Broadcasting on Faulty Hypercubes

Pao-Hwa Sui, Sheng-De Wang and Isaac Yi-Yuan Lee

Department of Electrical Engineering
National Taiwan University, Taiwan, R.O.C.

Abstract

In this paper we propose a method for constructing the maximum number of edge-disjoint spanning trees (in the directed sense) on a hypercube with arbitrary one faulty node. Each spanning tree is of optimal height. By taking the common neighbor of the roots of these edge-disjoint spanning trees as the new root and reversing the direction of the directed link from each root to the new root, a spanning graph, consisting of n - 1 edge-disjoint spanning trees, of optimal height is formed. Broadcasting based on the spanning graph has an optimal bandwidth utilization and an optimal latency.

Index Terms --- Broadcasting, edge-disjoint spanning trees, automorphism, dimension permutation, hypercubes.

I: Introduction

Distributed-memory multiprocessors are receiving increased attention to meet the demand for high-speed processing. Many interconnection structures have been proposed for these machines. The binary (Boolean) hypercube, due to its logarithmic diameter, simplicity, regularity, rich bandwidth and fault tolerance, has been one of the most popular topologies used in distributed-memory multiprocessors. One more important fact is that many interconnection networks such as trees and multidimensional meshes can be embedded in the hypercube [1]. A number of hypercube machines have been built such as Intel iPSC, NCUBE/10 and the Connection Machine [2]. It has possibility that processors may fail in a multiprocessor system. Because the probability of one or more processors that will fail in such system is quite large, algorithms designed for parallel systems with faulty processors are necessary. A faulty hypercube is caused by some nodes of a complete hypercube becoming faulty.

Broadcasting is one of the most important communication primitives used in distributed-memory multiprocessors. It is frequently used in many linear algebra algorithms, such as Gaussian elimination and matrix multiplication. Broadcasting on a graph can be realized by taking the root of a spanning tree, or a set of spanning trees, of the graph as the source node. More efficient broadcasting can be achieved if more spanning trees can be constructed provided that these spanning trees are edge-disjoint and have a common root. Graphs of minimal height have minimal propagation time which is an important concern for small data volumes. For large data volumes, efficient bandwidth utilization is important, in particular, if each processor can communicate on all its ports concurrently.

Several communication strategies for hypercubes have been proposed in previous researches [3, 4, 5, 8, 9, 10]. In [3], Johnsson and Ho define the binomial spanning tree and n edge-disjoint spanning trees which can be used for broadcasting on a *complete* hypercube. Tien [5] devises and analyzes a broadcasting algorithm based on edge-disjoint spanning trees in an *incomplete* hypercube of $2^n + 2^k$ nodes, where $0 \leq k < n$. Ramanathan and Shin [8] propose a reliable broadcasting algorithm which delivers multiple copies of the broadcast message through disjoint paths to all nodes of the hypercube. Nodes receive *many* copies of broadcast message and choose one by the majority vote. In [4], Kateseff describes a routing and broadcasting algorithm for *incomplete* hypercubes which are hypercubes with certain nodes removed. Lee and Hayes [10] present fault-tolerant communication approach to routing and broadcasting in hypercubes. *One* spanning tree of the faulty hypercube is found. In this paper, we find the maximum number of edge-disjoint spanning trees in a faulty hypercube. Broadcasting based on these edge-disjoint spanning trees is efficient in the sense of no redundant messages transmitted. The faulty hypercube concerned in this paper has arbitrary one faulty node. The incomplete hypercube defined in [4] has M nodes and is formed by removing the *last* N - M nodes from a complete n-cube. Also only *one* spanning tree of the incomplete hypercube is found in [4].

The communication model is assumed to be packet-switched or store-and-forward. Messages are broken down into smaller pieces, or packets, of the same size. All incident links of a node can be used for

0-8186-6555-6/94 $04.00

transmission or reception simultaneously. We use M to denote the number of packets to be broadcasted; τ is the transfer time of a packet over a hop; and t is the start-up time for communication of a packet. For simplicity, we assume that $\tau = 1$ and $t = 0$ in analyzing the communication complexity.

