Construction of Linear Network Codes that Achieve a Refined Singleton Bound

Shenghao Yang, Chi Kin Ngai, and Raymond W. Yeung
Department of Information Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{shyang5, ckngai2, whyeung}@ie.cuhk.edu.hk

***Abstract*—In this paper, we present a refined version of the Singleton bound for network error correction, and propose an algorithm for constructing network codes that achieve this bound.**

***Index Terms*—Network coding, error correction, Singleton bound.**

I. Introduction

The network error correction problem studied in [1]–[5] is to extend classical error correction coding theory to a general network setting. In Cai and Yeung [1]–[3], network generalizations of the Hamming bound, the Singleton bound, and the Gilbert-Varshamov bound in classical algebraic coding theory were obtained. In particular, the tightness of the Singleton bound is preserved. The minimum rank for linear network codes, which plays a role similar to that of the minimum distance in decoding classical error-correcting codes, was introduced by Zhang [4]. Recently, network generalizations of the Hamming weight, the Hamming distance, and the minimum distance of network codes have been obtained by Yang and Yeung [5]. In terms of the minimum distance, the capability of a network code for error correction, error detection, and erasure correction can be fully characterized. The relation between network coding and classical algebraic coding was clarified in [6].

There exist two classes of algorithms for constructing network error-correcting codes: deterministic algorithms and randomized algorithms. Jaggi *et al.* [7] studied the random design of network error-correcting codes, which can achieve the Singleton bound asymptotically. Deterministic algorithms have been discussed in Cai and Yeung [3]. Based on a given network code that achieves the max-flow bounds, they proposed a greedy algorithm for constructing the parity check matrix of the *message set*, namely the set of vectors that can be transmitted by the source node. The network codes constructed by this algorithm can achieve the Singleton bound. However, as we will explain, when the maximum flows to different sink nodes are not the same, the network codes so constructed cannot fully exploit the spacial redundancy provided by the network to combat the errors that occur.

In this paper, we propose a deterministic algorithm for constructing network error-correcting codes that achieve the Singleton bound for individual sink nodes. This is a refinement of the Singleton bound in [2]. Thus, the codes constructed by our algorithm are stronger than those constructed in [3]. Our algorithm first finds a message set that meets the minimum distance requirement, which can be obtained by usual constructions of classical error-correcting codes. Then it updates the global encoding kernels of each channel in a way such that the minimum distance is preserved in each step. We prove that if the algebraic operations are over a sufficiently large field, our algorithm can always construct a network code that meets the required minimum distances for all the sink nodes. Compared with the algorithm in [3], our algorithm renders the flexibility of the choice of the message set. Roughly speaking, any classical linear block codes that has the required minimum distance can be qualified as the message set.

This paper is organized as follows. Section II introduces the network error correction problem. Section III describes our algorithm for constructing network codes that achieves the Singleton bound for each individual sink node. Section IV proves the correctness of the algorithm. Section V is the concluding remarks.

II. Network Error Correction

A. The Formulation

We study network transmission in a directed acyclic communication network represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes in $\mathcal{G}$ and $\mathcal{E}$ is the set of edges in $\mathcal{G}$. We assume an order on $\mathcal{E}$ which is consistent with the associated partial order on $\mathcal{G}$. An edge from node a to node b, say edge e, represents a communication channel from node a to node b. We call node a (node b) the tail (head) of edge e, 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 input edges and output edges of node a, respectively. There can be multiple edges between a pair of nodes, and each edge can transmit one symbol in a finite field $\mathbb{F}_q$.

A multicast on $\mathcal{G}$ transmits information from a source node s to a set of sink nodes $\mathcal{T}$. Let $n_s = |Out(s)|$. The source node s modulates the information to be multicast into a row vector $\mathbf{x} \in \mathbb{F}_q^{n_s}$ called the *message vector*. The set of all message vectors is called the *message set*, denoted by $\mathcal{C}$. The source node s transmits the message vector by mapping its n_s components onto the edges in $Out(s)$. Define an $n_s \times |\mathcal{E}|$

matrix $A = [A_{i,j}]$ as

$$A_{i,j} = \begin{cases} 1 & e_j \text{ is the } i\text{th edge in } Out(s), \\ 0 & \text{otherwise.} \end{cases}$$

By applying the order on $\mathcal{E}$ to $Out(s)$, the n_s non-zero columns of A form an identity matrix. An *error vector* $\mathbf{z}$ is an $|\mathcal{E}|$-dimensional row vector with each component representing the error on the corresponding edge. An *error pattern* ρ is a set of edges on each of which an error can occur. An error vector is said to match an error pattern ρ if all the errors occur on the edges in ρ.

