Partitioning and Broadcasting in Hypercubes in the Presence of Faulty Communication Links

Suckjun Lee

November 1993

Abstract

The problem of broadcasting in faulty hypercubes is considered, based upon a strategy of partitioning the faulty hypercube into subcubes in which currently known algorithms can be implemented. Three similar partitioning and broadcasting algorithms for an n-dimensional hypercube in the presence of up to $(n^2 + 2n - c) / 4$ faulty communication links are presented, where $c = 4$ if n is an even number or $c = 3$ if n is an odd number. The most efficient algorithm is implemented in $1.3n + 6\log(n) + 9$ time units. To the best of our knowledge, this algorithm is the most efficient one for an n-dimensional hypercube in the presence of $O(n^2)$ faults.

1. Introduction

Parallel processing can improve computer performance executing programs simultaneously on several processors. Topologies used to interconnect processors vary from one computer system to another. Among several interconnecting topologies for parallel computers, the n-dimensional binary hypercube has been one of the most popular approaches. To maintain the high performance of parallel computers, efficient communications between processors are required. An optimal broadcasting algorithm for hypercube machines in the absence of faults has been proposed [2]. However, this algorithm can't be properly implemented in the presence of faults. The maintenance of high performance, fault-tolerant communications, has become an important issue as the use of hypercube machines has increased.

A number of fault-tolerant communication algorithms have been proposed recently [3, 4, 6 - 10]. These algorithms are capable of providing both simpe and efficient communications for hypercube machines in the presence of faults. However, when more than 2n - 3 faults occur on n-dimensional hypercube machines, an algorithm that can be properly used for communications hasn't been developed. Bruck, Cypher and Soroker have proposed a broadcasting scheme for n-dimensional hypercube machines in the presence of faults in excess of n - 1 [5]. However, this approach didn't present a concrete broadcasting algorithm. Also they presented an open problem which is to find an efficient fault-tolerent partitioning algorithm. Finding such an algorithm currently requires $\Omega\binom{n}{m}$ sequential time, where m is the dimension of subcubes obtained from partitioning an n-dimensional hypercube.

In this paper, three different broadcasting algorithms for n-dimensional hypercubes in the presence of up to $(n^2 + 2n - c) / 4$ faulty communication links, where c = 4 if n is an even number or c = 3 if n is an odd number, are presented. The most efficient algorithm is implemented in 1.3n + 6log(n) + 9 time units. Also an efficient partitioning algorithm is presented. The time complexity to partition an n-cube into k dimensional subcubes contained a certain number of faulty links is $O(n^4)$.

The basic idea of the proposed algorithms is as follows. Existing fault-tolerant broadcasting algorithms can be implemented successfully in hypercube machines when less than expected number of faults occur. When the broadcasting algorithms can't be implemented properly as new faults occur, the hypercube will be partitioned into subcubes in which the algorithms can be implemented properly. Therefore the broadcasting algorithms are implemented successfully in each subcube. By creating special connections between subcubes, every processors will be able to communicate efficiently.

This approach is based upon the assumption that each processor knows all the addresses of faulty communication links, and that each processor takes one time unit to send message to an adjacent processor. It is also assumed that all the processors in the hypercube machines are connected.

This paper is organized as follows. Section 2 describes the notation used in the paper and discusses some properties of hypercubes. Section 3 proves theorems which support the proposed partitioning algorithms. Sections 4 - 6 describe the proposed algorithm-A, algorithm-B, and algorithm-C respectively. The conclusions drawn from this research are presented in Section 7.

2. Properties and Notation

In this section, the following new terms are introduced. The term *n-cube* is used to indicate an n-dimensional hypercube, and the term *k-subcube* indicates a k-dimensional subcube of the n-cube. For the hypercube machine considered, *nodes* correspond to processors and *links* correspond to communication links.

An n-cube consists of 2^n nodes and $n2^{n-1}$ links. A node in the n-cube is represented by the binary sequence $(b_{n-1}\ b_{n-2}\ b_{n-3}\ .\ .\ .\ b_0)$. The Hamming weight of a node, $(b_{n-1}\ b_{n-2}\ b_{n-3}\ .\ .\ .\ b_0)$, is $\sum_{i=0,n-1} b_i$.

Let *node(k)* be a node of Hamming weight k. Similarly, let *link(d)* be a link between node(d)'s and node(d+1)'s.

Suppose that a link is on i^{th} dimension. Then the link is represented by the sequence $(b_{n-1}\ b_{n-2}\ .\ .\ .\ b_{i+1}\ -\ b_{i-1}.\ .\ .\ b_0)$, where b_j is 0 or 1 if j isn't i.

