ELSEVIER

Discrete Applied Mathematics 53 (1994) 3–13

DISCRETE APPLIED MATHEMATICS

On optimal broadcasting in faulty hypercubes

Jehoshua Bruck

IBM Research Division, Almaden Research Center, K54/802, 650 Harry Road, San Jose, CA 95120-6099, USA

Received 9 August 1991; revised 3 April 1992

Abstract

We construct a family of n spanning trees of the n-cube, called D_n, with the following properties: (i) All the trees in D_n are rooted at the same node (e.g. the all-0 node). (ii) Every tree in D_n is of depth n, which is the optimal depth. (iii) Every edge in the n-cube is shared by at most two trees in D_n.

One application of this construction is a technique to perform broadcasting in a hypercube with faulty links. Using D_n we obtain an efficient scheme—*it takes the same time as in a healthy cube*—to broadcast in the presence of up to $\lceil n/2 \rceil - 1$ edge faults.

1. Introduction

In this paper we present an interesting property of the n-cube, that is useful for performing broadcasting in the presence of faulty edges. In particular, we show how to perform broadcasting in an n-cube with $\lceil n/2 \rceil - 1$ edge faults in n communication steps. That is, without spending more time on communication with regard to a healthy hypercube. We make the following assumptions on the model:

(1) The goal is to broadcast a single packet from the source node to all other nodes. In each step a node can send packets to several of its neighbors. In particular, our broadcasting algorithm assumes a double-port model were a node can send packets to at most two of its neighbors at a time. This is a slightly more powerful model than the single-port model were a node can send a packet to a single neighbor at a time.

(2) Broadcasting is non-redundant in the sense that a minimal number of edges, namely $2^n - 1$, are used. This is useful for minimizing the congestion.

(3) Each node is allowed minimal local computation "for free". Our algorithms have at most nm bit operations per node, where m is the number of edge faults.

(4) Broadcasting is completed in optimal time, namely in n steps.

0166-218X/94/$07.00
SSDI 0166-218X(93)E0038-Z

(5) We address only the issue of edge (link) faults. The faults are static. In one of our broadcasting algorithms only the source node knows the location of the faults while in the other scheme the set of faults is known by all the nodes. We assume worst-case distribution of faults.

The key to our results is a construction of a family of n spanning trees of the n-cube, which we call D_n, with the following properties:

(1) All the trees are rooted at the same node (e.g. the all-0 node).

(2) Every tree is of depth n, which is the optimal depth.

(3) Every edge in the n-cube is shared by at most two trees in D_n.

Clearly, by the third property, when we have $\lceil n/2 \rceil - 1$ or less faulty edges, we can find a fault-free tree in D_n that will be used for broadcasting.

The hypercube (or n-cube) is one of the most popular interconnection topologies for parallel architectures [12]. Parallel machines like the iPSC/2 and iPSC/860 of Intel, NCUBE of NCUBE and CM-2 of Thinking Machines have a hypercube topology for the interconnection network. The hypercube has many interesting properties that on one hand allow to devise efficient parallel algorithms and on the other hand make it being amenable to efficient implementation. One of the important issues related to parallel architectures is fault tolerance. In particular, how do we compute in the presence of node/edge faults?

In this paper we will be interested in the operation of broadcasting information from a single node to all other nodes non-redundantly. Broadcasting is an important operation in algebraic computations [6] as well as in many other applications. For a complete list of references on the general problem of broadcasting on graphs the reader is referred to the survey paper by Hedetniemi et al. [5]. Various schemes were suggested to deal with the problem of routing and broadcasting in a faulty hypercube [1–4, 7, 8, 10, 11].

In [1–4] there are results on probabilistic analysis of routing in hypercubes with and without randomly distributed faults. In [11] the idea is to broadcast the packets with redundancy, and let the accepting node decide by majority voting. In [7] the authors define the concept of a safe/unsafe node and present a broadcasting algorithm that takes $n + 1$ steps and can deal with up to $\lceil n/2 \rceil$ node faults. In [8, 10] the assumption is that the broadcasting algorithm is fixed and oblivious and processors do not adapt their behavior based on knowledge of the fault distribution.

However, none of the results reported in the literature addresses the issue of broadcasting with edge faults under the above assumptions, in particular, all the known schemes handle the faults while spending some more time (than n) to completion.

