COMMUNICATIONS IN INFORMATION AND SYSTEMS
Vol. 6, No. 1, pp. 19-36, 2006

002

NETWORK ERROR CORRECTION, PART I: BASIC CONCEPTS AND UPPER BOUNDS

RAYMOND W. YEUNG* AND NING CAI†

Abstract. Error correction in existing point-to-point communication networks is done on a link-by-link basis, which is referred to in this paper as *classical error correction*. Inspired by network coding, we introduce in this two-part paper a new paradigm called *network error correction*. The theory thus developed subsumes classical algebraic coding theory as a special case. In Part I, we discuss the basic concepts and prove the network generalizations of the Hamming bound and the Singleton bound in classical algebraic coding theory. By studying a few elementary examples, the relation between network error correction and classical error correction is investigated.

Key words: Network coding, multicast, error correction, algebraic coding, Hamming bound, Singleton bound.

1. Introduction.

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 represents 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 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 a *code alphabet* $\mathcal{X}$ that can be transmitted on channel (a, b) in one transmission. $R_{(a,b)}$ is also referred to as the *capacity* (in the sense of graph theory, e.g., [1]) 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 one 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 one instead of by the single edge (a, b) of capacity 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.

*R. W. Yeung is with Department of Information Engineering, The Chinese University of Hong Kong, N.T., Hong Kong. Email: whyeung@ie.cuhk.edu.hk

†N. Cai is with The State Key Lab. of ISN, Xidian University, Xi'an, Shaanxi, 710071, China. He was with Department of Information Engineering, The Chinese University of Hong Kong, N.T., Hong Kong when this work was done. Email: caining@mail.xidian.edu.cn

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 and was first applied in satellite networks [2], may allow more information to be transmitted than what would be possible by routing alone [3]. In fact, routing is a special case of network coding.

Denote by $\text{maxflow}(s, u)$ the *maximum flow* between node s and node u, where $u \in \mathcal{U}$. In this paper, all logarithms are in the base 2 unless otherwise specified. The following fundamental theorem was proved in [3].

THEOREM 1. *For all sufficient large code alphabet $\mathcal{X}$, 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*

$$\log |\mathcal{Z}| \le n \log |\mathcal{X}|, \tag{1}$$

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

It was proved in [4] by a vector space approach and later in [5] by a matrix approach that it suffices to consider linear network codes. An algorithm for constructing optimal linear network codes was provided in [4], which was subsequently refined in [6] and [7]. For a tutorial on the subject, we refer readers to [7].

In the present two-part paper, we consider the situation that the channels in the network are not necessarily error-free. Error correction in all existing communication networks is done on a link-by-link basis. Inspired by network coding, we consider distributed error correction in a network, called *network error correction*. In Part I, the basic concepts are discussed and upper bounds on the size of the source alphabet of a network error-correcting code is proved. These upper bounds are the network generalizations of the Hamming bound and the Singleton bound. In Part II, lower bounds on the source alphabet that are network generalizations of the Gibert-Varshamov bound will be proved. The Hamming bound, the Singleton bound, and the Gibert-Varshamov bound are fundmental bounds in classical algebraic coding theory [9] [10] [11] [12].

The rest of Part I is organized as follows. In Section 2, we introduce the formulation for network error correction and define a network error-correcting code. The main results are in Sections 3 and 4, where we prove in Section 3 the Hamming bound and some related bounds and in Section 4 the Singleton bound for network error-correcting codes. Section 5 contains a few examples that illustrate the basic properties of network error-correcting codes. The conclusion is in Section 6.

2. Problem Formulation. 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 $d_+(a) = |\Gamma_+(a)|$ the in-degree and $d_-(a) = |\Gamma_-(a)|$ the out-degree of node a. Input and output edges of a set of nodes $A \subset V$, denoted respectively by $\Gamma_+(A)$ and $\Gamma_-(A)$, are called the boundary edges of A, and the other edges, i.e., the edges with both end nodes in A, are called inner edges. The set of inner edges of A is denoted by $\Im A$. Let (A, B) be a partition of the node set $\mathcal{V}$, and define the cut for the partition (A, B) by

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

In other words, $cut(A, B)$ is just $\Gamma_-(A)$ or $\Gamma_+(B)$. The quantity $\sum_{(a,b)\in cut(A,B)} R_{(a,b)}$ is called the volume of $cut(A, B)$. Furthermore, $cut(A, B)$ is called a cut between two nodes a and b if $a \in A$ and $b \in B$. 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 [1] in graph theory is equal to $\text{maxflow}(s, u)$.

In a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a sequence of edges $\{(a_1, b_1), (a_2, b_2), \ldots (a_k, b_k)\}$ is called a path if for $i = 2, \ldots, k$, $b_{i-1} = a_i$, and it is further called a cycle if $b_k = a_1$. A directed graph without a cycle is called acyclic and the corresponding network is called an acyclic network. In the present paper, we always assume that the network is acyclic unless otherwise specified.

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 node set. Under this assumption, a coding order always starts with the source node s. Unless otherwise specified, we assume that the code alphabet $\mathcal{X}$ is equal to $GF(q)$ for some prime power q.

DEFINITION 1. *Let $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ be a network, and $r_{(a,b)} \le R_{(a,b)}$ be positive integers for $(a, b) \in \mathcal{E}$. A network code for the network $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ is a family of local encoding 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 an $(s, b) \in \Gamma_-(s)$ and then sends the values of $\phi_{(s,b)}(z)$ to their output node b via

channels (s, b). Then the second node in the order 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 *global encoding function* $\tilde{\phi}_{(a,b)}$ from $\mathcal{Z}$ to $\mathcal{X}^{r_{(a,b)}}$ for each $(a, b) \in \mathcal{E}$ is induced in the natural way by such a scheme under the assumption that no error occurs when a symbol is transmitted through a channel (errors will be discussed below). Obviously, these functions induced 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. 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 channels 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. In the rest of the paper, we will regard an error as a value in $\mathcal{X}$, with the output of a channel being the modulo q sum of the input and the error applied to that channel. (There is no error when the value of the error is zero.)