For network $\mathcal{G}$, a linear network error-correcting code, or a linear network code for brevity, is specified by a set of *local encoding kernels* $\{k_{e',e} : e', e \in \mathcal{E}\}$ and the message set $\mathcal{C}$. The local encoding kernel $k_{e',e}$ can be non-zero only if $e' \in In(tail(e))$. The *global encoding kernel* of an edge e is an n_s-dimensional column vector f_e such that $\mathbf{x}f_e$ is the symbol transmitted on edge e when the message vector is $\mathbf{x}$ and the error vector $\mathbf{z}$ is the zero vector. The global encoding kernels of the outgoing edges of the source node s form the natural basis of $\mathbb{F}_q^{n_s}$. For other edges, the global encoding kernel can be obtained recursively by

$$f_e = \sum_{e' \in In(tail(e))} k_{e',e} f_{e'}.$$

Define the $|\mathcal{E}| \times |\mathcal{E}|$ one-step transition matrix $K = [K_{i,j}]$ for network $\mathcal{G}$ as $K_{i,j} = k_{e_i,e_j}$. For an acyclic network, $K^N = \mathbf{0}$ for some positive integer N. Define the transfer matrix of the network by $F = (I - K)^{-1}$ [8], so that the symbols transmitted on the edges are given by the components of $(\mathbf{x}A + \mathbf{z})F$.

For a sink node $t \in \mathcal{T}$, write $n_t = |In(t)|$, and define an $|\mathcal{E}| \times n_t$ matrix $B_t = [B_{i,j}]$ for sink node t as

$$B_{i,j} = \begin{cases} 1 & e_i \text{ is the } j\text{th edge in } In(t), \\ 0 & \text{otherwise.} \end{cases}$$

Again by applying the order on $\mathcal{E}$ to $In(t)$, the n_t nonzero rows of B_t form an identity matrix. The reception at a sink node t is given by

$$\mathbf{y}_t(\mathbf{x}, \mathbf{z}) = (\mathbf{x}A + \mathbf{z})FB_t. \tag{1}$$

Equation (1) is our formulation of the multicast network error correction problem. For linear network codes, message set $\mathcal{C}$ is a ω-dimensional linear space of $\mathbb{F}_q^{n_s}$. If so, then there exists a generator matrix G for the message set $\mathcal{C}$ in the sense of classical algebraic coding, i.e., $\mathcal{C}$ is the linear span of the row vectors of G. In this paper, we only consider this case, and we denote the dimension of $\mathcal{C}$ by ω. Equivalently, one can think of ω information symbols in $\mathbb{F}$ being transmitted from the source node to the sinks nodes.

B. Minimum Distance and the Singleton Bound

For a given linear network code, let $\Gamma_t = \{\mathbf{z} : \mathbf{y}_t(\mathbf{x}, -\mathbf{z}) = \mathbf{0}$ for some non-zero $\mathbf{x} \in \mathcal{C}\}$. The minimum distance at a sink node t of a network code with message set $\mathcal{C}$ is defined as

$$d_{\min,t} = \min\{w_H(\mathbf{z}) : \mathbf{z} \in \Gamma_t\},$$

where $w_H(\cdot)$ is the Hamming weight of a vector [5]. Zhang [4] defined the minimum rank of network codes which is equivalent to the minimum distance [5]. It was proved in [5] that a linear network code with minimum distance $d_{\min,t} \geq d_t$ at a sink node t can correct (detect) all error vectors with network Hamming weight less than $d_t/2$ (d_t), and can correct all erasures with network Hamming weight less than d_t at sink node t.