The paper is organized as follows. In the next section we give some background on spanning trees of the n-cube. In particular, we describe the two constructions that are the basis for our construction: the binomial spanning tree due to Sullivan and Bashkow [13] and the edge-disjoint spanning trees (of depth $n + 1$) due to Johnsson and Ho [6]. In Section 3 we describe the construction of D_n. In Section 4 we present efficient broadcasting algorithms which are based on the construction D_n. Finally, in

Section 5, we address the question of the optimality of our approach and present open problems.

2. Preliminaries

In this section we will describe the two fundamental constructions of spanning trees of the hypercube which are the basis for our construction. The first construction is the well-known *binomial spanning tree* that was first suggested by Sullivan and Bashkow [13]. The second construction—which is related to the binomial tree—is the n-disjoint spanning trees (of depth $n+1$) due to Johnsson and Ho [6].

A word about the notation. Q_n denotes an n-cube with 2^n nodes and $n2^{n-1}$ edges. The nodes of Q_n are labeled with binary vectors of length n; throughout the paper we will refer to nodes as being vectors and also to vectors as being the corresponding nodes. Two nodes are connected by an edge iff they differ in one bit. An edge between $V=(v_1v_2\dots v_i\dots v_n)$ and $\hat{V}=(v_1v_2\dots \bar{v}_i\dots v_n)$ is labeled as $(v_1v_2\dots x\dots v_n)$. Generalizing this notation, a set of nodes that form a subcube is denoted by a vector in $\{0,1,x\}^n$. The x's are in the locations that correspond to the locations in which the nodes of the subcube differ.

2.1. The binomial tree

The binomial spanning tree seems to be the natural way to broadcast information packets in Q_n. Without loss of generality we will assume that the node that wants to broadcast to all other nodes is the all-0 node. The idea is to recursively partition every subcube of dimension k (starting with $k=n$) to k subcubes and broadcast in every subcube. In an n-cube the subcubes are

$$\{10\dots00, x100\dots00, xx100\dots00, \dots, xxxxxxx1\}.$$

See Fig. 1 for an example of a binomial spanning tree of Q_3.

The broadcasting is performed as follows: There is an index which is attached to each packet. The index is an integer in $1\dots n$. P will denote the packet and i will denote the index. Note that the first bit corresponds to the most significant bit.

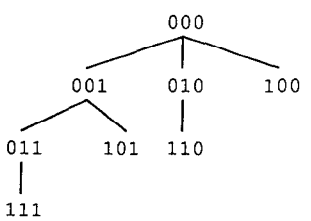

Fig. 1. A binomial spanning tree of a 3-cube.

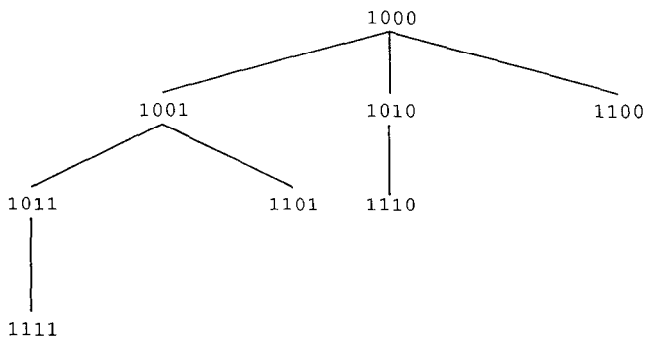

Fig. 2. The tree T_1 for the 4-cube.

Initialization: At the source node: For all $1 \leqslant j \leqslant n$ send $\{P, j\}$ to the node that differs in the jth bit.

At every node: Receive $\{P, i\}$. For all $1 \leqslant j < i$ send $\{P, j\}$ to the node that differs in the jth bit.

We note here that it takes n steps (which is optimal) to complete the broadcasting based on a binomial tree using a single port model.