The rest of this paper is organized as follows. In Section II, notations, properties of hypercubes and edge-disjoint spanning trees in hypercubes are introduced. The edge-disjoint spanning trees of a faulty hypercube are constructed in Section III. Section IV describes broadcasting based on the spanning graph and estimates the broadcasting complexity. A conclusion is given in Section V.

II. Preliminary

An n-dimensional hypercube Q_n (i.e., n-cube) can be modeled as a graph G(V,E), with $|V| = 2^n = N$ nodes, and $|E| = n2^{n-1}$ edges. Each node represents a processor and each edge represents a link between a pair of processors. The node address can be uniquely represented by an n-bit binary string from 0 to $N - 1$ such that there exists a link between two nodes if and only if their addresses differ in exactly one bit. Links are typically bidirectional, each edge in E corresponding to two directed links with opposite direction. A link which connects two nodes whose binary addresses differ in the ith bit (starting with the least significant bit as bit 0) is referred to as an i-th dimension link. The relative address of two nodes u and v is defined as the bitwise Exclusive-OR of their node addresses, $u \oplus v$. The Hamming distance between two nodes u and v, $H(u,v)$, is the number of ones in their relative address. The ith bit of the node address corresponds to the ith dimension in a Boolean space. In a node address symbol $*$ is used as a don't-care symbol whose value can be 0 or 1, and y^i stands for i consecutive y's. u_i denotes the ith neighbor of node u, i.e., $u_i = u \oplus 2^i$. Two nodes have the same lowest $k-1$ bits address value are referred to as the corresponding node to each other in a Q_k. In this paper, an n-dimensional hypercube Q_n that contains arbitrary one faulty node is denoted by Q_n'. The node with address (0^i) in Q_i', $0 \le i \le n-1$, is called the corresponding source node s'.

In [3], Johnsson and Ho defined n directed spanning trees in an n-cube and showed that they are all edge-disjoint (in the directed sense). The ith spanning tree, where $i \in \{0,1,\cdots,n-1\}$, is constructed by extending an edge from the root across dimension i to node $x = (0^{n-i-1}10^i)$ first, and then construct a spanning tree rooted at node x in subcube $(*^{n-i-1}\,1*^i)$ according to the sequence of dimension $(i+1)$ *mod* n, $(i+2)$ *mod* $n,\cdots$, $(i-1)$ *mod* n, and finally, by appending each node (except node 0^n) in subcube $(*^{n-i-1}\,0*^i)$ to its corresponding node in subcube $(*^{n-i-1}\,1*^i)$. Fig. 1 shows an example of three edge-disjoint spanning trees (EDST) in a 3-cube, where the labels on directed links are link numbers. The number of communication steps for the broadcasting algorithm based on n EDST's on an n-cube is shown to be $\lceil \frac{M}{n} \rceil + n$. Fig.2 shows an example of broadcasting three elements based on the three EDST's on a 3-cube, where labels on links are communication steps.

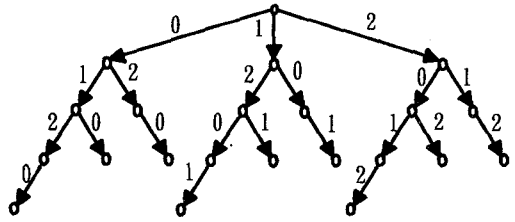

Fig.1. Three EDST's in 3-cube

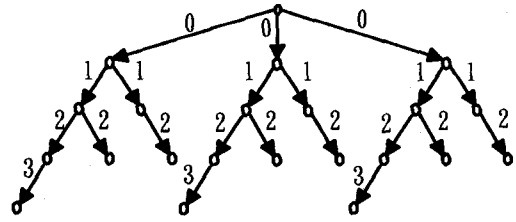

Fig.2. Broadcasting on a 3-cube

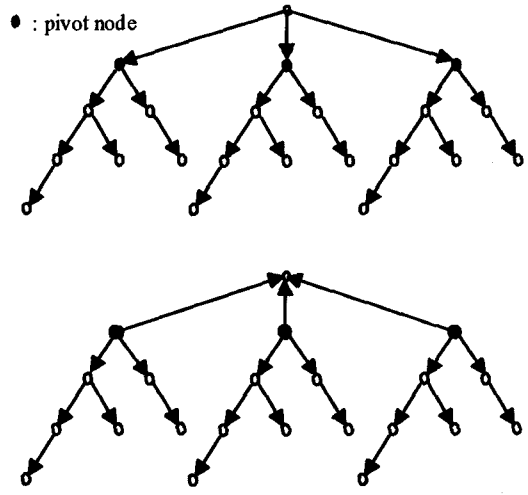

Fig.3. Three rooted EDST's and three pivoted EDST's

In [5], Tien generalized the construction method of edge-disjoint spanning trees for the incomplete

