

- [6] B. Ya. Ryabko, "Twice-universal coding," *Probl. Inform. Transm.*, vol. 20, no. 3, pp. 24–28, July–Sept. 1984.
- [7] —, "Prediction of random sequences and universal coding," *Probl. Inform. Transm.*, vol. 24, no. 2, pp. 3–14, Apr.–June 1988.
- [8] Yu. M. Shtarkov, "Switching discrete sources and its universal encoding," *Probl. Inform. Transm.*, vol. 28, no. 3, pp. 95–111, July–Sept. 1992.
- [9] M. J. Weinberger, N. Merhav, and M. Feder, "Optimal sequential probability assignment for individual sequences," *IEEE Trans. Inform. Theory*, vol. 40, no. 2, pp. 384–396, Mar. 1994.
- [10] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Trans. Inform. Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

## Fault-Tolerant Cube Graphs and Coding Theory

Jehoshua Bruck, *Senior Member, IEEE*,  
and Ching-Tien Ho, *Member, IEEE*

**Abstract**—Hypercubes, meshes, tori, and Omega networks are well-known interconnection networks for parallel computers. The structure of those graphs can be described in a more general framework called cube graphs. The idea is to assume that every node in a graph with  $q^\ell$  nodes is represented by a unique string of  $\ell$  symbols over  $\text{GF}(q)$ . The edges are specified by a set of offsets, those are vectors of length  $\ell$  over  $\text{GF}(q)$ , where the two endpoints of an edge are an offset apart. We study techniques for tolerating edge faults in cube graphs that are based on adding redundant edges. The redundant graph has the property that the structure of the original graph can be maintained in the presence of edge faults. Our main contribution is a technique for adding the redundant edges that utilizes constructions of error-correcting codes and generalizes existing *ad hoc* techniques.

**Index Terms**—Fault tolerance, parallel computing, interconnection networks, hypercubes, omega networks, error-correcting codes.

### I. INTRODUCTION

The mesh, torus, hypercube, and Omega networks are some of the most important interconnection networks for parallel computers. One of the most important issues in the design of a system which contains many components is the system's performance in the presence of faults. Hence, it is of major practical importance to develop efficient techniques (in terms of the cost of the redundancy) to handle faults in those architectures.

Our approach for handling faults is based on a graph model. In this model the architecture is viewed as a graph, where the nodes represent the processors and the edges represent communication links between the nodes. A target graph is selected and the required amount of fault tolerance  $f$  is determined. Then a fault-tolerant graph, that has the same number of nodes as the target graph, is defined with the property that given any set of  $f$  or fewer faulty edges, the remaining graph (after removal of the faulty edges) is guaranteed to contain the

Manuscript received September 3, 1995; revised February 16, 1996. This work was supported in part by the NSF Young Investigator Award CCR-9457811, by the Sloan Research Fellowship and under a Grant from the IBM Almaden Research Center, San Jose, CA.

J. Bruck is with the California Institute of Technology, MS 136-93, Pasadena, CA 91125 USA.

C.-T. Ho is with the IBM Almaden Research Center, San Jose, CA 95120 USA.

Publisher Item Identifier S 0018-9448(96)04824-9.

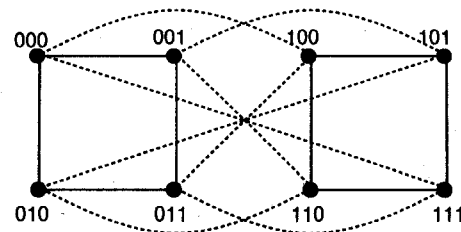


Fig. 1. A three-dimensional folded hypercube.

target graph as a subgraph. Note that this approach guarantees that any algorithm designed for the target graph will run *with no slowdown* in the presence of  $f$  or fewer edge faults in the fault-tolerant graph, regardless of their distribution. Minimizing the cost in this model amounts to constructing a fault-tolerant graph with minimum degree. (See [4] for constructions that address node faults.)

The key to our fault-tolerant constructions is a technique based on error-correcting codes for adding redundant edges using the so-called *wildcard dimensions*. Our technique is capable of handling an arbitrary number of faults over arbitrary alphabets. This is a generalization of the construction in [5] that is limited to the case of MDS (Maximum Distance Separable) codes. The key to our generalization is proving the equivalence between the constructions of fault-tolerant cube graphs and constructions of generator matrices of error-correcting codes (while in [5] the construction is based on parity check matrices of error-correcting codes.) First, we will describe the technique for constructing fault-tolerant hypercubes, then we will explain how it can be extended to Omega networks, tori, meshes, and cube-connected cycles (CCC).