The k-subcube of an n-cube will be represented by the sequence $(b_{n-1}\ b_{n-2}\ .\ .\ .\ b_0)$, where b_j is 0, or 1 if the n-cube is partitioned along the dimension j; otherwise b_j is *.

The number of node(k)'s in the n-cube is equal to $\binom{n}{k}$. From the following proposition, it is proved that the number of link(k)'s in the n-cube is $\binom{n}{k}(n - k)$.

Propositon 2.1: The number of link(k)'s in an n-cube is equal to

$$\binom{n}{k}(n - k)$$

Proof: Let the out-degree of a node(k) be the number of link(k)'s connected to the node(k). The out-degree of the node(k) is n - k. The number of link(k)'s is the number of node(k)'s times the out-degree of the node(k). Therefore, the number of link(k)'s is equal to $\binom{n}{k}(n - k)$.

There are n different ways to partition the n-cube into two (n-1)-subcubes. Let *major (n-1)-subcube* be an (n-1)-subcube which contains the larger number of faulty links among two (n-1)-subcubes of the n-cube. If the number of faulty links in two subcubes are equal, then the major subcube can be either of the subcubes. Let *minor (n-1)-subcube* be the other (n-1)-subcube that is not selected as the major subcube.

The subcubes of an n-cube can be logically disconnected. The nodes in the largest component of a disconnected subcube will be referred to as *connected nodes*, whereas the remaining nodes will be referred to as *isolated nodes*.

For an n-cube in the presence of f faulty links, there could be different number of faulty links in n different major subcubes. let *F(n, f)* be the minimum number of faulty links among all major subcubes. Given a dimension n and the number of faulty links f, let $\alpha(n, f)$ be the maximum number of faulty links among all possible F(n, f)'s. In other words, given any n-cube in the presence of f faulty links, the n-cube is gauranteed to be partitioned into two subcubes such that each subcube contains a maximum of $\alpha(n, f)$ faulty links. For example, $\alpha(5, 8)$ is 5 as will be shown in section 3. Then, given any 5-cube in the presence of 8 faulty links, there must be a way to partition the 5-cube into two 4-cubes such that each 4-cube contains at most 5 faulty links.

Similarly, let $\beta(n, f)$ be the maximum number of faulty links in an (n + 1)-cube such that $\alpha(n + 1, \beta(n, f)) = f$. For example, $\beta(4, 5) = 8$ since

$\alpha(5, 8) = 5$ and $\alpha(5, 9) = 6$ as will be shown in section 3. In other words, there exists a 5-cube in the presence of 9 faulty links such that it is impossible to partition the 5-cube into 4-cubes, each of which contains at most 5 faulty links.

Finally, let the *degree* of a node be the number of healthy links which are connected to that node. And let *distance* between two nodes be the minimum of time units required for a node to send message to the other node.

3. Fault-Tolerant Partitions

In this section, given any n-cube in the presence of a maximum of $(n^2 + 2n - c) / 4$ faulty links, it is proved that the n-cube can be partitioned into 3-subcubes such that each 3-subcube contains at most 3 faulty links, where $c = 4$ if n is an even number, or $c = 3$ if n is an odd number.

First, $\alpha(n, f)$ will be calculated to prove the claim. It is clear that $F(n, n) = n - 1$, when all of the faulty links are adjacent to a single node. For any n-cube in the presence of n faulty links, the maximum of $F(n, n)$ is $n - 1$ since the n-cube will be partitioned along a dimension on which at least one faulty link exists. Therefore, $\alpha(n, n) = n - 1$.

$\alpha(n, n)$ can be calculated in a different method. For an n-cube in the presence of a single faulty link(k), the number of (n - 1)-subcubes, each of which includes both the node(0) and the faulty link(k), is $(n - k - 1)$. Thus, for an n-cube in the presence of a single faulty link(0), the number of (n - 1)-subcubes, each including both the node(0) and the faulty link(k), is

$(n - 1)$. It follows that, for the n-cube in the presence of n faulty link(0)'s, the total number of faulty links in all subcubes containing the node(0) is $n(n - 1)$. It is clear that for each partition, the major subcube is always the subcube which contains the node(0). Thus the total number of faulty links in all major subcubes is $n(n - 1)$. Since there are n different major subcubes, by the Pigeon Hole Principle, there must exist at least one major subcube containing a maximum of $(n - 1)$ faulty links. Therefore, given the n-cube in the presence of n faulty link(0)'s, $F(n, n) = n - 1$.

