Construction Algorithm for Network Error-Correcting Codes Attaining the Singleton Bound

Ryutaroh MATSUMOTO\(^{(a)}\), Member

SUMMARY We give a centralized deterministic algorithm for constructing linear network error-correcting codes that attain the Singleton bound of network error-correcting codes. The proposed algorithm is based on the algorithm by Jaggi et al. We give estimates on the time complexity and the required symbol size of the proposed algorithm. We also estimate the probability of a random choice of local encoding vectors by all intermediate nodes giving a network error-correcting codes attaining the Singleton bound. We also clarify the relationship between the robust network coding and the network error-correcting codes with known locations of errors.

key words: error correction, MDS code, network coding, random network coding, Singleton bound

1. Introduction

Ahlswede et al. [1] proposed the notion of network coding that multicasts data from a single sender to multiple receivers at a rate at which the ordinary store and forward routing cannot multicast the data. Such high rate multicast becomes feasible by allowing intermediate nodes to encode and decode the data. A sender is usually called a source and a receiver is called a sink. A network coding is said to be linear if every intermediate node outputs a linear combination of its inputs [10].

A study of network coding usually assumes that an error does not occur in networks. Recently, Cai and Yeung [2], [13] considered errors in network coding, and proposed the network error correcting codes that allow sinks to recover the information even when errors occur on intermediate edges in the network. After formulating the network error correction, they proposed the lower and upper bounds on the number of messages in a network α-error correcting code, and one of their upper bound was a natural generalization of the Singleton bound for the ordinary error-correcting codes. Recently, Zhang [14] and Yang et al. [11] independently observed that the Singleton bound can be refined. We note that the problem formulation in [2], [13] was later independently presented in [4]. (The proceedings paper of [2], [13] appeared in 2002.)

Cai and Yeung mostly considered the case that intermediate nodes perform only simple encoding and decoding without delay, such as computing the output of the node as a linear combination of its inputs, and the sinks perform complex decoding computation. The network error correcting codes can avoid introducing decoding computation and delay into intermediate nodes, which is the advantage over use of ordinary error correcting codes between nodes.

Note that a similar type of network failure in a slightly different context was considered in [7, Sect. V] and [6, Sect. VI] in which every sink is assumed to know the set of failed edges and failed edges are assumed to emit zero symbols. Network error correction does not assume the knowledge of edges causing errors, and the problem formulation is different from [6], [7]. Note also that Kurihara [8] considered the different notion of robustness. In his paper, he considered network coding that allows sinks to recover partial information with edge failures.

For the construction of the network error-correcting codes, Jaggi et al. [5] proposed a randomized construction that uses coding among different time intervals. Their method produces codes attains the Singleton bound with high probability with sufficiently long block length, where the block length refers to the number of time intervals among which coding is done. It is desirable to have a network error-correcting code that does not code among different time intervals and thus does not introduce delay. Concurrently to this paper, Yang et al. [11] proposed an explicit construction algorithm that produces codes attaining the refined Singleton bound. The idea in [11] is similar to this paper in the sense that they also regard errors as information from the source and add extra components in the global encoding vectors corresponding to errors.

In this paper, we give a deterministic and centralized algorithm that constructs a network error-correcting code that attains the Singleton bound of network error-correcting codes obtained in [13]. We also give a relationship between the success probability and the field size for successful construction of network error-correcting codes when intermediate nodes choose their encoding coefficients randomly and independently. The proposed algorithms are based on [6]. Our network error-correcting codes make multicast robust to errors.
without introducing delay in the transmission, which is very attractive to delay sensitive multicast applications, such as multicast of video or audio. Our method is also useful for cryptographic applications, because it can tolerate modification and deletion of data by an adversary.