2.2. *Edge-disjoint trees*

The edge-disjoint trees (of depth $(n+1)$) is a remarkable construction due to Johnsson and Ho [6]. The motivation for this construction was to improve the complexity of certain communication primitives for algebraic operations. In what follows, we present a new derivation of this construction and show its strong relation with the binomial tree. We call this set of trees HJ-trees. The idea in the HJ-trees is to have n spanning trees that are disjoint in an n-cube in which every edge consists of two directed links, hence the trees are actually link-disjoint. Here we will assume that the n-cube consists of undirected edges and prove that every edge is shared by at most two trees out of the n HJ-trees. Consider the n subcubes of dimension $n-1$:

$$\{1x \ldots xx, x1 \ldots xx, \ldots, xx \ldots x1\}.$$

Let T_1 be the binomial spanning tree of the subcube $1xx \ldots xx$ rooted at $100 \ldots 00$. The depth of T_1 is $n-1$. See Fig. 2 for an example of T_1 of Q_4. Let T_i be the binomial spanning tree obtained from T_1 by right-cyclically shifting its node labels $i-1$ times. Hence, T_i, $1 \leqslant i \leqslant n$, is a spanning tree of the above ith subcube. The following lemma is the key for the construction of the HJ-trees.

Lemma 2.1. *The n binomial trees, T_i, $1 \leqslant i \leqslant n$, are edge disjoint.*

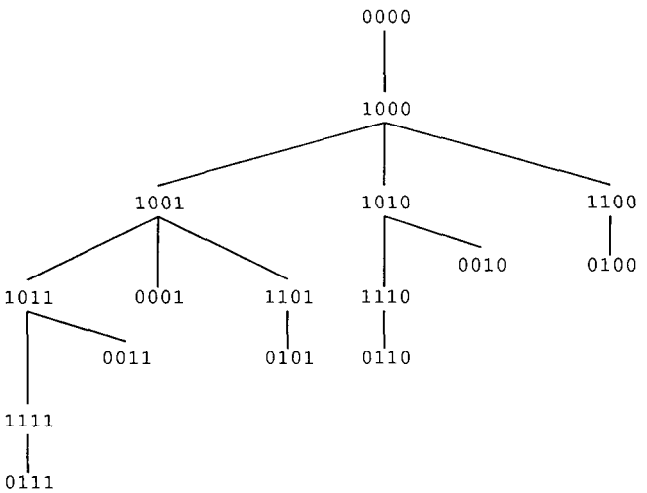

Fig. 3. The tree $\hat{T}_1$ for the 4-cube.

Proof. Let us assume, without loss of generality, that there is an edge $e = (V_1, V_2)$ that appears in both T_1 and T_2. By the definition of T_1 and T_2 both V_1 and V_2 have two 1's in the first two most significant bits. However, by the definition of a binomial tree, the parent node of $(11v_3 \dots v_n) \in T_1$ is $(10v_3 \dots v_n)$. Clearly, $(10v_3 \dots v_n) \notin T_2$. Hence, we have a contradiction—it cannot be that $V_1, V_2 \in T_1$ and $V_1, V_2 \in T_2$. □

Can we construct a spanning tree for Q_n from T_1? Clearly, we can do it by just adding 2^{n-1} leaves that correspond to those nodes not in T_1. Namely, we can connect every node $(1v_1v_2 \dots v_n) \in T_1$ with the node $(0v_2 \dots v_n)$. Let us denote the spanning tree of Q_n obtained from T_i using the above idea by $\hat{T}_i$. See Fig. 3 for an example of $\hat{T}_1$ of Q_4. The HJ-trees are basically the trees $\{\hat{T}_i | 1 \leqslant i \leqslant n\}$.

Lemma 2.2. *Every edge in Q_n appears in at most two trees out of $\{\hat{T}_i | 1 \leqslant i \leqslant n\}$.*

Proof. For a given i, the edges that are in $\hat{T}_i$ and not in the T_i are exactly all the edges in dimension i of Q_n. Since the T_i's are disjoint (Lemma 2.1), every edge in Q_n can be shared by at most two trees from $\{\hat{T}_i | 1 \leqslant i \leqslant n\}$. □

The construction of Johnsson and Ho is basically considering each of the $\hat{T}_i$ as a spanning tree of Q_n, rooted at the node $00 \dots 00$. Note that the depth of each of these spanning trees is $n + 1$. Namely, they are not optimal in terms of their depth (this construction is optimal for the application in presented in [6]). Is there a set of n spanning trees, each rooted at $00 \dots 00$ and of depth n, such that every edge in Q_n is shared by at most two of the trees? Such a construction provides a way to perform broadcasting, *with no penalty*, in the presence of at most $\lceil n/2 \rceil - 1$ edge faults. Our main result is a positive answer to this question—in the next section we describe a construction with the desired properties.