Let

$$d_{min} = \max_{t \in \mathcal{T}} d_{min,t}$$

and let m_t be the max-flow from source node s to sink node t. In terms of the notion of minimum distance, the Singleton bound obtained in [2] can be restated as

$$d_{min} \leq \min_{t \in \mathcal{T}} m_t - \omega + 1. \tag{2}$$

The tightness of (2) has been proved in [3]. In fact, it can readily be shown that

$$d_{min,t} \leq m_t - \omega + 1 \quad \text{for all } t \in \mathcal{T}, \tag{3}$$

which is more refined than (2), specifically when m_t are not the same for all $t \in \mathcal{T}$. The tightness of (3) will be proved by the construction in Section III.

For an (n_s, ω, (d_t:$t \in \mathcal{T}$)) network code, we refer to one for which the length of the message vector is n_s, the dimension of the message set is ω, and the minimum distance for sink node t is d_t, $t \in \mathcal{T}$. In Section III, we will present an algorithm that constructs an (n_s, ω, (d_t:$t \in \mathcal{T}$)) network code for any ω and d_t, $t \in \mathcal{T}$ satisfying (3), showing that this refined version of the Singleton bound continues to be tight.

C. An Example

The network shown in Fig. 1 has one source node s and two sink nodes t and u. The max-flow from the source node s to each sink node is 3. We want to design a network code with $\omega = 1$ and minimum distances $d_t = d_u = 3$. We begin by choosing the generator matrix of the message set as $G = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. Then we find the global encoding kernel for each edge e from upstream to downstream such that the minimum distance in each step is preserved.

One necessary condition for choosing the global encoding kernels is that for each sink node, the global encoding kernels of its three incoming edges are linearly independent. One example of such a network code over $\mathbb{F}_3$ is that all the non-zero local encoding kernels are 1. But this code does not meet the minimum distance requirement at the sink nodes as we now explain. Assume that the information symbol $x = 1$ and the errors are $z_{(a,t)} = 2$ and $z_{(s,b)} = 1$. We can check that the reception at sink node t is $\mathbf{0}$. Thus the minimum distance at sink node t must less than 3. So a stronger constraint must be imposed for choosing the global encoding kernels.

In fact, there exists no network codes over $\mathbb{F}_2$ and $\mathbb{F}_3$ that can meet the minimum distance requirement. Nevertheless, a solution exists over $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$, where $\alpha^2 + \alpha + 1 = 0$. In this solution, $G = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$, the local encoding kernels $k_{(a,d),(d,f)}$, $k_{(b,e),(e,g)}$ and $k_{(f,h),(h,i)}$ are α, while all other non-zero local encoding kernels are 1.

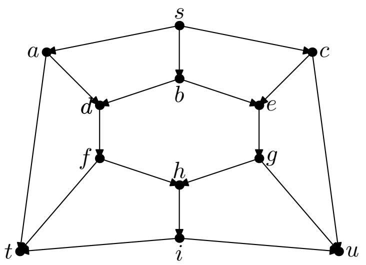

Fig. 1. A network error-correcting code example

III. The Proposed Algorithm

At the beginning, the algorithm finds m_t edge-disjoint paths from the source node s to each sink node t using a maximum flow algorithm, for example, the Ford-Fulkerson algorithm. Deleting the edges and nodes in $\mathcal{G}$ that are not on any paths, we obtain a network $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$. A network code for the network $\mathcal{G}^*$ can be extended to the network $\mathcal{G}$ without changing the minimum distances by assigning the zero global encoding kernel to all the edges in $\mathcal{G}$ but not in $\mathcal{G}^*$. Redefine $Out(s)$ as the set of outgoing edges of node s in $\mathcal{G}^*$, and denote $n_s = |Out(s)|$.

Then the algorithm chooses a message set and updates the global encoding kernels starting with the subgraph $\mathcal{G}_0^*$ of $\mathcal{G}^*$ consisting of the edges in $Out(s)$ (and the associated nodes). Following the order on $\mathcal{E}^*$ (inherited from $\mathcal{E}$), in the first step, $\mathcal{G}_0^*$ is expanded into $\mathcal{G}_1^*$ by appending the next edge in $\mathcal{E}^*$, and a global encoding kernel is assigned to this edge. This step is repeated until $\mathcal{G}_1^*$ eventually becomes $\mathcal{G}^*$.

Before descripting the algorithm formally in Subsection III-B, we first introduce a set of notation to facilitate the formulation and the analysis of the algorithm.