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 code symbol is sent on a channel, multiple errors can occur. (In terms of the edge set $\mathcal{E}^*$, however, only one code symbol is sent on each channel.) A τ-error is said to occur (in the network) if the total number of errors occurring in all the channels is equal to τ. In general an error-pattern set Υ is a collection of subsets of channels in the network. For a subset E of channels, an E-error is said to occur if an error occurs in every channel in E.

Definition 2. *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 $u \in \mathcal{U}$. A network code is Υ-error-correcting if it can correct E-errors for all $E \in \Upsilon$.*

As the empty set as an "error pattern" corresponds to the case when no error occurs, we always assume that it is in Υ when we refer to an Υ-error-correcting code. By definition, an Υ-error-correcting code is a t-error-correcting code if Υ is the collection of subsets of channels with cardinalities not larger than t. Throughout this two-part paper, we mainly treat t-error-correcting codes for single source acyclic

networks except in Section 2 in Part II where we present general constructions of good Υ-error-correcting codes. Nevertheless, most of our results on t-error-correcting codes can readily be extended to Υ-error-correcting codes.

Error correction in all existing communication networks is done on a link-by-link basis, which is a special case of the *network error correction* approach proposed in this work. In our subsequent discussion, we will refer to error-correcting codes for point-to-point communications as *classical error-correcting codes*. Specifically, these are network error-correcting codes on the network with one source node and one sink node.

As a preparation for further discussion, we first prove in the next lemma some basic properties of a network error-correcting code.

LEMMA 1. *Consider a fixed network code on a given acyclic network.*
i) For a channel $(a,b) \in \mathcal{E}^*$, the output of channel (a,b) is a function of the outputs of the channels in $\Gamma_+(a)$ and the error at channel (a,b).
ii) For $A \subset \mathcal{V} \setminus \{s\}$, if no error occurs at the channels in $\Im A$, the outputs of the channels in $\Im A$ are functions of the outputs of all the channels in $\Gamma_+(A)$, and the output of a channel $(a,b) \in \Gamma_-(A)$ is a function of the outputs of all the channels in $\Gamma_+(A)$ and the error at channel (a,b).

Proof.
i) Obvious.
ii) We first show that the second part follows from i) together with the first part. By i), the output of channel (a,b) is a function of the outputs of the channels in $\Gamma_+(a)$. Upon observing

$$\Gamma_+(a) = [\Gamma_+(a) \cap \Gamma_+(A)] \cup [\Gamma_+(a) \cap \Im(A)]$$

and invoking the first part, the second part follows. Therefore, we only need to prove the first part.

The first part is proved by induction on $|A|$, which by i) is obviously true for $|A| = 1$. Consider a set $A \subset \mathcal{V}\backslash\{s\}$ such that $|A| \geq 2$. Let $a \in A$ be the first node to encode according to any fixed coding order so that

$$\Gamma_+(a) \cap \Gamma_-(A\backslash\{a\}) = \emptyset,$$

i.e., there exists no channel from a node in $A\backslash\{a\}$ to node a. It follows that

$$\Gamma_+(a) \subset \Gamma_+(A). \tag{2}$$

Now consider

$$\Im(A) = \Im(A\backslash\{a\}) \cup [\Gamma_-(a)\backslash\Gamma_+(A)]$$

and note that the set in the square brackets above is a subset of $\Gamma_-(a)$. By i), the outputs of the channels in $\Gamma_-(a)$ are functions of the outputs of the channels in

$\Gamma_+(a) \subset \Gamma_+(A)$ (cf. (2)). Thus it remains to show that the outputs of the channels in $\Im(A\backslash\{a\})$ are also functions of the outputs of the channels in $\Gamma_+(A)$. By the induction hypothesis, the outputs of the channels in $\Im(A\backslash\{a\})$ are functions of the outputs of the channels in

$$\Gamma_+(A\backslash\{a\}) = [\Gamma_+(A\backslash\{a\}) \cap \Gamma_-(a)] \cup [\Gamma_+(A)\backslash\Gamma_+(a)].$$

In the above, the set in the second pair of square brackets is a subset of $\Gamma_+(A)$, and the set in the first pair of square brackets is a subset of $\Gamma_-(a)$, where we have shown in the foregoing that the outputs of the channels in $\Gamma_-(a)$ are functions of the outputs of the channels in $\Gamma_+(A)$. Therefore, we conclude that the outputs of the channels in $\Gamma_+(A\backslash\{a\})$ are functions of the outputs of the channels in $\Gamma_+(A)$. This completes the proof. □

3. The Hamming Bound. Upon defining a t-error-correcting code for an acyclic network $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$, we first present in this section an upper bound on the source alphabet $|\mathcal{Z}|$. This upper bound is a network generalization of the celebrated Hamming bound in classical algebraic coding theory. We then prove that the "projection" of a t-error-correcting code for a network across any so-called *regular* cut is a classical t-error-correcting code. This result renders a further upper bound on $|\mathcal{Z}|$ for certain types of networks.

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.

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,*

$$out(\phi, t, cut(A,B), z) \cap out(\phi, t, cut(A,B), z') = \emptyset \tag{3}$$

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

Based on this observation, we can prove the following "sphere-packing" bound[1], or the Hamming bound.