For an n-cube in the presence of one faulty link(k), the number of $(n - 1)$-subcubes, each of which includes both the node(0) and the faulty link(k), is $(n - k - 1)$. Similiarly, the number of $(n - 1)$-subcubes, which do not include the node(0), but do include the faulty link(k), is k. $n - 1$ is the maximum number of times a single faulty link can occur in subcubes for n different partitions. Thus $n(n - 1)$ is the maximum total number of faulty links in all major subcubes for n faulty links. Thus $F(n, n)$ can't be greater than $n(n - 1) / n = (n - 1)$. It follows that $\alpha(n, n) = n - 1$.

Now $\alpha(n, n + 1)$ will be calculated by using the same method used for calculating $\alpha(n, n)$. Since the number of faulty links is $n + 1$, at least one dimension exists on which there are at least two faulty links. By partitioning the n-cube along that dimension, the number of faulty links in the major subcube can't be larger than $n - 1$. Since $\alpha(n, n) = n - 1$, $\alpha(n, n + 1)$ must be $n - 1$. Therefore, $\alpha(n, n + 1) = n - 1$.

Now let's again calculate $\alpha(n, n + 1)$ in a different way. For n faulty links, the maximum total number of faulty links in all major subcubes is $n(n - 1)$. For the remaining faulty link, the maximum total number of faulty links in all major subcubes is $(n - 2)$. Thus, the maximum total number of faulty links in all major subcubes is $n(n - 1) + (n - 2)$ for $n + 1$ faulty links. From the Pigeon Hole Principle, at least one major subcube containing a maximum of $(n - 1)$ faulty links must exist and, for an n-cube in the presence of n faulty link(0)'s and one faulty link(1), it is clear that $F(n, n + 1) = n - 1$. Therefore, $\alpha(n, n + 1) = n - 1$.

Based upon this concept, it is easy to verify that $\alpha(n, n^2) = n(n - 1) / n + (n^2 - n)(n - 2) / n = (n - 1)^2$. Note that the total number of link(0)'s and link(1)'s is n^2. Similarly, $\alpha(n, f)$ is equal to $\sum_{i=0,j} \binom{n-1}{i}(n - i - 1)$, where $f = \sum_{i=0,j} \binom{n}{i}(n - i)$.

In general, $\alpha(n, f) = \sum_{i=0,k} \binom{n-1}{i}(n - i - 1) + \lfloor (f - \sum_{i=0,k} \binom{n}{i}(n - i))(n - k - 2) / n \rfloor$, where k is the largest number that satisfies $\sum_{i=0,k} \binom{n}{i}(n - i) \leq f \geq n$.

Theorem 3.1 : Given an n-cube in the presence of f faulty links, $\alpha(n, f) = \sum_{i=0,k} \binom{n-1}{i}(n - i - 1) + \lfloor (f - \sum_{i=0,k} \binom{n}{i}(n - i))(n - k - 2) / n \rfloor$, where $f \geq n$ and k is the largest number that satisfies $\sum_{i=0,k} \binom{n}{i}(n - i) \leq f$. For $f \leq n$, $\alpha(n, f) = f - 1$.

Proof: It is clear that $\alpha(n, f) = \lfloor f(n - 1) / n \rfloor = f - 1$, where $f \leq n$. Therefore, the proof of $\alpha(n, f)$ is considered only when $f \geq n$. Suppose that the claim is not true. Then, all n different major subcubes contain more than $\sum_{i=0,k} \binom{n-1}{i}(n - i - 1) + \lfloor (f - \sum_{i=0,k} \binom{n}{i}(n - i))(n - k - 2) / n \rfloor$ faulty links.

Let S be a set which contains n major subcubes. By changing subcubes in S and faulty links in the n-cube, it will be shown that all subcubes containing the node(0), can contain more than $\sum_{i=0,k} \binom{n-1}{i} (n - i - 1)$ $+ \lfloor (f - \sum_{i=0,k} \binom{n}{i} (n - i)) (n - k - 2) / n \rfloor$ faulty links.

Transfer Rule

If $(d_{n-1}, \ldots, d_{i+1}, 1, d_{i-1}, \ldots, d_0)$ is in S, then

it will be changed to $(d_{n-1}, \ldots, d_{i+1}, 0, d_{i-1}, \ldots, d_0)$.

at the same time,

if $(d_{n-1}, \ldots, d_{i+1}, 1, d_{i-1}, \ldots, d_0)$ is faulty

and if$(d_{n-1}, \ldots, d_{i+1}, 0, d_{i-1}, \ldots, d_0)$ is not faulty, then

they will be switched.