A. Iterative Formulation of Network Coding

In the ith step, the input set for a sink node t, denoted by $In^i(t)$, consists of m_t edges in $\mathcal{G}_i^*$, where the jth edge in $In^i(t)$ is the most downstream edge on the truncation in $\mathcal{G}_i^*$ of the jth edge-disjoint path from the source node s to sink node t in $\mathcal{G}^*$. The vector of symbols received on the edges in $In^i(t)$, denoted by $\mathbf{y}_t^i$, are referred to as the *current reception* of sink node t. We use F^i, A^i, B_t^i, and $\mathbf{z}^i$ to denoted the matrices (vectors) in the ith step for $\mathcal{G}_i^*$ that correspond to the matrices (vectors) F, A, B_t, and $\mathbf{z}$ for $\mathcal{G}^*$, respectively, so that

$$\mathbf{y}_t^i(\mathbf{x}, \mathbf{z}^i) = (\mathbf{x}A^i + \mathbf{z}^i)F^i B_t^i.$$

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

Let edge e be the edge to update in the $(i+1)$th step. We now describe the updating of F^i, A^i, B_t^i, and $\mathbf{z}^i$. Define an $(i+n_s)$-dimensional column vector $\beta_e = [k_{e',e}]$ where $e' \in \mathcal{G}_i^*$. Then

$$\begin{aligned} F^{i+1} &= \left(I - K^{i+1}\right)^{-1} \\ &= \left(I - \begin{bmatrix} K^i & \beta_e \\ \mathbf{0} & 0 \end{bmatrix}\right)^{-1} \\ &= \begin{bmatrix} F^i & F^i\beta_e \\ \mathbf{0} & 1 \end{bmatrix}. \end{aligned}$$

The matrix A^{i+1} is the same as A^i except that A^{i+1} has one more zero column than A^i, i.e.,

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

If the edge e is not on any path from the source node s to the sink node t, we only need to add a zero row to B_t^i to form B_t^{i+1}, i.e.,

$$B_t^{i+1} = \begin{bmatrix} B_t^i \\ \mathbf{0} \end{bmatrix}.$$

For this case, we can readily obtain

$$\mathbf{y}_t^{i+1}(\mathbf{x}, \mathbf{z}^{i+1}) = \mathbf{y}_t^i(\mathbf{x}, \mathbf{z}^{i+1\backslash e}). \tag{4}$$

If the edge e is on the jth edge-disjoint path from the source node s to sink node t, we need to first replace the jth column of B_t^i by the zero column vector and then add a row to select the last column of F^{i+1} to form B_t^{i+1}. That is, if $B_t^i = \begin{bmatrix} b_1 & b_2 & \cdots & b_{m_t} \end{bmatrix}$, then

$$B_t^{i+1} = \begin{bmatrix} b_1 & \cdots & b_{j-1} & \mathbf{0} & b_{j+1} & \cdots & b_{m_t} \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}, \tag{5}$$

where b_k is the kth column of B_t^i . It follows that

$$\begin{aligned} (\mathbf{y}_t^{i+1}(\mathbf{x}, \mathbf{z}^{i+1}))_j &= (\mathbf{x}A^{i+1} + \mathbf{z}^{i+1})F^{i+1}(B_t^{i+1})_j \\ &= (\mathbf{x}A^i + \mathbf{z}^{i+1\backslash e})F^i\beta_e + (\mathbf{z}^{i+1})_e, \end{aligned} \tag{6}$$

and

$$\begin{aligned} (\mathbf{y}_t^{i+1}(\mathbf{x}, \mathbf{z}^{i+1}))^{\backslash \{j\}} &= (\mathbf{x}A^{i+1} + \mathbf{z}^{i+1})F^{i+1}B_t^{i+1\backslash \{j\}} \\ &= (\mathbf{y}_t^i(\mathbf{x}, \mathbf{z}^{i+1\backslash e}))^{\backslash \{j\}}. \end{aligned} \tag{7}$$

B. The Weight Preserving Algorithm

Fig. 2 is a pseudo code of the algorithm, which, at the beginning, initializes F^0, A^0, and B_t^0, and chooses an $\omega \times n_s$ matrix G as the generator matrix for the message set $\mathcal{C}$ such that the classical linear block code generated by GB_t^0 has minimum Hamming distance larger than $d_t - 1$ for every sink node t. This can be achieved by an $(n_s, \omega, n_s - \min_{t \in \mathcal{T}}(m_t - d_t))$ classical linear block code. Note that such a code over a sufficiently large finite field exists because the classical Singleton bound is satisfied.

