network codes $1 - \frac{E}{C}H(2p) - o\left(\frac{\log(2pEmn+1)}{Cmn}\right)$.

Remark 1: To see the tightness of this upper bound, from Section V-A, $|\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}, 2pEmn)| \geq 2^{H(\frac{2E}{C}p)Cmn}$. By taking Taylor series expansion, asymptotically in n, the upper and lower bounds of $\frac{1}{Cmn}\log|\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}, 2pEmn)|$ differ by $(\frac{2E}{C}\log\frac{E}{C})p + (\frac{2E}{C}(\frac{E}{C}-1)\log e)p^2 + \mathcal{O}(p^3)$, which is smaller in order of p than the leading term $-\frac{2E}{C}p\log(2p)$. In other words, this upper bound is tighter for smaller value of p.

Remark 2: For some specific network topology and hence the matrix $\hat{\mathbf{T}}$, the rate achieved by this GV-type codes might be higher, since we use a "loose" bound for number of distince $\hat{\mathbf{T}}\mathbf{Z}$. However, for the non-coherent regime discussed below, $\hat{\mathbf{T}}$ is unknown *a priori*. Hence, the bound is tighter in the sense that one needs to consider all possible values of $\hat{\mathbf{T}}$.

Remark 3: For the scenario with multiple sinks $\{t \in \mathcal{D}\}$, for each sink t there is an impulse response matrix $\hat{\mathbf{T}}_t$. In the process of choosing codewords, more matrices from $\mathcal{A}$ are eliminated corresponding to all $\{\hat{\mathbf{T}}_t : t \in \mathcal{D}\}$. However, for finite number of sinks, by union bound, the same rate is achievable asymptotically in the block length.

2) Non-coherent GV-type network codes: The assumption that $\mathbf{T}$ and $\hat{\mathbf{T}}$ are known in advance to the receiver is often unrealistic, because random linear coding coefficients in the network are usually chosen on the fly. Hence we now consider the non-coherent setting, wherein $\mathbf{T}$ and $\hat{\mathbf{T}}$ are not known *a priori*. We demonstrate that despite this lack of information the same rate as in Section VI-A1 is achievable in the non-coherent setting.

The number of all possible $\hat{\mathbf{T}}$ is at most 2^{CEm} because $\hat{\mathbf{T}}$ is a $C \times E$ matrix over $\mathbb{F}_{2^m}$ – this number is independent of the block-length n. Hence in the non-coherent GV setting, we consider *all* possible values of $\hat{\mathbf{T}}$, and hence $\mathbf{T}$, since it comprises of a specific subset of C columns of $\hat{\mathbf{T}}$.

Codebook design: Initialize set $\mathcal{A}$ as the set of all binary $Cm \times n$ matrices. Choose a uniformly random matrix $\mathbf{X}_1$ from $\mathcal{A}$ as the first codeword. For each $C \times E$ matrix $\hat{\mathbf{T}}$ (over $\mathbb{F}_{2^m}$), eliminate from $\mathcal{A}$ all matrices $\{\mathbf{X}' : d_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}_1, \mathbf{T}\mathbf{X}') \leq 2pEmn\}$. Then choose a matrix $\mathbf{X}_2$ uniformly at random in the remaining set as the second codeword. Then further eliminate all matrices $\{\mathbf{X}' : d_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}_2, \mathbf{T}\mathbf{X}') \leq 2pEmn\}$ for all $\hat{\mathbf{T}}$. Repeat this procedure until the set $\mathcal{A}$ is empty.

Decoder: The decoder picks the codeword $\mathbf{X}$ which minimizes the transform metric distance $d_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}, \mathbf{Y})$ for all possible $\hat{\mathbf{T}}$.

The crucial difference from the coherent regime is in the process of choosing codewords – at each stage of the codeword elimination process, at most $2^{CEm} \cdot |\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{T}\mathbf{X}, 2pEmn)|$ potential codewords are eliminated. Hence the number of potential codewords that can be chosen in the codebook is at least $\frac{2^{Cmn}}{2^{CEm}(2pEmn+1)2^{H(2p)Emn}}$, which equals $2^{\left(1-\frac{E}{C}H(2p)-\frac{\log(2pEmn+1)+CEm}{Cmn}\right)Cmn}$. As can be verified, asymptotically in n this leads to the same rate of $1-\frac{E}{C}H(2p)-o\left(\frac{\log(2pEmn+1)+CEm}{Cmn}\right)$ in constant terms as in the coherent regime.