This paper is organized as follows. Section 2 introduces notations and the model of errors. Section 3 proposes an algorithm for constructing network error-correcting codes attaining the Singleton bound. Section 4 shows how to modify the algorithm in Sect. 3 to attain the refined Singleton bound, the success probability of the random construction of network error-correcting codes, and the relationship between the robust network coding [6], [7] and the network error-correcting codes with known locations of errors [12]. Section 5 gives concluding remarks.

2. Preliminary

2.1 Basic notations

We consider an acyclic directed graph \( G = (V, E) \) with possible parallel edges of unit capacity. \( V \ni s \) denotes the source and \( V \ni T \) denotes the set of sinks. Let \( n \) be the smallest min-cut separating \( s \) from any \( t \in T \) throughout this paper. For \( v \in V, \Gamma^+(v) \) (resp. \( \Gamma^-(v) \)) denotes the set of edges leaving (resp. reaching) the node \( v \), and \( \text{start}(e) \) (resp. \( \text{end}(e) \)) denotes the node at which the edge \( e \) starts (resp. ends).

We consider linear coding over a finite field \( \mathbb{F}_q \) with \( q \) elements. The source \( s \) gets \( k \) (\( \leq n \)) input symbols from \( \mathbb{F}_q \). The symbol \( y(e) \in \mathbb{F}_q \) carried by an edge \( e \) is a linear combination of the symbols carried by the edges entering \( \text{start}(e) \). The local encoding vector \( m_e : \Gamma^-(\text{start}(e)) \rightarrow \mathbb{F}_q \) determines the coefficients of this linear combination, that is,

\[
y(e) = \sum_{e' \in \Gamma^-(\text{start}(e))} m_{e'}(e') y(e').
\]

In this paper, a nonsink node performs only the computation of linear combination of its inputs, and they do not correct errors. An error is assumed to occur always at an edge. When an error occurs at an edge \( e \), the symbol received by \( \text{end}(e) \) is different from one sent by \( \text{start}(e) \), and \( \text{end}(e) \) computes its outputs as if there was no error at \( e \). The error value at an edge \( e \) is defined by the received symbol minus the transmitted symbol at \( e \). Note that we express a failure of a node \( v \in V \) in a real network as errors on edges in \( \Gamma^+(v) \) in our model. The number of errors is the number of edges at which errors occur. A network code is said to correct \( \alpha \) errors if every sink can recover the original information sent by the source when \( \alpha \) or less errors occur at arbitrary edges. We call the recovery of information by a sink decoding.

We represent errors occurred in the whole network by a vector \( \vec{c} \) in \( \mathbb{F}_q^{|E|} \), where \(|E|\) denotes the number of elements in \( E \). Fix some total ordering in \( E \), and enumeration of the error values gives \( \vec{c} \).

Regarding on the number of messages in a network \( \alpha \)-error correcting code, Cai and Yeung obtained the following result.

**Proposition 1:** [13] The number \( M \) of messages in a network \( \alpha \)-error correcting code, not necessarily linear, is upper bounded by

\[ M \leq q^{n-2\alpha}. \]

Very recently, Zhang [14] and Yang et al. [11] observed that the above proposition can be refined as follows.

**Proposition 2:** [11], [14] Let \( n_t \) be the min-cut from the source \( s \) to a sink \( t \). If the sink \( t \) can correct any \( \alpha_t \) errors then the number \( M \) of messages in the network correcting code, not necessarily linear, is upper bounded by

\[ M \leq q^{n_t-2\alpha_t}. \]

2.2 Jaggi et al.’s algorithm for construction of an ordinary network code

In this subsection, we review Jaggi et al.’s algorithm [6] for construction of an ordinary network coding. The proposed algorithm uses a modified version of their algorithm.

