Refined Coding Bounds and Code Constructions for Coherent Network Error Correction

Shenghao Yang, Raymond W. Yeung, *Fellow, IEEE*, Chi Kin Ngai

***Abstract*—Coherent network error correction is the error-control problem in network coding with the knowledge of the network codes at the source and sink nodes. With respect to a given set of local encoding kernels defining a linear network code, we obtain refined versions of the Hamming bound, the Singleton bound and the Gilbert-Varshamov bound for coherent network error correction. Similar to its classical counterpart, this refined Singleton bound is tight for linear network codes. The tightness of this refined bound is shown by two construction algorithms of linear network codes achieving this bound. These two algorithms illustrate different design methods: one makes use of existing network coding algorithms for error-free transmission and the other makes use of classical error-correcting codes. The implication of the tightness of the refined Singleton bound is that sink nodes with higher maximum flow values can have higher error correction capabilities.**

***Index Terms*—Network error correction, network coding, Hamming bound, Singleton bound, Gilbert-Varshamov bound, network code construction.**

I. Introduction

Network coding has been extensively studied for multicasting information in a directed communication network when the communication links in the network are error free. It was shown by Ahlswede *et al.* [1] that the network capacity for multicast satisfies the max-flow min-cut theorem, and this capacity can be achieved by network coding. Li, Yeung, and Cai [2] further showed that it is sufficient to consider linear network codes only. Subsequently, Koetter and Médard [3] developed a matrix framework for network coding. Jaggi *et al.* [4] proposed a deterministic polynomial-time algorithm to construct linear network codes. Ho *et al.* [5] showed that optimal linear network codes can be efficiently constructed by a randomized algorithm with an exponentially decreasing probability of failure.

A. Network Error Correction

Researchers also studied how to achieve reliable communication by network coding when links are not perfect. For example, network transmission may suffer from link failures [3], random errors [6] and maliciously injected errors [7]. We refer to these distortions in network transmission collectively as *errors*, and the network coding techniques for combating errors as *network error correction*.

Raymond W. Yeung and Chi Kin Ngai are with the Department of Information Engineering, The Chinese University of Hong Kong. Emails: whyeung, ckngai2@ie.cuhk.edu.hk

Shenghao Yang was with the Department of Information Engineering, The Chinese University of Hong Kong. He is now with the University of Waterloo. Email: shyang5@ie.cuhk.edu.hk

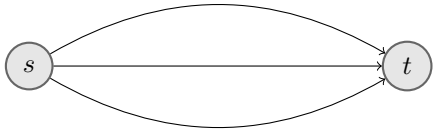


Fig. 1. This is a classical error correction example, where s is the source node and t is the sink node. This model is extensively studied by algebraic coding.

Fig. 1 shows one special case of network error correction with two nodes, one source node and one sink node, which are connected by parallel links. This is the model studied in classical algebraic coding theory [8], [9], a very rich research field for the past 50 years.

Cai and Yeung [6], [10], [11] have extended the study of algebraic coding from classical error correction to network error correction. They generalized the Hamming bound (sphere-packing bound), the Singleton bound and the Gilbert-Varshamov bound (sphere-covering bound) in classical error correction coding to network coding. Network error correction in packet networks has been studied by Zhang [12], [13], where they introduced an algebraic definition of the minimum distance for linear network codes and studied the decoding problem. The relation between network coding and maximum distance separation (MDS) codes in classical algebraic coding [14] was clarified in [15].

In [6], [10], [11], the common assumption is that the sink nodes know the network topology as well as the network code used in transmission. This class of works are referred to as *coherent network error correction*. By constract, network error correction without this assumption is referred to as *noncoherent network error correction*.[1] When using the deterministic construction of linear network codes [2], [4], the network transmission is usually regarded as "coherent". For random network coding, the network transmission is usually regarded as "noncoherent". However, it is possible to use noncoherent transmission for deterministicly constructed network codes and use coherent transmission for randomly constructed network codes.

In [16], Yang *et al.* developed a framework for characterizing error correction/detection capabilities of network codes for coherent network error correction. Their findings are summarized as follows. First, the error correction/detection capabilities of a network code is completely characterized

[1] Coherent and noncoherent transmissions for network coding are analogous to coherent and noncoherent transmissions for multiple antenna channels in wireless communications.

by a two-dimensional region of parameters which reduce to the minimum Hamming distance when 1) the network code is linear, and 2) the weight measure on the error vectors is the Hamming weight. For a nonlinear network code, two different minimum distances are needed for characterizing the capabilities of the code for error correction and for error detection. This leads to the discovery that for a nonlinear network code, the number of correctable errors can be more than half of the number of detectable errors. (For classical algebraic codes, the number of correctable errors is always the largest integer not greater than half of the number of detectable errors.) Further, for the general case, an equivalence relation on weight measures is defined and it is shown that weight measures belonging to the same equivalence class lead to the same minimum weight decoder. In the special case of network coding, four weight measures, including the Hamming weight and others that have been used by various authors [7], [12], [17], are proved to be in the same equivalent class for linear network codes.

Network error detection by random network coding has been studied by Ho *et al.* [18]. Jaggi *et al.* [7], [19], [20] have developed random algorithms for network error correction with various assumptions on the adversaries. Parts of the works of Zhang [12], [13] consider packet network error correction when the network code is not known by receivers. They studied decoding algorithms that can correct different kinds of errors and gave a sufficient condition for correct decoding in terms of the minimum distance. The distribution of the minimum distance when applying random network coding is bounded by Balli, Yan and Zhang [21]. They also studied decoding network error-correcting codes beyond the error correction capability [22].

Koetter and Kschischang [23] introduced a general framework for noncoherent network error correction. In their framework, messages are modulated as subspaces, so a code for noncoherent network error correction is also called a subspace code. They proved a Singleton bound, a sphere-packing bound and a sphere-covering bound for subspace codes. Using rank-metric codes, Silva and Kschischang [24] constructed suboptimal subspace codes and studied the decoding algorithms.

B. Paper Outline

In this paper, we follow the framework in [16] to study the design of linear network codes for coherent network error correction.

The coding bounds for coherent network error correction obtained in [6], [10], [11] take only one sink node with the smallest maximum flow from the source node into consideration. We observe that each sink node can be considered individually and the sink node with larger maximum flow can potentially have higher error correction/detection capability. These observations lead to the refined versions of the Hamming bound, the Singleton bound and the Gilbert-Varshamov bound for network error correction to be proved in this work. By way of the weight properties of network coding, the proof of these bounds are as transparent as the their classical counterparts for linear network codes. By contrast, the proofs of the original versions of these bounds (not necessarily for linear network codes) in [10], [11] are considerably more complicated. The refined Singleton bound is also implicitly obtained by Zhang [12] independently. When applying to classical error correction, these bounds reduce to the classical Hamming bound, the classical Singleton bound and the classical Gilbert-Varshamov bound, respectively.

Similar to its classical counterpart, this refined Singleton bound is tight for linear network codes. The tightness of this refined bound is shown by two construction algorithms of linear network codes achieving the bound. A linear network code consists of two parts, a codebook and a set of local encoding kernels (defined in Section II). Our first algorithm finds a codebook based on a given set of local encoding kernels. The set of local encoding kernels that meets our requirement can be found by the polynomial-time construction in [4]. The second algorithm finds a set of of local encoding kernels based on a given classical error-correcting code satisfying a certain minimum distance requirement as the codebook. These two algorithms illustrate different design methods: one makes use of existing network coding algorithms for error-free transmission and the other makes use of classical error-correcting codes.

Various parts of this paper have appeared in [25], [26]. Subsequent to [25], based on the idea of static network codes [3], Matsumoto [27] proposed an algorithm for constructing linear network codes achieving the refined Singleton bound. In contrast to ours, their algorithm designs the codebook and the local encoding kernels together. All these three algorithms are shown in this paper to have similar time complexity and similar field size requirements.

This paper is organized as follows. In Section II, we formulate the network error correction problem and review some previous works. The refined coding bounds for coherent network error correction are proved in Section III. In Section IV, the tightness of the refined Singleton bound is proved, and the first construction algorithm is given. In Section V, we introduce another construction algorithm that can achieve the refined Singleton bound. In the last section, we summarize our work and discuss future work.

II. Network Error-Correcting Problem

A. Problem Formulation

A communication network is represented by a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges in the graph. There can be multiple edges between a pair of nodes, and each edge represents a communication link with unit capacity, i.e., it can transmit one symbol in a finite field $\mathbb{F}$. We assume all edges are delay free, for which the network is still stable since the network is acyclic. We assume an order on the edge set $\mathcal{E}$ which is consistent with the associated partial order of the directed acyclic network $\mathcal{G}$. We call node a (node b) the tail (head) of edge $e = (a, b)$, denoted by $tail(e)$ ($head(e)$). Let $In(a) = \{e \in \mathcal{E} : head(e) = a\}$ and $Out(a) = \{e \in \mathcal{E} : tail(e) = a\}$ be the sets of incoming edges and outgoing edges of node a, respectively.

A *multicast network* $(\mathcal{G}, s, \mathcal{T})$ is a triple that includes a network $\mathcal{G}$, a source node $s \in \mathcal{V}$, which can inject symbols on its outgoing edges, and a set of sink nodes $\mathcal{T} \subset \mathcal{V}$, each of which can receive symbols on its incoming edges. Without loss of generality, we assume $In(s) = \emptyset$. Let $n_s = |Out(s)|$. The source node observes a message vector consisting of ω symbols taken from $\mathbb{F}$. The source node s encodes the message vector into a row vector $\mathbf{x} = [x_e, e \in Out(s)] \in \mathbb{F}^{n_s}$, called the *codeword*. The set of all codewords is the *codebook*, denoted by $\mathcal{C}$. Note that we do not require $\mathcal{C}$ to be a linear space. The source node s transmits the codeword by mapping its n_s components onto the edges in $Out(s)$.

An *error vector* $\mathbf{z}$ is an $|\mathcal{E}|$-dimensional row vector over the field $\mathbb{F}$ with the ith component representing the error on the ith edge in $\mathcal{E}$. An *error pattern* is a subset of $\mathcal{E}$. An error vector is said to match an error pattern if all the errors occur on the edges in the error pattern. The set of all error vectors that *match* error pattern ρ is denoted by ρ^*. Let $\rho_{\mathbf{z}}$ be the error pattern corresponding to the non-zero components of an error vector $\mathbf{z}$.

Let $\bar{F}_e$ and F_e be the input and output of edge e, respectively, and the error on the edge be z_e. The following relation holds:

$$F_e = \bar{F}_e + z_e. \tag{1}$$

For any set of edges ρ, form two row vectors

$$F_\rho = [F_e, e \in \rho], \tag{2}$$

and

$$\bar{F}_\rho = [\bar{F}_e, e \in \rho]. \tag{3}$$

A network code on network $\mathcal{G}$ is a codebook $\mathcal{C} \subseteq \mathbb{F}^{n_s}$ and a family of local encoding functions $\{\bar{\beta}_e : e \in \mathcal{E} \setminus Out(s)\}$, where $\bar{\beta}_e : \mathbb{F}^{|In(tail(e))|} \to \mathbb{F}$, such that

$$\bar{F}_e = \bar{\beta}_e(F_{In(tail(e))}). \tag{4}$$

Communication over the network with the code defined above is in an upstream-to-downstream order consistent with the partial order of the edges. For a codeword $\mathbf{x}$ and an error vector $\mathbf{z}$, once the network code is specified by (1) and (4), the symbol $\bar{F}_e$ can be determined inductively for all $e \in \mathcal{E}$ with the boundary condition $\bar{F}_{Out(s)} = \mathbf{x}$. When we want to indicate the dependence of $\bar{F}_e$ and F_e on $\mathbf{x}$ and $\mathbf{z}$ explicitly, we will write them as $\bar{F}_e(\mathbf{x}, \mathbf{z})$ and $F_e(\mathbf{x}, \mathbf{z})$, respectively.

