The Singleton Bound for Network Error-Correcting Codes

Ning Cai and Raymond W. Yeung
Department of Information Engineering
The Chinese University of Hong Kong
N.T., Hong Kong
Emails: ncai,whyeung@ie.cuhk.edu.hk

***Abstract*— Inspired by network coding, network error-correcting codes was introduced in [1] for multicasting a source message to a set of nodes on a network. The usual approach in existing networks, namely link-by-link error correction, is a special case of network error correction. In [1], network generalizations of the Hamming bound and the Gilbert-Varshamov bound were obtained. In this paper, we prove the network generalizations of the Singleton Bound and it's tightness.**

I. What is network coding?

An *acyclic* communication network is represented by a finite directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes in the network and $\mathcal{E}$ is the set of edges in $\mathcal{G}$ which represent the communication channels. An edge from node a to node b is denoted by (a, b). We call node a (node b) the input node (output node) of edge (a, b), and we call edge (a, b) an input (output) edge of node b (node a).

In the network, a message taken from an alphabet $\mathcal{Z}$ is generated by an information source at a node $s \in \mathcal{V}$, referred to as the *source node*. We call the set $\mathcal{Z}$ the *source alphabet* and the message generated the *source message*. The source message is transmitted through the network to each node $u \in \mathcal{U}$ for some $\mathcal{U} \subset \mathcal{V}$, and each node in $\mathcal{U}$ is referred to as a *sink node*.

Let $R_{(a,b)}$ be the maximum number of symbols from an alphabet $\mathcal{X}$ that can be transmitted on the channel (a, b). $R_{(a,b)}$ is also referred to as the *capacity* (in the sense of graph theory) of edge (a, b). Define $\mathcal{R} = \{R_{(a,b)} : (a, b) \in \mathcal{E}\}$. To simplify our discussion, we assume that $R_{(a,b)}$ are (nonnegative) integers for all $(a, b) \in \mathcal{E}$.

Such a network can be described alternatively by a graph in which all the edges have capacity 1 and there can be multiple edges between a pair of nodes. Specifically, if $R_{(a,b)} = r > 1$, we represent the channel (a, b) by r edges of capacity 1, denoted by $(a, b)(1), (a, b)(2), \ldots, (a, b)(r)$, instead of by the single edge (a, b) of capapcity r. In the rest of the paper, we will denote the edge set for such a representation by $\mathcal{E}^*$. With a slight abuse of notation, we will use (a, b) to refer to one of the channels in $\mathcal{E}^*$ from node a to node b.

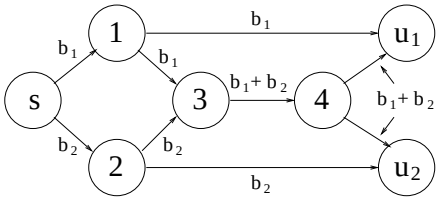

Fig. 1. A network coding example.

We will denote a network described above by $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$. For the time being, let us assume that all the channels are error-free. *Network coding*, which refers to coding at the nodes in a network, may allow more information to be transmitted than what would be possible by routing alone [2]. In fact, routing is a special case of network coding.

To illustrate the advantage of network coding, we consider the network in Fig. 1. Suppose $\mathcal{X} = \{0, 1\}$. Two bits, b_1 and b_2, are transmitted from node s to both nodes u_1 and u_2, with coding at node 3. Here '+' denotes modulo 2 addition. At node u_1 (u_2), the bit b_2 (b_1) can be recovered from the received bits b_1 (b_2) and $b_1 + b_2$. It is easy to verify that if coding is not allowed at node 3, then the above cannot be achieved.

Denote by $\mathrm{maxflow}(s,u)$ the maximum flow between node s and node u, where $u \in \mathcal{U}$. The following fundamental theorem was proved in [2].