hypercube I_k^n, which consists of an n-cube and a k-cube, and n > k >= 0. The EDST's constructed by Johnsson and Ho are referred to as *rooted EDST's,* for all spanning trees have a common root. In [5], a set of *pivoted EDST's* are formed by reversing the direction of the directed link between the common root and its immediate successors in the rooted EDST's. Pivoted EDST's are shown to be edge-disjoint [5]. Roots of pivoted EDST's are called *pivot nodes.* Fig.3 shows the rooted EDST's and pivoted EDST's in a 3-cube. Each EDST in Q_n' is referred to as a *light EDST,* because each spanning tree in Q_n' has one less node than the other spanning trees in Q_n.

III. Light edge-disjoint spanning trees in Q_n'

In this section, we show how $(n - 1)$ light EDST's are constructed in Q_n'. The root of each light spanning tree is a neighbor of a common node, the source node s.

Lemma 1: For any given node x in Q_n, there exists an automorphism which maps node x to node (0^n).

Proof: Let node x = $(x_{n-1}x_{n-2}\cdots x_0)$ and the automorphism of Q_n be defined as $\sigma_x : y \to x \oplus y$, where $\oplus$ denotes the bitwise Exclusive-OR operator. Thus, $\sigma_x(x) = (0^n)$.□

In the following, by lemma 1, we assume node (0^n) in Q_n' is the source node s. s_0 and s_1 denote the 0th and 1th neighbors of the source node s.

Lemma 2: For any node f in Q_n with Hamming distance d to the source node,(i.e., $H(f,s)=d$), there exists a dimension permutation which maps node f to (1^d0^{n-d}).

Proof: Without loss of generality, we assume node f in Q_5 is (00101), H(f,s) = 2. With dimension permutation π = [2 0 4 3 1], node f is mapped to node (1^20^3).□

A. Light EDST's in Q_3'

Light EDST's in Q_1' and Q_2' are obvious. For Q_3', the procedure to find the EDST's depends on the Hamming distance between the faulty node and the source node s. We consider three cases of H(f,s) = 1, H(f,s) = 2 and H(f,s) = 3. In each case, two light EDST's rooted at neighbors of source node are constructed.

Note that in case 2, the spanning tree rooted at s_0 has height n - 1, one less than that of the other spanning trees in all cases. A light EDST is referred to as the ith light EDST in if it's root address is $(0^{n-i-1}10^i)$, $0 \le i \le n-1$.

case 1: H(f,s) = 1

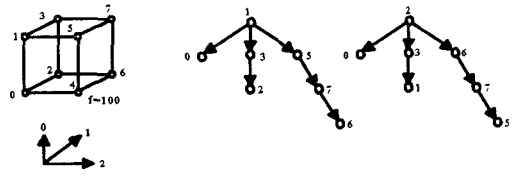

Fig.4. A Q_3' with H(f,s) = 1 and two light EDST's rooted at s_0 and s_1

case 2: H(f,s) = 2

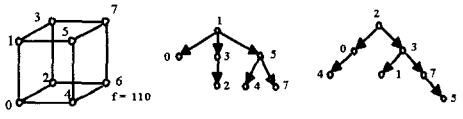

Fig.5. A Q_3' with H(f,s) = 2 and two light EDST's rooted at s_0 and s_1

case 3: H(f,s) = 3

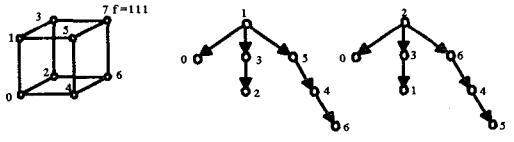

Fig.6. A Q_3' with H(f,s) = 3 and two light EDST's rooted at s_0 and s_1

Lemma 3: Two light EDST's can be constructed in an Q_3'.