3. The main result

In this section we present our main result—a construction of n spanning trees of the n-cube with the following three properties:

(1) the trees are all rooted at the same node (e.g. the all-0 node),

(2) the depth of every tree is n (this is the optimal depth),

(3) every edge in Q_n is shared by at most two trees.

Using this construction we can perform broadcasting in a faulty hypercube (with at most $\lceil n/2 \rceil - 1$ faulty edges) and spend the same time on communication as in a healthy hypercube (the trees are of depth n).

The main idea in the construction is to start with the HJ-trees defined in Section 2 (they satisfy properties (1) and (3) and construct a new set of trees that satisfy also property (2) (namely, all the trees are of depth n).

Definition 3.1. Let e be an edge of the n-cube. The *rotation class* of e, to be denoted by R_e, is the set of n edges that are obtained by cyclically shifting the edge e.

Example. Let $e = 100x$. Then

$$R_e = \{100x, x100, 0x10, 00x1\}.$$

Consider the tree $\hat{T}_1$ defined in the previous section. The set of trees, $\{\hat{T}_i | 1 \leqslant i \leqslant n\}$, are rotations of $\hat{T}_1$. The interesting property of $\hat{T}_1$ is that for every edge there is at most one other equivalent edge in the sense that they are in the same rotation class. More precisely, let e_1 be an edge in $\hat{T}_1$. Then there is at most one other edge, say $e_2 \in \hat{T}_1$ such that $R_{e_1} = R_{e_2}$. This property is just another way to state Lemma 2.2. Hence, an edge will appear at most twice in $\{\hat{T}_i | 1 \leqslant i \leqslant n\}$.

Definition 3.2. Let G_1 be a subgraph of Q_n (here we are interested mostly in trees). Let $\{G_i | 1 \leqslant i \leqslant n\}$ be the n subgraphs obtained from G_1 by cyclically shifting the labels of the nodes of G_1. The *rotation index* of G_1 is the maximum number of times an edge of Q_n appears in $\{G_i | 1 \leqslant i \leqslant n\}$.

Hence, our problem can be solved if we are able to obtain the following construction: a spanning tree of Q_n rooted at the all-0 node, of depth n, whose rotation index is 2.

Notice that in $\hat{T}_1$ there is a single node, labeled $011 \ldots 11$, in distance $n + 1$ from the root. If we are able to take care of connecting this node to the root, via a path of length $\leqslant n$, while preserving the other properties of $\hat{T}_1$ we obtain the desired construction.

Notation. v^i ($v = 0$ or 1) is a string of i bits all being equal to v.

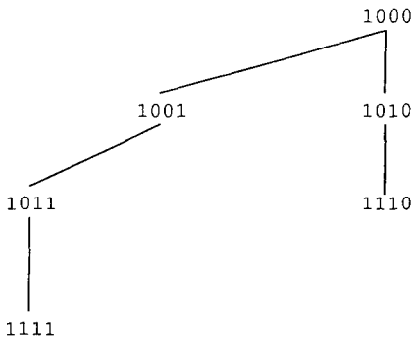

Fig. 4. The tree A_1 for the 4-cube.

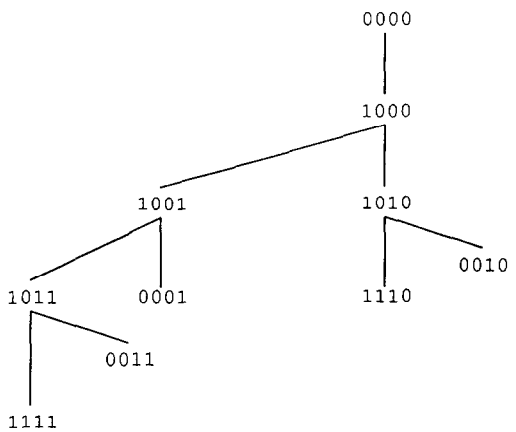

Fig. 5. The tree A_2 for the 4-cube.

Main Construction. Figs. 4–6 illustrate an example for the three steps of the construction associated with Q_4.