The main part of this algorithm is a loop, starting at line 7, for updating the global 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 global encoding kernel on edge e can be updated by choosing β_e,

```
1:  for each sink node $t$ do
2:      choose $m_t$ edge disjoint paths from $s$ to $t$;
3:      initialize $B_t$;
4:  end for
5:  initialize $F$ and $A$;
6:  Find a generator matrix $G$ such that the classical linear block code generated by $GB_t^0$ has minimum Hamming distance bigger than $d_t - 1$ for all sink node $t$;
7:  for each $e \in \mathcal{E}^* \backslash Out(s)$ from an upstream to downstream order do
8:      $D_{\beta_e} = \emptyset$;
9:      \\ the set of vectors that cannot be chosen as $\beta_e$.
10:     for each sink node $t$ do
11:         if there are no chosen paths from $s$ to $t$ through $e$ then
12:             $B_t = \begin{bmatrix} B_t \\ \mathbf{0} \end{bmatrix}$;
13:         else[$e$ is on the $j$th path from $s$ to $t$]
14:             for each $\mathcal{L}$ with $0 \le |\mathcal{L}| \le d_t - 1$ and $j \notin \mathcal{L}$ do
15:                 for each $\rho$ with $|\rho| = d_t - 1 - |\mathcal{L}|$ do
16:                     find $\mathbf{x}_0 \neq \mathbf{0}$ and $\mathbf{z}_0$ matching $\rho$ such that $(\mathbf{y}_t(\mathbf{x}_0, -\mathbf{z}_0))^{\backslash(\mathcal{L}\cup\{j\})} = \mathbf{0}$;
17:                     if such $\mathbf{x}_0$ and $\mathbf{z}_0$ exist then
18:                         $D_{\beta_e} = D_{\beta_e} \cup \{\beta\colon (\mathbf{x}_0 A - \mathbf{z}_0)F\beta = 0\}$;
19:                     end if
20:                 end for
21:             end for
22:             update $B_t$ as (5);
23:         end if
24:     end for
25:     choose a vector $\beta_e$ in $\mathbb{F}_q^{|In(tail(e))|} \setminus D_{\beta_e}$;
26:     $F = \begin{bmatrix} F & F\beta_e \\ \mathbf{0} & 1 \end{bmatrix}$;
27: end for
```

Fig. 2. A pseudo code of the proposed algorithm. The superscripts for F, A, B_t, $\mathbf{z}$, and $\mathbf{y}_t$ are omitted.

which is realized by the pseudo codes between line 8 and line 25. Concisely speaking, β_e is chosen such that for every sink node t,

$$(\mathbf{y}_t^i(\mathbf{x}, -\mathbf{z}^i))^{\backslash\mathcal{L}} \neq \mathbf{0} \tag{8}$$

for any $\mathcal{L} \subset \{1, 2, \ldots, m_t\}$ with $0 \le |\mathcal{L}| \le d_t - 1$, non-zero $\mathbf{x} \in \mathcal{C}$, and $\mathbf{z}^i$ with $w_H(\mathbf{z}^i) \le d_t - 1 - |\mathcal{L}|$.

We only need (8) to hold for $\mathcal{L}$ being the empty set, for all $t \in \mathcal{T}$ and for all $\mathbf{x}$ and $\mathbf{z}^i$ as prescribed, in order to preserve the minimum distances in each step, but it turns out that this is not enough to guarantee the existence of the required β_e. We will prove in Section IV that (8) is sufficient for this purpose provided that the field size q is sufficiently large.

C. Time Complexity of the Algorithm

The upper bound on the required field size is $\sum_{t\in\mathcal{T}}\binom{m_t+|\mathcal{E}^*|-2}{d_t-1}$. The linear equation in line 16 can be solved in polynomial time. The line 18 and 25 can be realized by the similar algorithm in [9] in polynomial time. The three levels of loops in this algorithm process these polynomial time algorithms $\sum_{k=n_s+1}^{|\mathcal{E}|}\sum_{t\in\mathcal{T}}\binom{m_t+k-1}{d_t-1}$ times. Thus, this algorithm is polynomial time to $|\mathcal{E}|$ of degree $d_{\min}$.