Theorem 1: It is possible to transmit a source message with alphabet $\mathcal{Z}$ in a network $\mathcal{G}$ from source node s to sink nodes $u \in \mathcal{U}$ if and only if

$$\log_{|\mathcal{X}|} |\mathcal{Z}| \leq n, \tag{1}$$

where $n = \min_{u\in\mathcal{U}} \mathrm{maxflow}(s,u)$.

Subsequently, it was proved in [3] by means of a vector space approach that linear network codes suffice to achieve the bound in Theorem 1. A similar result was proved in [4] by means of a matrix approach. In this paper, the vector space approach in [3] will be used.

Inspired by network coding, network error-correcting codes was introduced in [1] for multicasting a source message to a set of nodes on a network. The usual approach in existing networks, namely link-by-link error correction, is a special case of network error correction. In [1], network generalizations of the Hamming bound and the Gilbert-Varshamov bound were obtained.

In this paper, we prove the Singleton bound for network error-correcting codes and it's tightness. The rest of the present paper is organized as follows. Section II is an introduction to network error correction. Existing results are described in Section III, where preliminaries to the new results in Section IV are provided. Section IV presents the Singleton Bound, whose tightness is proved by means of a strengthened Varshamov bound. Concluding remarks are in Section V.

II. Network error correction

We first begin by defining a network code. Later on we will show how such a code can be designed so that it can be used for error correction. Basically, the source message is protected by the network code from distributed errors occurring in different channels in the network.

Let $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ be a given acyclic communication network. Then the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ naturally defines a partial order $\preceq$ ($\preceq_e$) on the node set $\mathcal{V}$ (edge set $\mathcal{E}$) i.e., for $a, b \in \mathcal{V}$ ($(a,c),(b,d) \in \mathcal{E}$), $a \preceq b$ ($(a,c) \preceq_e (b,d)$) if and only if there is path from a to b (from (a,c) to (b,d)). A partial order can be extended to a (total) order, and the extension, called a linear extension of the partial order in combinatorics, is usually not unique. Let us call an order on $\mathcal{V}$ a legal coding order, or simply a *coding order*, if it is a linear extension of $\preceq$.

Let $\Gamma_+(a) = \{(c,a) : (c,a) \in \mathcal{E}\}$ and $\Gamma_-(a) = \{(a,b) : (a,b) \in \mathcal{E}\}$ be the sets of input and output edges of node a, respectively. We also call $|\Gamma_+(a)|$ the in-degree and $|\Gamma_-(a)|$ the out-degree of node a. Without loss of generality, we can always assume that the in-degree of the source node s is 0 and all other nodes have positive in-degree because a non-source node with no input edge cannot obtain information from the network, and so it is useless for communication and can be deleted from the edge set. Under this assumption, a coding order always starts with the source node s. Let $\mathcal{Z}$ be the source alphabet and $\mathcal{X}$ be a finite set that serves as the *code alphabet* for the network.

Let $r_{(a,b)} \leq R_{(a,b)}$ for $(a,b) \in \mathcal{E}$ be positive integers. A network code for the network $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ is a family of functions $\{\phi_{(a,b)} : (a,b) \in \mathcal{E}\}$ such that $\phi_{(s,b)} : \mathcal{Z} \longrightarrow \mathcal{X}^{r_{(s,b)}}$ and $\phi_{(a,b)} : \prod_{(c,a)\in\Gamma_+(a)} \mathcal{X}^{r_{(c,a)}} \longrightarrow \mathcal{X}^{r_{(a,b)}}$ if a is not the source node s.