Let  $Q_\ell$  denote the  $\ell$ -dimensional hypercube. It consists of  $2^\ell$  nodes, where node  $i$ ,  $0 \leq i \leq 2^\ell - 1$ , is represented by the  $\ell$ -bit binary representation of  $i$ . Two nodes, say  $X = (x_{\ell-1}x_{\ell-2}\cdots x_0)$  and  $Y = (y_{\ell-1}y_{\ell-2}\cdots y_0)$ , are connected by an edge iff there is a single  $j$  for which  $x_j \neq y_j$ , and this edge is called a *dimension- $j$*  edge. Hence, the set of edges in the hypercube can be partitioned into  $\ell$  dimensions.

In [9], it was suggested to add another set of edges to the hypercube, called the *wildcard dimension*, resulting in a *folded* hypercube. The folded hypercube topology was also defined independently in [7], but with a different name (the bisectional interconnection network) and with a different addressing scheme for the nodes. A formal definition of the folded hypercube is given next.

**Definition 1:** An  $\ell$ -dimensional *folded* hypercube, denoted by  $F_\ell$ , is an  $\ell$ -dimensional hypercube to which extra links are added connecting every pair of nodes that are bit-wise complements of each other.

A *wildcard edge* in  $F_\ell$  is an edge that connects node  $X = (x_{\ell-1}x_{\ell-2}\cdots x_0)$  and node  $\bar{X} = (\bar{x}_{\ell-1}\bar{x}_{\ell-2}\cdots \bar{x}_0)$ . For example,  $F_3$  is depicted in Fig. 1; notice that the neighbors of node (000) are (001), (010), (100), and (111). The structure of  $Q_\ell$  and  $F_\ell$  can be described using a more general framework, see also [8].

**Definition 2:** Let  $\ell$  be a positive integer and let  $q$  be a prime number. Let  $S \subseteq \{0, 1, \dots, q-1\}^\ell$ . The graph  $N(q, \ell, S)$  is a graph with  $q^\ell$  nodes and with the edges specified by the set of vectors in  $S$ . Any two nodes, say  $X$  and  $Y$ , are connected by an edge iff there exists a vector  $V \in S$  such that  $X + V = Y$  where addition is vector addition performed over  $\text{GF}(q)$ .

Using this framework it is clear that  $Q_\ell = N(2, \ell, S_1)$ , where

$$S_1 = \{000 \cdots 001, 000 \cdots 010, \dots, 100 \cdots 000\}$$

namely, the set of  $\ell$  vectors with Hamming weight one. (The Hamming weight of a vector is the number of nonzero entries in a vector.) Also, the folded hypercube  $F_\ell = N(2, \ell, S_2)$  where  $S_2 = S_1 \cup \{1 \cdots 11\}$ , namely, the set  $S_1$  augmented by the all-1 vector.

The following theorem provides the basis for the fault-tolerant constructions (see [5] for a proof of a more general version of the theorem):

*Theorem 1:* Let  $\ell$  be a positive integer and let  $S \subseteq \{0, 1\}^\ell$  be a set of  $\ell$  vectors. The graph  $N(2, \ell, S)$  is isomorphic to the hypercube  $Q_\ell$  if and only if  $S$  is a set of  $\ell$  linearly independent vectors over  $\text{GF}(2)$ .

Namely, our approach for adding fault tolerance to hypercube is finding a set  $S$  of  $n$  vectors of length  $\ell$  over the finite field  $\text{GF}(2)$ , with the property that any  $n - f$  vectors in the set span the space  $\{0, 1\}^\ell$  (i.e., they have rank  $\ell$ ). Given such a set  $S$ , then, by Theorem 1, the graph  $N(2, \ell, S)$  is an  $f$ -edge-fault-tolerant hypercube. This follows because, in the worst case,  $f$  faulty edges can affect at most  $f$  distinct dimensions. We denote the set  $S$  by  $S(\ell, f)$  where  $n = |S(\ell, f)|$  is the number of vectors in the set. Clearly, we like to minimize  $n$ , given  $\ell$  and  $f$ .