Since linear coding is used, the information carried by an edge \( e \) is a linear combination of \( k \) information symbols in \( \mathbb{F}_q \). We can characterize the effect of all the local encoding vectors on an edge \( e \) independently of a concrete \( k \) information symbols using global encoding vectors \( \vec{b}(e) \in \mathbb{F}_q^k \). When the information from the source \( s \in \mathbb{F}_q^k \), the transmitted symbol on an edge \( e \) is equal to the inner product of \( \vec{t} \) and \( \vec{b}(e) \). In order to decide the encoding at the source node \( s \), we have to introduce an imaginary source \( s' \) and \( k \) edges of unit capacity from \( s' \) to \( s \). We regard that \( s' \) sends \( k \) symbols to \( s \) over \( k \) edges.

We initially computes an \( s' \)-flow \( f^t \) of magnitude \( k \) for each \( t \in T \) and decomposes this flow into \( k \) edge disjoint paths from \( s' \) to \( t \). If an edge \( e \) is on some flow path \( W \) from \( s' \) to \( t \), let \( f^t_{-}(e) \) denote the predecessor edge of the edge \( e \) on the path \( W \). Jaggi et al.’s algorithm steps through the nodes \( v \in V \) in a topological order induced by the directed graph \( G \). This ensures that the global encoding vectors of all edges reaching \( v \) are known when the local encoding vectors of the edges leaving \( v \) are determined. The algorithm defines the coefficients of \( m_e \) for one edge \( e \in \Gamma^+(v) \) after the other. There might be multiple flow paths to different sinks through an edge \( e \). Let \( T(e) \) denote the set of sinks using \( e \) in some flow \( f^t \) and let
Proof. Denote the Hamming weight of a vector \( \bar{x} \) by \( w(\bar{x}) \). The assumption of the lemma implies that for any \( \bar{t} \neq \bar{j} \) and \( \bar{e} \) with \( w(\bar{e}) \leq 2\alpha \) we have \[
\phi_t(\bar{t}, \bar{e}) \neq \phi_t(\bar{j}, \bar{0}). \tag{2}
\]
Equation (2) implies that for any \( \bar{t} \neq \bar{j} \) and \( \bar{e}_1, \bar{e}_2 \) with \( w(\bar{e}_1) \leq \alpha \) and \( w(\bar{e}_2) \leq \alpha \) we have \[
\phi_t(\bar{t}, \bar{e}_1) \neq \phi_t(\bar{j}, \bar{e}_2),
\]
which guarantees that \( t \) can decode the original information under the assumption that the number of errors is \( \leq \alpha \) by exhaustive search. \( \square \)

Remark 5: The above lemma does not guarantee the existence of an efficient decoding algorithm.

Fix \( F \subset E \) with \( |F| = 2\alpha \). We shall show how to construct a network error-correcting code that allows every sink to decode the original information when the errors can occur only at \( F \). We call \( F \) the error pattern. The following description is a condensed version of the proposed algorithm, which is equivalent to the full description with \( F = \{F\} \) in Fig. 2 on p. 5.

1. Add the imaginary source \( s' \) and draw \( k \) edges from \( s' \) to \( s \).
2. Add an imaginary node \( v \) at the midpoint of each \( e \in F \) and add an edge of unit capacity from \( s' \) to each \( v \).
3. For each sink \( t \), do the following:
   a. Draw as many edge disjoint paths from \( s' \) to \( t \) passing through the imaginary edges added at Step 2 as possible. Let \( m_t^F(\leq 2\alpha) \) be the number of paths.
   b. Draw \( k \) edge disjoint paths passing through \( s \) that are also edge disjoint from the \( m_t^F \) paths drawn in the previous step.
4. Execute the algorithm by Jaggi et al. with \( \sum_{t \in T} (k + m_t^F) \) edge disjoint paths constructed in Step 3.

Example 6: In Fig. 1, we give an example of addition...
of imaginary nodes and edges. The network structure in 
Fig. 1 is taken from [3, Fig. 2]. Nodes A and B are 
the imaginary nodes added in Step 2 and the dashed 
lines from s' to A and B represent the imaginary edges 
added in Step 2.