Communication over the network with the code defined above may be realized in a coding order as follows. The nodes in $\mathcal{V}$ encode and send codewords according to this order. The source node s first encodes the source message $z \in \mathcal{Z}$ into $\phi_{(s,b)}(z)$ for all $(s,b) \in \Gamma_-(s)$ and then sends the values of $\phi_{(s,b)}(z)$ to their output nodes b via channels (s,b). Then the second node in the order (whose input edges must be from node s by definition) encodes. According to this scheme, when a node a encodes, all nodes c such that $(c,a) \in \Gamma_+(a)$ have already encoded and sent their codewords. That is, node a has received a sequence $x^{r_{(c,a)}}$ in $\mathcal{X}^{r_{(c,a)}}$ sent by node c from each of its input edge $(c,a) \in \Gamma_+(a)$ before it encodes. Thus node a is able to encode the information it receives into the codewords $\phi_{(a,b)}(x^{r_{(c,a)}}, (c,a) \in \Gamma_+(a))$ and send them to nodes b on the output edges $(a,b) \in \Gamma_-(a)$. Communication ends at the last node in the order whose out-degree must be zero by definition.

Thus a function $\tilde{\phi}_{(a,b)}$ from $\mathcal{Z}$ to $\mathcal{X}^{r_{(a,b)}}$ for each $(a,b) \in \mathcal{E}$ is introduced in the natural way by such a scheme. Obviously, these functions introduced do not depend on the choice of the linear extension.

For a sink node $u \in \mathcal{U}$, we write $\Phi_u(z) = (\tilde{\phi}_{(a,u)},\ (a,u) \in \Gamma_+(u))$. Thus for a given code, the output of every edge is uniquely determined by the source message z if no error occurs (errors will be defined below). Then a code $\{\phi_{(a,b)} : (a,b) \in \mathcal{E}\}$ is uniquely decodable, or simply decodable, if $\Phi_u(z) \neq \Phi_u(z')$ for all $z \neq z'$ and all $u \in \mathcal{U}$.

We now consider the situation that the channels in the network are not necessarily error-free, i.e., a channel's output may be different from its input. A useful way to think of errors in a channel is that they are "applied" to the input upon transmission. Since the nodes in the network transmit according to a certain coding order, we can think of the errors in the network being applied to the channel inputs according to the same coding order.

An error is said to occur if an output symbol of a channel is different from the corresponding input symbol. Thus if a codeword consisting of more than one symbol is sent on a channel, multiple errors can occur. A τ-error is said to occur (in the network) if the total number of errors occur in all the channels is equal to τ.

Definition 1: A network code is t-error-correcting if it can correct all τ-errors for $\tau \leq t$, i.e., if the total number of errors in the network is at most t, then the source message can be recovered by all the sink nodes $u \in \mathcal{U}$.

III. Existing Bounds

Upon defining a t-error-correcting code for an acyclic network $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ in the last section, we present in this section existing bounds for network error-correcting codes, namely the Hamming bound and the Gilbert-Varshamov bound. The Gilbert-Varshamov bound is a preliminary to the Singleton bound to be discussed in Section IV.

A. The Hamming bound

For a given code $\phi = \{\phi_{(a,b)} : (a,b) \in \mathcal{E}\}$ and a set of channels $\mathcal{B}$, let us denote by $out(\phi, t, \mathcal{B}, z)$ the set of all possible output sequences of the channels in the set $\mathcal{B}$ (with length $\sum_{(a,b)\in\mathcal{B}} R_{(a,b)}$) when z is the source message and at most t errors occur in the network. For disjoint $A, B \subset \mathcal{V}$ such that $A \cup B = \mathcal{V}$, define

$$cut(A,B) = \{(a,b) \in \mathcal{E} : a \in A \text{ and } b \in B\}.$$

We say that $cut(A,B)$ is a cut between nodes s and u if $s \in A$ and $u \in B$. The quantity $\sum_{(a,b)\in cut(A,B)} R_{(a,b)}$ is called the volume of $cut(A,B)$. For a sink node $u \in \mathcal{U}$, denote by $c(s,u)$ the minimum volume of a cut between s and u, which by the max-flow min-cut theorem in graph theory is equal to $\text{maxflow}(s,u)$.

Observation 1: For a t-error-correcting network code ϕ, for any $cut(A,B)$ between the source node s and any sink node u,

$$out(\phi, t, cut(A,B), z) \cap out(\phi, t, cut(A,B), z') = \emptyset \quad (2)$$

for all $z, z' \in \mathcal{Z}$ such that $z \neq z'$.