Remark: In the non-coherent regime, the codebook design is agnostic to the choice of $\hat{\mathbf{T}}$. Hence, the non-coherent GV-type codes would work for the scenario with multiple sinks.

Note: We show in the following Section VI-A3 that random linear codes achieve the GV-type bound with high probability, which reduce the encoding complexity.

3) Linear GV-type Bound: Similar to Varshamov's linear construction [26] in classical coding theory, we show that for our worst-case binary-error network channel, random linear codes achieve the GV-type bound with high probability.

Let $\mathbf{G} \in \mathbb{F}_{2^m}^{k \times n}$ be the generator matrix of a random linear code, where each entry of $\mathbf{G}$ is chosen uniformly and independently at random from $\mathbb{F}_{2^m}$. With the mapping from $\mathbb{F}_{2^m}$ to $\mathbb{F}_2$ for the second symbol/matrix in a multiplication, $\mathbf{G}$ is equivalently a random $km \times n$ matrix over $\mathbb{F}_2$, where each entry is chosen uniformly i.i.d. from $\mathbb{F}_2$. Let $\mathbf{M} \in \mathbb{F}_{2^m}^{C \times (k-C)} \setminus \{\mathbf{0}\}$, with the mapping from $\mathbb{F}_{2^m}$ to $\mathbb{F}_2$ for the first symbol/matrix in a multiplication, $\mathbf{M} \in \mathbb{F}_2^{Cm \times (k-C)m} \setminus \{\mathbf{0}\}$. Note that the number of all possible $\mathbf{M}$ is $2^{C(k-C)m} - 1$, because $\mathbf{M}$ is chosen from $C \times (k-C)$ matrices over $\mathbb{F}_{2^m}$ then mapped to binary field. Now let $\bar{\mathbf{M}} = [\mathbf{I}|\mathbf{M}]$ be the $Cm \times km$ binary matrix by augmenting a $Cm \times Cm$ identity matrix $\mathbf{I}$ in front of $\mathbf{M}$. (The parameter k here should be sufficiently large so that $k - C > 0$.) We need to show that for any matrix $\mathbf{M}$ chosen in the way described above, $d_{\hat{\mathbf{T}}}(\bar{\mathbf{M}}\mathbf{G}, \mathbf{0}) \geq d$ with high probability, where $d = 2pEmn + 1$ is the minimum distance we require for the codebook.

Note that for any fixed matrix $\mathbf{M}$, by choosing $\mathbf{G}$ uniformly at random, $\bar{\mathbf{M}}\mathbf{G}$ is a uniformly random matrix from $\mathbb{F}_2^{Cm \times n}$. Hence, the probability over the choice of $\mathbf{G}$ of the code being "bad" can be bounded from above by $\Pr(d_{\hat{\mathbf{T}}}(\bar{\mathbf{M}}\mathbf{G}, \mathbf{0}) < d) = |\mathcal{B}_{\hat{\mathbf{T}}}(\mathbf{0}, d-1)|/2^{Cmn} \leq (2pEmn + 1)2^{H(2p)Emn}/2^{Cmn}$, where the inequality is bounded in Section VI-A1. By the union bound, $\Pr(\exists \mathbf{M}, d_{\hat{\mathbf{T}}}(\bar{\mathbf{M}}\mathbf{G}, \mathbf{0}) < d) \leq 2^{Cm(k-C)} \cdot (2pEmn + 1)2^{H(2p)Emn}/2^{Cmn} = (2pEmn + 1)2^{-\varepsilon Cmn}$, if we choose $k = \left(1 - \frac{E}{C}H(2p) - \varepsilon\right) n + C$. Since $(2pEmn + 1)2^{-\varepsilon Cmn} \ll 1$ for large enough n, we have shown that there exists a linear code with minimum distance $2pEmn + 1$ and rate at least $1 - \frac{E}{C}H(2p) - \varepsilon$.

Note: The advantage of this Varshamov-type construction is that the encoding complexity is $\mathcal{O}(n^2Cm^2)$, though the decoding complexity is still $\Omega(e^n)$. To deal with the high decoding complexity, we present a concatenated construction in the following Section VI-B so that the encoding and decoding complexity grows only polynomial in the block-length (albeit still exponentially in network parameters).