Proof: It is straightforward to verify that each pair of light EDST's in cases 1 to 3 are edge-disjoint.□

B. Light EDST's in $Q_n', n \ge 4$

A Q_n can be decomposed into two Q_{n-1} along dimension i, $0 \le i \le n-1$. With lemmas 1 and 2, it is clear that a Q_n' can be decomposed into a sequence of subcubes $\{Q_{n-1}, Q_{n-2}, \cdots, Q_i, Q_i'\}$, such that Q_i' is the subcube containing the faulty node. There are many ways to decompose Q_n' into a sequence of subcubes. For convenience, we fix the decomposition such that nodes in larger subcube have smaller addresses than that of the nodes in smaller subcube. For instance, a Q_5' with faulty node f = (11000) can be decomposed into $\{Q_4, Q_3, Q_3'\}$, where $(0*^4$, $10*^3$ and $11*^3)$ denote the addresses of Q_4, Q_3 and Q_3', respectively.

For constructing light spanning trees for higher dimensional cube, the two light EDST's in Q_3' are taken as basis, and then used for constructing three light EDST's in $\{Q_3, Q_3'\}$, which composes Q_4'. The three light EDST's are in turn taken as basis for constructing

four light EDST's in $\{Q_4, Q'_4\}$, which compose Q'_5. Continue this process until n - *1* light EDST's for Q'_n are constructed. By lemmas 1 and 2, only n cases (cases of H(f,s) = d, $1 \le d \le n$) are considered. These n cases can be categorized into case *A* and case *B*. Case *A*: $2 \le H(f,s) \le n$; case *B*: H(f,s) = 1. A faulty node f having H(f,s') = d in Q'_{n-1}, $1 \le d \le (n-1)$, has H(f,s) = $d + 1$ in Q'_n. Constructing three light EDST's in Q'_4 is described as an example.

Case *A*: H(f,s) = d, $2 \le d \le 4$.

For a Q'_4, which can be decomposed into $\{Q_3, Q'_3\}$, according to the value of H(f,s) in subcube Q'_3, we first construct two light EDST's rooted at s'_0 and s'_1. Then construct two pivoted EDST's in subcube Q_3 rooted at s_0 and s_1 according to [5]. Appending nodes s'_0 and s'_1 to nodes s_0 and s_1, respectively, across dimension 3, the first two light EDST's are formed. For the last light EDST in Q'_4, attach nodes s'_0 and s'_1 to node s', and then attach all nodes (except nodes s'_0 and s'_1) in subcube Q'_3 to their corresponding nodes in Q_3, together with the last pivoted EDST in subcube Q_3, the last light EDST is formed.

Case *B*: H(f,s) = 1

When the faulty node is the corresponding source node s' in subcube Q'_3, constructing three pivoted EDST's in subcube Q'_3 by removing the faulty node from each pivoted EDST. The removal of the faulty node in each pivoted EDST does not destroy the spanning tree. Then, construct three pivoted EDST's in subcube Q_3 and appending the three pivot nodes in Q'_3 to it's corresponding nodes in subcube Q_3, respectively. Three light EDST's are formed in Q'_4.The procedure described is generalized as Algorithm A.

Algorithm A:

Case A:

Input: n - *2* light EDST's, denoted $ST' = \{ST'_i\}$, in Q'_{n-1} rooted at s'_i, $0 \le i \le n-3$.

Output: n - *1* light EDST's, denoted $T = \{T_i\}$, in Q'_n rooted at s_i, $0 \le i \le n-2$.

1. Decompose Q'_n into subcubes Q_{n-1} and Q'_{n-1}.

2. Construct n - *2* pivoted EDST's, denoted $ST = \{ST_i\}$, in subcube Q_{n-1} rooted at s_i, where $0 \le i \le n-3$.

3. For each i, $0 \le i \le n-3$, append ST'_i to ST_i via a link from s'_i to s_i across dimension n - *1*. The first n - *2* light EDST's are constructed.