THEOREM 2 (Hamming Bound). *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 $\mathcal{Z}$, where $t \leq n$, then*

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

[1]It is in general not exactly a sphere as in the case of a classical t-error-correcting code. See the discussion following the proof of Theorem 2

Proof. Let $\phi = \{\phi_{(a,b)} : (a,b) \in \mathcal{E}^*\}$ be a t-error-correcting code for the network. By Observation 1, it is sufficient for us to show that for all sink nodes $u \in \mathcal{U}$, any cut $cut(A,B)$ between s and u with volume m (say), where $m \geq t$, and all $z \in \mathcal{Z}$,

$$|out(\phi, t, cut(A,B), z)| \geq \sum_{i=0}^{t} \binom{m}{i} (q-1)^i. \tag{4}$$

We first fix a cut $cut(A,B)$ and an output $z \in \mathcal{Z}$ of the information source. Let w_H denote the Hamming weight, i.e., for $x^m = (x_1, x_2, \ldots, x_m)$, $w_H(x^m) := |\{i : x_i \neq 0\}|$. To facilitate our discussion, we will regard the errors in the network as injected by a jammer in following sense. The jammer labels the channels in $cut(A,B)$ as $(a_1,b_1),(a_2,b_2),\ldots,(a_m,b_m)$ according to the coding order, chooses an "error sequence" $e^m := (e_1, e_2, \ldots, e_m) \in \mathcal{X}^m$ with $w_H(e^m) \leq t$, and adds $e_1, e_2, \ldots, e_m$ in modulo q to the outputs of the channels $(a_1,b_1),(a_2,b_2),\ldots,(a_m,b_m)$ successively. Although the jammer may as well inject errors at channels not in $cut(A,B)$, we consider for the time being only scenarios as prescribed above because this will suffice to prove the bound in (4). Observe that for a fixed source message z, for $1 \leq j \leq m$, once $e_1, e_2, \ldots, e_j$ are fixed, the outputs of the channels $(a_1,b_1),(a_2,b_2),\ldots,(a_j,b_j)$ are determined. Based on this observation, we now show that the outputs of the channels in $cut(A,B)$ cannot be identical for two distinct error sequences $e^m = (e_1, e_2, \ldots, e_m)$ and $(e^m)' = (e'_1, e'_2, \ldots, e'_m)$. Let i_0 be the smallest i such that $e_i \neq e'_i$, i.e., $e_{i_0} \neq e'_{i_0}$ and $e_i = e'_i$ for all $i < i_0$, and consider the jammer choosing e^m or $(e^m)'$ and add it to the inputs of the channels in $cut(A,B)$ according to the coding order. Then in either case, the outputs of channels $(a_1,b_1),(a_2,b_2),\ldots,(a_{i_0-1},b_{i_0-1})$ are identical, because the operations of the process are exactly the same until e_{i_0} or e'_{i_0} is added to the input of channel (a_{i_0},b_{i_0}). Since $e_{i_0} \neq e'_{i_0}$, the output of channel (a_{i_0},b_{i_0}) must be different for the two cases. Thus we have shown that the outputs of the channels in $cut(A,B)$ cannot be identical for two distinct error sequences e^m and $(e^m)'$. As the set of outputs of the channels in $cut(A,B)$ caused by input z as prescribed above is a subset of $out(\phi, t, cut(A,B), z)$ and there are totally $\sum_{i=0}^{t} \binom{m}{i}(q-1)^i$ possibilities for e^m, (4) holds. This completes our proof. □

Although the RHS of (4) is exactly equal to the volume of a sphere in $\mathcal{X}^m$ with radius t, (4) by no means implies 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 (for a counterexample, see Example 3 in Section 5). 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. In the rest of the section, we will develop further results along this line.

A pair $(\mathcal{A}, \preceq_{\mathcal{A}})$ is called a partial order set, or a poset, if $\preceq_{\mathcal{A}}$ is a partial order

in $\mathcal{A}$. For a poset $(\mathcal{A}, \preceq_{\mathcal{A}})$, $a, b \in \mathcal{A}$ are said to be incomparable if neither $a \preceq_{\mathcal{A}} b$ nor $b \preceq_{\mathcal{A}} a$. A pairwise incomparable subset of $\mathcal{A}$ is called an *antichain*.

DEFINITION 3. *Consider an acyclic network $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$. For a partition (A, B) of the node set $\mathcal{V}$, $cut(A, B)$ is a regular cut if its members form an antichain for the poset $(\mathcal{E}, \preceq_e)$, i.e., if $(a, b), (c, d) \in cut(A, B)$, then there exists no path either from (a, b) to (c, d) or from (c, d) to (a, b).*

DEFINITION 4. *An acyclic network is regular if $\min_u c(s, u) = \min_u rg(s, u)$, where $rg(s, u)$ is the minimum volume of a regular cut between s and u.*

THEOREM 3. *Let $\{\phi_{(a,b)} : (a, b) \in \mathcal{E}^*\}$ be a t-error-correcting code for an acyclic network $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ with source alphabet $\mathcal{Z}$ and code alphabet $\mathcal{X}$, and let $n = \min_{u \in \mathcal{U}} rg(s, u)$. Then*
i) if $cut(A, B)$ is a regular cut between the source node s and a sink node u, then $\{(\tilde{\phi}_{(a,b)}(z), (a, b) \in cut(A, B)) : z \in \mathcal{Z}\}$ is a classical t-error correcting code with alphabet $\mathcal{X}$, and consequently
ii)

$$|\mathcal{Z}| \leq A(n, t, q),$$

and in the case that the code is linear,

$$|\mathcal{Z}| \leq L(n, t, q),$$

where $q = |\mathcal{X}|$, and $A(n, t, q)$ and $L(n, t, q)$ are the size of an optimal classical q-ary t-error-correcting code of length n and the size of an optimal classical linear q-ary t-error-correcting code of length n, respectively.

Proof. Note that if the code $\{\phi_{(a,b)} : (a, b) \in \mathcal{E}^*\}$ is linear, so is the code $\{(\tilde{\phi}_{(a,b)}(z), (a, b) \in cut(A, B)) : z \in \mathcal{Z}\}$ for any cut $cut(A, B)$. Then ii) follows from i).