Based on this observation, by showing that for all sinks $u \in \mathcal{U}$, any $cut(A,B)$ between s and u with volume m (say), and all $z \in \mathcal{Z}$,

$$|out(\phi, t, cut(A,B), z)| \geq \sum_{i=0}^{t} \binom{m}{i} (q-1)^i, \quad (3)$$

we can prove the following sphere-packing bound, or the Hamming bound.

Theorem 2 (Hamming Bound): [1] Let $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ be an acyclic network and $n = \min_{u\in\mathcal{U}} c(s,u)$. Let the code alphabet $\mathcal{X}$ be q-ary, i.e., $|\mathcal{X}| = q$. If there exists a t-error-correcting code on $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ for an information source with alphabet $\mathcal{Z}$, then

$$|\mathcal{Z}| \leq \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i}.$$

Although the RHS of (3) is exactly equal to the volume of a sphere in $\mathcal{X}^m$ with radius t, (3) by no means imply that $out(\phi, t, cut(A,B), z)$ contains a sphere in $\mathcal{X}^m$ with center at $(\tilde{\phi}_{(a,b)}(z), (a,b) \in cut(A,B))$ and radius t. If this is true, then together with Observation 1, $\{(\tilde{\phi}_{(a,b)}(z), (a,b) \in cut(A,B)) : z \in \mathcal{Z}\}$ would form a classical t-error-correcting code in $\mathcal{X}^m$. However, it turns out that this is actually the case when $cut(A,B)$ satisfies a certain property. We refer the reader to [5] for a detailed discussion.

B. The Gilbert-Varshamov Bound

Throughout this section, we assume that the code alphabet $\mathcal{X}$ is $GF(q)$ for some sufficiently large prime q, and we will work in an n-dimensional

linear space $GF^n(q)$ spanned by a *linear-code multicast* (LCM) defined shortly. The source alphabet $\mathcal{Z}$ will be a subset of $GF^n(q)$ for a general code and a k-dimensional subspace of $GF^n(q)$ for some positive integer $k \leq n$ for a linear code. Boldfaced letters (e.g., $\mathbf{a}, \mathbf{b}, \ldots, \mathbf{z}$) stand for row vectors whose dimensions are understood from the context. The transpose operation on vectors and matrices will be denoted by "τ". So $\mathbf{v}^\tau, \mathbf{w}^\tau$, etc, are column vectors. Addition and subtraction of vectors are understood to be in the linear spaces over $GF(q)$. With a slight abuse of notation, we also use $GF^n(q)$ to denote the linear spaces of n-dimensional row vectors and column vectors in $GF(q)$.

The definition of an LCM we give below has been simplified for acylic networks. Recall the definition of $\mathcal{E}^*$ in Section I.

Definition 2: [3] A linear code multicast (LCM) V for an acyclic network $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ is an assignment of linear subspace $\mathcal{L}_V(a)$ of (column space) $GF^n(q)$ to a node $a \in \mathcal{V}$ and a column vector $\mathbf{v}_V^\tau((a,b))$ of dimension n to a channel $(a,b) \in \mathcal{E}^*$ over a sufficiently large finite field $GF(q)$ for a positive integer n, such that

1) for all nodes $a \in \mathcal{V}$ and channels $(b,c) \in \mathcal{E}^*$, $\mathcal{L}_V(a) \subset \mathcal{L}_V(s)$, $\mathbf{v}_L^\tau((b,c)) \in \mathcal{L}_V(s)$, and $\mathcal{L}_V(s) \subset GF^n(q)$;

2) $\mathbf{v}_V^\tau((a,b)) \in \mathcal{L}_V(a)$ if $(a,b) \in \Gamma_-(a)$;

3) $\mathbf{v}_V^\tau((b,c))$ is a linear combination of $\mathbf{v}_V^\tau((a,b))$, $(a,b) \in \Gamma_+(b)$ for all output channels $(b,c) \in \Gamma_-(b)$.