The min-cut from s to every sink is 4 in the original 
network. The set F of edges with errors consists of the 
edge from s to node 1 and the edge from node 1 to node 5.

We denote a path by enumerating nodes on the 
path. In Step 3a for t1 we can find two edge dis-
joint paths, namely (s’, A, 1, t1) and (s’, B, 5, 8, t1). On 
the other hand, in Step 3a for t2, we can find only 
one edge disjoint path, namely (s’, A, 1, 5, t2) or 
(s’, B, 5, 8, t2). Therefore n = 2 while m = 1.

In Step 3b for t1, we find two edge disjoint paths 
as (s’, 3, 6, 9, t1) and (s’, 4, 7, 10, t1). In Step 3b for 
t2, we find three edge disjoint paths as (s’, s, 2, 6, 9, t2), 
(s’, s, 3, 7, 10, t2), and (s’, s, 4, t2). We can use arbitrary 
two paths among the three paths. In either case, we can 
find n – m = 2 paths in Step 3b.

In Step 3b, we can guarantee the existence of k 
paths as follows: Suppose that edges in m = 2 paths 
used in Step 3a are removed from the original network 
(V, E). Then the min-cut from s to a sink t in the 
original network (V, E) is at least n – m = 2, which is larger 
than or equal to k.

In Step 4 we use the algorithm by Jaggi et al. as 
if the imaginary source s’ sent information on the α 
imaginary edges added in Step 2. We denote by B^F
the set B^F of global encoding vectors for k + m = 2 
edge disjoint paths. B^F consists of k + m = 2 vectors of length 
k + 2α. We require that every sink t is able to decode k 
information symbols, while t may be unable to decode 2α error 
symbols in general because m = 2

There are always two edges end at the added imagi-
inary node v and one edge starts from v in Step 2. Since 
v is imaginary, we cannot choose local encoding vectors 
at v. Therefore, in Step 4, all components in the local 
coding vector at v must be selected to 1, which keeps 
B_t linearly independent. The reason is as follows: Let 
e be the edge from s’ to v added in Step 2. The global 
encoding vector of e is of the form

(0j−1, 1, 0n−j),

that is, it has only 1 at the j-th component. All other 
global encoding vectors in B^F have zero at the j-th component, since they are not in downstream of e when we choose local encoding vectors at v. Therefore, the added imaginary node v does not interfere with the execution of Jaggi et al.’s algorithm.

Observe also that q > |T| guarantees the successful execution of the algorithm as with the original version of Jaggi et al.’s algorithm.

We shall show how each sink t can decode the origi-
na information sent from the source s. After executing 
Step 4 we have decided all the local encoding vectors in 
the original network (V, E). Consider the three linear 
spaces defined by

V_1 = \{φ_t(⃗i, ⃗e) | ⃗i ∈ F^q_k, ⃗e ∈ F^|E|_q^F\},
V_2 = \{φ_t(⃗i, ⃗0) | ⃗i ∈ F^q_k\},
V_3 = \{φ_t(⃗0, ⃗e) | ⃗e ∈ F^q|E|_k\},

where components in ⃗e corresponding to E \ F are zero, 
and φ_t is as defined in Definition 3. We consider V_1, 
V_2, and V_3 in the original network (V, E) without added 
imaginary nodes and edges. Then we have

V_1 = V_2 + V_3, \text{dim} V_2 ≤ k.

Since we keep B^F linearly independent,

\text{dim} V_1 ≥ k + m.

Since the maximum number of edge disjoint paths pass-
ning through the imaginary edges added in Step 2 is m = 2,

\text{dim} V_3 ≤ m.

Equations (3–5) imply

\text{dim} V_2 = k + m,
\text{dim} V_2 = k,
\text{dim} V_3 = m,
\text{dim} V_2 \cap V_3 = 0.