Step 1. Construct a binomial spanning tree of the subcube $1xx\ldots xx$ rooted at $100\ldots00$, omitting the set N_1 of $n-2$ nodes, where

$$N_1 = \{1100^i1^j | i + j = n - 3; i, j \geqslant 0\}.$$

For instance, if $n = 4$, $N_1 = \{1100, 1101\}$. Note that the nodes in N_1 are all leaves in the binomial spanning tree of $1xx\ldots xx$, that is why omitting them is possible. Call this tree A_1. See Fig. 4 for an example of A_1 of Q_4.

Step 2. Connect every node $V = (1v_2\ldots v_n) \in A_1$, such that $V \notin N_2$, with a node $\hat{V} = (0v_2\ldots v_n)$. The set N_2 consists of $n-2$ nodes,

$$N_2 = \{1111^i0^j | i + j = n - 3; i, j \geqslant 0\}.$$

For instance, for $n = 4, N_2 = \{1111, 1110\}$. Call this tree A_2. See Fig. 5 for an example of A_2 of Q_4. Note that the tree A_2 contains almost all the nodes in Q_n. There are

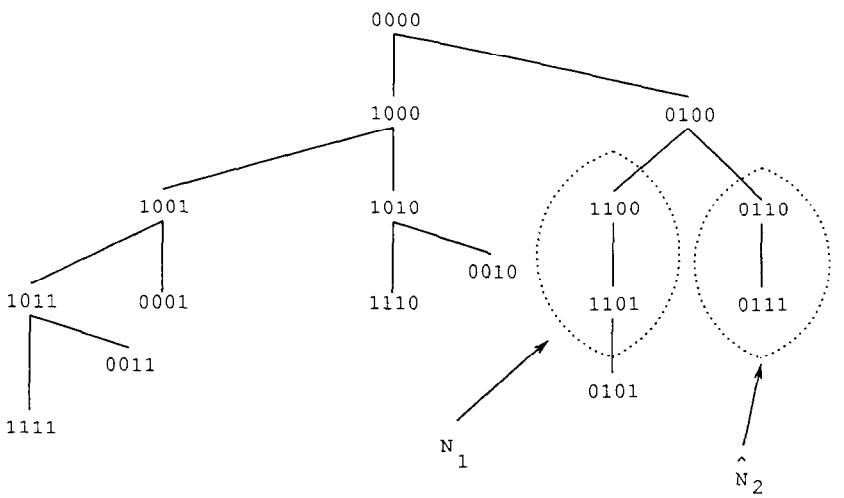

Fig. 6. The tree $\tilde{T}_1$ for the 4-cube.

$3n-6$ nodes in the Q_n which are not in A_2: the sets $N_1, \hat{N}_1$ and $\hat{N}_2$, where $\hat{N}_1$ (also $\hat{N}_2$) is the set of nodes that is obtained from N_1 (N_2) by complementing the first bit.

Step 3. In this step we obtain a spanning tree of Q_n out of the tree A_2 obtained in the previous step. Observe that both N_1 and $\hat{N}_2$ correspond to paths of length $n-2$ in Q_n. The path that correspond to N_1 is:

$$\{1100^{n-3}, 1100^{n-4}1, \ldots, 1101^{n-3}\}.$$

Also the path that correspond to $\hat{N}_2$ is:

$$\{0110^{n-3}, 01110^{n-4}, \ldots, 0111^{n-3}\}.$$

We connect the paths defined by N_1 and $\hat{N}_2$ to the root, i.e. 0^n, via the node $010^{n-2} \in \hat{N}_1$. We connect the rest of the set $\hat{N}_1$ to the corresponding nodes in N_1. We call the resulting tree $\tilde{T}_1$. See Fig. 6 for an example of $\tilde{T}_1$ of Q_4.

Lemma 3.3 (main). *The tree $\tilde{T}_1$ has the following properties:*
(1) *It is a spanning tree of the n-cube rooted at the all-0 node.*
(2) *Its depth is n.*
(3) *Its rotations index is 2. Namely, for any edge $e_1 \in \tilde{T}_1$ there exists at most one other edge $e_2 \in \tilde{T}_1$, $e_1 \neq e_2$, such that $R_{e_1} = R_{e_2}$.*

Proof. The first two properties follow directly from the construction. We prove here the third property by showing that $\tilde{T}_1$ and $\hat{T}_1$ have the same rotation classes. Since we know that property (3) is true in $\hat{T}_1$ the result will follow.