Based on the folded hypercube construction given that  $f = 1$  then  $n = \ell + 1$  (this is minimal). In [11], selected  $S(\ell, f)$  were given for  $f \in \{2, 3\}$  and  $\ell \in \{4, 11, 26\}$ , using a computer search. The question of constructing nontrivial  $S(\ell, f)$  for  $f = 2$  and  $\ell > 26$ , and for any  $f > 3$  was left as an open problem.

In this correspondence, we show that the problem of constructing  $S(\ell, f)$  is equivalent to the problem of constructing error-correcting codes. This connection provides the following results:

- when  $f = 2$  and  $\ell \leq 2^r - r - 1$  for some positive integer  $r$ , then  $n = 2^r - 1$  based on Hamming codes;
- when  $f = 3$  and  $\ell \leq 2^r - r - 1$  for some positive integer  $r$ , then  $n = 2^r$  based on extended Hamming codes; and
- in general, given  $f$  and  $\ell$ , one can find  $n$  such that  $n - \ell$  is about  $\lfloor f/2 \rfloor \log n$ , based on BCH codes.

In the next section we will describe a method based on error-correcting codes for constructing  $S(\ell, f)$ . In Section III we will indicate how the technique can be generalized to include Omega networks, tori, meshes, and CCC's.

## II. CONSTRUCTING OFFSETS USING ERROR-CORRECTING CODES

Following the discussion in the preceding section, the method for constructing  $f$ -edge fault-tolerant hypercubes is based on the existence of a set of  $n$  vectors of length  $\ell$  over the finite field  $\text{GF}(2)$ , that has the property that any  $n - f$  vectors in the set span the space  $\{0, 1\}^\ell$ , namely, they have rank  $\ell$ . We will represent the set  $S(\ell, f)$  by the set of  $n$  columns of an  $\ell \times n$  matrix.

Clearly, one can construct  $S(\ell, f)$  by taking  $f + 1$  copies of the  $\ell \times \ell$  identity matrix. For example

$$S(4, 1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Namely,  $n = 8$  and any seven columns in the above matrix span  $\{0, 1\}^4$ . However, this approach results in a fault-tolerant hypercube with very high node degree where every edge is duplicated  $f + 1$  times.

A more efficient construction for the case  $f = 1$  was presented in [9] by adding the all-1 vector to the identity matrix. Namely

$$S(4, 1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \quad (1)$$

In this section we show that the construction of  $S(\ell, f)$  is equivalent to the construction of generator matrices for error-correcting codes and show how to construct  $S(\ell, f)$  for any  $\ell$  and  $f$ . We note here that the main idea in the construction of [5] is based on parity check matrices of error-correcting codes and is limited to MDS codes.

The main issue in the theory of error-correcting codes is to construct a large set of vectors (code) with the property that the Hamming distance between any two vectors in the code is larger than a predefined parameter  $d$  (the minimum distance). A minimum distance  $d$  allows the correction of up to  $\lfloor (d - 1)/2 \rfloor$  errors or the detection of up to  $d - 1$  errors. *Binary linear block codes* are codes that have the property that the set of codewords is a *vector subspace* over  $\text{GF}(2)$ . In particular, an  $(n, k, d)$  code is a binary code of length  $n$  (the length of vector in the code), dimension  $k$  (consists of  $2^k$  codewords), and minimum distance  $d$ . These codes can be described by a matrix known as the *generator matrix* which spans the code. In particular, an  $(n, k, d)$  code can be described by a  $k \times n$  generator matrix that is denoted by  $G(n, k, d)$ .

A vector  $V$  of length  $n$  is in the code if and only if there exists a vector  $X \in \{0, 1\}^k$  such that  $X \cdot G = V$ , where computations are performed over  $\text{GF}(2)$ . In other words, any linear combination of the  $k$  row vectors (of length  $n$ ) in  $G$  forms a vector in the code. Clearly, the all-0 vector is also in the code. For example, the following is the generator matrix of a  $(7, 4, 3)$  Hamming code:

$$G(7, 4, 3) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Notice, that any linear combination of the rows of  $G(7, 4, 3)$  has Hamming weight of at least 3. In general, we have the following lemma.

*Lemma 1:* The Hamming weight of every codeword in an  $(n, k, d)$  code is at least  $d$ .