B. Proof of Theorem 5 (Concatenated Codes and Zyablov-Type Bound)

The codes which achieve the Gilbert-Varshamov-type bound in Section VI-A take running time $2^{\mathcal{O}(n)}$. This section provides a code concatenation strategy using the GV-type code from Section VI-A1 as the *inner code* and a Reed-Solomon code as the *outer code*. This type of *concatenated network codes* have encoding/decoding complexity that is polynomial in the block length n (albeit still exponentially in the network parameter C and the coding parameter m).

Also, a Zyablov-type lower bound stated in Theorem 5 is derived, which characterizes the rate achieved by the concatenated network codes.

1) Code Concatenation Construction: Consider the outer code and the inner code as follows,

$$\begin{aligned} C_{\text{out}} &: \left[\mathbb{F}_{2^{Cm\times R_{\text{in}}\log n}}\right]^{R_{\text{out}}\frac{n}{\log n}} \rightarrow \left[\mathbb{F}_{2^{Cm\times R_{\text{in}}\log n}}\right]^{\frac{n}{\log n}}, \\ C_{\text{in}} &: \left[\mathbb{F}_2\right]^{Cm\times R_{\text{in}}\log n} \rightarrow \left[\mathbb{F}_2\right]^{Cm\times \log n}, \end{aligned}$$

where R_{out} and R_{in} are the corresponding rates of the outer and inner codes to be characterized later. The concatenated code is denoted by $C_{\text{con}} = C_{\text{out}} \circ C_{\text{in}}$, and conducts the following steps.

- Firstly, the encoder breaks the messages from $\mathbb{F}_2^{Cm\times R_{\text{out}}R_{\text{in}}n}$ into $R_{\text{out}}\frac{n}{\log n}$ such many chunks with size $Cm\times R_{\text{in}}\log n$, and treats each chunk as an element from the large field $\mathbb{F}_{2^{Cm\times R_{\text{in}}\log n}}$. The field size is much larger than the block length $\frac{n}{\log n}$, hence one can take an $\left[\frac{n}{\log n}, R_{\text{out}}\frac{n}{\log n}, d_{\text{out}}\right]_{2^{Cm\times R_{\text{in}}\log n}}$ Reed-Solomon code as the outer code. Therefore, the minimum distance of the outer code is $d_{\text{out}} = (1-R_{\text{out}})\frac{n}{\log n}+1$. The outer code converts the messages into codewords of length $\frac{n}{\log n}$ over the large alphabet $\mathbb{F}_{2^{Cm\times R_{\text{in}}\log n}}$.
- Secondly, the encoder takes the output codewords from the outer code and converts the symbols from the field $\mathbb{F}_{2^{Cm\times R_{\text{in}}\log n}}$ into binary matrices of size $Cm \times R_{\text{in}}\log n$. For the inner code, the encoder takes the block binary matrices from $\left[\mathbb{F}_2\right]^{Cm\times R_{\text{in}}\log n}$ as messages, then uses the same codebook design for the GV-type bound as in Section VI-A1. Hence the minimum distance d_{in} and the rate R_{in} of the inner code satisfy $R_{\text{in}} = 1 - \frac{E}{C}H\left(\frac{d_{\text{in}}}{Em\log n}\right)$, which gives that $d_{\text{in}} = H^{-1}\left(\frac{C}{E}(1-R_{\text{in}})\right)Em\log n$. This completes the whole concatenated coding process and outputs codewords from $\mathbb{F}_2^{Cm\times n}$.

2) Zyablov-Type Bound: The Reed-Solomon outer code has minimum distance $d_{\text{out}} = (1 - R_{\text{out}})\frac{n}{\log n}+1$. The GV-type inner code has minimum distance $d_{\text{in}} = H^{-1}\left(\frac{C}{E}(1-R_{\text{in}})\right)Em\log n$. The overall distance $\bar{d}$ of the code C_{con} satisfies $\bar{d} \geq d_{\text{out}}{\cdot}d_{\text{in}} \geq (1-R_{\text{out}})H^{-1}\left(\frac{C}{E}(1-R_{\text{in}})\right)Emn$.