If $\bar{\beta}_e$ is a linear function for all $e \in \mathcal{E} \setminus Out(s)$, i.e.,

$$\bar{F}_e = \sum_{e' \in \mathcal{E}} \beta_{e',e} F_{e'}, \tag{5}$$

we say that the network code is *linear*, where $\beta_{e',e}$ is called the *local encoding kernel* from edge e' to edge e. The local encoding kernel $\beta_{e',e}$ can be non-zero only if $e' \in In(tail(e))$. Define the $|\mathcal{E}| \times |\mathcal{E}|$ one-step transformation matrix $\mathbf{K} = [K_{i,j}]$ in network $\mathcal{G}$ as $K_{i,j} = \beta_{e_i,e_j}$. For an acyclic network, $\mathbf{K}^N = \mathbf{0}$ for some positive integer N. Define the transfer matrix of the network by $\mathbf{F} = (\mathbf{I} - \mathbf{K})^{-1}$ [3].

For a set of edges ρ, define a $|\rho| \times |\mathcal{E}|$ matrix $\mathbf{A}_\rho = [A_{i,j}]$ by

$$A_{i,j} = \begin{cases} 1 & e_j \text{ is the } i\text{th edge in } \rho, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

By applying the order on $\mathcal{E}$ to ρ, the $|\rho|$ nonzero columns of $\mathbf{A}_\rho$ form an identity matrix. To simplify notation, we write $\mathbf{F}_{\rho,\rho'} = \mathbf{A}_\rho \mathbf{F} \mathbf{A}_{\rho'}^\top$. For input $\mathbf{x}$ and error vector $\mathbf{z}$, the output of the edges in ρ is

$$F_\rho(\mathbf{x}, \mathbf{z}) = (\mathbf{x}\mathbf{A}_{Out(s)} + \mathbf{z})\mathbf{F}\mathbf{A}_\rho^\top \tag{7}$$

$$= \mathbf{x}\mathbf{F}_{Out(s),\rho} + \mathbf{z}\mathbf{F}\mathbf{A}_\rho^\top. \tag{8}$$

Writing $F_v(\mathbf{x}, \mathbf{z}) = F_{In(v)}(\mathbf{x}, \mathbf{z})$ for a node v, the received vector for a sink node t is

$$F_t(\mathbf{x}, \mathbf{z}) = \mathbf{x}\mathbf{F}_{s,t} + \mathbf{z}\mathbf{F}_t, \tag{9}$$

where $\mathbf{F}_{s,t} = \mathbf{F}_{Out(s),In(t)}$, and $\mathbf{F}_t = \mathbf{F}\mathbf{A}_{In(t)}^\top$. Here $\mathbf{F}_{s,t}$ and $\mathbf{F}_t$ are the transfer matrices for message transmission and error transmission, respectively, for sink node t.

For coherent network error correction, $\mathbf{F}_{s,t}$ and $\mathbf{F}_t$ are known by the sink nodes for decoding. In the network shown in Fig. 1, $\mathbf{F}_{s,t}$ and $\mathbf{F}_t$ are the identity matrix, and the problem becomes that of classical error correction.

B. Existing Results

In [16], Yang *et al.* developed a framework for characterizing error correction/detection capabilities of linear network codes for coherent network error correction. Further, equivalence classes of weight measures on error vectors are defined. Weight measures in the same equivalence class have the same characterizations of error correction/detection capabilities and induce the same minimum weight decoder. Four weight measures, including the Hamming weight and others that have been used by various authors [7], [12], [17], are proved to be in the same equivalent class for linear network codes. Hence they are all equivalent for error correction and detection.

In the rest of this paper, we only consider the Hamming weight on error vectors. For sink node t, define

$$\Phi_t(c) = \{\mathbf{z}\mathbf{F}_t : \mathbf{z} \in \mathbb{F}^{|\mathcal{E}|},\ w_H(\mathbf{z}) \le c\}, \tag{10}$$

where $w_H(\mathbf{z})$ is the Hamming weight of an error vector $\mathbf{z}$.

Definition 1: Consider a linear network code with codebook $\mathcal{C}$. For each sink node t, define the *distance measure*

$$D_t(\mathbf{x}_1, \mathbf{x}_2) = \min\{c : (\mathbf{x}_1 - \mathbf{x}_2)\mathbf{F}_{s,t} \in \Phi_t(c)\}, \tag{11}$$

and define the *minimum distance* of the codebook

$$d_{\min,t} = \min_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}} D_t(\mathbf{x}_1, \mathbf{x}_2). \tag{12}$$

Definition 2: Minimum Weight Decoder I at a sink node t with respect to w_H, denoted by MWD_t^I, decodes a received vector $\mathbf{y}$ as follows: First, find all the solutions of the equation

$$F_t(\mathbf{x}, \mathbf{z}) = \mathbf{y} \tag{13}$$

with $\mathbf{x} \in \mathcal{C}$ and $\mathbf{z} \in \mathbb{F}^{|\mathcal{E}|}$ as variables. A pair $(\mathbf{x}, \mathbf{z})$, consisting of the message part $\mathbf{x}$ and the error part $\mathbf{z}$, is said to be a solution if it satisfies (13), and furthermore a minimum weight solution if $w_H(\mathbf{z})$ achieves the minimum among all the solutions. If there exists at least one solution and all the minimum weight solutions have identical message parts, then the decoder outputs the common message part as the decoded

message. Otherwise, the decoder outputs a warning that errors have occurred.

A code is *c-error-correcting* at sink node t if all error vectors $\mathbf{z}$ with $w_H(\mathbf{z}) \leq c$ are correctable by MWD_t^I.

Theorem 1 ([16]): A linear network code is c-error-correcting at sink node t if and only if $d_{\min,t} \geq 2c+1$.

For two subsets $V_1, V_2 \subset \mathbb{F}^{n_s}$, their sum is the set defined by

$$V_1 + V_2 = \{\mathbf{v}_1 + \mathbf{v}_2 : \mathbf{v}_1 \in V_1, \mathbf{v}_2 \in V_2\}. \tag{14}$$

For $\mathbf{v} \in \mathbb{F}^{n_s}$ and $V \subset \mathbb{F}^{n_s}$, we also write $\{\mathbf{v}\} + V$ as $\mathbf{v} + V$.

With respect to the Hamming weight, define the *decoding sphere* of a codeword $\mathbf{x}$ as

$$\Phi_t(\mathbf{x}, c) = \{F_t(\mathbf{x}, \mathbf{z}) : \mathbf{z} \in \mathbb{F}^{|\mathcal{E}|}, w_H(\mathbf{z}) \leq c\} \tag{15}$$

$$= \mathbf{x}\mathbf{F}_{s,t} + \Phi_t(c) \tag{16}$$

for a sink node t and a nonnegative integer c.

Definition 3: If $\Phi_t(\mathbf{x}, c)$ for all $\mathbf{x} \in \mathcal{C}$ are nonempty and disjoint, *Minimum Weight Decoder II* at sink node t, denoted by $\mathrm{MWD}_t^{II}(c)$, decodes a received vector $\mathbf{y}$ as follows: if $\mathbf{y} \in \Phi_t(\mathbf{x}, c)$ for some $\mathbf{x} \in \mathcal{C}$, the decoder outputs $\mathbf{x}$ as the decoded message. If $\mathbf{y}$ is not in any of the decoding spheres, the decoder outputs a warning that errors have occurred.

A code is *c-error-detecting* at sink node t if $\mathrm{MWD}_t^{II}(0)$ exists and all error vector $\mathbf{z}$ with $0 < w_H(\mathbf{z}) \leq c$ are detectable by $\mathrm{MWD}_t^{II}(0)$.

Theorem 2 ([16]): A code is c-error-detecting at sink node t if and only if $d_{\min,t} \geq c+1$.

Erasure correction is error correction with the potential positions of the errors in the network known by the decoder. We can similarly characterize the erasure correction capability of linear network codes by $d_{\min,t}$. Readers are referred to [16] for the details about erasure correction.

The coding bounds on network codes that corresponding to the classical Hamming bound, Singleton bound and Gilbert-Varshamov bound have been proved in [6], [10], [11]. Some of these results are reviewed in the following. Let

$$d_{\min} = \min_{t \in \mathcal{T}} d_{\min,t}, \tag{17}$$

and

$$n = \min_{t \in \mathcal{T}} \mathrm{maxflow}(s, t), \tag{18}$$

where $\mathrm{maxflow}(s,t)$ is the maximum flow value from the node s to node t. In terms of the notion of minimum distance, the Hamming bound and the Singleton bound for network codes obtained in [10] can be restated as

$$|\mathcal{C}| \leq \frac{q^n}{\sum_{i=0}^{\tau} \binom{n}{i}(q-1)^i}, \tag{19}$$

where $\tau = \lfloor \frac{d_{\min}-1}{2} \rfloor$, and

$$|\mathcal{C}| \leq q^{n - d_{\min} + 1}. \tag{20}$$

The tightness of (20) has been proved in [11].

III. Refined Coding Bounds

In this section, we present refined versions of the coding bounds in [6], [10], [11] for linear network codes. In terms of the distance measures developed in [16], the proofs of these bounds are as transparent as the their classical counterparts.

A. Hamming Bound and Singleton Bound

Theorem 3: Consider a linear network code with codebook $\mathcal{C}$, $\mathrm{rank}(\mathbf{F}_{s,t}) = r_t$ and $d_{\min,t} > 0$. Then $|\mathcal{C}|$ satisfies the

1) refined Hamming bound

$$|\mathcal{C}| \leq \min_{t \in \mathcal{T}} \frac{q^{r_t}}{\sum_{i=0}^{\tau_t} \binom{r_t}{i}(q-1)^i}, \tag{21}$$

where $\tau_t = \lfloor \frac{d_{\min,t}-1}{2} \rfloor$, and the

2) refined Singleton bound

$$|\mathcal{C}| \leq q^{r_t - d_{\min,t} + 1}, \tag{22}$$

for all sink node t.

Remark: The refined Singleton bound can be rewritten as

$$d_{\min,t} \leq r_t - \log_q |\mathcal{C}| + 1 \leq \mathrm{maxflow}(s,t) - \log_q |\mathcal{C}| + 1, \tag{23}$$

for all sink node t. Thus the refined Singleton bound suggests that the sink nodes with larger maximum flow values can potentially have higher error correction capabilities. We present algorithms that can achieve this bound in Section IV and V.