Denote by $M(a)$ the matrix whose columns are the vectors assigned to the input channels of node a. For any LCM V, by 3) in the above definition, there exists a column vector $\mathbf{a}^\tau$ such that $\mathbf{v}_V^\tau((a,b)) = M(a)\mathbf{a}^\tau$. For the time being, let $GF^n(q)$ plays the role of the source alphabet and call a vector $\mathbf{w} \in GF^n(q)$ an input to the network. Then we can define a linear network code ϕ based on any LCM V by

1) $\phi_{(s,a)}(\mathbf{w}) = \langle \mathbf{w}, \mathbf{v}_V((s,a)) \rangle$ for all $a \in \Gamma_-(s)$;
2) $\phi_{(a,b)}(\mathbf{u}(a)) = \mathbf{u}(a)\mathbf{a}^\tau$, where $\mathbf{u}(a)$ is the row vector whose ith component is the output of the ith channel in $\Gamma_+(a)$ in the same order as the columns of $M(a)$.

It is easy to verify inductively that

$$\tilde{\phi}_{(a,b)}(\mathbf{w}) = \langle \mathbf{w}, \mathbf{v}_V((a,b)) \rangle$$

for all $(a,b) \in \mathcal{E}^*$.

We now define a *generic* LCM which we will use for code construction. The existence of a generic LCM is guaranteed by the theorem that follows.

Definition 3: [3] An LCM V assigning n-dimensional column vectors to channels in a network $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ is generic if for all $k \leq n$ and any subset of k channels $\{(a_1,b_1),(a_2,b_2),\ldots,(a_k,b_k)\}$, that $\mathcal{L}_V(a_j) \not\subset span[\mathbf{v}_V^\tau((a_i,b_i)),\ i \neq j]$ for all $j \in \{1,2,\ldots,k\}$ implies that $\mathbf{v}_V^\tau((a_1,b_1)), \mathbf{v}_V^\tau((a_2,b_2)) \ldots, \mathbf{v}_V^\tau((a_k,b_k))$ are independent.

Theorem 3: [3]

i) For a given network with source node s, for all $n \leq \Gamma_-(s)$ and sufficiently large q (depending on the network and n), there exists a generic LCM assigning n-dimensional column vectors over $GF(q)$ to channels in the network.

ii) For the generic LCM in i) with $n = \Gamma_-(s)$ and all nodes $a \in \mathcal{V} \setminus \{s\}$, $dim(\mathcal{L}_V(a))$ is equal to $\mathrm{maxflow(s,a)} = c(s,a)$.

To construct error correcting codes via a generic LCM, we need the following preparation. Consider the given network and let $n = \min_{u \in \mathcal{U}} c(s,u)$. Then we can obtain a subnetwork by deleting some channels if necessary, such that for all $u \in \mathcal{U}$,

$$cut(s,u) = d_+(u) = n. \tag{4}$$

For simplicity of notation, without loss of generality, we assume that (4) holds for the given network. Now by Theorem 3, we can find a generic LCM V assigning n-dimensional column vectors to the channels in the network. This generic LCM V induces a network code specified by ϕ and $\tilde{\phi}$ as described above. Since for any $u \in \mathcal{U}$,

$$dim(\mathcal{L}_V(u)) = cut(s,u) = n = dim(\mathcal{L}_V(s))$$

by Theorem 3 and (4), and $\mathcal{L}_V(u) \subset \mathcal{L}_V(s)$, we see that

$$\mathcal{L}_V(u) = \mathcal{L}_V(s) = GF^n(q).$$

This implies that the matrix $M(u)$ is a full rank square matrix of size n.