The Transfer Rule is applied, one dimension at a time for $0 \leq i \leq n - 1$, to all subcubes in S and all faulty links in the n-cube. Then, all subcubes in S are transferred to the subcubes which contain the node(0).

When a subcube is transferred to another subcube, some of the faulty links in the former subcube are switched with healthy links in new subcube. It is easy to verify that the number of faulty links in new subcube can't be less than the number of faulty links in the older subcube. And the number of faulty links in the n-cube isn't changed.

Therefore, all subcubes containing the node(0) contain more than

$\sum_{i=0,k}\binom{n-1}{i}(n-i-1) + \lfloor(f - \sum_{i=0,k}\binom{n}{i}(n-i))\ (n-k-2)\ /\ n\rfloor$ faulty links.

Therefore, the sum of faulty links in n subcubes containing the node(0) can be greater than $\sum_{i=0,k}\binom{n-1}{i}(n-i-1)n + (f - \sum_{i=0,k}\binom{n}{i}(n-i))$ $(n-k-2)$. If the the sum of faulty links in n subcubes containing the node(0) can't be greater than $\sum_{i=0,k}\binom{n-1}{i}(n-i-1)n + (f - \sum_{i=0,k}\binom{n}{i}(n-i))$ $(n-k-2)$, then the claim is proved by contradiction.

By counting the maximum number of faulty links in all (n - 1)-subcubes containing the node(0), the proof will be done. It is known that, given an n-cube in the presence of a single faulty link(k), the total number of faulty link(k)'s in all (n - 1)-subcubes containing the node(0) is (n - k - 1). Therefore, the total number of faulty links in all (n - 1)-subcubes containing the node(0) will be maximized if all faulty links occur at the nearest distance from the node(0). Thus, the total number of faulty links in all (n - 1)-subcubes containing the node(0) is equal to $\sum_{i=0,j}\binom{n}{i}$ $(n-i)(n-i-1) = \sum_{i=0,k}\binom{n-1}{i}(n-i-1)n$ if the number of faulty links is equal to $\sum_{i=0,j}\binom{n}{i}(n-i)$.

It is easy to verify that, given any n-cube in the presence of f faulty links, the total number of faulty links in all (n - 1)-subcubes containing the node(0) can't be greater than $\sum_{i=0,k} \binom{n-1}{i}(n - i - 1)n + (f - \sum_{i=0,k} \binom{n}{i}(n - i))(n - k - 2)$. Contradiction.

Lemma 3.1 : For $n \le f \le (n^2 + 1)(n - 1) / (n + 1)$ and $n \ge 2$, $\beta(n, f) = \lceil f(n + 1) / (n - 1) \rceil - 1$.

Proof: $\alpha(n, f) = n - 1 + \lfloor (f - n)(n - 2) / n \rfloor = \lfloor f(n - 2) / n \rfloor + 1$, where $n \le f \le n^2$. Assume the claim is false. Then, $\alpha(n + 1, \lceil f(n + 1) / (n - 1) \rceil) \le f$. $\alpha(n + 1, \lceil f(n + 1) / (n - 1) \rceil) = \lfloor (f(n + 1) / (n - 1))((n - 1) / (n + 1)) \rfloor + 1 = f + 1$. Contradiction.

Theorem 3.2: Given any n-cube in the presence of a maximum of $(n^2 + 2n - c) / 4$ faulty links, where $n \ge 3$, and $c = 4$ if n is an even number, or $c = 3$ if n is an odd number, the n-cube is gaurranteed to be partitioned into 3-subcubes such that each subcube has at most three faulty links.

Proof: The claim will be proved by induction on n. For $n = 3$, the number of faulty links in 3-cube is at most 3 since $(9 + 6 - 3) / 4 = 3$.

Assume that this claim is true for n. By Lemma 3.1, $\beta(n, (n^2 + 2n - c) / 4) = (n^3 + 3n^2 - (2 + c)n + 4 - c) / 4(n - 1)$. Two cases are considered. First, suppose that n is odd, then $c = 3$. Therefore, $(n^3 + 3n^2 - 5n + 1) / 4(n - 1) = ((n + 1)^2 + 2(n + 1) - 4) / 4$. Suppose that n is even, then $\beta(n, (n^2 + 2n - 4) / 4) = ((n + 1)^2 + 2(n + 1) - 3) / 4$.

Therefore, the n-cube in the presence of at most $(n^2 + 2n - c) / 4$ faulty links is gaurranteed to be partitioned into 3-subcubes such that each subcube contains at most three faulty links.