Proof: Fix a sink node t. Find r_t linearly independent rows of $\mathbf{F}_{s,t}$ and let ρ_t be the set of edges in $Out(s)$ that corresponds to these r_t linearly independent rows. Note that $\rho_t \subset Out(s) \subset \mathcal{E}$, so that ρ_t can be regarded as an error pattern. Define the set

$$\mathcal{C}_t = \{\mathbf{x}' \in \mathbb{F}^{n_s} : \mathbf{x}'\mathbf{A}_{Out(s)} \in \rho_t^*,\ \mathbf{x}'\mathbf{F}_{s,t} = \mathbf{x}\mathbf{F}_{s,t}, \text{ for some } \mathbf{x} \in \mathcal{C}\}, \tag{24}$$

where the matrix $\mathbf{A}_{Out(s)}$ is defined as (6). Define a mapping

$$\phi_t : \mathcal{C} \to \mathcal{C}_t \tag{25}$$

by $\phi_t(\mathbf{x}) = \mathbf{x}'$ if $\mathbf{x}'\mathbf{F}_{s,t} = \mathbf{x}\mathbf{F}_{s,t}$. Since the rows of $\mathbf{F}_{s,t}$ indexed by ρ_t form a basis for the row space of $\mathbf{F}_{s,t}$, ϕ_t is well-defined. The mapping ϕ_t is onto by the definition of $\mathcal{C}_t$. The mapping ϕ_t is also one-to-one because otherwise there exists $\mathbf{x}' \in \mathcal{C}_t$ such that $\mathbf{x}'\mathbf{F}_{s,t} = \mathbf{x}_1\mathbf{F}_{s,t} = \mathbf{x}_2\mathbf{F}_{s,t}$ for distinct $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$, a contradiction to the assumption that $d_{\min,t} > 0$. Thus the mapping ϕ_t is a one-to-one and onto mapping, which implies that $|\mathcal{C}_t| = |\mathcal{C}|$.

Let

$$\mathcal{Z}_t = \{\mathbf{z} \in \rho_t^* : w_H(\mathbf{z}) \leq \tau_t\}. \tag{26}$$

By Theorem 1, the network code with $\mathcal{C}$ being the codebook that can correct all the errors in $\mathcal{Z}_t$ at sink node t. Since sink node t has the same reception for the transmission of either $\mathbf{x} \in \mathcal{C}$ or $\phi_t(\mathbf{x}) \in \mathcal{C}_t'$ for the same error vector, the network code with $\mathcal{C}_t'$ being the codebook can also correct all the errors in $\mathcal{Z}_t$ at sink node t.

Consider the problem of finding a subset of ρ_t^* as an error-correcting code that can correct all the errors in $\mathcal{Z}_t$. This problem is equivalent to the problem in classical algebraic coding of finding a block code with codeword length r_t that can correct τ_t errors. The vectors in the set

$$\mathcal{C}_t' = \{\mathbf{x}\mathbf{A}_{Out(s)} : \mathbf{x} \in \mathcal{C}_t\} \tag{27}$$

must form such a code, otherwise the network code with $\mathcal{C}_t'$ being the codebook cannot possibly correct all the error vectors in $\mathcal{Z}_t$ at sink node t. Applying the Hamming bound

and the Singleton bound for classical error-correcting codes to $\mathcal{C}'_t$, we have

$$|\mathcal{C}'_t| \leq \frac{q^{r_t}}{\sum_{i=0}^{\tau_t}\binom{r_t}{i}(q-1)^i}, \tag{28}$$

and

$$|\mathcal{C}'_t| \leq q^{r_t - d_{\min,t}+1}. \tag{29}$$

The proof is completed by noting that $|\mathcal{C}| = |\mathcal{C}_t| = |\mathcal{C}'_t|$. ∎

Lemma 4:

$$\frac{q^m}{\sum_{i=0}^{\tau}\binom{m}{i}(q-1)^i} < \frac{q^{m+1}}{\sum_{i=0}^{\tau}\binom{m+1}{i}(q-1)^i} \tag{30}$$

for $\tau \leq m/2$.

Proof: This inequality can be established by considering

$$\frac{q^m}{\sum_{i=0}^{\tau}\binom{m}{i}(q-1)^i} = \frac{q^{m+1}}{\sum_{i=0}^{\tau}\frac{q(m-i+1)}{m+1}\binom{m+1}{i}(q-1)^i} \tag{31}$$

$$< \frac{q^{m+1}}{\sum_{i=0}^{\tau}\binom{m+1}{i}(q-1)^i}, \tag{32}$$

where (32) holds because $\frac{q(m-i+1)}{m+1} > 1$ given that $q \geq 2$ and $i \leq \tau \leq m/2$. ∎

The refined Hamming bound and the refined Singleton bound, as we will show, imply the bounds shown in (19) and (20) but not vice versa.

The refined Hamming bound implies

$$|\mathcal{C}| \leq \frac{q^{r_t}}{\sum_{i=0}^{\tau_t}\binom{r_t}{i}(q-1)^i} \tag{33}$$

$$\leq \frac{q^{r_t}}{\sum_{i=0}^{\tau}\binom{r_t}{i}(q-1)^i} \tag{34}$$

$$\leq \frac{q^{\mathrm{maxflow}(s,t)}}{\sum_{i=0}^{\tau}\binom{\mathrm{maxflow}(s,t)}{i}(q-1)^i} \tag{35}$$

for all sink nodes t, where (34) follows from $\tau \leq \tau_t$, and (35) follows from $r_t \leq \mathrm{maxflow}(s,t)$ and the inequality proved in Lemma 4. By the same inequality, upon minimizing over all sink nodes $t \in \mathcal{T}$, we obtain (19).

Toward verifying the condition for applying the inequality in Lemma 4 in the above, we consider the refined Singleton bound, and obtain

$$1 \leq |\mathcal{C}| \leq q^{r_t - d_{\min,t}+1} \tag{36}$$

or

$$d_{\min,t} - 1 \leq r_t \tag{37}$$

for all $t \in \mathcal{T}$. Then

$$\tau = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor \leq \left\lfloor \frac{d_{\min,t}-1}{2} \right\rfloor \leq \frac{d_{\min,t}-1}{2} \leq \frac{r_t}{2} \tag{38}$$

for all $t \in \mathcal{T}$.

For the refined Singleton bound, we first note that it is maximized when $r_t = \mathrm{maxflow}(s,t)$ for all $t \in \mathcal{T}$. This can be achieved by a *linear broadcast* whose existence was proved in [2], [15]. To show that the refined Singleton bound implies (20), consider

$$|\mathcal{C}| \leq q^{r_t - d_{\min,t}+1} \tag{39}$$

$$\leq q^{r_t - d_{\min}+1} \tag{40}$$

$$\leq q^{\mathrm{maxflow}(s,t) - d_{\min}+1} \tag{41}$$

for all sink nodes t. Then (20) is obtained upon minimizing over all $t \in \mathcal{T}$.

B. Sphere-Packing Bound

Let

$$\Delta_t(\mathbf{x}, d) = \{\mathbf{x}' \in \mathbb{F}^{n_s} : D_t(\mathbf{x}', \mathbf{x}) \leq d\}. \tag{42}$$

Here $D_t(\cdot,\cdot)$ is defined as in Definition 1. Since D_t is a translation invariant metric [16], we have $\Delta_t(\mathbf{x}, d) = \mathbf{x} + \Delta_t(\mathbf{0}, d)$, which implies $|\Delta_t(\mathbf{x}, d)| = |\Delta_t(\mathbf{0}, d)|$. Another fact is that $\Delta_t(\mathbf{0}, d)$ is closed under scalar multiplication, i.e.,

$$\alpha\Delta_t(\mathbf{0}, d) = \{\alpha\mathbf{x} : \mathbf{x} \in \Delta_t(\mathbf{0}, d)\} = \Delta_t(d), \tag{43}$$

where $\alpha \in \mathbb{F}$ and $\alpha \neq 0$.

Lemma 5:

$$\binom{|\mathcal{E}|}{d} q^d \geq |\Delta_t(\mathbf{0}, d)| q^{-(n_s - r_t)} \geq \sum_{i=0}^{d}\binom{r_t}{i}(q-1)^i, \tag{44}$$

where $r_t = \mathrm{rank}(\mathbf{F}_{s,t})$ and $d \leq r_t$.

Proof: Applying the definition of D_t in Definition 1, $\Delta_t(\mathbf{0}, d)$ can be rewrited as

$$\Delta_t(\mathbf{0}, d) = \{\mathbf{x}' \in \mathbb{F}^{n_s} : \mathbf{x}'\mathbf{F}_{s,t} \in \Phi_t(d)\}, \tag{45}$$

where Φ_t is defined in (10). Since in (45), $\mathbf{x}'$ is an n_s-dimensional row vector and the rank of $\mathbf{F}_{s,t}$ is r_t, we have

$$|\Delta_t(\mathbf{0}, d)| = q^{n_s - r_t}|\Phi_t(d)|, \tag{46}$$

i.e.,

$$|\Delta_t(\mathbf{0}, d)| q^{-(n_s - r_t)} = |\Phi_t(d)|. \tag{47}$$

In the case of linear network coding,

$$\Phi_t(d) = \{\mathbf{z}\mathbf{F}_t : w_H(\mathbf{z}) \leq d\}. \tag{48}$$

Since $\mathbf{z}$ is an $|\mathcal{E}|$-dimensional row vector,

$$|\Phi_t(d)| \leq \binom{|\mathcal{E}|}{d} q^d, \tag{49}$$

so that from (47), we obtain the first inequality in (44).

Recall that $\mathbf{F}_{s,t}$ is a submatrix of $\mathbf{F}_t$ containing the rows of $\mathbf{F}_t$ which correspond to the outgoing edges of the source node s. Thus,

$$\Phi_t(d) = \{\mathbf{z}\mathbf{F}_t : w_H(\mathbf{z}) \leq d\} \supset \{\mathbf{x}\mathbf{F}_{s,t} : \ w_H(\mathbf{x}) \leq d\}. \tag{50}$$

Since the rank of $\mathbf{F}_{s,t}$ is r_t, we can lower bound $|\{\mathbf{x}\mathbf{F}_{s,t} : w_H(\mathbf{x}) \leq d\}|$ by

$$|\{\mathbf{x}\mathbf{F}_{s,t} : \ w_H(\mathbf{x}) \leq d\}| \geq \sum_{i=0}^{d}\binom{r_t}{i}(q-1)^i, \tag{51}$$

where the RHS is the number of r_t-dimensional row vectors with the Hamming weight less than or equal to d. Hence

$$|\Phi_t(d)| \geq |\{\mathbf{x}\mathbf{F}_{s,t} : \ w_H(\mathbf{x}) \leq d\}| \geq \sum_{i=0}^{d}\binom{r_t}{i}(q-1)^i. \tag{52}$$

Again by (47), we obtain the second inequality in (44). The lemma is proved. ∎

Using the idea of sphere packing, we have the following stronger version of the refined Hamming bound in Theorem 3.

Theorem 6 (Sphere-packing bound): A linear network code with codebook $\mathcal{C}$ and positive minimum distance $d_{\min,t}$ for all sink node t satisfies

$$|\mathcal{C}| \leq q^{n_s}/|\Delta_t(\mathbf{0}, \tau_t)|, \tag{53}$$

where $\tau_t = \lfloor \frac{d_{\min,t}-1}{2} \rfloor$, for all sink node t.

Proof: For different codewords $\mathbf{x}_1$ and $\mathbf{x}_2$, if there exists

$$\mathbf{x} \in \Delta_t(\mathbf{x}_1, \tau_t) \cap \Delta_t(\mathbf{x}_2, \tau_t), \tag{54}$$

from (42), we have

$$\begin{aligned} D_t(\mathbf{x}_1, \mathbf{x}_2) &\leq D_t(\mathbf{x}_1, \mathbf{x}) + D_t(\mathbf{x}_2, \mathbf{x}) && (55)\\ &\leq 2\tau_t && (56)\\ &\leq d_{\min,t} - 1, && (57)\end{aligned}$$

which is a contradiction to the definition of $d_{\min,t}$. Thus, $\Delta_t(\mathbf{x}_1, \tau_t)$ and $\Delta_t(\mathbf{x}_2, \tau_t)$ are disjoint for different codewords $\mathbf{x}_1$ and $\mathbf{x}_2$. Therefore, $q^{n_s} \geq \sum_{\mathbf{x}\in\mathcal{C}} |\Delta_t(\mathbf{x}, \tau_t)| = |\mathcal{C}||\Delta_t(\mathbf{0}, \tau_t)|$. ∎

Applying the second inequality in Lemma 5, Theorem 6 implies the refined Hamming bound in Theorem 3. Thus Theorem 6 gives a potentially tighter upper bound on $|\mathcal{C}|$ than the refined Hamming bound, although the former is not as explicit as the latter.