*Proof:* This follows from the fact that the all-0 vector is in the code and the Hamming weight is equal to the Hamming distance to the all-0 vector.  $\square$

The following is our main theorem.

*Theorem 2*  $G(n, k, d)$  and  $S(\ell, f)$  are equivalent when  $\ell = k$  and  $f = d - 1$ . Namely

- 1) A generator matrix,  $G(n, k, d)$ , of an  $(n, k, d)$  binary linear error-correcting block code is also an  $S(k, d - 1)$  matrix.
- 2) A matrix  $S(\ell, f)$  with  $n$  columns can be used as a generator matrix for an  $(n, \ell, f + 1)$  binary linear error-correcting block code.

*Proof:*

*Proof of 1):* We need to prove that every  $n - d + 1$  columns of  $G(n, k, d)$  span  $\{0, 1\}^k$ . The proof is by contradiction. Assume that there exists a set of  $n - d + 1$  columns of  $G(n, k, d)$  which do not span  $\{0, 1\}^k$ . Let  $S_1$  be the matrix associated with those columns. Namely, the column rank of the matrix  $S_1$  is less than  $k$  and the row rank of  $S_1$  is also less than  $k$ . Hence, there is a linear combination of the rows of  $G(n, k, d)$  that is 0 in the coordinates that correspond to  $S_1$ . However, this linear combination has at least  $n - d + 1$  0's and its Hamming weight is smaller than  $d$ . This is a contradiction to the fact that every nonzero codeword has Hamming weight of  $d$  or more.

*Proof of 2):* We need to prove that the subspace that is spanned by a matrix  $S(\ell, f)$  has minimum Hamming weight of  $f + 1$ . Again, the proof is by contradiction. Assume that the subspace that is spanned by  $S(\ell, f)$  contains a vector, denoted by  $V$ , with Hamming weight of  $f$ . Now consider the matrix  $S_2$  that is obtained from  $S(\ell, f)$  by deleting the  $f$  columns that correspond to the 1's in  $V$ . The matrix  $S_2$  has  $n - f$  columns and its rank is smaller than  $\ell$ ; this is a contradiction.  $\square$

Theorem 2 shows that the problem of constructing  $S(\ell, f)$  matrices is equivalent to constructing error-correcting codes. Hence, the theory of error-correcting codes can be applied to construct  $S(\ell, f)$  matrices for arbitrary sets of parameters as well as to prove bounds on the size of those constructions. In the following we provide a number of examples that utilize these connections.

*Example 1 (Simple Parity Codes):* By Theorem 2, the case  $f = 1$  can be dealt with by a matrix  $G(k + 1, k, 2) = (I_k 1_k)$  which corresponds to the simple parity code, where  $I_k$  is the  $k \times k$  identity matrix and  $1_k$  is the all-1 vector of length  $k$ . This result corresponds to the so-called wildcard dimensions in hypercubes [5], [8], [9]. For instance, when  $k = 4$ ,  $G(5, 4, 2) = S(4, 1)$  which is given in (1).

*Example 2 (Hamming Codes and Extended Hamming Codes):* By Theorem 2, the case  $f = 2$  and  $f = 3$  can be dealt with by a  $G(n, k, 3)$  and  $G(n, k, 4)$ , respectively. This can be solved by utilizing the construction of Hamming codes and extended Hamming codes, respectively. Hamming codes ( $d = 3$ ) exist for  $n = 2^r - 1$  and  $k = 2^r - r - 1$ , for every  $r \geq 2$ . Extended Hamming codes ( $d = 4$ ) exist for  $n = 2^r$  and  $k = 2^r - r - 1$ , for every  $r \geq 2$ . This result generalizes the selected constructions for  $S(\ell, f)$  that were found in [11] using *ad hoc* computer search. For example, we consider the case  $r = 3$ .

$$S(4, 2) = G(7, 4, 3) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$S(4, 3) = G(8, 4, 4) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \quad (2)$$

We note here that Hamming codes can be constructed for arbitrary values of  $k$

$$2^{r-1} - (r - 1) - 1 < k < 2^r - r - 1$$

by deleting rows (and resulting all-zero columns) in  $G(2^r - 1, 2^r - r - 1, 3)$ .