From the proof of Theorem 3.2 , any n-cube in the presence of at most $(n^2 + 2n - c) / 4$ faulty links can be partitioned into k-subcubes such that each k-subcube contains a maximum of $(k^2 + 2k - c) / 4$ faulty links.

In the following section, based upon the results developed in this section, a fault-tolerant partitioning and broadcasting algorithm are discussed.

4. Fault-Tolerant Partitioning and Broadcasting Algorithm

In this section, a broadcasting algorithm for n-cubes in the presence of up to $(n^2 + 2n - c) / 4$ faulty links is presented. The general approach consists of partitioning a cube into subcubes in which broadcasting algorithms of proven workability can be implemented, then embedding trees into the cube for use as communication between the subcubes.

Park and Bose presented a broadcasting algorithm implemented in any connected n-cube in the presence of a maximum of $2n - 3$ faulty links [4]. This algorithm is implemented in $n + 3$ time units.

If a disconnected n-cube contains at most $2n - 3$ faulty links, then a single node is disconnected to the other nodes. Even if the n-cube in the presence of at most $2n - 3$ faulty links is disconnected physically, the Park and Bose algorithm can be successfully implemented in a connected component if the disconnected node is not a source node.

Let m-cube be *fault-tolerant* if an m-cube contains at most $2m - 3$ faulty links. The following method of partitioning a hypercube will be referred to as the *fault-tolerant partitioning method-A*.

If an n-cube isn't fault-tolerant, then the cube will be partitioned into two subcubes such that the number of faulty links in the major subcube is the smallest among n different major subcubes. If a subcube is fault-tolerant, then the subcube will not be partitioned. If a subcube isn't fault-tolerant, then the subcube continues to be partitioned until all subcubes are fault-tolerant.

From Theorem 3.2, if the n-cube contains at most $(n^2 + 2n - c) / 4$ faulty links, then this fault-tolerant partitioning method will be terminated before subcubes become 3-cubes. Note that $(n^2 + 2n - c) / 4 = 2n - 3$ for $n = 3$ or 4.

To design the proposed broadcasting algorithm by using the fault-tolerant partitioning method, two major problems must be considered. One is how to communicate from one fault-tolerant subcube to another fault-tolerant subcube. The second one is how to broadcast to logically isolated nodes obtained by partitioning the cubes. These two problems are solved by embedding trees into the cubes.

While partitioning cubes, a tree will be embedded into cubes to communicate between the fault-tolerant subcubes.

Suppose that an n-cube is partitioned into two subcubes, sbc_a and sbc_b, along the j dimension. Since the n-cube contains at most $(n^2 + 2n - c) / 4$ faulty links, there are two nodes, nd_a and nd_b, such that the degree of each node is n and the distance between these is 1. And suppose that nd_a is in sbc_a and nd_b is in sbc_b.

Suppose that sbc_a does not tolerate faulty links. Then sbc_a will be partitioned into two cubes, sbc_a' and sbc_a''. Without loss of generality, suppose that nd_a is in sbc_a'. Then, there is at least one node, nd_a', in sbc_a'' such that the degree of nd_a' is n and that the distance between nd_a' and nd_a is a maximum of 3. The tree will be extended from nd_a to nd_a'. While partitioning the cube, the tree will be constructed on the cube from repetition of the steps described above.

For each fault-tolerant subcube, there is exactly one special node on the tree, which will be referred to as the *subsource node*. The subsource node is a source node for a fault-tolerant subcube. For example, the nd_a will be a subsource node of sbc_a' if sbc_a' is fault-tolerant.

The concept of embedding trees for isolated nodes is described as follows. Let *S_in* be a set which contains all isolated nodes after partitioning a cube into fault-tolerant subcubes. Then at least one of the nodes in S_in is directly connected to connected nodes since all of the nodes are connected physically. Some isolated nodes in S_in, di_nodes, are directly connected to connected nodes. They are deleted from S_in. If S_in is not empty, some nodes exist which are directly connected to di_ nodes. They are deleted from S_in. If S_in is not empty, then above steps are repeated until S_in is empty. All the isolated nodes will be connected to the connected nodes.

The fault-tolerant broadcasting algorithm-A is described as follows. There are three different types of nodes: connected nodes, isolated nodes, and subsource nodes. When an isolated node is a source node, it transmits messages to a connected node. When a connected node receives message from an isolated node, or when a connected node is a source node, the connected node will broadcast it to all of the connected nodes in a fault-tolerant subcube by implementing the Park and Bose algorithm.