Take $(1-R_{\text{out}})H^{-1}\left(\frac{C}{E}(1-R_{\text{in}})\right)Emn = 2pEmn$, we have $R_{\text{out}} = 1 - \frac{2p}{H^{-1}\left(\frac{C}{E}(1-R_{\text{in}})\right)}$. The overall rate of the concatenated code is $R = R_{\text{out}} \cdot R_{\text{in}}$, replace R_{in} by an adjustable variable r, optimized over the choice of r, the rate of the concatenated code satisfies

$$R \geq \max_{0<r<1-\frac{E}{C}H(2p)} r \cdot \left(1 - \frac{2p}{H^{-1}\left(\frac{C}{E}(1-r)\right)}\right),$$

where the constraint $r < 1 - \frac{E}{C}H(2p)$ is necessary to guarantee that $R > 0$.

3) Generalized Minimum Distance Decoding: A natural decoding algorithm is to reverse the encoding process as described in Section VI-B1. Briefly, the algorithm uses the inner code to decode each chunk with possibly wrongly decoded chunks, then uses the outer code to correct the wrongly decoded chunks. Denote the input matrix to the decoder as $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_{n/\log n}) \in \left[(\mathbb{F}_2)^{Cm \times \log n}\right]^{n/\log n}$. The natural decoding algorithm is described as follows.

Natural decoding algorithm:

Step1: Decode each $\mathbf{Y}_i$ to $\mathbf{V}_i \in \mathbb{F}_2^{Cm \times R_{\text{in}} \log n}$ such that $\mathbf{V}_i$ minimizes $d_{\hat{\mathbf{T}}}\left(C_{\text{in}}(\mathbf{V}_i), \mathbf{Y}_i\right)$.

Step2: Decode $\mathbf{V} = (V_1, V_2, \ldots, V_{n/\log n}) \in (\mathbb{F}_{2^{Cm \times R_{\text{in}} \log n}})^{\frac{n}{\log n}}$ using decoding algorithms for the RS outer code.

It can be easily shown that the natural decoding algorithm can correct up to $(d_{\text{out}} \cdot d_{\text{in}})/4$ errors. Briefly, the outer code fails only if the number of wrongly decoded inner chunks is greater than $d_{\text{out}}/2$. An inner chunk is decoded wrongly only when there are more than $d_{\text{in}}/2$ errors.

To improve the decodability to correct up to half the minimum distance $(d_{\text{out}} \cdot d_{\text{in}})/2$, we develop the algorithm below mimicking the *generalized minimum distance decoding* [33] for classical concatenated codes.

Generalized minimum distance (GMD) decoding algorithm:

Step1: Decode each $\mathbf{Y}_i$ to $\mathbf{V}_i \in \mathbb{F}_2^{Cm \times R_{\text{in}} \log n}$ such that $\mathbf{V}_i$ minimizes $d_{\hat{\mathbf{T}}}\left(C_{\text{in}}(\mathbf{V}_i), \mathbf{Y}_i\right)$. Let $\omega_i = \min\left(d_{\hat{\mathbf{T}}}\left(C_{\text{in}}(\mathbf{V}_i), \mathbf{Y}_i\right), d_{\text{in}}/2\right)$.

Step2: With probability $2\omega_i/d_{\text{in}}$, set $V_i' = ?$ to be an erasure; otherwise, set $V_i' = V_i$ in $\mathbb{F}_{2^{Cm \times R_{\text{in}} \log n}}$.

Step3: Decode $\mathbf{V}' = (V_1', V_2', \ldots, V_{n/\log n}')$ with both errors and erasures using decoding algorithms for the RS outer code.

Denote the number of errors by e and number of erasures by s, an RS code with minimum distance d_{out} can decode correctly if $2e + s < d_{\text{out}}$. The following Lemma shows that in expectation it is indeed the case if the total number of errors is less than $(d_{\text{out}} \cdot d_{\text{in}})/2$.

Lemma 10. *Let* $\mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_{n/\log n})$ *be the codeword sent, suppose* $d_{\hat{\mathbf{T}}}(\mathbf{W}, \mathbf{Y}) < (d_{\text{out}} \cdot d_{\text{in}})/2$ *holds. If* $\mathbf{V}'$ *has* e *errors and* s *erasures compared to* $\mathbf{W}$*, then* $E[2e + s] < d_{\text{out}}$.