*Example 3 (BCH codes and Beyond):* By Theorem 2, for general  $f$  we can use known constructions, like those of BCH codes [10]. For a table of the best known constructions please see [2].

*Example 4 (Upper and Lower Bounds):* We can apply Theorem 2 to derive bounds on  $n$  in  $S(\ell, f)$  using known results in coding theory [10].

- By the Hamming Bound:

$$2^{n-\ell} \geq 1 + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{\lfloor f/2 \rfloor}$$

From which it follows that the construction based on Hamming codes ( $f = 2$ ) is optimal.

- By the Singleton Bound:

$$n \geq f + \ell$$

- By the Gilbert-Varshamov bound,  $S(\ell, f)$  exists for every  $n$  such that

$$1 + \binom{n-1}{1} + \binom{n-1}{2} + \dots + \binom{n-1}{f-1} \leq 2^{n-\ell}$$

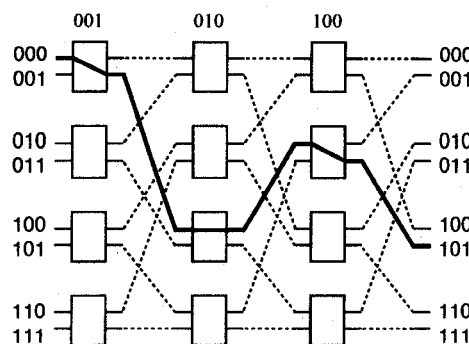


Fig. 2. An 8-input unshuffle network.

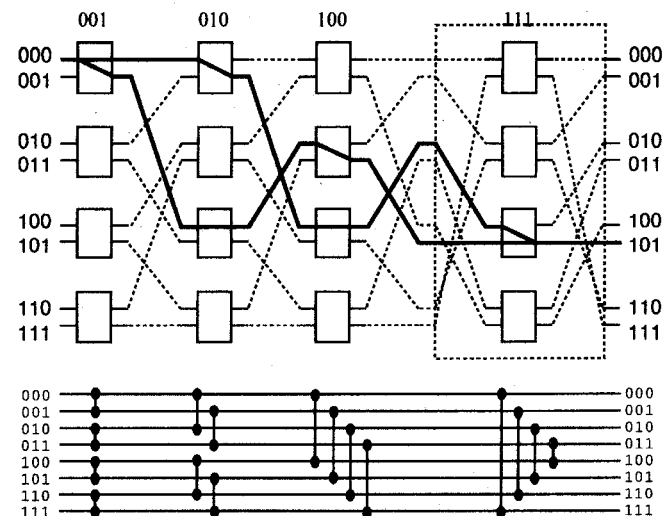


Fig. 3. An 8-input unshuffle network enhanced with a wildcard-dimension stage.

- By the Griesmer bound

$$n \geq \sum_{i=0}^{\ell-1} \left\lceil \frac{f+1}{2^i} \right\rceil$$

### III. EXTENSIONS AND CONCLUSIONS

We have described a technique for constructing  $f$ -edge fault-tolerant hypercube graphs. Those are graphs that have the property that they contain a hypercube as a subgraph in the presence of arbitrary  $f$  edge faults. The key to the technique is the construction of a set of redundant dimensions using the columns of a generator matrix of error-correcting codes. The technique can be generalized in a number of ways.

#### A. The Class of Omega Networks

Omega networks are networks that consist of stages of switches (as opposed to the hypercube that is a point-to-point network). It is a well-known fact that a stage in an Omega network corresponds to a dimension in a hypercube. Fig. 2 depicts an eight-input unshuffle network, which belongs to the class of Omega networks. Stages are labeled by binary vectors called *masks* and are shown above the stages in the figure. A switch in a stage with a mask  $M$  has as inputs the lines that correspond to  $X$  and  $X + M$  (over  $GF(2)$ ). There is a unique path between an input and output of the network. For example, the path between (000) and (101) is highlighted.