The number of nonzero components in φ_t(⃗i, ⃗e) is k + m
and the number of unknowns in φ_t(⃗i, ⃗e) is k + 2α, which 
can be larger than k + m. However, by Eq. (7), the sink 
t can compute φ_t(⃗i, ⃗0) from φ_t(⃗i, ⃗e) as follows: Write 
φ_t(⃗i, ⃗e) as ⃗u + ⃗v such that ⃗u ∈ V_2 and ⃗v ∈ V_3. By Eq. (7) 
⃗u and ⃗v are uniquely determined [9, p.19, Theorem 4.1].
We have ⃗u = φ_t(⃗i, ⃗0) and the effect of errors is removed.
The sink t can also compute the original information ⃗i 
from φ_t(⃗i, ⃗0) by Eq. (6).

We shall describe how to construct a network error-
correcting code that can correct errors in any edge set 
F \subset E with |F| = 2α. Let \mathcal{F} = \{F \subset E : |F| = 2α\}. 
The idea in this paragraph is almost the same as the 
construction of the robust network coding in [6, Sect. VI]. Recall that B^F is the set of global encoding vectors 
on edge disjoint paths to a sink t with an edge set F 
of errors. Execute Jaggi et al.’s algorithm keeping B^F
linearly independent for all t ∈ T and all F ∈ \mathcal{F}. Then 
every sink t can decode the original information with 
the knowledge of the edge set F on which errors actually 
 occur. As in [6, Sect. VI],

q > |T| \cdot |\mathcal{F}| = |T| \left(\frac{|E|}{2α}\right)

guarantees the successful execution of the algorithm.

We present a pseudo programming code of the
foreach error pattern $F \in \mathcal{F}$ do
  Initialize global encoding vector $\vec{u}^F(e_i) = ((0, -1, 1)^{\beta+2\alpha_i}) \in \mathbb{F}_q^{\beta+2\alpha}$.
endforeach

foreach edge $e \in E$ do
  Add an imaginary node $v$ at the midpoint of $e \in E$. $O(1)$
  Divide $e$ into an edge to $v$ and an edge from $v$. $O(1)$
  Draw an imaginary edge from $s'$ to $v$. (*) $O(1)$
endforeach

draw sink $t \in T$ do
  Draw as many edge disjoint paths from $s'$ to $t$ as possible
  passing through the edge added in (*).
  $O(2\alpha(|E| + k + 4\alpha))$
  Draw $k$ edge disjoint path from $s'$ to $t$ passing through $s$ and also disjoint from paths made in the previous step.
  $O((k(|E| + k + 4\alpha))$
  Initialize the basis $B^t_F = \{\vec{u}^F(e_i) | e_i \text{ is on a path to } t\}$. $O((k + 2\alpha)^2)$
endforeach

endforeach

(* Main loop *)
foreach edge $e \in \bigcup_{F \in \mathcal{F}} E_F \setminus \{e_1, \ldots, e_k\}$ in a topological order do
  if $\text{start}(e) \in V$ then
    Choose a linear combination $\vec{u}^F(e) = \sum_{p \in P^F(e)} m_e(p)\vec{u}(p)$
    such that $B^t_F$ remains linearly independent for all $t$ and $F$ by the method in [6, Sect. III.B]. (**)
  else
    $m_e(p) = 1$ for all $p \in P^F(e)$ and $\vec{u}^F(e) = \sum_{p \in P^F(e)} \vec{u}(p)$. $O((k + 2\alpha)^2)$
  endif
endforeach
return $\{m_e(\cdot) | \text{start}(e) \in V\}$.

Fig. 2 Construction algorithm for a network $\alpha$-error correcting code. The rightmost $O(\cdot)$ indicates the time complexity executing the step.

4. Variants of the proposed method and its relation to the robust network coding

We shall introduce two variants of the proposed method in this section.