Proof: For $1 \leq i \leq n/\log n$, let $\delta_i = d_{\hat{\mathbf{T}}}(\mathbf{W}_i, \mathbf{Y}_i)$, then

$$\sum_{i=1}^{n/\log n} \delta_i < \frac{d_{\text{out}} \cdot d_{\text{in}}}{2}. \tag{11}$$

Define two indicator random variables $\mathbb{1}_i^{err}$ and $\mathbb{1}_i^{ers}$ for the event of an error and an erasure at V_i' respectively. In the following, we show through case analysis that

$$E[2 \cdot \mathbb{1}_i^{err} + \mathbb{1}_i^{ers}] \leq \frac{2\delta_i}{d_{\text{in}}}. \tag{12}$$

Case 1 ($\mathbf{W}_i = C_{\text{in}}(\mathbf{V}_i)$). For the erasure event, we have $E[\mathbb{1}_i^{ers}] = \Pr(\mathbb{1}_i^{ers} = 1) = 2\omega_i/d_{\text{in}}$. For the error event, if $V_i' = ?$ is an erasure, then $\mathbb{1}_i^{err} = 0$; otherwise $\mathbf{W}_i = C_{\text{in}}(\mathbf{V}_i) = C_{\text{in}}(\mathbf{V}_i')$, which means that there is no error $\mathbb{1}_i^{err} = 0$. By the definition of ω_i, we have $\omega_i \leq d_{\hat{\mathbf{T}}}\left(C_{\text{in}}(\mathbf{V}_i), \mathbf{Y}_i\right) = d_{\hat{\mathbf{T}}}\left(\mathbf{W}_i, \mathbf{Y}_i\right) = \delta_i$. Hence, in this case $E[2 \cdot \mathbb{1}_i^{err} + \mathbb{1}_i^{ers}] = 2\omega_i/d_{\text{in}} \leq 2\delta_i/d_{\text{in}}$.

Case 2 ($\mathbf{W}_i \neq C_{\text{in}}(\mathbf{V}_i)$). In this case, still we have $E[\mathbb{1}_i^{ers}] = 2\omega_i/d_{\text{in}}$. When $\mathbf{V}_i'$ is not an erasure we have $\mathbf{W}_i \neq C_{\text{in}}(\mathbf{V}_i')$, which means that $\mathbf{V}_i'$ has an error. Hence, $E[\mathbb{1}_i^{err}] = 1 - \Pr(\mathbb{1}_i^{ers} = 1) = 1 - 2\omega_i/d_{\text{in}}$ and $E[2 \cdot \mathbb{1}_i^{err} + \mathbb{1}_i^{ers}] = 2 - 2\omega_i/d_{\text{in}}$. In the following, we show that $\omega_i + \delta_i \geq d_{\text{in}}$ through case analysis.

- Case 2.1 ($\omega_i = d_{\hat{\mathbf{T}}}\left(C_{\text{in}}(\mathbf{V}_i), \mathbf{Y}_i\right) < d_{\text{in}}/2$). In this case, $\omega_i + \delta_i = d_{\hat{\mathbf{T}}}\left(C_{\text{in}}(\mathbf{V}_i), \mathbf{Y}_i\right) + d_{\hat{\mathbf{T}}}(\mathbf{W}_i, \mathbf{Y}_i) \geq d_{\hat{\mathbf{T}}}(C_{\text{in}}(\mathbf{V}_i), \mathbf{W}_i) \geq d_{\text{in}}$, where the first inequality is by triangle inequality and the second inequality follows by the minimum distance of the codebook since $\mathbf{W}_i \neq C_{\text{in}}(\mathbf{V}_i)$ are two different codewords.
- Case 2.2 ($\omega_i = d_{\text{in}}/2 \leq d_{\hat{\mathbf{T}}}\left(C_{\text{in}}(\mathbf{V}_i), \mathbf{Y}_i\right)$). In this case, $\delta_i = d_{\hat{\mathbf{T}}}(\mathbf{W}_i, \mathbf{Y}_i) \geq d_{\hat{\mathbf{T}}}\left(C_{\text{in}}(\mathbf{V}_i), \mathbf{Y}_i\right) \geq d_{\text{in}}/2$, where the first inequality is by the fact that we decode $\mathbf{Y}_i$ to $\mathbf{V}_i$ which minimize the transform metric distance. Hence, $\omega_i + \delta_i \geq d_{\text{in}}$.

Hence for Case 2, we have shown that $E[2 \cdot \mathbb{1}_i^{err} + \mathbb{1}_i^{ers}] = 2 - 2\omega_i/d_{\text{in}} \geq 2\delta_i/d_{\text{in}}$.