Let T be a communication tree between subcubes. When a subsource node is a source node, or when a subsource node receives a broadcasting message, the subsource node will transmit it to adjacent nodes on the T. Before transmitting message, the message will be specially marked so that each node on the T recognizes whether the message needs to be broadcast or transmitted to the adjacent nodes on the T. Let specially marked messages be *t-messages*.

Whenever each node on the T receives a t-message, the node transmits the t-message to adjacent nodes on the T. When a subsource node in each subcube receives a t-message, the subsource node changes the t-message to an original broadcasting message, and then the node broadcasts it to all the other connected nodes in a fault-tolerant subcube by implementing the Park and Bose algorithm. Before changing the t-message to original broadcasting message, the subsource node broadcasts the t-message to adjacent nodes on the T if they exist.

Let T' be communication trees between isolated nodes and connected nodes. The nodes on T' transmit broadcasting messages to adjacent nodes on T' whenever the messages are received.

Therefore, each node receives broadcasting messages from any of the source nodes.

Theorem 4.1: Given any n-cube in the presence of at most $(n^2 + 2n - c) / 4$ faulty links, the fault-tolerant broadcasting algorithm-A is implemented in $6.3n$ time units.

Proof: The broadcasting time in the worst case is bounded by the sum of following three cases:

1) the diameter of the embedded tree,

2) broadcasting time for two 4-subcube, and

3) broadcasting time for isolated nodes.

The diameter of the embedded tree is less than $6n - 29$ time units. Note that $2*3(n - 5) + 1 = 6n - 29$. 2) is at most 14 time units. 3) is at most $0.3n + 2$ time units.

Therefore, less than $6.3n$ time units are required to broadcast messages in the n-cube.

The length of a path between an isolated node and a connected node is less than $0.3n + 2$. Let *weight of a faulty link* be the number of isolated nodes which are adjacent to the faulty link minus 1. If the weights of faulty links are 0, then the maximum length is not longer than $(n / 4) + 1$. However, the weights of faulty links can be larger than 0.

Exactly one link between two isolated nodes can be adjacent to both isolated nodes. Thus, if there are n isolated nodes, the sum of weights of all faulty links can not be larger than $n(n - 1) / 2$. Therefore the lengh of paths between isolated nodes and connected nodes cannot be longer than k, where $(n - 1)k - k(k - 1) / 2 - (n - 4) = (n^2 + 2n - c) / 4$.

The time complexity of the fault-tolerant partitioning method-A is $O(n^4)$. By counting 1's on the i^{th} bits of the faulty links, whether the cube can be partitioned along the i^{th} dimension can be determined. Since there are $O(n^2)$ faulty links and each link has n bits, it takes $O(n^3)$. In the worst case, the above process is repeated n - 4 times. Therefore, the time complexity for partitioning an n-cube is $O(n^4)$, and the time complexity for finding a tree for communication between fault-tolerant subcubes is $O(n^3)$.

The time complexity for finding routes for isolated nodes is $O(n^4 \log(n))$. Since there are at most n^2 isolated nodes, the isolated nodes can be sorted in $O(n^3)$. By checking sorted isolated nodes, it is easily known that a node is an isolated node or not. The time complexity for finding isolated nodes, di_nodes, which are directly connected to connected nodes, is $O(n^3 \log(n))$. Let S_nodes be a set of isolated nodes which are not di_nodes. The time complexity for finding isolated nodes in the S_nodes directly connected to nodes in di_nodes is also $O(n^3 \log(n))$. The length of the routes is not longer than n. Therefore, the time complexity to find routes for the isolated nodes is $O(n^4 \log(n))$.

5. Fault-Tolerant Partitioning and Broadcasting Algorithm

From the fault-tolerant partitioning method-A, cubes are partitioned until all subcubes are fault-tolerant. Once a subcube is fault-tolerant, it is not again partitioned. Thus the size of the subcubes may differ and each subcube may be partitioned along different dimensions. These conditions make it difficult to establish simple and efficient communication between fault-tolerant subcubes. By eliminating these problems, a simpler and more efficient broadcasting strategy is presented in this section.

Again, we assume that connected n-cubes contain at most $(n^2 + 2n - c) / 4$ faulty links. Fault-tolerant partitioning method-B is described as follows. First, the n-cube is partitioned into two subcubes such that the number of faulty links in the major subcube is at most $((n - 1)^2 + 2(n - 1) - c) / 4$.