From the construction, the tree A_2 is a subgraph of the tree $\hat{T}_1$. Thus, the rotation index of A_2 is at most 2. Hence, we only need to show that the rotation classes that

correspond to the edges that were added to A_2 in step 3 of the main construction are such that property 3 is not violated.

Let E_1 be the set of edges that were added in step 3 of the main construction. Then E_1 consists of

(1) $0x0^{n-2}$ due to the connection of 010^{n-2} to 0^n,

(2) $\{1100^i x1^j | i+j=n-4;\ i,j \geqslant 0\}$ due to the path defined by the set N_1,

(3) $\{x100^i1^j | i+j=n-3;\ i,j \geqslant 0\}$ due to the connection of $\hat{N}_1$ with the set N_1,

(4) $\{011^i x0^j | i+j=n-3;\ i,j \geqslant 0\}$, due to the path defined by the set $\hat{N}_2$.

Let E_2 be the set of edges in $\hat{T}_1$ but not in A_2, it consists of

(1) $\{1x00^i1^j | i+j=n-3;\ i,j \geqslant 0\}$ due to the nodes in N_1,

(2) $\{x100^i1^j | i+j=n-3;\ i,j \geqslant 0\}$ due to the nodes in $\hat{N}_1$,

(3) $\{x111^i0^j | i+j=n-3;\ i,j \geqslant 0\}$ due to the nodes in $\hat{N}_2$.

When we compare the rotation classes that correspond to E_1 and E_2 we find that, for almost every edge $e_1 \in E_1$ there is an edge $e_2 \in E_2$ such that $R_{e_1} = R_{e_2}$. The only exception is the edge $0x0^{n-2}$ which does not have an equivalent edge in E_2 (in the sense of having the same rotation class). However, since in $\hat{T}_1$ the rotation class that corresponds to the edge $0x0^{n-2}$ appears only once the result follows. Note that the edge $x1^{n-1}$ appears only in E_2; clearly this can only reduce the rotation index of $\tilde{T}_1$. □

Given the construction of the tree $\tilde{T}_1$ we define D_n to be the family of n trees that are obtained from $\tilde{T}_1$ by cyclic shifts. Given the set D_n it is clear that we can construct an isomorphic family of trees which is rooted in an arbitrary node V by simply adding the label of V to the labels of the nodes in D_n (which is rooted at the all-0 node).

4. Broadcasting in a faulty n-cube

Given the construction of the trees D_n it is clear how to perform broadcasting in a faulty cube. Let us assume that a node $S = (s_1 s_2 \ldots s_n)$ wants to broadcast a packet to all other nodes. Assume that there are $m, m < \lceil n/2 \rceil$, faulty edges whose labels are known to S. Node S finds out which tree in D_n is healthy and broadcasts the packet along this tree. It is quite easy to select a healthy tree—since most of the edges in D_n are edges in the binomial tree (or its rotated version) and only the sets N_1 and $\hat{N}_2$ correspond to special edges. In fact, finding a healthy tree takes order of nm bits operations—linear in the size of the input (the details of the algorithm to do that are omitted).

Now given a healthy tree, say $\tilde{T}_1$, we make the following observations:

- Using a single-port model for communication it takes $n+1$ steps to complete the broadcasting. This is because the tree $\tilde{T}_1$ consists of two subtrees one of which is a subtree of $\hat{T}_1$ and the other consists of the sets $N_1, \hat{N}_1$ and $\hat{N}_2$. Using a single-port model it takes n steps to complete broadcasting in $\hat{T}_1$ and $n-1$ steps to complete broadcasting in the second subtree of $\tilde{T}_1$. Hence, it takes a total of $n+1$ steps for $\tilde{T}_1$.

• It is not hard to prove that if we assume a double-port model, namely, a node can communicate with up to two of its neighbors at the same time, then broadcasting can be completed in n steps.

We suggest two possible schemes for broadcasting.

(1) Given the set of faults, the source node, S, selects a healthy tree and transmits the packet to its children in the chosen tree along with its label (n bits). At any other node: given the set of faults and the label of the source node it is possible to compute the healthy tree (the trees in D_n can be numbered and the convention is that the chosen tree is the first tree that is healthy in D_n). Knowing the chosen tree, every node can transmit the packet (along with the label of the source node) to its children in this tree. Note that the overhead in the communication is n bits per packet. Also we have to perform order of nm bit-operations at every node to determine the children.