Hence we have shown (12), and combining with (11) we have

$$\begin{aligned} E[2e + s] &= E[\sum_{i=1}^{n\log n} 2 \cdot \mathbb{1}_i^{err} + \mathbb{1}_i^{ers}] \\ &= \sum_{i=1}^{n\log n} E[2 \cdot \mathbb{1}_i^{err} + \mathbb{1}_i^{ers}] \\ &\leq \sum_{i=1}^{n\log n} \frac{2\delta_i}{d_{\text{in}}} \\ &< \frac{2}{d_{\text{in}}} \cdot \frac{d_{\text{out}} \cdot d_{\text{in}}}{2} = d_{\text{out}}. \end{aligned}$$

□

VII. Conclusion

In this work we investigate upper and lower bounds on the coding rates of end-to-end error-correcting codes for worst-case binary-error networks. We discuss that this model is appropriate for highly dynamic wireless networks, wherein the noise-levels on individual links might be hard to accurately estimate. The abstracted network channel in Section II-C and the transform metric in Section IV might be of independent interests for other applications. We demonstrate significantly better performance for our proposed schemes, compared to prior benchmark schemes. We also discuss the practicality by considering regimes where network topology and coding coefficients are unknown, and also methods to reduce encoding/decoding complexity.

A. Discussion

While for ease of exposition the focus of this paper has been on binary extension fields, our techniques translate well to general q-ary base fields. As can be verified via direct computation, each of the corresponding bounds in Theorems 1- 5 change as follows, where $H_q(\cdot)$ denotes the q-ary entropy function $H_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)$.

Theorem 11 (q-ary Hamming-Type Bound)**.** *For all p less than $\frac{C}{E}$, an upper bound on the achievable rate of any code over the worst-case q-ary-error channel is $1 - H_q\left(\frac{E}{C}p\right) + o\left(\frac{\log(Cmn+1)}{Cmn}\right)$.*

Theorem 12 (q-ary Plotkin-Type Bound)**.**

1) For networks with $E \geq \frac{q}{q-1}C$,

i. for all p less than $\left(1 - \frac{qC}{2(q-1)E}\right)\frac{C}{E}$, an upper bound on the achievable rate of any code over the worst-case q-ary-error network channel is $1 - \frac{2(q-1)E^2}{2(q-1)CE - qC^2}p$;

ii. for all p greater than $\left(1 - \frac{qC}{2(q-1)E}\right)\frac{C}{E}$, the asymptotic rate achieved by any code over the worst-case q-ary-error network channel is 0.

2) For networks with $E < \frac{q}{q-1}C$,

i. for all p less than $\frac{q-1}{2q}$, an upper bound on the achievable rate of any code over the worst-case q-ary-error network channel is $1 - \frac{2q}{q-1}p$;

ii. for all p greater than $\frac{q-1}{2q}$, the asymptotic rate achieved by any code over the worst-case q-ary-error network channel is 0.

Theorem 13 (q-ary Elias-Bassalygo-Type Bound)**.** *For all* p *less than* $\frac{C}{E}\left(1-\frac{qC}{2(q-1)E}\right)$*, an upper bound on the achievable rate of any code over the worst-case* q*-ary-error network channel is* $1-H_q\left(\frac{(q-1)E}{qC}(1-\sqrt{1-\frac{2q}{q-1}p})\right)+o\left(\frac{\log(Cmn+1)}{Cmn}\right)$.

Theorem 14 (q-ary Gilbert-Varshamov-Type Bound)**.**

1) Coherent GV-type network codes achieve a rate of at least $1-\frac{E}{C}H_q(2p)-o\left(\frac{\log(2pEmn+1)}{Cmn}\right)$.

2) Non-coherent GV-type network codes achieve a rate of at least $1-\frac{E}{C}H_q(2p)-o\left(\frac{\log(2pEmn+1)+CEm}{Cmn}\right)$.

Theorem 15 (q-ary Zyablov-Type Bound)**.** *Concatenation network codes achieve a rate of at least*

$$\max_{0<r<1-\frac{E}{C}H_q(2p)} r\cdot\left(1-\frac{2p}{H_q^{-1}\left(\frac{C}{E}(1-r)\right)}\right).$$

References

[1] Q. Wang, S. Jaggi, and S.-Y. R. Li, "Binary error correcting network codes," in *Proc. of IEEE Information Theory Workshop*, Paraty, Brazil, October 2011.