Then, the minor (n - 1)-subcube contains less than $(n^2 + 2n - c) / 8$ faulty links. All the $(n / (2^{0.5}))$-subcubes, which are obtained by partitioning the minor $(n - 1)$-subcube along any dimensions, obviously contain at most $(n^2 + 2n - c) / 8$ faulty links. Since $(((n / (2^{0.5}))^2 + 2(n / (2^{0.5})) - c) / 4)$ is always larger than $(n^2 + 2n - c) / 8$, where $n > 4$, the number of faulty links in each $(n / (2^{0.5}))$-subcube cannot be more than $(((n / (2^{0.5}))^2 + 2(n / (2^{0.5})) - c) / 4)$. Therefore, although the minor $(n - 1)$-subcube is partitioned along any dimension, all $(n / (2^{0.5}))$-subcubes contain a maximum of $(((n / (2^{0.5}))^2 + 2(n / (2^{0.5})) - c) / 4)$ faulty links.

The major $(n - 1)$-subcube is partitioned into two $(n - 2)$-subcubes such that the number of faulty links in the major $(n - 2)$-subcube is a maximum of $((n - 2)^2 + 2(n - 2) - c) / 4$. And the minor $(n - 1)$-subcube is also partitioned along the same dimension. Then, four $(n - 2)$-subcubes contain at most $((n - 2)^2 + 2(n - 2) - c) / 4$ faulty links.

In general, suppose that there are $2^{(n-k)}$ k-subcubes after the n-cube is partitioned $n - k$ times, where $k > n / (2^{0.5})$. Let the mm-cube be a k-subcube which is a major subcube obtained from other major subcubes. The mm-cube is partitioned into two $(k - 1)$-subcubes such that the number of faulty links in the major $(k - 1)$-subcube is at most $((k - 1)^2$

$+ 2(k - 1) - c) / 4$. And the other k-subcubes will also be partitioned along the same dimension.

All subcubes are partitioned continuously until each subcube becomes $(n / (2^{0.5}))$ dimensional. After partition is completed, there are $2^{(n- (n / (2^{0.5})))}$ $(n / (2^{0.5}))$-subcubes. All $2^{(n- (n / (2^{0.5})))}$ $(n / (2^{0.5}))$-subcubes contain a maximum of $(((n / (2^{0.5}))^2 + 2(n / (2^{0.5})) - c) / 4)$ faulty links.

All $(n / (2^{0.5}))$-subcubes are connected by $(n - (n / (2^{0.5})))$-cubes. Let *the comcube of an n-cube* be a healthy $(n - (n / (2^{0.5})))$-cube, each node of which is in each $(n / (2^{0.5}))$-subcube and the degree of each node in the comcube is greater than $n - 2$. There are $2^{(n / (2^{0.5}))}$ different $(n - (n / (2^{0.5})))$-cubes, each node of which is in each $(n / (2^{0.5}))$-subcube. The links in all $2^{(n / (2^{0.5}))}$ different $(n - (n / (2^{0.5})))$-cubes are disjointed. Since $2^{(n / (2^{0.5}))}$ is larger than $(n^2 + 2n - c) / 4$ for $n > 4$, there exists a comcube which connects all $(n / (2^{0.5}))$-subcubes. Therefore, communications between all $(n / 2^{0.5})$-subcubes are enabled from use of the comcube.

If some of the (n / $2^{0.5}$)-subcubes do not tolerate faulty links, then partitioning these subcubes will be completed by repeated performance of the same partitioning strategy.

For communication between fault-tolerant subcubes, it is possible to have more than a single comcube. Then, a comcube will be connected to another comcube by use of links. It is easy to verify that the number of links which are used to connect two comcubes is a maximum of 2.

The time complexity of fault-tolerant partitioning method-B is same as the time complexity of fault-tolerant partitioning method-A which is $O(n^4)$.

A strategy for finding the comcubes is described as follows. Suppose that some k-cubes are not fault-tolerant. After each k-cube is partitioned into (k / $2^{0.5}$)-subcubes, the partitioned dimensions are known. All of the faulty links in each k-cube will be sorted on non-partitioned dimensions by using the radix sort. In the worst case, the time complexity for sorting them is $O(n^3)$. If k is not n, there is a comcube, CC, which is used for communication between k-cubes. Each node of CC is in each k-cube. The time complexity for finding new comcubes at distance 2 from CC is $O(n^2)$. These steps are repeated $O(\log(n))$ times for the worst case. Therefore the time complexity for finding all comcubes is $O(n^3 \log(n))$.

The fault-tolerant broadcasting algorithm-B is similar to the fault-tolerant broadcasting algorithm-A. The only difference is that cubes and links are used for communication between fault-tolerant subcubes in place of the tree used for the fault-tolerant broadcasting algorithm-A.