IV. Algorithm Verification

The algorithm is verified by induction. At the initialization, for each sink node t, since GB_t^0 has minimum distance larger than $d_t - 1$, we have $\mathbf{y}_t^0(\mathbf{x}, -\mathbf{z}^0) = (\mathbf{x} - \mathbf{z}^0)B_t^0 \neq \mathbf{0}$ for any non-zero $\mathbf{x}$ and any $\mathbf{z}^0$ with $w_H(\mathbf{z}^0) \le d_t - 1$. Let $\mathcal{L}$ be a subset of $\{1, 2, \ldots, m_t\}$ with $0 \le |\mathcal{L}| \le d_t - 1$. We have $(\mathbf{x} - \mathbf{z}^0)B_t^{0\backslash\mathcal{L}} \neq \mathbf{0}$ for all $\mathbf{z}^0$ with $w_H(\mathbf{z}^0) \le d_t - 1 - |\mathcal{L}|$. Thus (8) holds for $i = 0$.

Assume (8) holds for $i \le k$, where $k \ge 0$. In the $(k+1)$th step, let e be the edge appended to the graph. We will find β_e such that (8) continues to hold for $i = k+1$ and all $\mathcal{L}$, $\mathbf{x}$, and $\mathbf{z}$ as prescribed. We first consider a sink node t for which edge e is not on any path from the source node s to t. Then by (4), (8) holds for $i = k+1$ if and only if $(\mathbf{y}_t^k(\mathbf{x}, -\mathbf{z}^{k+1\backslash e}))^{\backslash\mathcal{L}} \neq \mathbf{0}$ for all $\mathcal{L}$, $\mathbf{x}$, and $\mathbf{z}$ as prescribed. This is true by the induction hypothesis because $w_H(\mathbf{z}^{k+1\backslash e}) \le w_H(\mathbf{z}^{k+1}) \le d_t - 1 - |\mathcal{L}|$. Thus, any β_e works for such a sink node t, i.e., no constraint is imposed on the choice of β_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 $\mathcal{L}$ such that $j \in \mathcal{L}$, by (7) and using the same argument as the previous case, we see that again no constraint is imposed on the choice of β_e.

For $\mathcal{L}$ such that $j \notin \mathcal{L}$, then (8) holds if and only if

$$(\mathbf{y}_t^{k+1}(\mathbf{x}, -\mathbf{z}^{k+1}))^{\backslash\mathcal{L}\cup\{j\}} \neq 0 \tag{9}$$

or

$$(\mathbf{y}_t^{k+1}(\mathbf{x}, -\mathbf{z}^{k+1}))_j \neq 0. \tag{10}$$

By (7) and (6), (9) and (10) are equivalent to

$$(\mathbf{y}_t^{k}(\mathbf{x}, -\mathbf{z}^{k+1\backslash e}))^{\backslash\mathcal{L}\cup\{j\}} \neq \mathbf{0}, \tag{11}$$

and

$$(\mathbf{x}A^k - \mathbf{z}^{k+1\backslash e})F^k\beta_e - (\mathbf{z}^{k+1})_e \neq \mathbf{0}, \tag{12}$$

respectively. We observe that the LHS in (11) does not involve β_e. This means that for $(\mathbf{x}, \mathbf{z}^{k+1})$ satisfying (11), there is no constraint on the choice of β_e. Otherwise, we need to choose β_e such that (12) holds. Let Σ^{k+1} be the set of all $(\mathbf{x}, \mathbf{z}^{k+1})$ that do not satisfy (11), with $\mathbf{x} \in \mathcal{C}$, $\mathbf{x} \neq \mathbf{0}$, and $w_H(\mathbf{z}^{k+1}) \le d_t - 1 - |\mathcal{L}|$.

Lemma 1: Let t be a sink node such that edge e is on the jth path from the source node s to t, and $\mathcal{L}$ be a subset of $\{1, 2, \ldots, m_t\}$ such that $j \notin \mathcal{L}$. Then there exist at most $\binom{k+n_s}{d_t-1-|\mathcal{L}|}q^{|In(tail(e))|-1}$ possible β_e such that (12) does not hold for some $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma^{k+1}$.