Now we prove i). Let $\{\phi_{(a,b)} : (a, b) \in \mathcal{E}^*\}$ be any given t-error correcting code, and denote the *down set* of a channel (a, b) by

$$down[(a, b)] := \{(c, d) : (c, d) \preceq_e (a, b), \text{ and } (c, d) \in \mathcal{E}^*\}.$$

Consider a regular cut $cut(A, B)$ between the source node s and a sink node u with volume n. The input of any channel (a, b) on $cut(A, B)$ depends only on the source message z and the errors occurring in the channels on $down[(a, b)]$. Since $cut(A, B)$ is regular for all $(a, b) \in cut(A, B)$, $down[(a, b)]$ is disjoint with $cut(A, B)$. In particular, when no error occurs in all the channels that are not on $cut(A, B)$, the inputs of the channels on $cut(A, B)$ depend only on z, and they are exactly equal to $(\tilde{\phi}_{(a,b)}(z), (a, b) \in \mathcal{E}^*)$.

As in the proof of Theorem 2, we consider a jammer injecting errors in the channels on $cut(A, B)$. Now assume that $\{(\tilde{\phi}_{(a,b)}(z), (a, b) \in cut(A, B)) : z \in \mathcal{Z}\}$ is not a

classical t-error correcting code, i.e., there exists $z, z' \in \mathcal{Z}$ such that the Hamming distance between $(\tilde{\phi}_{(a,b)}(z), (a,b) \in cut(A,B))$ and $(\tilde{\phi}_{(a,b)}(z'), (a,b) \in cut(A,B))$ is less than $2t+1$. Then with the inputs of the channels on $cut(A,B)$ being respectively $(\tilde{\phi}_{(a,b)}(z), (a,b) \in cut(A,B))$ and $(\tilde{\phi}_{(a,b)}(z'), (a,b) \in cut(A,B))$, by injecting at most t errors in each case (we assume that no error occurs in all the channels not on $cut(A,B)$), it is possible for the jammer to produce a common set of values at the outputs of the channels on $cut(A,B)$. By ii) of Lemma 1, it is not always possible for the sink node u to distinguish whether the source message is z or z' when less than or equal to t errors occur. This is a contradiction because $\{\phi_{(a,b)} : (a,b) \in \mathcal{E}^*\}$ is a t-error correcting code. Hence, we have shown that $\{(\tilde{\phi}_{(a,b)}(z), (a,b) \in cut(A,B), z \in \mathcal{Z}\}$ is a classical t-error correcting code. □

The bounds in Theorem 3 and their tightness will be investigated in Section 5.

4. The Singleton Bound. In this section, we prove the network generalization of the Singleton bound for classical error-correcting codes.

THEOREM 4 (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*

$$\log |\mathcal{Z}| \leq (n-2t) \log q, \tag{5}$$

where $n - 2t > 0$.

Proof. Let $\{\phi_{(a,b)} : (a,b) \in \mathcal{E}^*\}$ be a t-error-correcting network code transmitting an information source with alphabet $\mathcal{Z}$ and assume that

$$|\mathcal{Z}| > q^{n-2t}. \tag{6}$$

Let $u \in \mathcal{U}$ be a sink such that there exists a partition (A,B) of $\mathcal{V}$ with $s \in A, u \in B$ and $|cut(A,B)| = n$. We order the the channels on $cut(A,B)$ according to a linear extension of $\preceq_e$ as $(a_1,b_1), (a_2,b_2), \ldots, (a_n,b_n)$ such that that there is a path from (a_i,b_i) to (a_j,b_j) for $i \neq j$ implies that $i < j$. By (6) there exist $z, z' \in \mathcal{Z}$ such that $z \neq z'$ and $\tilde{\phi}_{(a_i,b_i)}(z) = \tilde{\phi}_{(a_i,b_i)}(z')$ for $i = 1, 2, ..., n-2t$. Thus we can write

$$(\tilde{\phi}_{(a_1,b_1)}(z), \tilde{\phi}_{(a_2,b_2)}(z), ..., \tilde{\phi}_{(a_n,b_n)}(z) = (x_1, x_2, ..., x_{n-2t}, u'_1, u'_2, ...u'_t, y_1, y_2, ..., y_t)$$

and

$$(\tilde{\phi}_{(a_1,b_1)}(z'), \tilde{\phi}_{(a_2,b_2)}(z'), ..., \tilde{\phi}_{(a_n,b_n)}(z') = (x_1, x_2, ..., x_{n-2t}, u_1, u_2, ...u_t, y'_1, y'_2, ..., y'_t).$$

In the following, like in the proof of Theorem 2, we regard the errors occurring in the network as being injected by a jammer according to the coding order. We will show that it is possible for the jammer to produce exactly the same outputs at all

the channels on $cut(A, B)$ when the message is z or z' by injecting at most t errors in either case.

Suppose the message is z. The jammer will inject t errors at channels $(a_{n-2t+1}, b_{n-2t+1}), (a_{n-2t+2}, b_{n-2t+2}), \ldots, (a_{n-t}, b_{n-t})$ in this order as follows. First, the jammer injects an error at channel (a_{n-2t+1}, b_{n-2t+1}) to change the output from u'_1 to u_1. By doing so, the outputs of channels $(a_{n-2t+2}, b_{n-2t+2}), (a_{n-2t+3}, b_{n-2t+3}), \ldots, (a_n, b_n)$ may be affected, but not the outputs of channels $(a_1, b_1), (a_2, b_2), \ldots, (a_{n-2t}, b_{n-2t})$. Let $u'_i(j)$ and $y_i(j)$ denote the outputs of channels (a_{n-2t+i}, b_{n-2t+i}) and (a_{n-t+i}, b_{n-t+i}), respectively after the jammer has injected an error at channel (a_{n-2t+i}, b_{n-2t+i}), where $i = 1, 2, \ldots, t$ (with $u'_i(1) = u_1$). Then the jammer injects an error at channel (a_{n-2t+2}, b_{n-2t+2}) to change its output from $u'_i(1)$ to u_2. The process continues until the jammer finishes injecting t errors at channels (a_{n-2t+1}, b_{n-2t+1}), (a_{n-2t+2}, b_{n-2t+2}), $\ldots$, (a_{n-t}, b_{n-t}), and the outputs of these channels are $(x_1, x_2, ..., x_{n-2t}, u_1, u_2, ...u_t, y'_1(t), y'_2(t), ..., y'_t(t))$.