4.1 Attaining the refined Singleton bound

Network error-correcting codes constructed by the proposed method attains the Singleton bound (Proposition 1), while they do not necessarily attain the refined Singleton bound (Proposition 2). Yang et al. [11] concurrently proposed a construction algorithm that produces a code attaining the refined Singleton bound. In this subsection we modify the proposed method so that it can produce a code attaining the refined Singleton bound.

Let $n_t$ be the min-cut from $s$ to $t$, and suppose that the source $s$ emits $k$ symbols within unit time interval. A sink $t$ can correct $\alpha$ errors if $2\alpha \leq n_t - k$. Let $T_i = \{F \subset E : |F| = n_t - k\}$ and $\mathcal{F} = \bigcup_{t \in T} T_i$. For fixed $F \in \mathcal{F}$ and $t \in T$, we cannot guarantee that there exists $k$ edge disjoint paths in Step 3b. For such $F$, the sink $t$ cannot decode information with errors occurred at $F$. We exclude $B^t_F$ with such $(t, F)$ from the algorithm. Note that if $|F| \leq n_t - k$ then there always exist $k$ edge disjoint paths in Step 3b.

In order to attain the refined Singleton bound we keep the linear independence of all bases in $\{B^t_F | t \in T, F \in \mathcal{F}, |F| \leq n_t - k\}$ in Step (***) in Fig. 2. By the exactly same argument, we see that the produced code attains the refined Singleton bound.

By almost the same argument as Sect. 3, we see that the modified proposed algorithm runs in time complexity $O(|\mathcal{F}|T(k + 2\alpha_{\max}) + |E| + k + 4\alpha_{\max} + |\mathcal{F}|T|T|)$, where $\alpha_{\max} = \lceil (\max_{t \in T} n_t - k)/2 \rceil$. The required field size for successful execution of the algorithm is $|T| \cdot |\mathcal{F}|$, and in this case $|\mathcal{F}|$ depends on the structure of the network $(V, E)$. 
On the other hand, the time complexity of constructing local encoding vectors by the method of Yang et al. [11] is

\[
O\left( |E|^k \sum_{t \in T} \sum_{j=0}^{n_t-k} \binom{|E|}{j} (q-1)^j \right),
\]

and the required field size is

\[
\sum_{t \in T} \left( n_t + |E| - 2 \right).
\]

The time complexity of the proposed algorithm can be smaller or larger depending on the network structure and \(q\) than Yang et al. [11]. The required field size of the proposed algorithm can also be smaller or larger depending on the network structure. However, for the special case \(n_t = n\) for all \(t \in T\), the required field size of the proposed method is smaller than Yang et al. [11].

### 4.2 Completely randomized construction

By using the idea in the previous section, we can estimate the success probability of constructing a network error-correcting code by randomly choosing local encoding vectors as follows. The idea behind its proof is almost the same as [6, Theorem 12]. Observe that the random choice of local encoding vectors completely remove the time complexity of selecting encoding vectors in the centralized manner at the expense of larger required field size \(q\).

**Proposition 7:** Suppose that the source \(s\) transmits \(k\) symbols within unit time interval, and let \(\mathcal{F} = \{ F \subset E : |F| = 2\alpha \}\) be the set of edges on which errors can occur. Suppose also that local encoding vector coefficients are generated at random independently and uniformly over \(\mathbb{F}_q\). With this network error-correcting code, all sinks can correct errors in any edge set \(F \in \mathcal{F}\) with probability at least \(1 - \delta\) if \(q \geq |E||T||\mathcal{F}|/\delta\).

**Proof:** First pick independent random local encoding vectors for all edges in the network simultaneously. Then pick an error pattern \(F \in \mathcal{F}\). For this \(F\), execute Steps 1 and 2 in page 3 and compute the global encoding vectors \(\vec{b}(e)\)'s belonging to \(\mathbb{F}_q^{k+1}|\mathcal{F}|\). For each cut in the network, test whether \(B^F_t\)'s are linearly independent for all \(t\). This test fails with probability at most \(|T|/q\) by the proof of [6, Theorem 9] provided that this tests succeed on all the upstream cuts and \(n \geq k + 2\alpha\).