C. Gilbert Bound and Varshamov Bound

We have the following sphere-covering type bounds for linear network codes.

Theorem 7 (Gilbert bound): Given a set of local encoding kernels, let $|\mathcal{C}|_{\max}$ be the maximum possible size of the codebook such that the network code has positive minimum distance $d_{\min,t}$ for each sink node t. Then,

$$|\mathcal{C}|_{\max} \geq \frac{q^{n_s}}{|\Delta(\mathbf{0})|}, \tag{58}$$

where

$$\Delta(\mathbf{0}) = \cup_{t\in\mathcal{T}} \Delta_t(\mathbf{0}, d_{\min,t} - 1). \tag{59}$$

Proof: Let $\mathcal{C}$ be the codebook with the maximum possible size. Then for any $\mathbf{x} \in \mathbb{F}^{n_s}$, there exists a codeword $\mathbf{c} \in \mathcal{C}$ and a sink node t such that

$$D_t(\mathbf{x}, \mathbf{c}) \leq d_{\min,t} - 1, \tag{60}$$

since otherwise we could add $\mathbf{x}$ to the codebook while keeping the minimum distance larger than or equal to $d_{\min,t}$ for each sink node t, which is a contradiction to the maximality of $|\mathcal{C}|$.

Let

$$\Delta(\mathbf{c}) = \cup_{t\in\mathcal{T}} \Delta_t(\mathbf{c}, d_{\min,t} - 1). \tag{61}$$

Hence, the whole space $\mathbb{F}^{n_s}$ is contained in the union of $\Delta(\mathbf{c})$ over all codewords $\mathbf{c} \in \mathcal{C}$, i.e.,

$$\mathbb{F}^{n_s} = \cup_{\mathbf{c}\in\mathcal{C}} \Delta(\mathbf{c}). \tag{62}$$

Since $\Delta(\mathbf{c}) = \mathbf{c} + \Delta(\mathbf{0})$, we have $|\Delta(\mathbf{c})| = |\Delta(\mathbf{0})|$. So we deduce that $q^{n_s} \leq |\mathcal{C}||\Delta(\mathbf{0})|$, that is

$$|\mathcal{C}| \geq \frac{q^{n_s}}{|\Delta(\mathbf{0})|}. \tag{63}$$

∎

Theorem 8 (Varshamov bound): Given a set of local encoding kernels, let $\omega_{\max}$ be the maximum possible dimension of the *linear* codebook such that the network code has positive minimum distance $d_{\min,t}$ for each sink node t. Then,

$$\omega_{\max} \geq n_s - \log|\Delta(\mathbf{0})|, \tag{64}$$

where $\Delta(\mathbf{0})$ is defined in (59).

Proof: Let $\mathcal{C}$ be the linear codebook with the maximum possible dimension. We claim that

$$\mathbb{F}^{n_s} = \Delta(\mathbf{0}) + \mathcal{C}. \tag{65}$$

If the claim is true, then

$$q^{n_s} = |\Delta(\mathbf{0}) + \mathcal{C}| \leq |\Delta(\mathbf{0})||\mathcal{C}| = |\Delta(\mathbf{0})|q^{\omega_{\max}}, \tag{66}$$

proving (64).

It is obviously that $\mathbb{F}^{n_s} \supset \Delta(\mathbf{0}) + \mathcal{C}$. So we prove $\mathbb{F}^{n_s} \subset \Delta(\mathbf{0}) + \mathcal{C}$ by contradiction. Assume there exists

$$\mathbf{g} \in \mathbb{F}^{n_s} \setminus (\Delta(\mathbf{0}) + \mathcal{C}). \tag{67}$$

Let $\mathcal{C}' = \mathcal{C} + \langle\mathbf{g}\rangle$. Then $\mathcal{C}'$ is a subspace with dimension $\omega_{\max} + 1$. If $\mathcal{C}' \cap \Delta(\mathbf{0}) \neq \{\mathbf{0}\}$, then there exists a non-zero vector

$$\mathbf{c} + \alpha\mathbf{g} \in \Delta(\mathbf{0}), \tag{68}$$

where $\mathbf{c} \in \mathcal{C}$ and $\alpha \in \mathbb{F}$. Here, $\alpha \neq 0$, otherwise we have $\mathbf{c} = \mathbf{0}$ because $\mathcal{C} \cap \Delta(\mathbf{0}) = \{\mathbf{0}\}$. Since $\Delta_t(\mathbf{0}, d_{\min,t} - 1)$ is closed under scalar multiplication for all $t \in \mathcal{T}$, see from (59) that the same holds for $\Delta(\mathbf{0})$. Thus from (68),

$$\mathbf{g} \in \Delta(\mathbf{0}) - \alpha^{-1}\mathbf{c} \subset \Delta(\mathbf{0}) + \mathcal{C}, \tag{69}$$

which is a contradiction to (67). Therefore, $\mathcal{C}' \cap \Delta(\mathbf{0}) = \{\mathbf{0}\}$, i.e., $\mathcal{C}'$ is a codebook such that the network code has unicast minimum distance larger than or equal to $d_{\min,t}$, which is a contradiction on the maximality of $\mathcal{C}$. The proof is completed.

∎

IV. Tightness of the Singleton Bound and Code Construction

For an $(\omega, (r_t : t \in \mathcal{T}), (d_t : t \in \mathcal{T}))$ linear network code, we refer to one for which the codebook $\mathcal{C}$ is an ω-dimensional subspace of $\mathbb{F}^{n_s}$, the rank of the transfer matrix $\mathbf{F}_{s,t}$ is r_t, and the minimum distance for sink node t is at least d_t, $t \in \mathcal{T}$.

In this section, two algorithms are proposed to construct an $(\omega, (r_t : t \in \mathcal{T}), (d_t : t \in \mathcal{T}))$ linear network code. In the first algorithm, a codebook is constructed after finding a set of local encoding kernels. In the second algorithm, a set of local encoding kernels is found after specifying a codebook. Over sufficiently large finite fields, both algorithms can achieve the refined Singleton bound.

A. Tightness of the Singleton Bound

Theorem 9: Given a set of local encoding kernels with $r_t = \mathrm{rank}(\mathbf{F}_{s,t})$ over a finite field with size q, for every

$$0 \le \omega \le \min_{t\in\mathcal{T}} r_t, \tag{70}$$

there exists a codebook $\mathcal{C}$ with $|\mathcal{C}| = q^\omega$ such that

$$d_{\min,t} = r_t - \omega + 1 \tag{71}$$

for all sink nodes t, provided that q is sufficiently large.

Proof: We start with any given set of local encoding kernels, which determines r_t for all sink nodes t. Fix an ω which satisfies (70). We will then construct an ω-dimensional linear codebook which together with the given set of local encoding kernels constitutes a linear network code that satisfy (71) for all t. Note that (70) and (71) imply

$$d_{\min,t} \ge 1. \tag{72}$$

We now construct the codebook $\mathcal{C}$. Let $\mathbf{g}_1, \cdots, \mathbf{g}_\omega \in \mathbb{F}^{n_s}$ be a sequence of vectors obtained as follows. For each i, $1 \le i \le \omega$, choose $\mathbf{g}_i$ such that

$$\mathbf{g}_i \notin \Delta_t(\mathbf{0}, r_t - \omega) + \langle \mathbf{g}_1, \cdots, \mathbf{g}_{i-1}\rangle \tag{73}$$

for each sink node t. As we will show, this implies

$$\Delta_t(\mathbf{0}, r_t - \omega) \cap \langle \mathbf{g}_1, \cdots, \mathbf{g}_i\rangle = \{\mathbf{0}\} \tag{74}$$

for each sink node t. If such $\mathbf{g}_1, \cdots, \mathbf{g}_\omega$ exist, then we claim that $\mathcal{C} = \langle \mathbf{g}_1, \cdots, \mathbf{g}_\omega\rangle$ is a codebook with the desired properties. To verify this claim, first, we see that $\mathbf{g}_1, \cdots, \mathbf{g}_\omega$ are linearly independent since (73) holds for $i = 1, \cdots, \omega$; second, we have $d_{\min,t} \ge r_t - \omega + 1$ since (74) holds for $i = \omega$. Note that by (22), the refined Singleton bound, we indeed have $d_{\min,t} = r_t - \omega + 1$, namely (71) for any sink node t.

Now we show that $\mathbf{g}_i$ satisfying (73) exists if the field size q is sufficiently large. Observe that

$$\begin{aligned} &|\Delta_t(\mathbf{0}, r_t - \omega) + \langle \mathbf{g}_1, \cdots, \mathbf{g}_{i-1}\rangle| \\ &\le |\Delta_t(\mathbf{0}, r_t - \omega)| q^{i-1} &(75)\\ &\le \binom{|E|}{r_t - \omega} q^{r_t-\omega} q^{n_s - r_t} q^{i-1} &(76)\\ &= \binom{|E|}{r_t - \omega} q^{n_s - \omega + i - 1}, &(77)\end{aligned}$$

where (76) follows from Lemma 5. Considering all sink nodes, we have at most

$$\sum_{t\in\mathcal{T}} \binom{|E|}{r_t - \omega} q^{n_s - \omega + i - 1} \tag{78}$$

vectors that cannot be chosen as $\mathbf{g}_i$. Thus, if

$$q > \sum_{t\in\mathcal{T}} \binom{|E|}{r_t - \omega}, \tag{79}$$

then there exists a vector that can be chosen as $\mathbf{g}_i$ for $i = 1, \cdots, \omega$.

Fix $\mathbf{g}_1, \cdots, \mathbf{g}_i$ that satisfy (73). We now prove by induction that (74) holds for $\mathbf{g}_1, \cdots, \mathbf{g}_i$. If (74) does not hold for $i = 1$, then there exists a non-zero vector $\alpha\mathbf{g}_1 \in \Delta_t(\mathbf{0}, r_t - \omega)$, where $\alpha \in \mathbb{F}$. Since $\Delta_t(\mathbf{0}, r_t - \omega)$ is closed under scalar multiplication and $\alpha \ne 0$, we have $\mathbf{g}_1 \in \Delta_t(\mathbf{0}, r_t - \omega)$, a contradiction to (73) for $i = 1$. Assume (74) holds for $i \le k-1$. If (74) does not hold for $i = k$, then there exists a non-zero vector

$$\sum_{i=1}^{k} \alpha_i \mathbf{g}_i \in \Delta_t(\mathbf{0}, r_t - \omega), \tag{80}$$

where $\alpha_i \in \mathbb{F}$. If $\alpha_k = 0$,

$$\sum_{i=1}^{k-1} \alpha_i \mathbf{g}_i \in \Delta_t(\mathbf{0}, r_t - \omega), \tag{81}$$

a contradiction to the assumption that (74) holds for $i = k-1$. Thus $\alpha_k \ne 0$. Again, by $\Delta_t(\mathbf{0}, r_t - \omega)$ being closed under scalar multiplication, we have

$$\begin{aligned} \mathbf{g}_k &\in \Delta_t(\mathbf{0}, r_t - \omega) - \alpha_k^{-1} \sum_{i=1}^{k-1} \alpha_i \mathbf{g}_i &(82)\\ &\subset \Delta_t(\mathbf{0}, r_t - \omega) + \langle \mathbf{g}_1, \cdots, \mathbf{g}_{k-1}\rangle, &(83)\end{aligned}$$

a contradiction to $\mathbf{g}_k$ satisfying (73). The proof is completed. ■

B. The First Construction Algorithm

The proof of Theorem 9 gives a constrution algorithm for an $(\omega, (r_t : t \in \mathcal{T}), (d_t : t \in \mathcal{T}))$ linear network code for a given set of local encoding kernels such that $\mathrm{rank}(\mathbf{F}_{s,t}) = r_t$ for all $t \in \mathcal{T}$ and it also verifies the correctness of the algorithm when the field size is sufficiently large. The pseudo code for the algorithm is shown below.