Now suppose the message is z'. In exactly the same way as we have described above, by injecting t errors at channels $(a_{n-t+1}, b_{n-t+2}), (a_{n-t+3}, b_{n-t+3}), \ldots, (a_n, b_n)$, the jammer can produce the outputs $(x_1, x_2, ..., x_{n-2t}, u_1, u_2, ...u_t, y'_1(t), y'_2(t), ..., y'_t(t))$. By Observation 1, the sink u cannot distinguish between the messages z and z', which is a contradiction because the network code $\{\phi_{(a,b)} : (a, b) \in \mathcal{E}^*\}$ is t-error-correcting Hence, the assumption in (6) cannot hold. This completes the proof. □

It is well-known that the Singleton bound for classical error-correcting code on a sufficiently large field is achievable, i.e., the bound is tight. We will show in Part II that the tightness of the Singleton bound is preserved in the network setting.

5. Some Examples. We proved in Theorem 2 an explicit upper bound on the size of the source alphabet of a network error-correcting code, namely the Hamming Bound. In Theorem 3, we proved further upper bounds in terms of upper bounds for classical error-correcting codes. In this section, we explore these bounds by studying in details a few examples, from which we can see that the construction of network error-correcting codes is considerably more complicated than that of classical error-correcting codes.

EXAMPLE 1. *Let $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ be an acyclic network with a single sink and minimum cut n, i.e., $\mathcal{U} = \{u\}$ and $c(s, u) = n$. Then a message from an information source with alphabet size $|\mathcal{Z}|$ no larger than $A(n, t, q)$ ($L(n, t, q)$) can be transmitted by a (linear) t-error correcting code as follows. Given a t-error-correcting non-linear or linear code, let the source message be encoded into a codeword $c = (c_1, c_2, \ldots, c_n)$. By the Max-flow Min-cut theorem, there exist n channel-disjoint paths from source node s to the unique sink node u. Then we construct a network code by simply using the jth path to carry the symbol c_j. This network code obviously can correct t errors that may occur at any channels in the network.*

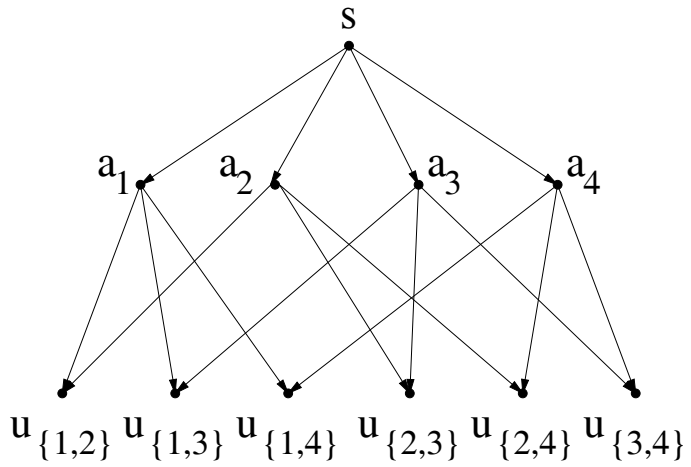

FIG. 1. *The* $\binom{4}{2}$ *combination network.*

The scheme in the above example works for any single sink network. In particular, if the network is regular, i.e., $n = c(s,u) = rg(s,u)$, then the scheme further shows that the bounds in Theorem 3 are tight. However, we will show in Example 3 that these bounds are generally not tight.

EXAMPLE 2. *Consider the* $\binom{n}{k}$ *combination network* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, where* $k < n$*, defined as follows. Let the node set*

$$\mathcal{V} = \{s\} \cup \{a_1, a_2, \ldots, a_n\} \cup \mathcal{U},$$

where s *and* $\mathcal{U}$ *are respectively the source node and the set of sink nodes with* $|\mathcal{U}| = \binom{n}{k}$*. The members of* $\mathcal{U}$ *are labeled as* u_A*, where* A *are* k*-element subsets of* $\{1, 2, \ldots, n\}$*. Let the edge set*

$$\mathcal{E} = \{(s, a_1), (s, a_2), \ldots, (s, a_n)\} \cup \bigcup_{A:|A|=k} \{(a_i, u_A) : a_i \in A\}.$$

Each edge in $\mathcal{E}$ *is assigned capacity one. The* $\binom{4}{2}$ *combination network is illustrated in Fig. 1. Evidently, the* $\binom{n}{k}$ *combination network is regular.*

Consider an arbitrary t*-error-correcting network code* $\{\phi_{(a,b)} : (a,b) \in \mathcal{E}^*\}$ *on the* $\binom{n}{k}$ *combination network, where* $2t + 1 \leq k$*. We adopt the abbreviation* $a_i(z)$ *for* $\phi_{(s,a_i)}(z)$ *for all message* $z \in \mathcal{Z}$*, and write* $a^n(z) = (a_1(z), a_2(z), \ldots, a_n(z))$*. We will show that* $\{a^n(z) : z \in \mathcal{Z}\}$ *is a classical error-correcting code with minimum distance at least* $n - k + 2t + 1$*.*

Since there is only one input edge to each node a_i*, we assume without loss of generality that the symbol sent on a channel* (a_i, u_A)*, where* $a_i \in A$*, is simply the symbol received by node* a_i*. Observe that by accessing a distinct subset of* k *nodes in* $\{a_1, a_2, \ldots, a_n\}$*, each sink node can decode the message correctly when at most* t *errors occur at the channels* (s, a_i)*. Therefore, we see that* $\{a^n(z) : z \in \mathcal{Z}\}$ *is a*

classical error-correcting code that can correct $n-k$ erasures and t errors. Hence, the minimum distance of this code is at least $n-k+2t+1$ (see for example [10], p.256).