4. Construct the last pivoted EDST ST_{n-2} in subcube Q_{n-1} rooted at s_{n-2} .

5. Append each node s'_i, $0 \le i \le n-3$, to the corresponding source node s' in subcube Q'_{n-1}.

6. Append all nodes, except node s'_i, $0 \le i \le n-3$, in subcube Q'_{n-1} to its corresponding node in subcube Q_{n-1}, respectively. The last light EDST is constructed.

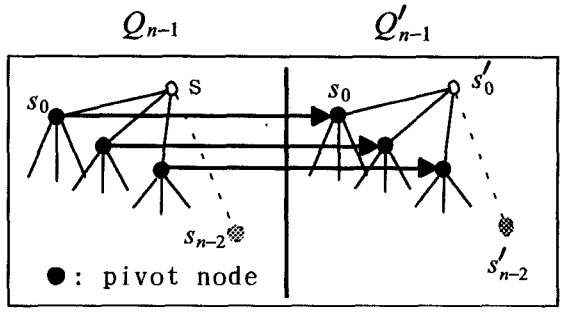

Fig. 7 Steps 1 to 3 for case A

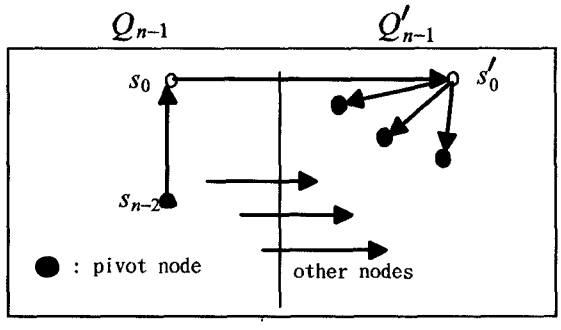

Fig. 8 Steps 4 to 6 for case A

Case B: H(f,s) = 1

Input: Q'_n.

Output: n -*1* light EDST's, denoted $T = \{T_i\}$, in Q'_n rooted at s_i, $0 \le i \le n-2$.

1. Decompose Q'_n into subcubes $\{Q_{n-1}, Q'_{n-1}\}$.

2. Construct n - *1* pivoted EDST's, denoted by $ST' = \{ST'_i\}$, $0 \le i \le n-2$, in subcube Q'_{n-1} with the faulty node removed from each pivoted EDST.

3. Construct n - *1* pivoted EDST's, denoted by $ST = \{ST_i\}$, $0 \le i \le n-2$, in subcube Q_{n-1} rooted at s_i.

4. For each i, $0 \le i \le n-2$, appending ST'_i to ST_i from s'_i to s_i via a link across dimension n - *1*.

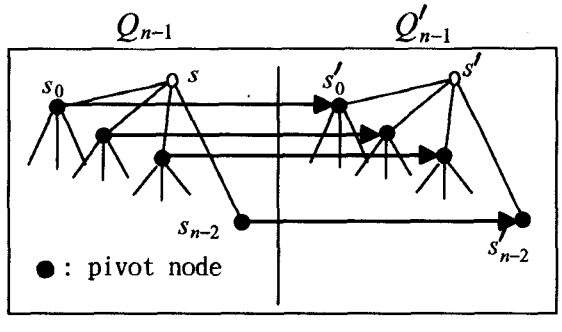

Fig. 9 Steps 1 to 4 for case B

Lemma 4: The outputs of Algorithm A finds n - *1* light EDST's, denoted $T = \{ T_i \}$, rooted at s_i, $0 \le i \le n-2$, in Q'_n.

Proof: It is clear that the n - *1* light EDST's constructed by Algorithm A are spanning trees for Q'_n. We are going to show that they are edge-disjoint in cases A and B. Links in Q'_n are partitioned into five sets (1) L1: the set of links in subcube Q_{n-1}, (2) L2: the set of links with link number n - *1*, i.e., links connecting subcube Q_{n-1} and subcube Q'_{n-1}, (3) L3: the set of n - *1* links from node s' to node s'_i, $0 \le i \le n-2$, in subcube Q'_{n-1}, (4) L4: the set of n - *1* links from node s'_i to node s', $0 \le i \le n-2$, in subcube Q'_{n-1}, (5) L5: the other links in subcube Q'_{n-1}.