In the proposed algorithm in Fig. 2, we test linear independence of \(B^F_t\)'s on \(|E|\) cuts in Step (**), which is sufficient to guarantee the decodability of the information by every sink. By the same reason, for each sink to be able to correct errors in \(F\), one needs to consider linear independence only on at most \(|E|\) such cuts with random choice of local encoding vectors. By the union bound, the probability that the the independence tests fails for any of \(|T|\) sinks in any of the \(|E|\) cuts in any of the \(|\mathcal{F}|\) error patterns is at most \(\delta\) if \(q \geq |E||T||\mathcal{F}|/\delta\). □

Jaggi et al. [5] do not provide an estimate on the relation between the success probability of their algorithm and the field size \(q\). Their method [5] uses coding among different time intervals and thus introduces delays while our methods do not introduce extra delay. In addition to this, \(\alpha\)-error correcting codes by constructed by the proposed methods allow sinks to correct less than \(\alpha\) errors, while the method in [5] does not. The advantage of the method in [5] over the proposed methods in this paper is that their method allows efficient decoding of information by every sink, while our proposed methods require exhaustive search of transmitted information.

We summarize the comparison among the proposed algorithms and [5], [11] in Table 1.

| Method       | Delay          | Required field size for the success probability of code construction | Time complexity of code construction | Time complexity of decoding by sinks |
|--------------|----------------|---------------------------------------------------------------------|-------------------------------------|-------------------------------------|
| Figure 2     | none           | \(|T|/\binom{|E|}{2\alpha}\)                                        | \(O(|\binom{|E|}{2\alpha}|T|(|k+2\alpha||E|+k+4\alpha+\binom{|E|}{2\alpha}|T|))\) | \(O(q^k \sum_{j=0}^{n} \binom{|E|}{j}(q-1)^j(k+\alpha)^2)\) |
| Sect. 4.2    | none           | \(|E||T|/\binom{|E|}{2\alpha}\)/\delta                            | \(O(I)\)                           | \(O(q^k \sum_{j=0}^{n} \binom{|E|}{j}(q-1)^j(k+\alpha)^2)\) |
| Paper [11]   | none           | \(|T|(|k+|E|-2|)/2\alpha\)                                        | \(O(|E||T|q^k \sum_{j=0}^{n} \binom{|E|}{j}(q-1)^j)\) | \(O(q^k \sum_{j=0}^{n} \binom{|E|}{j}(q-1)^j(k+\alpha)^2)\) |
| Paper [5]    | large          | not estimated                                                      | \(O(I)\)                           | \(O((n \times \text{delay})^2)\) |

### 4.3 Relation to the robust network coding

We clarify the difference between the robust network coding in [7, Sect. V],[6, Sect. VI] and the network error-correcting codes with known locations of errors [12]. A network error correcting codes that can correct errors on a known locations \(F \subset E\) is a robust network coding tolerating edge failures on \(F\). However, the converse is not always true. Consider the network consists of three nodes \(\{s,t,v\}\) with two directed edges from \(s\) to \(v\) and one directed edge from \(v\) to \(t\). The source is
s and the sink is t. The intermediate node v sends to t the sum of two inputs from s. This network coding tolerate single edge failure between s and v but cannot correct single error between s and v.

5. Concluding remarks

In this paper, we proposed an algorithm constructing network error-correcting codes attaining the Singleton bound, and clarified its relation to the robust network coding [6, Sect. VI].

There are several research problems that have not been addressed in this paper. Firstly, the proposed deterministic algorithm requires tests of linear independence against \( \binom{|E|}{2\alpha} \) sets consisting of \( k + m_i \) vectors, which is really time consuming. It is desirable to have a more efficient deterministic construction algorithm.