Conversely, it is readily seen that if $\{a^n(z) : z \in \mathcal{Z}\}$ is a classical error-correcting code with minimum distance at least $n-k+2t+1$, then $\{\phi_{(a,b)} : (a,b) \in \mathcal{E}^*\}$ is a t-error-correcting network code on the $\binom{n}{k}$ combination network. Thus we conclude that an optimal t-error-correcting (linear) network code on the $\binom{n}{k}$ combination network is equivalent to an optimal classical error-correcting code with minimum distance at least $n-k+2t+1$.

A natural way to think of a message being transmitted from the source node to a sink node over a network is to imagine its "components" being carried by the paths in a flow from the source node to the sink node. Theorem 3 gives the important characterization that in order for the network code to be able to correct t errors, the projection of the code across any regular cut between the source node and any sink node must be a classical t-error-correcting code. For the regular networks in Examples 1 and 2, an optimal network error-correcting code is constructed from a classical error-correcting code. Roughly speaking, we first decompose the maximum flows to all the sink nodes into paths and then let each path carry a symbol of the codeword of a properly chosen classical error-correcting code. One may wonder whether this theme applies to all regular networks. It turns out that the problem of constructing network error-correcting codes, even for regular networks, is much more complicated, especially when the code alphabet is small. This is illustrated in the next example.

EXAMPLE 3. *Consider the network $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ in Fig. 2 which is specified by*

$$\mathcal{U} = \{u_1, u_2\}$$

$$\mathcal{V} = \{s\} \cup \{a, b, c, d, e, f, g\} \cup \mathcal{U}$$

and

$$\mathcal{E} = \{(s,a), (s,b), (s,c), (a,d), (a,u_1), (b,f), (b,u_1), (c,d), (c,u_2), (d,e), (e,f), (e,u_2), (f,g), (g,u_1), (g,u_2)\}.$$

All the channels in $\mathcal{E}$ have capacity one. Let us consider binary codes for this network, i.e., the encoding alphabet is given by $\mathcal{X} = \{0,1\}$.

It is easy to verify for this network that $c(s,u_i) = rg(s,u_i) = 3$ for $i = 1,2$, so it is regular. In light of the existence of a classical binary 1-error-correcting (3,1) code, if the bounds in Theorem 3 are tight, then there would exist a binary 1-error-correcting network code

$$\{\phi_{(s,a)}, \phi_{(s,b)}, \phi_{(s,c)}, \ldots, \phi_{(g,u_1)}, \phi_{(g,u_2)}\} \tag{7}$$

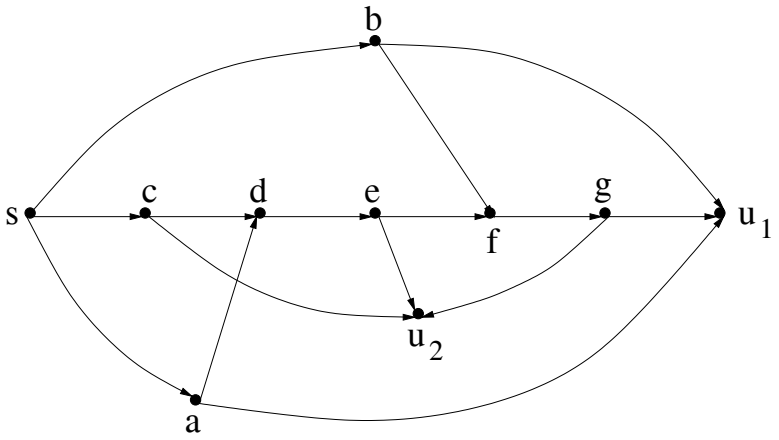

FIG. 2. *The network $\mathcal{G}$ for Example 3.*

that multicasts a message from the binary source alphabet $\mathcal{Z} = \{0, 1\}$.

Assume that the network code in (7) is 1-error-correcting. We will show that this leads to a contradiction. Without loss of generality, we let $\phi_{(s,a)}(0) = \phi_{(s,b)}(0) = \phi_{(s,c)}(0) = 0$ and $\phi_{(s,a)}(1) = \phi_{(s,b)}(1) = \phi_{(s,c)}(1) = 1$ since by symmetry one can exchange the roles of 0 and 1 componentwise. We observe that for a particular network code, a channel can be removed if its local encoding function can only take one value because such a channel does not convey any information. For the network in Fig. 2, if the local encoding function of any channel can take only one value, then by removing that channel from the network, we will find a sink node such that the minimum cut between the source node and this sink node is reduced to 2. This contradicts Theorem 3 because of the nonexistence of a (2,1) code that can correct 1 error. This means that the local encoding functions of all the channels must take two values. In particular, a local encoding function of a channel whose input-nodes has in-degree one must be a bijection, so we may assume with loss of generality that it is the identity function.

Let us consider the local encoding function $\phi_{(d,e)}$ with the first and the second arguments being the outputs of channels (a, d) and (c, d), respectively. We will show that there is no way to choose the function $\phi_{(d,e)}$ such that the code is able to correct one error.

First, without loss of generality, let

$$\phi_{(d,e)}(0, 1) = 0. \tag{8}$$

Let us consider the case that the source message is 1 and an error occurs at channel (s, a). It is easy to see that the outputs of channels (a, d) and (c, d) are 0 and 1, respectively, so that by (8), channel (d, e) outputs a 0. Then across the cut

$$cut(\{s, c, d, u_2\}, \mathcal{V} \setminus \{s, c, d, u_2\}) = \{(s, a), (s, b), (d, e)\} \tag{9}$$

between s and u_1, the outputs of the channels are $(0,1,0)$ if the source message is 1 and an error occurs at channel (s,a).