(2) In this scheme the selection of a healthy tree is performed by the source node. This scheme is useful when a host node wants to broadcast to all the other nodes and the information about the faulty links is stored in the host node, say as a result of running a diagnostic procedure. This algorithm can be used, for example, to distribute the knowledge about the location of the faults to all the nodes. The extra information which is transmitted with a packet consists of the label of the chosen tree ($\log n$ bits) and the actual address in the tree (n bits). At every node, the children can be determined in order of n bit-operations.

5. Conclusions and extensions

Our main result is a construction of a family of n spanning trees, rooted at the all-0 node, each of depth n, with the property that an edge in Q_n is shared by at most two trees.

One of the applications of this construction is to broadcasting in a hypercube with at most $\lceil n/2 \rceil - 1$ faulty edges. The interesting property of this approach is that the time spent on communication is the same as in a healthy hypercube—the depth of the spanning tree is n.

The main problem is whether we can do better than dealing with $\lceil n/2 \rceil - 1$ faulty edges? A trivial upper bound on the number of edge faults is,

Proposition 5.1. *Using depth n spanning trees we can deal with at most $n - 2$ edge faults.*

Proof. Assume there are $n - 1$ faults (e.g. in the first $n - 1$ dimensions) in the edges incident to the source (e.g. the all-0 node) then there is a node in distance $n + 1$ from the source (the node $1^{n-1}0$). □

Also, we can prove the following proposition.

Proposition 5.2. *Given any set of four spanning trees of Q_4 each of depth 4. We can find two edges such that every tree contains at least one edge.*

Hence, our construction is optimal in the sense that a general construction of n trees that will be able to deal with $\lceil n/2 \rceil$ faults (or more) does not exist. Actually we can prove (details are omitted) that a family of trees with the desired properties of size cn (where c is an arbitrary constant) does not exist. We note here that recently Peleg [9] devised a scheme for broadcasting in optimal time on faulty hypercubes. Peleg's scheme is not based on disjoint trees and can tolerate up to $n - 2$ faults (both nodes and edges). The added assumptions in the new scheme are an all-port model for communication and global knowledge of faults. We conclude with the following open problem.

Open problem. Is it possible to achieve fault-tolerant broadcasting in n steps, using a single-port model, in the presence of a certain number of edge/node faults?

Acknowledgement

I would like to thank C.T. Ho for the useful discussions, comments and suggestions and to the referees and the guest editor for their remarks that improved the presentation.

References

[1] D. Bienstock, Broadcasting with random faults, Discrete Appl. Math. 20 (1988) 1–7.
[2] M.S. Chen and K.G. Shin, Message routing in an injured hypercube, in: Proceedings 3rd Conference on Hypercube Computers and Applications, Pasadena (1988) 312–317.
[3] U, Feige, D. Peleg, P. Raghavan and E. Upfal, Randomized broadcast in networks, Random Structures Algorithms 4 (1990) 447–460.
[4] J.M. Gordon and Q.F. Stout, Hypercube message routing in the presence of faults, in: Proceedings 3rd Conference on Hypercube Computers and Applications, Pasadena (1988) 318–327.
[5] S.M. Hedetniemi, S.T. Hedetniemi and A.L. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (1985) 319–349.
[6] S.L. Johnsson and C.T. Ho, Optimum broadcasting and personalized communication in hypercubes, IEEE Trans. Comput. (1989) 1249–1268.
[7] T.C. Lee and J.P. Hayes, Routing and broadcasting in faulty hypercube computers, in: Proceedings 3rd Conference on Hypercube Computers and Applications, Pasadena (1988).
[8] A.L. Liestman, Fault-tolerant broadcast graphs, Networks 15 (1985) 159–171.
[9] D. Peleg, A note on optimal time broadcast in faulty hypercubes, manuscript.
[10] D. Peleg and A. Schäffer, Time bounds on fault-tolerant broadcasting, Networks 19 (1989) 803–822.
[11] P. Ramanathan and K.G. Shin, Reliable broadcast in hypercube multicomputers, IEEE Trans. Comput. 37 (1988) 1654–1657.
[12] C.L. Seitz, The cosmic cube, Comm. ACM 28 (1985) 22–33.
[13] H. Sullivan and T.R. Bashkow, A large scale homogeneous, fully distributed parallel machine. I, in: Proceedings 4th Annual Symposium on Computer Architecture (1977) 105–117.