A link in L1 belongs to at most one spanning tree, since pivoted EDST's $ST_0, ST_1, \cdots, ST_{n-2}$ are all edge-disjoint [5]. In case A, a link in L2 belongs to T_i, $0 \le i \le n-3$, if it is incident to node s_i and it belongs to T_{n-2} otherwise. In case B, a link in L2 belongs to T_i, $0 \le i \le n-2$, if it is incident to node s_i. Links in L3, except the link incident to node s'_{n-2}, belong to T_{n-2} only in case A. Links in L3 are not used in case B. A link in L4 belongs to T_i for case A, if it is incident to node s'_i, where $0 \le i \le n-3$. Links in L4 are not used in case B. Any link in L5 belongs to at most one spanning tree, since $ST'_0, ST'_1, \cdots, and\ ST'_{n-2}$ are edge-disjoint.□

Lemma 5: For case A, heights of n - *2* light EDST's out of the n - *1* light EDST's are n; the other one has height n - *1*. For case B, each of the n - *1* EDST's has height n. Each spanning tree is of optimal height.

Proof: The two light EDST's in Q'_3 shown in Fig. 4 to Fig. 6 are used as our basis. Note that each light EDST in Q'_3 has height 3, except when the address of the faulty node is $(1^2 0)$; in this case, the height of the light EDST rooted at $(0^2 1)$ is 2.

Case A: A Q'_n is first decomposed into a sequence of n - *2* subcubes $\{Q_{n-1}, Q_{n-2}, \cdots, Q_3, Q'_3\}$, before calling Algorithm A for constructing light EDST's. It takes one hop to append a light EDST ST'_i, $0 \le i \le n-3$, in Q'_{n-1} to a pivoted EDST ST_i in Q_{n-1} to form a light EDST T_i in Q'_n. For light EDST T_{n-2}, appending nodes in subcube Q'_{n-1} to its corresponding nodes in subcube Q_{n-1} also increases the height by one. Algorithm A is called *(n - 3)* times while constructing n - *1* light EDST's for Q'_n from Q'_3. Therefore, the height of each light EDST in Q'_n is $3 + 1 + \cdots + 1 = n$ and the height of the light EDST is $2 + 1 + \cdots + 1 = n - 1$ for the exception.

Case B: The height of each pivoted EDST in Q'_{n-1} is n - 1 [5]. Removal of the fault node from each pivoted EDST in Q'_{n-1} does not affect the height. For constructing T_i, $0 \le i \le n-2$, appending node s'_i to node s_i increases the height by one. The height of each light EDST is $n - 1 + 1 = n$.

The address of the deepest node in a spanning tree is the complementary of that of the root node. The addresses of the n - *1* roots are $(0^{n-i-1}10^i)$, $0 \le i \le n-2$. Only when the unique faulty node is located at $(1^{n-1}0)$, the spanning tree rooted at $(0^{n-1}1)$ has height n-1. For all other situations, each spanning tree has height n, diameter of Q_n, the minimal possible value.□

Theorem 1: The output of Algorithm A finds maximum number, n - *1*, of edge-disjoint spanning trees in Q'_n, and each spanning tree is of optimal height.

Proof: For the minimum indegree (outdegree) of Q'_n is n - *1*, at most n - *1* EDST's can be expected. With lemmas 4 and 5, Theorem 1 is proved.□

IV. Broadcasting on Q'_n

The performance of broadcasting is determined mainly by two factors. One is the bandwidth utilization, denoted α, which is the average number of packets can be sent per step from the root. The other factor is the latency, denoted β, which is the propagation delay for a packet. The latency can be measured by the number of steps for a packet to reach the farthest destination. The communication complexity of a broadcast is $\lceil \frac{M}{\alpha} \rceil + \beta - 1$ if both α and β are fixed. The optimal communication complexity of a broadcast can be obtained if α is maximal and β is minimal.

By taking the common neighbor of the roots of T_i, $0 \le i \le n-2$, as the new root and reversing the direction of the directed link from each root to the new root a spanning graph, not a spanning tree, consists of n - 1 EDST's, is formed. Broadcasting on Q'_n can be realized based on the spanning graph with the new root being source node.