Next, consider the case that the source message is 0 and an error occurs at channel (s,b). Then we must have

$$\phi_{(d,e)}(0,0) = 1, \tag{10}$$

otherwise the outputs of the channels across the cut in (9) would again be $(0,1,0)$ so that the sink node u_1 cannot distinguish the source messages 0 and 1.

We now consider the cut

$$cut(\{s,a,d,u_1\}, \mathcal{V} \setminus \{s,a,d,u_1\}) = \{(s,b),(s,c),(d,e)\} \tag{11}$$

between s and u_2. It is easy to verify that if $\phi_{(d,e)}(1,1) = 0$, then the outputs of the channels across the cut in (11) is $(0,1,0)$ if the source message is 0 and an error occurs at channel (s,c), or if the source message is 1 and an error occurs at channel (s,b). Thus we must have

$$\phi_{(d,e)}(1,1) = 1. \tag{12}$$

Now again consider the cut in (9). With (8), (10), and (12), it can readily be verified that

$$d_H((\tilde{\phi}_{(s,a)}(0), \tilde{\phi}_{(s,b)}(0), \tilde{\phi}_{(d,e)}(0)), (\tilde{\phi}_{(s,a)}(1), \tilde{\phi}_{(s,b)}(1), \tilde{\phi}_{(d,e)}(1))) \tag{13}$$

$$= d_H((0,0,1),(1,1,1)) \tag{14}$$

$$= 2. \tag{15}$$

By Theorem 3, $\{(\tilde{\phi}_{(s,a)}(z), \tilde{\phi}_{(s,b)}(z), \tilde{\phi}_{(d,e)}(z)) : z \in \{0,1\}\}$ is a classical 1-error-correcting code so that its minimum distance is at least 3, a contradiction to (15). Therefore, the assumption that the code in (7) is 1-error-correcting is incorrect, and we conclude that there exists no binary 1-error-correcting network code that can transmit 1 bit. This in turn shows that the upper bound in Theorem 3 is not tight.

In the above example, the volume of a maxflow to either sink is equal to 3. There are a 3-flow $\{\mathcal{P}_{1,1}, \mathcal{P}_{1,2}, \mathcal{P}_{1,3}\}$ with $\mathcal{P}_{1,1} := \{(s,a),(a,u_1)\}$, $\mathcal{P}_{1,2} := \{(s,b),(b,u_1)\}$, and $\mathcal{P}_{1,3} := \{(s,c),(c,d),(d,e),(e,f),(f,g),(g,u_1)\}$ from the source node s to the sink node u_1, and a 3-flow $\{\mathcal{P}_{2,1}, \mathcal{P}_{2,2}, \mathcal{P}_{2,3}\}$ with $\mathcal{P}_{2,1} := \{(s,a),(a,d),(d,e),(e,u_2)\}$, $\mathcal{P}_{2,2} := \{(s,b),(b,f),(f,g),(g,u_2)\}$, and $\mathcal{P}_{2,3} := \{(s,c),(c,u_2)\}$ from the source node s to the sink node u_2. All the paths in the first flow intersect with the paths in the second flow. In particular, $\mathcal{P}_{1,3}$ intersects with all the paths in the flow to u_2: it intersects $\mathcal{P}_{2,3}$ at channel (s,c), intersects $\mathcal{P}_{2,1}$ at channel (d,e), and intersects $\mathcal{P}_{2,2}$ at channel (f,g). As such, an error that occurs at a channel can disturb the delivery of the message to the sinks u_1 and u_2 in some entangled ways. For example, a single error occurring at different channels on the path $\mathcal{P}_{1,3}$ would have different effects.

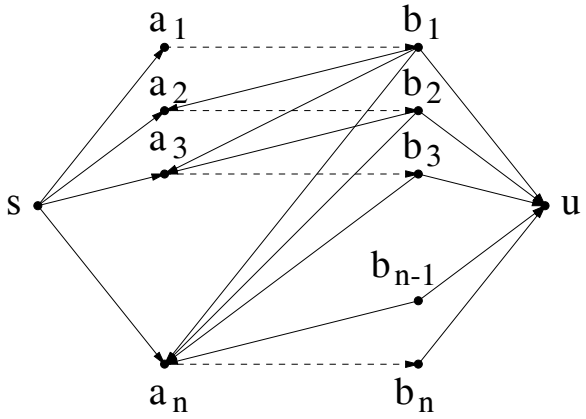

FIG. 3. *A network simulating a feedback channel.*

Specifically, an error at channel (s, c) would change the symbols carried by all the paths to the sink u_2, an error at channel (c, d) or (d, e) would change the symbols carried by the two paths $\mathcal{P}_{2,1}$ and $\mathcal{P}_{2,2}$, an error at channel (e, f) or (f, g) would change the symbols carried by $\mathcal{P}_{2,2}$, while an error at channel (g, u_1) would have no influence on the symbol carried by any path to the sink u_2. The multiple effects of an error pattern makes it generally much harder to construct network error-correcting codes than classical error-correcting codes. The above example simply shows that all these possible effects of a single error cannot be circumvented simultaneously when the code alphabet is small. Nevertheless, we will see in Part II that this can be achieved when the code alphabet is sufficiently large.