Algorithm 1: Construct network codes that achieve the refined Singleton bound.

input : $(\mathcal{G}, s, \mathcal{T})$, $(r_t : t \in \mathcal{T})$, ω, $(d_t : t \in \mathcal{T})$ with $r_t \le \mathrm{maxflow}(s, t)\ \forall t \in \mathcal{T}$
output: local encoding kernels, and $\mathcal{C}$

```
begin
    for i ← 1, ω do
        find g_i such that
        g_i ∉ ∪_t Δ_t(0, d_t − 1) + ⟨g_1, ··· , g_{i−1}⟩ ;
    end
end
```

The analysis of the complexity of the algorithm requires the following lemma implied by Lemmas 5 and 8 in [4].

Lemma 10: Suppose $m \le q$, the field size, and $\mathcal{B}_k \subset \mathbb{F}^n$, $k = 1, \cdots, m$, are subspaces with $\dim(\mathcal{B}_k) < n$. A vector $\mathbf{u} \in \mathbb{F}^n \setminus \cup_{k=1}^m \mathcal{B}_k$ can be found in time $\mathcal{O}(n^3 m + nm^2)$.

Proof: For each $\mathcal{B}_k$ find a vector $\mathbf{a}_k \in \mathbb{F}^n$ such that $\mathbf{a}_k \mathbf{b}^\top = 0$, $\forall \mathbf{b} \in \mathcal{B}_k$. This vector $\mathbf{a}_k$ can be obtained by solving the system of linear equations

$$\mathbf{B}_k \mathbf{a}_k^\top = \mathbf{0}, \tag{84}$$

where $\mathbf{B}_k$ is formed by juxtaposing a set of vectors that form a basis of $\mathcal{B}_k$. The complexity of solving this system of linear equations is $\mathcal{O}(n^3)$.

We inductively construct $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_m$ such that $\mathbf{u}_i\mathbf{a}_k^\top \neq 0$ for all $1 \leq k \leq i \leq m$. If such a construction is feasible, then $\mathbf{u}_m \notin \mathcal{B}_k$, $\forall k \leq m$. Thus, $\mathbf{u} = \mathbf{u}_m \notin \cup_{k=1}^m \mathcal{B}_k$ is the desired vector.

Let $\mathbf{u}_1$ be any vector such that $\mathbf{u}_1\mathbf{a}_1^\top \neq 0$. For $1 \leq i \leq m-1$, if $\mathbf{u}_i\mathbf{a}_{i+1}^\top \neq 0$, we set $\mathbf{u}_{i+1} = \mathbf{u}_i$. Otherwise, find $\mathbf{b}_{i+1}$ such that $\mathbf{b}_{i+1}\mathbf{a}_{i+1}^\top \neq 0$. We choose

$$\alpha \in \mathbb{F} \setminus \{-(\mathbf{b}_{i+1}\mathbf{a}_j^\top)/(\mathbf{u}_i\mathbf{a}_j^\top) : 1 \leq j \leq i\}, \tag{85}$$

and define

$$\mathbf{u}_{i+1} = \alpha\mathbf{u}_i + \mathbf{b}_{i+1}. \tag{86}$$

The existence of such an α follows from $q \geq m > i$.

By construction, we know that

$$\mathbf{u}_{i+1}\mathbf{a}_{i+1}^\top = \alpha\mathbf{u}_i\mathbf{a}_{i+1}^\top + \mathbf{b}_{i+1}\mathbf{a}_{i+1}^\top \tag{87}$$

$$= \mathbf{b}_{i+1}\mathbf{a}_{i+1}^\top \tag{88}$$

$$\neq 0. \tag{89}$$

If $\mathbf{u}_{i+1}\mathbf{a}_j^\top = \alpha\mathbf{u}_i\mathbf{a}_j^\top + \mathbf{b}_{i+1}\mathbf{a}_j^\top = 0$ for some $1 \leq j \leq i$, we have $\alpha = -(\mathbf{b}_{i+1}\mathbf{a}_j^\top)/(\mathbf{u}_i\mathbf{a}_j^\top)$, a contradiction to (85). So, $\mathbf{u}_{i+1}\mathbf{a}_j^\top \neq 0$ for all j such that $1 \leq j \leq i+1$.

Similar to the analysis in [4, Lemma 8], the construction of $\mathbf{u}$ takes time $\mathcal{O}(nm^2)$. Therefore, the overall time complexity is $\mathcal{O}(n^3m + nm^2)$. ∎

We now analyze the time complexity of the algorithm for the representative special case that $r_t = r$ and $d_t = d$ for all $t \in \mathcal{T}$, where $r \leq \min_{t\in\mathcal{T}} \text{maxflow}(s,t)$ and $d \leq r - \omega + 1$.

In the algorithm, Line 3 can be realized using the algorithm in the proof of Lemma 10. Consider $\Delta_t(\mathbf{0}, d-1)$ as the union of $\binom{|\mathcal{E}|}{d-1}$ subspaces of $\mathbb{F}^r$. Line 3 can be realized in time $\mathcal{O}(n_s^3|\mathcal{T}|\binom{|\mathcal{E}|}{d-1} + n_s(|\mathcal{T}|\binom{|\mathcal{E}|}{d-1})^2)$ as proved in Lemma 10, and this line is repeated ω times. Thus the complexity of this algorithm with deterministic realization is

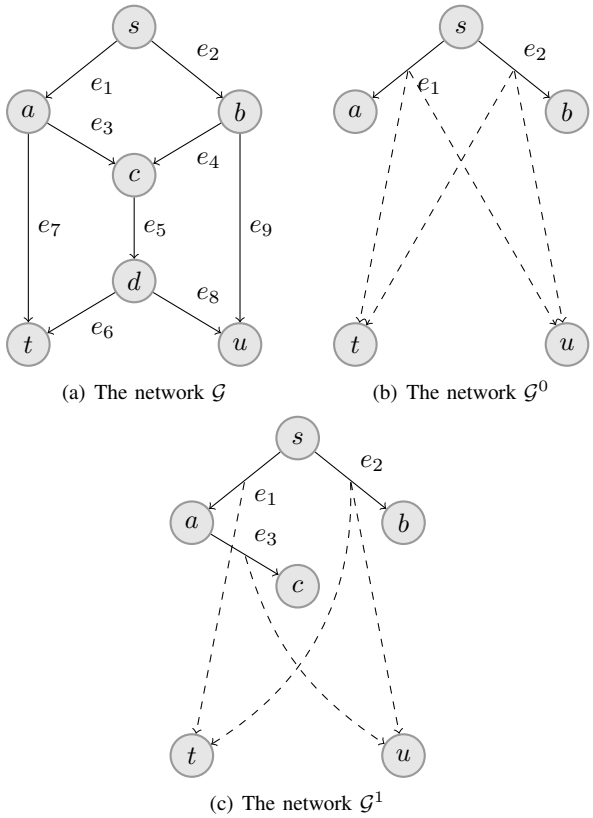
$$\mathcal{O}(\omega n_s^3|\mathcal{T}|\xi + \omega n_s|\mathcal{T}|^2\xi^2), \tag{90}$$

where $\xi = \binom{|\mathcal{E}|}{d-1}$. The deterministic algorithm to construct local encoding kernels by Jaggi *et al.* [4] has complexity $\mathcal{O}(|\mathcal{E}||\mathcal{T}|m(m+|\mathcal{T}|))$, where $m = \min_{t\in\mathcal{T}} \text{maxflow}(s,t)$. Comparing the complexities for constructing the local encoding kernels and finding the codebook (this algorithm), the latter dominates.

For the existence of the code, we require the field size to be sufficiently large. From (79) in the proof of Theorem 9, all finite fields with size larger than $|\mathcal{T}|\binom{|E|}{r-\omega}$ are sufficient. It is straightforward to show that this algorithm can also be realized randomly with high success probability if the field size is much larger than necessary.

V. The Second Construction Algorithm

In this section, we consider another algorithm that constructs an $(\omega, (r_t : t \in \mathcal{T}), (d_t : t \in \mathcal{T}))$ linear network code. At the beginning, the algorithm finds r_t edge-disjoint paths from the source node s to each sink node t using a maximum flow algorithm (for example, finding augmenting paths). We assume that every edge in the network is on at least one of the $\sum_{t\in\mathcal{T}} r_t$ paths we have found. Otherwise, we delete the edges and nodes that are not on any such path, and consider the coding problem for the new network. Note that a network code for the new network can be extended to the original network without changing the minimum distances by assigning zeros to all the local encoding kernels associated with the deleted edges.



(a) The network $\mathcal{G}$ (b) The network $\mathcal{G}^0$

(c) The network $\mathcal{G}^1$

Fig. 2. An example of $\mathcal{G}^0$ and $\mathcal{G}^1$. The dashed lines are not new edges but indicate the incoming edges of t and u. In $\mathcal{G}^0$, both t and u have e_1 and e_2 as their incoming edges. In $\mathcal{G}^1$, $In(t) = \{e_1, e_2\}$ and $In(u) = \{e_3, e_2\}$.

Before describing the algorithm formally, we first introduce a set of notations that facilitate the description of the algorithm. We construct a series of graphs $\mathcal{G}^i, i = 0, 1, \cdots, |\mathcal{E}| - n_s$ as follows. First, $\mathcal{G}^0$ consists of a subgraph of $\mathcal{G}$ containing only the edges in $Out(s)$ (and the associated nodes) and all the sink nodes. Following the order on $\mathcal{E}$, in the ith step, $\mathcal{G}^{i-1}$ is expanded into $\mathcal{G}^i$ by appending the next edge (and the associated node) in $\mathcal{E}$. This step is repeated until $\mathcal{G}^i$ eventually becomes $\mathcal{G}$. Note that $\mathcal{G}^i$ contains $n_s + i$ edges. A sink node t has r_t incoming edges in $\mathcal{G}^i$, where the jth edge is the most downstream edge in the truncation in $\mathcal{G}^i$ of the jth edge-disjoint path from the source node s to sink node t in $\mathcal{G}$. With a slight abuse of notation, we denote the set of incoming edges of the sink node t in $\mathcal{G}^i$ as $In(t)$, when $\mathcal{G}^i$ is implied by the context. Fig. 2 illustrates $\mathcal{G}^0$ and $\mathcal{G}^1$ when $\mathcal{G}$ is the butterfly network.