[2] L. Song, R. W. Yeung, and N. Cai, "A separation theorem for single-source network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 1861–1871, 2006.

[3] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.

[4] D. Silva, F. R. Kschischang, and R. Kötter, "A rank-metric approach to error control in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 3951–3967, 2008.

[5] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient network coding in the presence of Byzantine adversaries," in *Proc. 26th IEEE Int. Conf. on Computer Commun.*, Anchorage, AK, May 2007, pp. 616–624.

[6] N. Cai and R. W. Yeung, "Network coding and error correction," in *Proc. of IEEE Information Theory Workshop*, Bangalore, India, October 2002.

[7] R. W. Yeung, N. Cai *et al.*, "Network error correction, i: Basic concepts and upper bounds," *Communications in Information & Systems*, vol. 6, no. 1, pp. 19–35, 2006.

[8] N. Cai, R. W. Yeung *et al.*, "Network error correction, ii: Lower bounds," *Communications in Information & Systems*, vol. 6, no. 1, pp. 37–54, 2006.

[9] S. Yang, R. W. Yeung, and C. K. Ngai, "Refined coding bounds and code constructions for coherent network error correction," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1409–1424, 2011.

[10] S. Yang, R. W. Yeung, and Z. Zhang, "Weight properties of network codes," *European Transactions on Telecommunications*, vol. 19, no. 4, pp. 371–383, 2008.

[11] Z. Zhang, "Linear network error correction codes in packet networks," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 209–218, 2008.

[12] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. of IEEE International Symposium on Information Theory*, Yokohama, Japan, June 2003.

[13] S. P. Borade, "Network information flow: Limits and achievability," in *Proc. of IEEE International Symposium on Information Theory*, Lausanne, Switzerland, June 2002.

[14] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[15] S. Katti, D. Katabi, H. Balakrishnan, and M. Médard, "Symbol-level network coding for wireless mesh networks," in *Proc. ACM SIGCOMM*, Seattle, WA, 2008.

[16] D. Silva and F. R. Kschischang, "Universal secure network coding via rank-metric codes," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1124–1135, 2011.

[17] ——, "Using rank-metric codes for error correction in random network coding," in *Proc. of IEEE International Symposium on Information Theory*, Nice, June 2007, pp. 796–800.

[18] ——, "On metrics for error correction in network coding," *IEEE Transactions on Information Theory*, vol. 55, no. 12, pp. 5479–5490, 2009.

[19] T. Etzion and A. Vardy, "Error-correcting codes in projective space," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1165–1173, 2011.

[20] V. Skachek, O. Milenkovic, and A. Nedić, "Hybrid noncoherent network coding," *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3317–3331, 2013.

[21] U. Martínez-Peñas, "On the similarities between generalized rank and hamming weights and their applications to network coding," *IEEE Transactions on Information Theory*, vol. 62, no. 7, pp. 4081–4095, 2015.

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

[23] R. Lidl and H. Niederreiter, *Finite fields*. Cambridge university press, 1997, vol. 20.

[24] S. Jaggi, M. Effros, T. Ho, and M. Médard, "On linear network coding," in *Proceedings of 42nd Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, 2004.

[25] E. N. Gilbert, "A comparison of signalling alphabets," *Bell System Technical Journal*, vol. 31, no. 3, pp. 504–522, 1952.

[26] R. Varshamov, "Estimate of the number of signals in error correcting codes," in *Dokl. Akad. Nauk SSSR*, vol. 117, no. 5, 1957, pp. 739–741.

[27] R. Jurrius and R. Pellikaan, "The extended coset leader weight enumerator," in *Proc. of IEEE International Symposium on Information Theory*, Benelux, 2009.

[28] T. Helleseth, "The weight distribution of the coset leaders for some classes of codes with related parity-check matrices," *Discrete Mathematics*, vol. 28, no. 2, pp. 161–171, 1979.

[29] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, pp. 147–160, 1950.

[30] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley and Sons, 1991.

[31] M. Plotkin, "Binary codes with specified minimum distance," *IRE Transactions on Information Theory*, vol. 6, no. 4, pp. 445–450, 1960.

[32] L. A. Bassalygo, "New upper bounds for error correcting codes," *Problemy Peredachi Informatsii*, vol. 1, no. 4, pp. 41–44, 1965.

[33] G. D. Forney, "Generalized minimum distance decoding," *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125–131, 1966.