Fix a coding order in the network and choose any generic LCM V as prescribed above. We now consider the situation that the channels are not necessarily error-free. Since the code alphabet $\mathcal{X}$

is $GF(q)$, we can regard the output of a channel $(a,b) \in \mathcal{E}^*$ as the sum of the input of the channel and an error symbol $e_{(a,b)} \in GF(q)$. Define $\mathbf{e} = (e_{(a,b)} : (a,b) \in \mathcal{E}^*)$, which we will refer to as an *error vector*. Note that if an error vector $\mathbf{e}$ occurs, its components are added to the channel inputs according to the coding order. Then the output of a channel (a,b) is a function of both the input $\mathbf{w}$ to the network and the error vector $\mathbf{e}$ that occurs, and we denote it by $\psi_{(a,b)}(\mathbf{w},\mathbf{e})$.

Lemma 1: [1] For all $(a,b) \in \mathcal{E}^*$, any inputs $\mathbf{w}$ and $\mathbf{w}'$ to the network, and any error vectors $\mathbf{e}$ and $\mathbf{e}'$,

$$\psi_{(a,b)}(\mathbf{w}+\mathbf{w}', \mathbf{e}+\mathbf{e}') = \psi_{(a,b)}(\mathbf{w},\mathbf{e}) + \psi_{(a,b)}(\mathbf{w}',\mathbf{e}').$$

By this lemma, for any network input $\mathbf{w}$ and error vector $\mathbf{e}$, we have,

$$\psi_{(a,b)}(\mathbf{w},\mathbf{e}) = \psi_{(a,b)}(\mathbf{w},\mathbf{0}) + \psi_{(a,b)}(\mathbf{0},\mathbf{e}).$$

Upon observing that $\psi_{(a,b)}(\mathbf{w},\mathbf{0}) = \tilde{\phi}_{(a,b)}(\mathbf{w})$ and defining $\theta_{(a,b)}(\mathbf{e}) = \psi_{(a,b)}(\mathbf{0},\mathbf{e})$, we can write

$$\psi_{(a,b)}(\mathbf{w},\mathbf{e}) = \tilde{\phi}_{(a,b)}(\mathbf{w}) + \theta_{(a,b)}(\mathbf{e}).$$

For each $u \in \mathcal{U}$, define the set of *t-error patterns*

$$\Xi(V,t,u) = \{(\theta_{(a,u)}(\mathbf{e})M^{-1}(u), (a,u) \in \Gamma_+(u)) : w_H(\mathbf{e}) \le t\},$$

where $w_H(\mathbf{e})$ denotes the Hamming weight of $\mathbf{e}$. Define

$$\Delta(V,t) = \cup_{u \in \mathcal{U}} \{\mathbf{f} = \mathbf{g}' - \mathbf{g} : \mathbf{g}, \mathbf{g}' \in \Xi(V,t,u)\}.$$

Theorem 4 (Gilbert-Varshamov Bound): [1] For all positive integer A with

$$(A-1)|\Delta(V,t)| < q^n,$$

there exists a t-error-correcting code with source alphabet size A (i.e., $|\mathcal{Z}| = A$). In particular, for all positive integers k such that

$$|\Delta(V,t)| < q^{n-k},$$

one can construct a linear code of at least k dimensions (i.e., $|\mathcal{Z}| = q^k$) via the given generic LCM V.

To obtain a lower bound on $|\mathcal{Z}|$ which does not depend on the particular choice of LCM V, we employ the upper bound

$$|\Delta(V,t)| \le \sum_{u \in \mathcal{U}} |\Xi(V,t,u)|^2 \le |\mathcal{U}| \left[\sum_{j=0}^{t} \binom{K}{j} (q-1)^j\right]^2.$$

This allows us to obtain a lower bound on $|\mathcal{Z}|$ as follows.

Corollary 1: For a network with $\min_{u \in \mathcal{U}} c(s,u) = n$, for all $\epsilon > 0$ and sufficiently large q (depending on the network and ϵ), one can construct a t-error correcting code with source alphabet $\mathcal{Z}$ such that

$$\log |\mathcal{Z}| \ge (n-2t)(1-\epsilon)\log q. \tag{5}$$

Moreover, for all sufficiently large prime power q and $k = n - 2t - 1$ one can find a k-dimensional linear t-error correcting code for the network.