Proof: See Appendix. ■

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, we have at most

$$\begin{aligned} & \sum_{t\in\mathcal{T}} \sum_{l=0}^{d_t-1} \binom{m_t-1}{l} \binom{k+n_s}{d_t-1-l} q^{|In(tail(e))|-1} \\ = & \sum_{t\in\mathcal{T}} \binom{m_t+k+n_s-1}{d_t-1} q^{|In(tail(e))|-1} \end{aligned} \tag{13}$$

vectors that cannot be chosen as β_e when $i = k+1$, where $k \leq |\mathcal{E}^*| - n_s - 1$. Hence, if $q > \sum_{t\in\mathcal{T}} \binom{m_t+|\mathcal{E}^*|-2}{d_t-1}$, we have a choice of β_e for all i.

V. Concluding Remarks

In this paper, we present a refinement of the Singleton bound previously obtained by Cai and Yeung [2] for network error-correcting codes. Unlike the bound in [2], this refined Singleton bound imposes different constraints on the individual sink nodes when the maximum flows from the source node to the sink nodes are not the same. The tightness of the refined Singleton bound is proved by a construction of linear network codes that achieve the bound. The network codes so constructed can fully exploit the spacial redundancy provided by the network to combat the errors that occur. Besides, our construction takes advantage of the rich results in classical algebraic coding theory.

Appendix
Proof of Lemma 1

We first prove the following 3 lemmas.

Lemma 2: If a linear equation $\bar{x}M = \mathbf{0}$ has only the zero solution, then $\bar{x}M^{\setminus\{j\}} = \mathbf{0}$ has at most a one-dimensional solution space, where M is an $m \times n$ matrix and j is the index of any column of M.

Proof: Define the linear mapping $J : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^{n-1}$ that delete the jth component of a vector in $\mathbb{F}_q^n$. The matrix representation of J is a $n \times (n-1)$ matrix formed by deleting the jth column of the $n \times n$ identity matrix. Since $M^{\setminus\{j\}} = MJ$, the solution space of $\bar{x}M^{\setminus\{j\}} = \mathbf{0}$ is simply the null space of MJ. Since the null space of J is a one-dimensional subspace and M is an injective mapping, the dimension of the null space of MJ is at most 1. ■

Lemma 3: Any $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma^{k+1}$ satisfies $w_H(\mathbf{z}^{k+1}) = d_t - 1 - |\mathcal{L}|$ and $(\mathbf{z}^{k+1})_e = 0$.

Proof: 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 $0 \leq |\mathcal{L}| < d_t - 1$, by induction, all error vectors $\mathbf{z}^{k+1}$ with $w_H(\mathbf{z}^{k+1\setminus e}) \leq d_t - 2 - |\mathcal{L}|$ satisfy (11). Hence $d_t - 1 - |\mathcal{L}| = w_H(\mathbf{z}^{k+1\setminus e}) \leq w_H(\mathbf{z}^{k+1}) \leq d_t - 1 - |\mathcal{L}|$. This completes the proof. ■

Lemma 4: Let ρ be an error pattern with $|\rho| = d_t - 1 - |\mathcal{L}|$. The set of all $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma^{k+1}$ with $\mathbf{z}^{k+1}$ matching ρ, if nonempty, spans a one-dimensional linear space. In other words, there exists $(\mathbf{x}_0, \mathbf{z}_0^{k+1})$ such that all $(\mathbf{x}, \mathbf{z}^{k+1})$ in the set can be written as $\alpha(\mathbf{x}_0, \mathbf{z}_0^{k+1})$ for some non-zero $\alpha \in \mathbb{F}$.

Proof: For any $(\mathbf{x}, \mathbf{z}^{k+1})$ in the prescribed set, we can write $\mathbf{x} = \mathbf{m}G$ where $\mathbf{m}$ is an ω-dimensional row vector, and $\mathbf{z}^{k+1} = \begin{bmatrix} \tilde{\mathbf{z}}A_\rho^k & 0 \end{bmatrix}$ where $\tilde{\mathbf{z}}$ is a $(d_t - 1 - |\mathcal{L}|)$-dimensional row vector and A_ρ^k is a matrix similar to A^k that chooses the rows of F^k corresponding to the edges in ρ. Since $(\mathbf{x}, \mathbf{z}^{k+1})$ does not satisfy (11), $(\mathbf{m}, \tilde{\mathbf{z}})$ satisfies

$$\begin{bmatrix} \mathbf{m} & -\tilde{\mathbf{z}} \end{bmatrix} \begin{bmatrix} GA^k \\ A_\rho^k \end{bmatrix} F^k B_t^{k\setminus(\mathcal{L}\cup\{j\})} = \mathbf{0}. \tag{14}$$