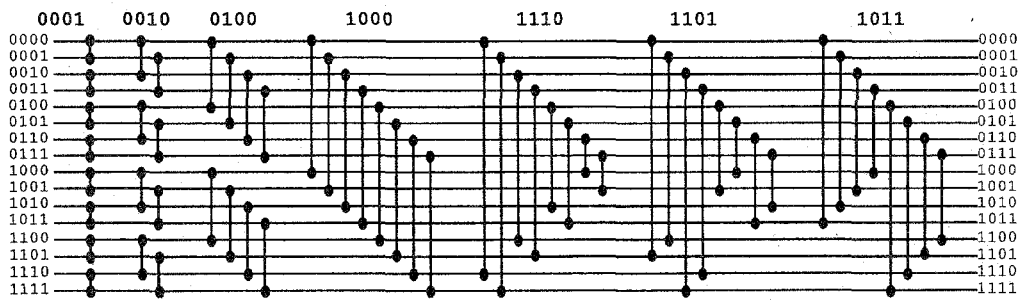


Fig. 4. A 16-input regular multistage network enhanced with three extra stages, based on Hamming codes  $G(7, 4, 3)$  for tolerating any two-stage faults.

Fig. 3 shows an eight-input unshuffle network enhanced with a wildcard-dimension stage. The additional stage, depicted within the big dotted box as the last stage, has a mask vector  $M = (11 \cdots 1)$ , which corresponds to the inputs of the form  $X$  and  $\bar{X}$ . The two paths from node (000) to node (101) are highlighted in the figure. For clarity, the functionality of the network can be represented by the figure below it, using the so-called Knuth's sorting network notation. Using this notation, a vertical line represents a switch. Our goal is to be able to tolerate  $f$  faults in vertical lines and still maintain the same structure. Clearly, the condition is similar to the case when the network is a hypercube, namely, the set of healthy masks (stages) should span the space of the inputs.

Fig. 4 shows a 16-input 4-stage network enhanced by three extra stages, based on Hamming code  $G(7, 4, 3)$ , (2), for tolerating any two-stage faults. The network has the property that if any two stages are faulty, the remaining network is functionally equivalent to a 16-input 4-stage network. In general, for an Omega network with  $2^\ell$  inputs and  $\ell$  stages, one can add  $n - \ell$  extra stages, where  $n = |S(\ell, f)|$ , such that given any  $f$  switch faults the remaining network is still functionally equivalent to the original Omega network. We note here that Omega networks with switches with more inputs can be handled using error-correcting codes over larger fields.

A couple remarks regarding the implementation of our method. Our technique can be applied to construct fault-tolerant Omega networks by adding extra stages of switches and a by-passing capability to each switch. The by-passing capability of a switch can be implemented by adding demultiplexers and multiplexers outside the switch chips [1]. Note that switches are currently designed for 8 or 16 inputs and consist of complex circuitry to support buffer management and flow control. While faults can occur at switches it is less likely they will occur at the by-passing circuits which are considerably simpler.

### B. The Meshes and Tori Networks

Let  $q$  be the width of the tori/meshes and assume  $q$  is prime throughout this subsection. We now generalize our construction of  $f$ -edge fault-tolerant hypercubes to  $f$ -edge fault-tolerant tori. First, define  $S_q(\ell, f)$  as a set of  $n$  vectors of length  $\ell$  over the finite field  $\text{GF}(q)$  with the property that any  $n - f$  vectors in  $S$  span the space  $\{0, 1, \dots, q - 1\}^\ell$ . Given such a set  $S$ , then the graph  $N(q, \ell, S)$  is an  $f$ -edge fault-tolerant tori of width  $q$  and dimension  $\ell$ . If we denote the generator matrix of a code with length  $n$ , dimension  $k$ , minimum distance  $d$ , and alphabet set  $\{0, 1, \dots, q - 1\}$  by  $G_q(n, k, d)$ , then we have the following generalization of Theorem 2:

**Theorem 3:**  $G_q(n, k, d)$  and  $S_q(\ell, f)$  are equivalent when  $\ell = k$  and  $f = d - 1$ . Namely

- 1) A generator matrix,  $G_q(n, k, d)$ , of an  $(n, k, d)$  linear error-correcting block code over  $\text{GF}(q)$  is also an  $S_q(k, d - 1)$  matrix.

- 2) A matrix  $S(\ell, f)$  with  $n$  columns can be used as a generator matrix for an  $(n, \ell, f + 1)$  linear error-correcting block code over  $\text{GF}(q)$ .