A. Iterative Formulation of Network Coding

The network $\mathcal{G}^i$ is a multicast network with the source node s and the set of sinks $\mathcal{T}$. The algorithm chooses a proper codebook, and then constructs local encoding kernels starting with $\mathcal{G}^0$. Except for the new edge, all the local encoding

kernels in $\mathcal{G}^{i+1}$ are inherited from $\mathcal{G}^i$.

We define $\mathbf{K}^i$, $\mathbf{F}^i$, F^i_ρ, $\mathbf{z}^i$ and $\mathbf{A}^i_\rho$ for $\mathcal{G}^i$ in view of $\mathbf{K}$, $\mathbf{F}$, F_ρ, $\mathbf{z}$ and $\mathbf{A}_\rho$ defined for $\mathcal{G}$ in Section II, respectively. Writing $F^i_t = F^i_{In(t)}$, $\mathbf{A}^i_s = \mathbf{A}^i_{Out(s)}$ and $\mathbf{A}^i_t = \mathbf{A}^i_{In(t)}$, we have

$$F^i_t(\mathbf{x}, \mathbf{z}^i) = (\mathbf{x}\mathbf{A}^i_s + \mathbf{z}^i)\mathbf{F}^i(\mathbf{A}^i_t)^\top, \tag{91}$$

in view of (7). Further, we can define the minimum distance $d^i_{\min,t}$ corresponding to the sink node t at the ith step as in (12).

Let M be a matrix and $\mathcal{L}$ be any subset of the column index set of M. Let $(M)_j$ be the jth column of M, and $M^{\setminus\mathcal{L}}$ be the matrix obtained by deleting the columns of M indexed by $\mathcal{L}$. Let $\mathbf{z}^{i+1}$ be any error vector in the $(i+1)$th step. Let $(\mathbf{z}^{i+1})_e$ be the component of $\mathbf{z}^{i+1}$ corresponding to edge e, and $\mathbf{z}^{i+1\setminus e}$ be an error vector in the ith step obtained by removing the component of $\mathbf{z}^{i+1}$ corresponding to edge e.

Let e be the new edge in $\mathcal{G}^{i+1}$, i.e., the local encoding kernels associated with e are to be determined in the $(i+1)$th step. Let $\mathbf{k}_e = [\beta_{e',e} : e' \in \mathcal{E}_i]$ be an $(n_s + i)$-dimensional column vector containing the local encoding kernels associated with e. Then

$$\mathbf{F}^{i+1} = \left(\mathbf{I} - \mathbf{K}^{i+1}\right)^{-1} \tag{92}$$

$$= \left(\mathbf{I} - \begin{bmatrix} \mathbf{K}^i & \mathbf{k}_e \\ \mathbf{0} & 0 \end{bmatrix}\right)^{-1} \tag{93}$$

$$= \begin{bmatrix} \mathbf{I} - \mathbf{K}^i & -\mathbf{k}_e \\ \mathbf{0} & 1 \end{bmatrix}^{-1} \tag{94}$$

$$= \begin{bmatrix} (\mathbf{I} - \mathbf{K}^i)^{-1} & (\mathbf{I} - \mathbf{K}^i)^{-1}\mathbf{k}_e \\ \mathbf{0} & 1 \end{bmatrix} \tag{95}$$

$$= \begin{bmatrix} \mathbf{F}^i & \mathbf{F}^i\mathbf{k}_e \\ \mathbf{0} & 1 \end{bmatrix}. \tag{96}$$

The matrix $\mathbf{A}^{i+1}_s$ has one more column with zero components than $\mathbf{A}^i_s$, i.e.,

$$\mathbf{A}^{i+1}_s = \begin{bmatrix} \mathbf{A}^i_s & \mathbf{0} \end{bmatrix}. \tag{97}$$

If the edge e is not on any path from the source node s to sink node t, we only need to append a column with zero components to $\mathbf{A}^i_t$ to form $\mathbf{A}^{i+1}_t$, i.e.,

$$\mathbf{A}^{i+1}_t = \begin{bmatrix} \mathbf{A}^i_t & \mathbf{0} \end{bmatrix}. \tag{98}$$

For this case, we can readily obtain from (91) (with $i+1$ in place of i), (96), (97) and (98) that

$$F^{i+1}_t(\mathbf{x}, \mathbf{z}^{i+1}) = (\mathbf{x}\mathbf{A}^{i+1}_s + \mathbf{z}^{i+1})\mathbf{F}^{i+1}(\mathbf{A}^{i+1}_t)^\top, \tag{99}$$

$$= (\mathbf{x}\mathbf{A}^i_s + (\mathbf{z}^{i+1})^{\setminus e})\mathbf{F}^i(\mathbf{A}^i_t)^\top, \tag{100}$$

$$= F^i_t(\mathbf{x}, (\mathbf{z}^{i+1})^{\setminus e}). \tag{101}$$

If edge e is on the jth edge-disjoint path from the source node s to sink node t, to form $\mathbf{A}^{i+1}_t$, we need to first append a column with zero components to $\mathbf{A}^i_t$, and then move the '1' in the jth row to the last component of that row. That is, if $(\mathbf{A}^i_t)^\top = \begin{bmatrix} \mathbf{b}_1^\top & \mathbf{b}_2^\top & \cdots & \mathbf{b}_{r_t}^\top \end{bmatrix}$, then

$$(\mathbf{A}^{i+1}_t)^\top = \begin{bmatrix} \mathbf{b}_1^\top & \cdots & \mathbf{b}_{j-1}^\top & \mathbf{0} & \mathbf{b}_{j+1}^\top & \cdots & \mathbf{b}_{r_t}^\top \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}. \tag{102}$$

We can then obtain $F^{i+1}_t(\mathbf{x}, \mathbf{z}^{i+1})$ from (91) (with $i+1$ in place of i), (96), (97) and (102) as

$$(F^{i+1}_t(\mathbf{x}, \mathbf{z}^{i+1}))_j = (\mathbf{x}\mathbf{A}^{i+1}_s + \mathbf{z}^{i+1})\mathbf{F}^{i+1}((\mathbf{A}^{i+1}_t)^\top)_j \tag{103}$$

$$= (\mathbf{x}\mathbf{A}^i_s + \mathbf{z}^{i+1\setminus e})\mathbf{F}^i\mathbf{k}_e + (\mathbf{z}^{i+1})_e, \tag{104}$$

and

$$(F^{i+1}_t(\mathbf{x}, \mathbf{z}^{i+1}))^{\setminus\{j\}}$$

$$= (\mathbf{x}\mathbf{A}^{i+1}_s + \mathbf{z}^{i+1})\mathbf{F}^{i+1}((\mathbf{A}^{i+1}_t)^\top)^{\setminus\{j\}} \tag{105}$$

$$= (F^i_t(\mathbf{x}, \mathbf{z}^{i+1\setminus e}))^{\setminus\{j\}}. \tag{106}$$

B. The Weight Preserving Algorithm

Algorithm 2 shows the pseudo code for our second algorithm which at the beginning finds r_t edge disjoint paths from the source node to each sink node t, and initializes $\mathbf{F}$, $\mathbf{A}_s$, and $\mathbf{A}_t, t \in \mathcal{T}$ by $\mathbf{F}^0$, $\mathbf{A}^0_s$, and $\mathbf{A}^0_t, t \in \mathcal{T}$, respectively. The algorithm takes a linear codebook $\mathcal{C}$, such that $d^0_{\min,t} \geq d_t$ for all sink node t. Such a codebook can be effectively constructed by using Reed-Solomon codes.

The main part of this algorithm is a loop starting at Line 7 for updating the local encoding kernels for the edges in $\mathcal{E} \setminus Out(s)$ in an upstream-to-downstream order. Let e be the edge appended to the graph in the ith step. The choice of $\mathbf{k}_e$ is realized by the pseudo codes between Line 7 and Line 24.

For a fixed i, we choose $\mathbf{k}_e$ such that the following *feasible condition* is satisfied:

$$(F^i_t(\mathbf{x}, -\mathbf{z}^i))^{\setminus\mathcal{L}} \neq \mathbf{0} \tag{107}$$

for all the combinations of

C1) $t \in \mathcal{T}$,
C2) $\mathcal{L} \subset \{1, 2, \ldots, r_t\}$ with $0 \leq |\mathcal{L}| \leq d_t - 1$,
C3) non-zero $\mathbf{x} \in \mathcal{C}$, and
C4) error vector $\mathbf{z}^i$ with $w_H(\mathbf{z}^i) \leq d_t - 1 - |\mathcal{L}|$.

The feasible condition is sufficient for $d^i_{\min,t} \geq d_t, t \in \mathcal{T}$, which is equivalent to (107) with $\mathcal{L} = \emptyset$ holding for all $t \in \mathcal{T}$, for all non-zero $\mathbf{x} \in \mathcal{C}$, and all error vector $\mathbf{z}^i$ with $w_H(\mathbf{z}^i) \leq d_t - 1$. So when the algorithm terminates, the code constructed for $\mathcal{G}$ satisfies $d_{\min,t} \geq d_t$ for all $t \in \mathcal{T}$. Even though the feasible condition is stronger than necessary for $d^i_{\min,t} \geq d_t, t \in \mathcal{T}$, as we will see, it is needed for the existence of the local encoding kernels for $k > i$ such that $d^k_{\min,t} \geq d_t$ for all $t \in \mathcal{T}$.

Theorem 11: Given a linear codebook with $d^0_{\min,t} \geq d_t$ for all $t \in \mathcal{T}$, there exist local encoding kernels such that the feasible condition is satisfied for $i = 0, \cdots, |\mathcal{E}| - n_s$ when the field size is larger than $\sum_{t\in\mathcal{T}} \binom{r_t + |\mathcal{E}| - 2}{d_t - 1}$.

Proof: See Section V-C. ∎

Again, we analyze the time complexity of the algorithm for the representative special case that $r_t = r$ and $d_t = d$ for all $t \in \mathcal{T}$, where $r \leq \min_{t\in\mathcal{T}} \text{maxflow}(s, t)$ and $d \leq r - \omega + 1$. For Line 3, the augmenting paths for all the sinks can be found in time $\mathcal{O}(|\mathcal{T}||\mathcal{E}|r)$ [4]. Line 15 and 17 can be realized by solving a system of linear equations which take time $\mathcal{O}(r^3)$ and $\mathcal{O}(1)$, respectively, and each of these two lines is repeated $\mathcal{O}(d|\mathcal{E}||\mathcal{T}|\binom{|\mathcal{E}|}{d-1})$ times. Line 25 can be solved by the method

Algorithm 2: A pseudo code of the proposed algorithm. The superscripts of $\mathbf{F}$, $\mathbf{A}_s$, $\mathbf{A}_t$, $\mathbf{z}$, and F_t are omitted.