Secondly, since there seems no structure in the constructed code, the decoding of the original information at a sink t requires the exhaustive search by t for possible information from the source and possible errors. It is desirable to have a code with structure that allows efficient decoding.

Finally, the case \(|T| = 1\) and \(|E| = n\) includes the ordinary error correcting codes as a special case. Substituting \(|T| = 1, |E| = n\) and \(2\alpha = n - k\) into Eq. (1) gives \( q > \binom{n-\alpha}{n-k} \), which can be regarded as a sufficient condition for the existence of the MDS linear code. On the other hand, a well-known sufficient condition for the existence of the MDS linear code is \( q > n - 2 \), which suggests that Eq. (1) is loose and that there is a room for improvement in Eq. (1).

Acknowledgment

The author thanks for constructive criticisms by reviewers that improved the presentation of the results very much. He also thanks Dr. Masazumi Kurihara for pointing out ambiguity in the earlier manuscript. He would like to thank Prof. Kaoru Kurosawa for drawing his attention to the network error correction, Prof. Olav Geil, Prof. Toshiya Itoh, Prof. Tomohiko Uyematsu, Mr. Akisato Kinura, and Dr. Shigeaki Kuzuoka for helpful discussions. He also would like to thank Dr. Sidharth Jaggi and Mr. Allen Min Tan for informing the papers [5], [11]. Part of this research was conducted during the author’s stay in the Department of Mathematical Sciences, Aalborg University.

References

[1] R. Ahlswede, N. Cai, S.Y.R. Li, and R.W. Yeung, “Network information flow,” IEEE Trans. Inf. Theory, vol.46, no.1, pp.1204–1206, July 2000.
[2] 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.
[3] K. Harada and H. Yamamoto, “Strongly secure network coding schemes with a ramp threshold,” Proc. SITA 2005, vol.2, pp.741–744, Okinawa, Japan, Nov. 20–23, 2005.
[4] S. Jaggi, M. Langberg, T. Ho, and M. Médard, “Correction of adversarial errors in networks,” Proc. ISIT 2005, pp.1455–1459, Adelaide, Australia, Sept. 4–9, 2005.
[5] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, “Resilient network coding in the presence of Byzantine adversaries.” to appear in Proc. INFOCOM 2007, Anchorage, Alaska, USA, May 6–12, 2007.
[6] S. Jaggi, P. Sanders, P.A. Chou, M. Effros, S. Egner, K. Jain, and L.M.G.M. Tolhuizen, “Polynomial time algorithms for multicast network code construction,” IEEE Trans. Inf. Theory, vol.61, no.6, pp.1204–1206, July 2006.
[7] R. Koetter and M. Médard, “An algebraic approach to network coding,” IEEE/ACM Trans. on Networking, vol.11, no.5, pp.782–795, Oct. 2003.
[8] M. Kurihara, “On some robust and secure transformations for linear network coding,” IEICE Technical Report, vol.106, no.185, pp.37–42, July 2006.
[9] S. Lang, Linear Algebra, 3rd ed., Undergraduate Texts in Mathematics, Springer-Verlag, New York, USA, 1987.
[10] S.Y.R. Li, R.W. Yeung, and N. Cai, “Linear network coding,” IEEE Trans. Inf. Theory, vol.49, no.2, pp.371–381, Feb. 2003.
[11] S. Yang, C.K. Ngai, and R.W. Yeung, “Construction of linear network codes that achieve a refined Singleton bound,” to appear in Proc. ISIT 2007, Nice, France, June 24–29, 2007.
[12] S. Yang and R.W. Yeung, “Characterizations of network error correction/detection and erasure correction,” Proc. NetCod 2007, UCSD, San Diego, California, USA, Jan. 29, 2007, available from http://code.ucsd.edu/netcod07/.
[13] 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.
[14] Z. Zhang, “Network error correction coding in packetized networks,” Proc. ITW’06, Chengdu, China, pp.433–437, Oct. 22–26, 2006.