For example, if  $q = 3$  one can derive the generator matrix  $G_q(n, k, d) = G_3(13, 10, 3)$ , from a Hamming code over  $\text{GF}(3)$ , as

$$G_3(13, 10, 3)$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 2 \\ 1 & 0 & 1 \\ 1 & 0 & 2 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \end{pmatrix}$$

To derive  $S_q(\ell, f) = S_3(3, 2)$ , the set of vectors defining 2-edge fault-tolerant  $3 \times 3 \times 3$  tori, we use  $G_q(n, k, d) = G_3(6, 3, 3)$  by letting  $q = 3, k = \ell$  and  $d = f + 1$  and finding a minimum  $n$  available from the codes. In this case,  $G_3(6, 3, 3)$  can be derived by removing any seven rows (and resulting seven all-zero columns) from  $G_3(13, 10, 3)$  as, for instance,

$$G_3(6, 3, 3) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Note that our previous result in [5] only works for  $f$ -edge fault-tolerant tori of width  $q$  and dimension  $\ell$  where  $f \leq q + 1 - \ell$ . Thus for this case, where  $q = 3$  and  $\ell = 3$ , our previous result can only tolerate one fault.

Finally, using the technique of wildcard dimensions, any  $f$ -edge fault-tolerant torus is also a  $(2f + 1)$ -edge fault-tolerant mesh of the same width and dimension [5].

### C. The CCC Network

In [3] we have described a method for handling edge faults in cube-connected cycle (CCC) graphs. The approach described here for fault-tolerant hypercubes can be similarly applied for fault-tolerant CCC's. This yields a fault-tolerant CCC of smaller degree than those constructions in [3] for the case of multiple cube-edge faults. More specifically, let  $\text{CCC}(\ell, n), n \geq \ell$ , be the family of CCC's with  $2^\ell$  cycles and  $n$  nodes per cycle and in which the  $\ell$  nodes with corresponding cube edges are not necessarily contiguous. It is possible to create a fault-tolerant CCC of degree 4 and of the same size as  $\text{CCC}(\ell, n)$ , where  $n = |S(\ell, f)|$ , such that it tolerates any single cycle-edge fault and any  $f$  cube-edges fault when  $n$  is even. (When  $n$  is odd,  $n = |S(\ell, f)| + 1$ .) The construction can be easily derived from Construction I of a fault-tolerant CCC described in [3] and the construction of cube dimensions given by  $S(\ell, f)$  here.

## ACKNOWLEDGMENT

The authors wish to thank the Editor and the referees for their comments and suggestions that helped to improve the correspondence.

## REFERENCES

- [1] G. B. Adams and H. J. Siegel, "The extra stage cube: a fault-tolerant interconnection network for supersystems," *IEEE Trans. Comput.*, vol. C-31, no. 5, pp. 443-454, May 1982.
- [2] A. E. Brouwer and T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 662-677, Mar. 1993. See also the WWW at: <http://www.win.tue.nl/win/math/dw/voorlincod.html>.
- [3] J. Bruck, R. Cypher, and C.-T. Ho, "On the construction of fault-tolerant cube connected cycles networks," *J. Parallel Distrib. Comput.*, vol. 5, pp. 98-106, 1995.
- [4] —, "Fault-tolerant meshes with minimal numbers of spares," *IEEE Trans. Comput.*, vol. 42, no. 9, pp. 1089-1104, Sept. 1993.
- [5] —, "Wildcard dimensions, coding theory and fault-tolerant meshes and hypercubes," *IEEE Trans. Comput.*, vol. 44, no. 1, pp. 150-155, Jan. 1995.
- [6] S. K. Chen, " $n^+$ -cube: The extra dimensional  $n$ -cube," in *Proc. Int. Conf. on Parallel Processing*, vol. I (Pennsylvania State University, University Park, PA, 1990), pp. 583-584.
- [7] A. Ghafoor, T. R. Bashkow, and I. Ghafoor, "Bisectional fault-tolerant communication architecture for supercomputer systems," *IEEE Trans. Comput.*, vol. 38, no. 10, pp. 1425-1446, Oct. 1989.
- [8] C.-T. Ho, "An observation on the bisectional interconnection networks," *IEEE Trans. Comput.*, vol. 41, no. 7, pp. 873-877, July 1992.
- [9] S. Latifi and A. El-Amawy, "On folded hypercubes," in *Proc. Int. Conf. on Parallel Processing*, vol. I (Pennsylvania State University, University Park, PA, 1989), pp. 180-187.
- [10] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.
- [11] C. J. Shih and K. E. Batcher, "Adding multiple-fault tolerance to generalized cube networks," *IEEE Trans. Parallel Distributed Syst.*, vol. 5, no. 8, pp. 785-792, Aug. 1994.