input : $(\mathcal{G}, s, \mathcal{T})$, $(r_t : t \in \mathcal{T})$, ω, $(d_t : t \in \mathcal{T})$, a linear codebook $\mathcal{C}$ with $d^0_{\min,t} \geq d_t$, $\forall t \in \mathcal{T}$
output: local encoding kernels, and $\mathcal{C}$
begin
  for *each sink node* t **do**
    choose r_t edge disjoint paths from s to t;
    initialize $\mathbf{A}_t$;
  end
  $\mathbf{F} \leftarrow \mathbf{I}$, $\mathbf{A}_s \leftarrow \mathbf{I}$;
  for *each* $e \in \mathcal{E} \setminus Out(s)$ *in an upstream to downstream order* **do**
    $\Gamma \leftarrow \emptyset$;
    for *each sink node* t **do**
      if *no chosen path from* s *to* t *crosses* e **then**
        $\mathbf{A}_t \leftarrow \begin{bmatrix}\mathbf{A}_t & \mathbf{0}\end{bmatrix}$;
      else e is on the jth path from s to t
        for *each* $\mathcal{L}$ *with* $|\mathcal{L}| \leq d_t - 1$ *and* $j \notin \mathcal{L}$ **do**
          for *each* ρ *with* $|\rho| = d_t - 1 - |\mathcal{L}|$ **do**
            find $\mathbf{x}_0 \neq \mathbf{0}$ and $\mathbf{z}_0$ matching ρ such that $(F_t(\mathbf{x}_0, -\mathbf{z}_0))^{\setminus(\mathcal{L}\cup\{j\})} = \mathbf{0}$;
            if *exist* $\mathbf{x}_0$ *and* $\mathbf{z}_0$ **then**
              $\Gamma \leftarrow \Gamma \cup \{\mathbf{k}: (\mathbf{x}_0\mathbf{A} - \mathbf{z}_0)\mathbf{F}\mathbf{k} = 0\}$;
            end
          end
        end
      end
      update $\mathbf{A}_t$ using (102);
    end
    choose a vector $\mathbf{k}_e$ in $\mathbb{F}_q^{|In(tail(e))|} \setminus \Gamma$;
    $\mathbf{F} \leftarrow \begin{bmatrix}\mathbf{F} & \mathbf{F}\mathbf{k}_e \\ \mathbf{0} & 1\end{bmatrix}$;
  end
end

in Lemma 10 in time $\mathcal{O}(\delta|\mathcal{T}|\binom{r+|\mathcal{E}|-2}{d-1}(\delta^2 + |\mathcal{T}|\binom{r+|\mathcal{E}|-2}{d-1}))$, where δ is the maximum incoming degree of $\mathcal{G}$, and this line is repeated $\mathcal{O}(|\mathcal{E}|)$ times. Under the assumption that each edge is on some chosen path from the source to the sinks, $\delta \leq r|\mathcal{T}|$. Summing up all the parts, we obtain the complexity

$$\mathcal{O}(\delta|\mathcal{E}||\mathcal{T}|\xi'(\delta^2 + |\mathcal{T}|\xi') + r^3 d|\mathcal{E}||\mathcal{T}|\xi), \tag{108}$$

where $\xi' = \binom{r+|\mathcal{E}|-2}{d-1}$.

In Table I, we compare these two algorithms with another algorithm subsequently proposed by Matsumoto [27] for the same purpose. As we can see, all these algorithms have similar complexities.

C. Algorithm Verification

In this section, Theorem 11 is proved by induction on i. After the initialization, for each sink node t, it follows from

TABLE I
COMPARISON OF DETERMINISTIC CONSTRUCTION ALGORITHMS FOR NETWORK ERROR-CORRECTING CODES. $\xi = \binom{|\mathcal{E}|}{d-1}$ AND $\xi' = \binom{r+|\mathcal{E}|-2}{d-1}$.

	field size	Time complexity
Algorithm 1	$\|\mathcal{T}\|\xi$	$\mathcal{O}(n_s\|\mathcal{T}\|\omega\xi(n_s^2 + \|\mathcal{T}\|\xi))$
Algorithm 2	$\|\mathcal{T}\|\xi'$	$\mathcal{O}(\delta\|\mathcal{E}\|\|\mathcal{T}\|\xi'(\delta^2 + \|\mathcal{T}\|\xi') + r^3 d\|\mathcal{E}\|\|\mathcal{T}\|\xi)$
[27, Fig. 2]	$\|\mathcal{T}\|\xi$	$\mathcal{O}(r\|\mathcal{E}\|\|\mathcal{T}\|\xi(\|\mathcal{T}\|\xi + r + d))$

$d^0_{\min,t} \geq d_t$ that (107) holds for $i = 0$, all $\mathcal{L}$ satisfying C2), all $\mathbf{x}$ satisfying C3) and all $\mathbf{z}^0$ satisfying C4). This verifies the feasible condition for $i = 0$.

Assume that up to the kth step, where $0 \leq k < |\mathcal{E}| - n_s$, we can find local encoding kernels such that the feasible condition is satisfied for all $i \leq k$. In the $(k+1)$th step, let e be the edge appended to $\mathcal{G}^k$ to form $\mathcal{G}^{k+1}$. We will show that there exists $\mathbf{k}_e$ such that the feasible condition continues to hold for $i = k + 1$.

We first consider a sink node t (which does not necessarily exist) for which edge e is not on any path from the source node s to t. For all $\mathcal{L}$ satisfying C2), all $\mathbf{x}$ satisfying C3) and all $\mathbf{z}^{k+1}$ satisfying C4) with $k+1$ in place of i, we have

$$(F_t^{k+1}(\mathbf{x}, -\mathbf{z}^{k+1}))^{\setminus\mathcal{L}} = (F_t^k(\mathbf{x}, -(\mathbf{z}^{k+1})^{\setminus e}))^{\setminus\mathcal{L}} \tag{109}$$
$$\neq \mathbf{0}, \tag{110}$$

where (109) follows from (101), and (110) follows from $w_H(\mathbf{z}^{k+1\setminus e}) \leq w_H(\mathbf{z}^{k+1}) \leq d_t - 1 - |\mathcal{L}|$ and the feasible condition (107) is satisfied for $i = k$ by the induction hypothesis. Therefore, (107) holds for $i = k + 1$ regardless of the choice of $\mathbf{k}_e$.

For a sink node t such that edge e is on the jth edge-disjoint path from the source node s to t, we consider two scenarios for $\mathcal{L}$, namely $j \in \mathcal{L}$ and $j \notin \mathcal{L}$. For all $\mathcal{L}$ satisfying C2) and $j \in \mathcal{L}$, all $\mathbf{x}$ satisfying C3) and all $\mathbf{z}^{k+1}$ satisfying C4) for $i = k + 1$,

$$(F_t^{k+1}(\mathbf{x}, -\mathbf{z}^{k+1}))^{\setminus\mathcal{L}} = (F_t^k(\mathbf{x}, -(\mathbf{z}^{k+1})^{\setminus e}))^{\setminus\mathcal{L}} \tag{111}$$
$$\neq \mathbf{0}, \tag{112}$$

where (111) follows from (106) and (112) follows from the same argument as the previous case. Therefore, (107) again holds for $i = k + 1$ regardless of the choice of $\mathbf{k}_e$.

For all $\mathcal{L}$ satisfying C2) and $j \notin \mathcal{L}$, all $\mathbf{x}$ satisfying C3) and all $\mathbf{z}^{k+1}$ satisfying C4) with $i = k + 1$, (107) holds for $i = k + 1$ if and only if either

$$(F_t^{k+1}(\mathbf{x}, -\mathbf{z}^{k+1}))^{\setminus\mathcal{L}\cup\{j\}} \neq \mathbf{0} \tag{113}$$

or

$$(F_t^{k+1}(\mathbf{x}, -\mathbf{z}^{k+1}))_j \neq 0.eq : 22rpo \tag{114}$$

By (106) and (103), (113) and (114) are equivalent to

$$(F_t^k(\mathbf{x}, -\mathbf{z}^{k+1\setminus e}))^{\setminus\mathcal{L}\cup\{j\}} \neq \mathbf{0}, \tag{115}$$

and

$$(\mathbf{x}\mathbf{A}_s^k - \mathbf{z}^{k+1\setminus e})\mathbf{F}^k\mathbf{k}_e - (\mathbf{z}^{k+1})_e \neq 0, \tag{116}$$

respectively. Note that $\mathbf{k}_e$ is involved in (116) but not in (115).

For an index set $\mathcal{L}$ satisfying C2) and $j \notin \mathcal{L}$, let $\Sigma_{\mathcal{L}}^{k+1}$ be the set of all $(\mathbf{x}, \mathbf{z}^{k+1})$ that do not satisfy (115), where

$\mathbf{x}$ satisfies C3) and $\mathbf{z}^{k+1}$ satisfies C4) for $i = k+1$. We need to find a proper $\mathbf{k}_e$ such that for any $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$, $(\mathbf{x}, \mathbf{z}^{k+1})$ satisfies (116). In the following technical lemmas, we first prove some properties of $\Sigma_{\mathcal{L}}^{k+1}$.

Lemma 12: If the feasible condition holds for $i = k$, then for any $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$, $w_H(\mathbf{z}^{k+1}) = d_t - 1 - |\mathcal{L}|$ and $(\mathbf{z}^{k+1})_e = 0$.

Proof: Fix $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$. If $|\mathcal{L}| = d_t - 1$, since $w_H(\mathbf{z}^{k+1}) \leq d_t - 1 - |\mathcal{L}| = 0$, the lemma is true. If $0 \leq |\mathcal{L}| < d_t - 1$, we now prove that $w_H(\mathbf{z}^{k+1\backslash e}) > d_t - 2 - |\mathcal{L}|$. If $w_H(\mathbf{z}^{k+1\backslash e}) \leq d_t - 2 - |\mathcal{L}|$, by the assumption that the feasible condition holds for $i = k$,

$$(F_t^k(\mathbf{x}, -\mathbf{z}^{i+1\backslash e}))^{\mathcal{L}\cup\{j\}} \neq \mathbf{0}, \tag{117}$$

i.e., $(\mathbf{x}, \mathbf{z}^{k+1})$ satisfies (115), a contradiction to $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$. Therefore

$$\begin{aligned} d_t - 1 - |\mathcal{L}| &\leq w_H(\mathbf{z}^{k+1\backslash e}) && (118)\\ &\leq w_H(\mathbf{z}^{k+1}) && (119)\\ &\leq d_t - 1 - |\mathcal{L}|. && (120) \end{aligned}$$

Hence, $w_H(\mathbf{z}^{k+1\backslash e}) = w_H(\mathbf{z}^{k+1}) = d_t - 1 - |\mathcal{L}|$. This also implies that $(\mathbf{z}^{k+1})_e = 0$. ∎

Lemma 13: Let M be a matrix, $\mathbf{x}$ be a row vector, and j be a column index of M. If a system of linear equations $\mathbf{x}M = \mathbf{0}$ has only the zero solution, then $\mathbf{x}M^{\backslash\{j\}} = \mathbf{0}$ has at most a one-dimensional solution space.

Proof: The lemma is proved by contradiction. Assume $\mathbf{x}M^{\backslash\{j\}} = \mathbf{0}$ has a solution space with more than one dimension. Then there exist nonzero, linearly independent $\mathbf{x}_1$ and $\mathbf{x}_2$ as its solutions. Since $\mathbf{x}M = \mathbf{0}$ has only the zero solution, we have $\mathbf{x}_1(M)_j \neq 0$ and $\mathbf{x}_2(M)_j \neq 0$. Let $\alpha = -\mathbf{x}_1(M)_j/\mathbf{x}_2(M)_j$. Then, $\mathbf{x}_1 + \alpha\mathbf{x}_2 \neq 0$ and $(\mathbf{x}_1 + \alpha\mathbf{x}_2)(M)_j = 0$. Together with $(\mathbf{x}_1 + \alpha\mathbf{x}_2)M^{\backslash\{j\}} = \mathbf{0}$, we have a contradiction to the assumption that $\mathbf{x}M = \mathbf{0}$ has only the zero solution. ∎

Lemma 14: Let ρ be an error pattern with $|\rho| = d_t - 1 - |\mathcal{L}|$, where $0 \leq |\mathcal{L}| \leq d_t - 1$. If the feasible condition holds for $i = k$, the span of all $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$ with $\mathbf{z}^{k+1} \in \rho^*$ is either empty or a one-dimensional linear space.