IV. The Singleton Bound

In this section, we present the Singleton bound for network error-correcting codes. The tightness of this bound can be shown via an enhancement of the Varshamov bound in Theorem 4.

Theorem 5 (Singleton Bound): Let $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ be an acyclic network and $n = \min_{u \in \mathcal{U}} c(s,u)$. If there exists a t-error-correcting code for the network with source alphabet $\mathcal{Z}$, then

$$\log |\mathcal{Z}| \le (n-2t)\log q. \tag{6}$$

Comparing the Singleton bound (upper bound on $|\mathcal{Z}|$) with the bound in Corollary 1 (lower bound on $|\mathcal{Z}|$), which is a consequence of the Varshamov bound, we see that the two bounds differ only by the ϵ in the latter. However, the gap between the two bounds can be quite large because according to the proof of Corollary 1, $q \to \infty$ as $\epsilon \to 0$. By employing a more elaborate technique to estimate the size of the difference set $\Delta(V,t)$, we can prove the following strengthened Varshamov bound, showing the tightness of the Singleton bound.

Theorem 6 (Strengthened Varshamov Bound): For a fixed arbitrary acyclic network with $\min_{u \in \mathcal{U}} c(s,u) = n$ and all sufficiently large q, there exists an $(n-2t)$-dimensional linear t-error-correcting code for the network.

For the proofs of these results, we refer the reader to [5].

V. Discussion

Together with our previous work in [1], we have generalized the most important bounds in classical algebraic coding theory, namely the Hamming

bound, the Gilbert-Varshamov bound, and the Singleton bound, for network error-correcting codes. Moreover, the tightness of the Singleton bound is preserved. Conceivably, similar network generalizations can be obtained for classical convolutional codes, turbo codes, LDPC codes, etc. These are very important problems for future research.

Algebraic network error-correcting codes can potentially be applied in networks for which the nodes are connected by noisy channels. When these channels are independent and memoryless, separation theorems for network coding and channel coding have been proved [6] [7] [8]. In other words, link-by-link error correction does not sacrafice asymptotic optimality, and one can simply employ the best known channel codes for point-to-point communication (e.g., turbo codes, LDPC codes). However, when the channels either are not independent or have memory, no separation theorem for network coding and channel coding exists. In such cases, algebraic network error-correcting codes can be very useful.

In addition to bandwidth optimality, computational complexity is an important issue in real implementations. With link-by-link error correction, decoding needs to be done at each intermediate node, which is computationally expensive. By contrast, by employing linear network error-correcting codes, only a very simple linear transformation is needed at each intermediate node, and decoding is done only at the sink nodes.

In certain network applications, errors are injected by malicious nodes in the network instead of being caused by noise in the channels. In such cases, algebraic network error-correcting codes simply render the natural solution.

REFERENCES

[1] N. Cai and R. W. Yeung, "Network coding and error correction," 2002 IEEE Information Theory Workshop, Bangalore, India, Oct 20-25, 2002.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, IT-46: 1204-1216, 2000.

[3] S.-Y. R. Li, R. W. Yeung and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, IT-49: 371-381, 2003.

[4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on network coding*, vol. 11, 782-795, 2003.

[5] "Network error correction," preprint.

[6] L.Song and R. W. Yeung, "A separation principle of communication network," 1999 IEEE Information Theory and Networking Workshop, Metsovo, Greece, Jun 27-Jul 1, 1999.

[7] S. Borade, "Network Information Flow: Limits and Achievability," 2002 IEEE International Symposium on Information Theory, Lausanne, Switzerland, Jun 30-Jul 5, 2002.

[8] L. Song and R. W. Yeung, and N. Cai, "A separation theorem for single source network coding," to appear in *IEEE Transactions on Information Theory*.