## A Simple Class of Capacity-Achieving Strategies for Discrete Memoryless Channels with Feedback

Thijs Veugen, *Student Member, IEEE*

**Abstract**—A class of multiple-repetition strategies is presented for discrete memoryless channels with noiseless feedback. This extends results obtained by Schalkwijk to nonbinary, asymmetric memoryless channels. An interesting characterization of the attainability of a rate is presented in terms of a generating polynomial, from which follows that for each strategy in our class, there exists a discrete memoryless channel such that capacity can be achieved on this channel.

**Index Terms**—Feedback, constrained sequences, discrete memoryless channel, capacity, repetition strategy.

Manuscript received November 15, 1994; revised May 7, 1996. The material in this correspondence was presented at the 1994 IEEE International Symposium on Information Theory, Trondheim, Norway.

The author is with the Department of Electrical Engineering, Technological University, 5600 MB Eindhoven, The Netherlands.

Publisher Item Identifier S 0018-9448(96)06888-5.

## I. INTRODUCTION

In [1], Schalkwijk presented repetition strategies for the binary symmetric channel with noiseless feedback. These block-coding schemes achieve capacity for several values of the channel error probability and are easy to implement. Later, Schalkwijk and Post [2] showed that similar recursive coding schemes exist with fixed coding delay as well as variable coding delay. The computed bit error exponent for their variable coding delay case is comparable to Horstein's [3].

Several other related schemes have been proposed and analyzed. We mention a few of them. Some only use decision feedback, i.e., the decoder can ask for retransmission, while others use full feedback, i.e., each received symbol is sent back to the encoder via a noiseless feedback link. Forney [4] presented the (inverse) concatenation construction, a graphical construction to relate block error exponents and convolutional error exponents. Fang [5] showed that, for any nonpathological discrete memoryless channel with noiseless decision feedback, there exists a variable-length convolutional code such that the reliability function is not less than channel capacity for all transmission rates less than channel capacity. Recently, Telatar and Gallager [6] obtained, for block-coding schemes on discrete memoryless channels with noiseless decision feedback, a new lower bound of the largest possible error exponent while the erasure exponent is positive, which improves results of Forney [7]. Kudryashov [8] presented a scheme achieving the largest known error exponent for convolutional coding on discrete memoryless channels with decision feedback. Yamamoto and Itoh [9] presented block codes for discrete memoryless channels with noiseless feedback attaining high reliability functions. The largest known error exponent for discrete memoryless channels with noiseless feedback is obtained by Horstein [3] who used a variable coding delay system. In general, the complexity of the schemes increases when the error exponent increases.

In Section II, it is described how Schalkwijk's repetition strategies can be extended to multiple-repetition strategies which can be used on arbitrary discrete memoryless channels. It is explained how they can be used for block coding and fixed delay coding, respectively. Then in Section III, the rate of a multiple-repetition strategy is computed. Finally, we prove that multiple-repetition strategies can achieve capacity for several values of the channel error probabilities, as was conjectured by Becker [10, p. 93]. Moreover, equations are given from which these values could be computed. Our results are illustrated by some examples.

## II. MULTIPLE-REPETITION STRATEGIES

Consider a discrete memoryless channel and number the inputs from 0 to  $I - 1$ , and the corresponding outputs from 0 to  $J - 1$ . Without loss of generality,  $I \leq J$  because otherwise it is possible to eliminate a suitably chosen input symbol without affecting the capacity of the channel [11].

The idea behind repetition strategies is the following: suppose that during the transmission of a message an  $i \rightarrow j$  error occurs ( $0 \leq i < I$ ,  $0 \leq j < J$ ,  $i \neq j$ ). At that moment, the encoder stops transmitting the following message symbols and inserts the symbol  $i$   $k_{ij}$  times at this position in the message, where  $k_{ij}$  is a constant positive integer depending on  $i$  and  $j$ . After the insertions, the error is "corrected" and the transmission is resumed. The decoder scans the received sequence from right to left and looks for forbidden subsequences  $j^i k_{ij}$  ( $0 \leq i < I$ ,  $0 \leq j < J$ ,  $i \neq j$ ) and substitutes