Lemma 6: The diameter of the spanning graph is n + *1*.

Proof: From lemma 5, the heights of the n - *1* EDST's are at most n. Reversing the direction of the directed link from each root to their common neighbor increases the height of each spanning by one; thus the maximal height become $n + 1$.□

Lemma 6 can also be proved by using the notion of the fault diameter [7]. In [7], Latifi showed that $d_{n-1} = n + 1$, i.e., even if n - *1* processors fail, the diameter increases only by 1. In Q'_n, only 1 outgoing link of the source node can be used by each of the n - *1* EDST's. To each spanning tree this is equivalent to the case that the other n - *1* neighbors of the source node are not used, i.e., the n - *1* neighbors can be thought as faulty nodes, in the first step of broadcasting. Because there is only one faulty

node in Q_n^f, at least n - 2 out of the n - 1 EDST's, rooted at the source node, are of height n + 1, which is equivalent to d_{n-1} (= n + 1).

Theorem 2: The communication complexity of the broadcasting algorithm based on the spanning graph is $\lceil \frac{M}{n-1} \rceil + n$, which is the minimal complexity.

Proof: From Theorem 1, the value of α is n - 1 on a Q_n^f. From lemma 6, the value of β is n + 1. The communication complexity of a broadcast is $\lceil \frac{M}{n-1} \rceil + n$ because n - 1 is the maximal value for α and n + 1 is the minimal value for β and the complexity is minimal.□

V. Conclusion

The hypercube topology has been one of the most popular topologies used in distributed-memory multiprocessor. Because the probability that a processor may fail in such a parallel system is quite large, algorithms designed for faulty hypercubes are necessary.

This paper has proposed an efficient method for constructing the maximum number of edge-disjoint spanning trees on a faulty hypercube, which is a hypercube containing one arbitrary faulty node. Each of the spanning trees is of optimal height. With these edge-disjoint spanning trees as our basis, a spanning graph is formed. Broadcasting based on the spanning graph sends messages to each node without any redundant messages. The communication complexity of a broadcast is $\lceil \frac{M}{n-1} \rceil + n$, which has an optimal bandwidth utilization and an optimal latency.

References :

[1]. Y. Saad and M. H. Schultz, "Topological properties of hypercubes," IEEE Trans. Computers.,Vol. 37, No. 7, pp. 867-872, July 1988

[2]. W. D. Hills, The Connection Machine. Cambridge, MA: M.I.T. Press, 1985.

[3]. S. L. Johnsson and C.-T. Ho, "Optimum broadcasting and personalized communication in hypercubes," IEEE Trans. Computers, Vol. 38, No.9, pp. 1249-1268, Sept. 1989.

[4]. H. P. Katseff, "Incomplete hypercubes," IEEE Trans. Computers, Vol. 37, pp. 604-608, May 1988.

[5] J. Y. Tien, C. -T. Ho, and W. -P. Yang, "Broadcasting on Incomplete Hypercubes," IEEE Trans. Computers, Vol. 42, No. 11, pp. 1393-1398, Nov. 1993.

[6]. C. L. Seitz, "The cosmic cube," Commun. ACM, Vol. 28, pp. 22-23, Jan. 1985.

[7]. S. Latifi, "Combinatorial Analysis of the Fault-Diameter of the n-cube," IEEE Trans. Computers, Vol. 42, No. 1, pp. 27-33, Jan. 1993.

[8]. P. Ramanathan, Kang G. Shin, "Reliable Broadcast in Hypercube Multicomputers," IEEE Trans. Computers, Vol. 37, No. 12 pp. 1654-1657, Dec. 1988.

[9]. P. Fraigniaud, "Asymptotically Optimal Broadcasting and Gossiping in Faulty Hypercube Multicomputers," IEEE Trans. Computers, Vol. 41, No. 11, pp. 1410-1419, Nov. 1992.

[10]. T. C. Lee and J. P. Hayes, "A Fault-Tolerant Communication scheme for Hypercube Computers," IEEE Trans. Computers, Vol.41, No. 10, pp. 1242-1256, Oct. 1992.