EXAMPLE 4. *Consider the network* $(\mathcal{G}, s, \mathcal{U}, \mathcal{R})$ *in Fig. 3 which is specified by*

$$\mathcal{U} = \{u\},\ \mathcal{V} = \{a_i, b_j : i, j = 1, 2, \ldots, n\} \cup \{s, u\}$$

and

$$\begin{aligned}\mathcal{E} = &\{(s, a_i), (a_i, b_i), (b_i, u) : i = 1, 2, \ldots, n\} \cup \\ &\{(b_i, a_j) : i = 1, 2, \ldots, n-1 \text{ and } j = i+1, i+2, \ldots, n\}.\end{aligned}$$

For $i = 1, 2, \ldots, n$, *the channels* (a_i, b_i) *are assigned capacity one while all other edges are assigned capacity infinity (or a sufficiently large integer). Let the code alphabet be binary. In Fig. 3, a channel with capacity one and a channel with capacity infinity are represented by a dotted arrow and a solid arrow, respectively.*

Consider any t*-error-correcting code for this network. Since the channels* (s, a_i) *for* $i = 1, 2, \ldots, n$ *have infinite capacity, we assume without loss of generality that the source node* s *employs a classical* t*-error-correcting code (with sufficiently large block length) to send the message* $z \in \mathcal{Z}$ *to each node* a_i *through channel* (s, a_i)*, so that node* a_i *can decode the message* z*. Similarly, we assume that for* $i = 1, 2, \ldots, n$,

node b_i employs a classical t-error-correcting code to send the output of channel (a_i, b_i), denoted by w_i, to nodes $a_{i+1}, a_{i+2}, \ldots a_n$ and the sink node u through channels $(b_i, a_{i+1}), (b_i, a_{i+2}), \ldots, (b_i, a_n)$ and (b_i, u), respectively, so that nodes $a_{i+1}, a_{i+2}, \ldots, a_n$ and the sink node u can decode w_i. Therefore, when at most t errors occur in the network, all the channels (s, a_i), (a_i, b_i), and (b_i, a_j) (for $i = 1, 2, \ldots, n$ and $j = i+1, i+2, \ldots, n$), which have infinite capacity, can be regarded as noiseless.

Thus we see that this network actually simulates the operation of an error-correcting code with feedback and block length equal to n (see Fig. 8.12, p. 176 in [8]). For $i = 1, 2, \ldots, n$, let us denote the local encoding mapping for channel (a_i, b_i) by $\phi_{(a_i,b_i)}(z, w_1, w_2, \ldots, w_{i-1})$, where $z \in \mathcal{Z}$ and $w_j \in \{0, 1\}$ for $j = 1, 2, \ldots, n-1$. Then in order for the network code to be t-error-correcting,

$$\{\phi_{(a_1,b_1)}, \phi_{(a_2,b_2)}, \ldots, \phi_{(a_n,b_n)}\} \tag{16}$$

must form the collection of encoding functions of a classical t-error-correcting code with feedback.

Owing to the existence of classical binary t-error-correcting codes with feedback of length n with a message alphabet strictly larger than $A(n, t, 2)$ (e.g., [9] or [13]), we see that

$$\{(\tilde{\phi}_{(a_1,b_1)}(z), \tilde{\phi}_{(a_2,b_2)}(z), \ldots, \tilde{\phi}_{(a_n,b_n)}(z)) : z \in \mathcal{Z}\}$$

cannot be a classical binary t-error-correcting code (without feedback). By noting that $(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)$ are the set of channels across the cut

$$cut(\{s\} \cup \{a_i : i = 1, 2, \ldots, n\}, \mathcal{V} \setminus (\{s\} \cup \{a_i : i = 1, 2, \ldots, n\})),$$

which is irregular, we have obtained an example showing that $\{(\tilde{\phi}_{(a,b)}(z), (a, b) \in cut(A, B)) : z \in \mathcal{Z}\}$ is not a classical t-error-correcting code if $cut(A, B)$ is irregular, necessitating the assumption of irregularity on the cut in Theorem 3.

As a summary, in Examples 1 and 2, we see the relation between network error-correcting codes and classical error-correcting codes, in particular that the former contains the later as a special case. In Examples 3 and 4, we see that network error correction is considerably more complicated than classical error correction.

6. Conclusion. In Part I of this paper, we have introduced network error correction as a generalization of classical point-to-point error correction. We have obtained the network generalization of the Hamming bound for classical error-correcting codes. We also have obtained further upper bounds on the size of the source alphabet of a network error-correcting code in terms of upper bounds for classical error-correcting codes. Along another line, we have obtained the network generalization of the Singleton bound for classical error-correcting codes. The tightness of this bound will be proved in Part II. By studying in details a few elementary examples, the relation

between network error correction and classical error correction is investigated, and it is seen that the former is considerably more complicated than the latter.

Acknowledgment

The work of Raymond W. Yeung was partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (RGC Ref. No. CUHK4214/03E).

REFERENCES

[1] B. Bollobas, *Graph Theory, An Introductory Course*, Springer-Verlag, 1979.

[2] R. W. Yeung and Z. Zhang,*Distributed source coding for satellite communications*, IEEE Trans. Inform. Theory, IT-45(1999), pp. 1111–1120.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, *Network information flow*, IEEE Trans. Inform. Theory, IT-46(2000), pp. 1204–1216.

[4] S.-Y. R. Li, R. W. Yeung and N. Cai, *Linear network coding*, IEEE Trans. Inform. Theory, IT-49(2003), pp. 371–381.

[5] R. Koetter and M. Médard, *An algebraic approach to network coding*, IEEE/ACM Transactions on network coding, 11(2003), pp. 782–795.

[6] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, *Polynomial time algorithms for multicast network code construction*, IEEE Transactions on Information Theory, IT-51, pp. 1973–1982.

[7] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, *Network Coding Theory*, now Publishers, 2006.

[8] R. W. Yeung, *A First Course in Information Theory*, Kluwer Academic/Plenum Publishers, 2002.

[9] E. R. Berlekamp, *Block coding for the binary symmetric channel with noiseless, delayless feedback*, in ıMann, H.B. H. B. Mann, Error Correcting Codes, Wiley, New York, 1968.

[10] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, Massachusetts, 1983.

[11] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[12] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[13] K. S. Zigangirov, *Numbers of correctable errors for transmission over a binary symmetrical channel with feedback*, Problems Inform. Transmission, V. 12, pp85-97, translated from Probemi Peredachi Informassi V. 12, pp3-9 (in Russian), 1976.