Proof: Consider the equation

$$(F_t^k(\mathbf{x}, -\mathbf{z}^{k+1\backslash e}))^{\backslash\mathcal{L}} = \mathbf{0} \tag{121}$$

with $\mathbf{x} \in \mathcal{C}$ and $\mathbf{z}^{k+1} \in \rho^*$ as variables. Since $\mathcal{C}$ and ρ^* are both vector spaces, (121) is a system of linear equations. By the assumption that the feasible condition holds for $i = k$, (121) has only the zero solution. By Lemma 13, the system of linear equations

$$(F_t^k(\mathbf{x}, -\mathbf{z}^{k+1\backslash e}))^{\backslash\mathcal{L}\cup\{j\}} = \mathbf{0}, \tag{122}$$

with $\mathbf{x} \in \mathcal{C}$ and $\mathbf{z}^{k+1} \in \rho^*$ as variables, has at most a one-dimensional solution space. ∎

Lemma 15: If the feasible condition holds for $i = k$, there exist at most $\binom{n_s+k}{d_t-1-|\mathcal{L}|}q^{|In(tail(e))|-1}$ values of $\mathbf{k}_e$ such that (116) does not hold for some $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$.

Proof: For $(\mathbf{x}_0, \mathbf{z}_0^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$, by Lemma 12, $(\mathbf{z}_0^{k+1})_e = 0$. Thus, all the $\mathbf{k}_e$ satisfying

$$(\mathbf{x}_0\mathbf{A}_s^k - \mathbf{z}_0^{k+1\backslash e})\mathbf{F}^k\mathbf{k}_e = 0 \tag{123}$$

do not satisfy (116) for $(\mathbf{x}_0, \mathbf{z}_0^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$.

To count the number of solutions of (123), we notice that

$$(F_t^k(\mathbf{x}_0, -\mathbf{z}_0^{k+1\backslash e}))^{\backslash\mathcal{L}} \neq \mathbf{0}, \tag{124}$$

by the feasible condition holding for $i = k$, and

$$(F_t^k(\mathbf{x}_0, -\mathbf{z}_0^{k+1\backslash e}))^{\backslash\mathcal{L}\cup\{j\}} = \mathbf{0}, \tag{125}$$

since $(\mathbf{x}_0, \mathbf{z}_0^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$. Thus,

$$(F_t^k(\mathbf{x}_0, -\mathbf{z}_0^{k+1\backslash e}))_j = ((\mathbf{x}_0\mathbf{A}_s^k - \mathbf{z}_0^{k+1\backslash e})\mathbf{F}^k(\mathbf{A}_t^k)^\top)_j \neq \mathbf{0}. \tag{126}$$

This nonzero component of $(\mathbf{x}_0\mathbf{A}_s^k - \mathbf{z}_0^{k+1\backslash e})\mathbf{F}^k$ corresponds to the edge that precedes edge e on the jth path from s to t. This shows that the components of $(\mathbf{x}_0\mathbf{A}_s^k - \mathbf{z}_0^{k+1\backslash e})\mathbf{F}^k$ in (126) corresponding to the edges in $In(tail(e))$ are not all zero. On the other hand, a component of $\mathbf{k}_e$ can possibly be nonzero if and only if it corresponds to an edge in $In(tail(e))$. Therefore, the solution space of $\mathbf{k}_e$ in (123) is an $\mathbb{F}_q^{|In(tail(e))|-1}$-dimensional subspace.

By Lemma 12, for each $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$, $\mathbf{z}^{k+1}$ must match an error pattern ρ with $|\rho| = d_t - 1 - |\mathcal{L}|$ and $e \notin \rho$. Since there are totally $n_s + k$ edges in $\mathcal{G}^{k+1}$ excluding e, there are $\binom{n_s+k}{d_t-1-|\mathcal{L}|}$ error patterns with size $d_t - 1 - |\mathcal{L}|$.

Consider an error pattern ρ with $|\rho| = d_t - 1 - |\mathcal{L}|$ and $e \notin \rho$. By Lemma 14, if $(\mathbf{x}_0, \mathbf{z}_0^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$ with $\mathbf{z}_0^{k+1} \in \rho^*$, all $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$ with $\mathbf{z}^{k+1} \in \rho^*$ can be expressed as $(\alpha\mathbf{x}_0, \alpha\mathbf{z}_0^{k+1})$ with nonzero $\alpha \in \mathbb{F}$. Since we obtain the same solutions of $\mathbf{k}_e$ in (123) when $\mathbf{x}_0$ and $\mathbf{z}_0^{k+1}$ are replaced by $\alpha\mathbf{x}_0$ and $\alpha\mathbf{z}_0^{k+1}$, respectively, for a particular pattern ρ, we only need to consider any $(\mathbf{x}_0, \mathbf{z}_0^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$ with $\mathbf{z}_0^{k+1} \in \rho^*$.

Upon considering all error patterns ρ with $|\rho| = d_t - 1 - |\mathcal{L}|$ and $e \notin \rho$, we conclude that there exist at most $\binom{n_s+k}{d_t-1-|\mathcal{L}|}q^{|In(tail(e))|-1}$ values of $\mathbf{k}_e$ not satisfying (116) for some $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma_{\mathcal{L}}^{k+1}$. ∎

Considering the worst case that for all $t \in \mathcal{T}$, edge e is on an edge-disjoint path from the source node s to sink node t, and considering all the index set $\mathcal{L}$ with $0 \leq |\mathcal{L}| \leq d_t - 1$ and $j \notin \mathcal{L}$ for each sink node t, we have at most

$$\begin{aligned} &\sum_{t\in\mathcal{T}}\sum_{l=0}^{d_t-1}\binom{r_t-1}{l}\binom{n_s+k}{d_t-1-l}q^{|In(tail(e))|-1}\\ =\ &\sum_{t\in\mathcal{T}}\binom{r_t+n_s+k-1}{d_t-1}q^{|In(tail(e))|-1} && (127)\\ \leq\ &\sum_{t\in\mathcal{T}}\binom{r_t+|\mathcal{E}|-2}{d_t-1}q^{|In(tail(e))|-1} && (128) \end{aligned}$$

vectors that cannot be chosen as $\mathbf{k}_e$. Note that (128) is justified because $0 \leq k \leq |\mathcal{E}| - n_s - 1$. Since $q > \sum_{t\in\mathcal{T}}\binom{r_t+|\mathcal{E}|-2}{d_t-1}$, there exists a choice of $\mathbf{k}_e$ such that for all $\mathcal{L}$ satisfying C2) and $j \notin \mathcal{L}$, all $\mathbf{x}$ satisfying C3), and all $\mathbf{z}^{k+1}$ satisfying C4) for $i = k+1$, (107) holds for $i = k+1$. Together with the other cases (where the choice of $\mathbf{k}_e$ is immaterial), we have proved the existence of a $\mathbf{k}_e$ such that the feasible condition holds for $i = k+1$.

VI. Concluding Remarks

This work, together with the previous work [16], gives a framework for coherent network error correction. In [16], they showed how to characterize the error correction/detection capability of a general transmission system with network coding being a special case. The problems concerned here are what the best a network code can do for error correction is and how to construct an optimal network code for this purpose.

Toward the first goal, refined versions of the Hamming bound, the Singleton bound and the Gilbert-Varshamov bound for network error correction have been presented with simple proofs based on the distance measures developed in [16]. These bounds are improvements over the ones in [6], [10], [11] in the linear network coding case. Even though these bounds are stated based on the Hamming weight as the weight measure on the error vectors, they can also be applied to the weight measures in [7], [12], [17] because of the equivalence relation among all these weight measures (See [16], [28]).

Like the original version of the Singleton bound [6], [10], the refined Singleton bound for linear network codes proved in this paper continues to be tight. Two different construction algorithms have been presented and both of them can achieve the refined Singleton bound. The first algorithm finds a codebook based on a given set of local encoding kernels, which simply constructs an MDS code when the problem setting is the classical case. The second algorithm constructs a set of of local encoding kernels based on a given classical error-correcting code satisfying a certain minimum distance requirement by recursively choosing the local encoding kernels that preserves certain minimum distance properties.

Acknowledgment

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[4] S. Jaggi, P. Sandrs, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973 – 1982, Jun. 2005.

[5] T. Ho, B. Leong, M. Medard, R. Koetter, Y. Chang, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE ISIT'03*, Jun. 2003.

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

[7] S. Jaggi, M. Langberg, T. Ho, and M. Effros, "Correction of adversarial errors in networks," in *Proc. IEEE ISIT'05*, Jul. 2005.

[8] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. north-holland publishing, 1978.

[9] S. Lin and D. J. Costello, *Error Control Coding: fundamentals and applications*, 2nd ed. Pearson Prentice Hall, 2004.

[10] R. W. Yeung and N. Cai, "Network error correction, part I: basic concepts and upper bounds," *Communications in Information and Systems*, vol. 6, no. 1, pp. 19 – 36, 2006.

[11] N. Cai and R. W. Yeung, "Network error correction, part II: lower bounds," *Communications in Information and Systems*, vol. 6, no. 1, pp. 37 – 54, 2006.

[12] Z. Zhang, "Network error correction coding in packetized networks," in *Proc. IEEE Information Theory Workshop 2006*, Oct. 2006.

[13] ——, "Linear network error correction codes in packet networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 209–218, Jan. 2008.

[14] R. C. Singleton, "Maximum distance Q-nary code," *IEEE Trans. Inf. Theory*, vol. IT-10, pp. 116–118, 1964.

[15] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Network coding theory," *Foundation and Trends in Communications and Information Theory*, vol. 2, no. 4 and 5, pp. 241–381, 2005.

[16] 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, invited paper.

[17] S. Yang and R. W. Yeung, "Characterizations of network error correction/detection and erasure correction," in *Proc. Netcod Workshop 2007*, Jan. 2007.

[18] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks using randomized network coding," in *Proc. IEEE ISIT'04*, Jun. 2004.

[19] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Mdard, and M. Effros, "Resilient network coding in the presence of byzantine adversaries," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2596–2603, Jun. 2008.

[20] S. Jaggi and M. Langberg, "Resilient network codes in the presence of eavesdropping byzantine adversaries," in *Proc. IEEE ISIT'07*, Jun. 2007.

[21] H. Balli, X. Yan, and Z. Zhang, "Error correction capability of random network error correction codes," in *Proc. IEEE ISIT'07*, Jun. 2007.

[22] X. Yan, H. Balli, and Z. Zhang, "Decoding beyond error correction capability for random network error correction codes," submitted to IEEE Transaction on Information Theory.

[23] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," in *Proc. IEEE ISIT'07*, Jun. 2007.

[24] D. Silva and F. R. Kschischang, "Using rank-metric codes for error correction in random network coding," in *Proc. IEEE ISIT'07*, Jun. 2007.

[25] S. Yang, C. K. Ngai, and R. W. Yeung, "Construction of linear network codes that achieve a refined singleton bound," in *Proc. ISIT'07*, Jan. 2007.

[26] S. Yang and R. W. Yeung, "Refined coding bounds for network error correction," in *Proc. IEEE Information Theory Workshop 2007*, Bergen, Norway, 2007.

[27] R. Matsumoto, "Construction algorithm for network error-correcting codes attaining the singleton bound," *IEICE Trans. Fundamentals*, vol. E90-A, no. 9, pp. 1729 – 1735, Nov. 2007.

[28] S. Yang, "Network coding and error correction," Ph.D. dissertation, The Chinese University of Hong Kong, 2008.