If the row vectors in $A_\rho^k F^k B_t^{k\setminus(\mathcal{L}\cup\{j\})}$ are linearly dependent, then it is a contradiction to Lemma 3. Thus the row vectors in $A_\rho^k F^k B_t^{k\setminus(\mathcal{L}\cup\{j\})}$, and hence the row vectors in $A_\rho^k F^k B_t^{k\setminus\mathcal{L}}$, are linearly independent. Therefore, we see that only $\mathbf{m} = \mathbf{0}$ and $\tilde{\mathbf{z}} = \mathbf{0}$ can satisfy

$$\begin{bmatrix} \mathbf{m} & -\tilde{\mathbf{z}} \end{bmatrix} \begin{bmatrix} GA^k \\ A_\rho^k \end{bmatrix} F^k B_t^{k\setminus\mathcal{L}} = \mathbf{0}. \tag{15}$$

By Lemma 2, the set of all non-zero $\begin{bmatrix} \mathbf{m} & \tilde{\mathbf{z}} \end{bmatrix}$ satisfying (14) spans a one-dimensional subspace. Let $\begin{bmatrix} \mathbf{m}_0 & \tilde{\mathbf{z}}_0 \end{bmatrix}$ be any non-zero vector in this subspace. Then $\mathbf{x}_0 = \mathbf{m}_0 G$ and $\mathbf{z}_0^{k+1} = \begin{bmatrix} \tilde{\mathbf{z}}_0 A_\rho^k & 0 \end{bmatrix}$ satisfy the requirement of the lemma. ■

For an error pattern ρ with $(\mathbf{x}_0, \mathbf{z}_0^{k+1})$ as prescribed in Lemma 4, all the β_e satisfying

$$(\mathbf{x}_0 A^k - \mathbf{z}_0^{k+1\setminus e}) F^k \beta_e = \mathbf{0} \tag{16}$$

do not satisfy (12). By induction, we obtain $(\mathbf{y}_t^k(\mathbf{x}_0, -\mathbf{z}_0^{k+1\setminus e}))^{\setminus\mathcal{L}} \neq \mathbf{0}$, and hence $(\mathbf{y}_t^k(\mathbf{x}_0, -\mathbf{z}_0^{k+1\setminus e}))_j = (\mathbf{x}_0 A^k - \mathbf{z}_0^{k+1\setminus e}) F^k (B_t^k)_j \neq \mathbf{0}$. Thus the solution set of (16) with β_e being the unknown is a $\mathbb{F}_q^{|In(tail(e))|-1}$-dimensional linear subspace. Since there are a total of $\binom{k+n_s}{d_t-1-|\mathcal{L}|}$ error patterns with cardinality $d_t - 1 - |\mathcal{L}|$, there exist at most $\binom{k+n_s}{d_t-1-|\mathcal{L}|} q^{|In(tail(e))|-1}$ possible β_e not satisfying (12) for some $(\mathbf{x}, \mathbf{z}^{k+1}) \in \Sigma^{k+1}$.

Acknowledgement

The authors would like to thank Sidharth Jaggi, Min Tan and Silas Fong for stimulating discussions.

References

[1] N. Cai and R. W. Yeung, "Network coding and error correction," in *Proc. IEEE ITW'02*, 2002.

[2] 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.

[3] 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.

[4] Z. Zhang, "Network error correction coding in packetized networks," in *Proc. IEEE ITW'06*, Oct. 2006.

[5] S. Yang and R. W. Yeung, "Characterizations of network error correction/detection and erasure correction," in *Proc. Netcod'07*, Jan. 2007.

[6] 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.

[7] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Medard, "Resilient network coding in the presence of byzantine adversaries," submitted to *INFOCOM'06*.

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

[9] 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. Inform. Theory*, vol. 51, no. 6, pp. 1973 – 1982, June 2005.