Theorem 5.1: Given any n-cube in the presence of a maximum of $(n^2 + 2n - c) / 4$ faulty links, the fault-tolerant broadcasting algorithm-B is implemented in $2n + 8\log(n) + 9$ time units.

Proof: Let *ICC* be interconnected comcubes which are used for communication between all fault-tolerant subcubes of the n-cube. In the worst case, the broadcasting time is the sum of 1), 2), and 3), as follows:

1) is the possible largest diameter of ICC,

2) is the broadcasting time for two 4-cubes, and

3) is the broadcasting time for isolated nodes.

1) is less than $2(n - 4) - (n - (n / 2^{0.5})) + 2*2(\log(n^2)) = 1.7n + 8\log(n) - 8$,

and 2) is 14 time units in the worst case. In the worst case, 3) is $0.3n + 2$ time units as explained in section 4.

Therefore, the fault-tolerant broadcasting algorithm-B for the n-cube in the presence of a maximum of $(n^2 + 2n - c) / 4$ faulty links is implemented in $2n + 8\log(n) + 10$ time units.

6. Fault-Tolerant Partitioning and Broadcasting Algorithm

In section 5, the largest comcube for an n-cube is $n / 2^{0.5}$ dimensional. By finding larger comcube, a more efficient broadcasting algorithm can be developed. In this section, the most efficient broadcasting algorithm among the proposed algorithms is presented.

Let *the m-partition* of an n-cube represent a partition into (n - m)-subcubes such that all the subcubes are obtained by partitioning the n-cube along same dimensions.

Lemma 6.1: Given any n-cube in the presence of at most $(n^2 + 2n - c) / 4$ faulty links, there exists an m-partition such that each (n - m)-subcube contains a maximum of $((n - m)^2 + 2(n - m) - c) / 4$ faulty links, where $m \leq n - 3$.

Proof: Suppose that the claim is not true. Then, from each m-partition, there is at least one (n - m)-subcube that contains more than $((n - m)^2 + 2(n - m) - c) / 4$ faulty links. Note that there are $\binom{n}{m}$ different m-partitions. Let S be a set which contains an (n - m)-subcube in the presence of more than $((n - m)^2 + 2(n - m) - c) / 4$ faulty links from each m-partition.

By changing subcubes in S and faulty links in the n-cube, it will be shown that all (n - m)-subcubes containing the node(0) can contain more than $((n - m)^2 + 2(n - m) - c) / 4$ faulty links.

Transfer Rule

If $(d_{n-1}, \ldots, d_{i+1}, 1, d_{i-1}, \ldots, d_0)$ is in S, then

it is changed to $(d_{n-1}, \ldots, d_{i+1}, 0, d_{i-1}, \ldots, d_0)$.

if $(d_{n-1}, \ldots, d_{i+1}, 1, d_{i-1}, \ldots, d_0)$ is faulty

and if$(d_{n-1}, \ldots, d_{i+1}, 0, d_{i-1}, \ldots, d_0)$ is not faulty, then

they are switched.

The Transfer Rule is applied to all subcubes in S and all faulty links, one dimension at a time, for $0 \leq i \leq n - 1$. Then, all subcubes in S are transferred to subcubes which contain the node(0). When a subcube is transferred to another subcube, some of the faulty links in the former subcube are switched with healthy links in new subcube. It is easily verified that the number of faulty links in the new subcube will not be less than the number of faulty links in the former subcube, and the number of faulty links in the n-cube will not change. Therefore, all $(n - m)$-subcubes containing the node(0) can contain more than $((n - m)^2 + 2(n - m) - c) / 4$ faulty links.

However, it is known that an n-cube in the presence of $(n^2 + 2n - c) / 4$ faulty links can be partitioned into two subcubes each of which contains at most $((n - 1)^2 + 2(n - 1) - c) / 4$ faulty links. Thus, the transferred n-cube can be partitioned into two subcubes such that each subcube contains at most $((n - 1)^2 + 2(n - 1) - c) / 4$ faulty links. Also, an $(n - 1)$-subcube containing the node(0) can be partitioned into two $(n - 2)$-subcubes such that each $(n - 2)$-subcube contains a maximum of $((n - 2)^2 + 2(n - 2) - c) / 4$ faulty links. By continuously partitioning subcubes containing the node(0), an $(n - m)$-subcubes containing the node(0) and a maximum of $((n - m)^2 + 2(n - m) - c) / 4$ faulty links must be determined. This gives contradiction.

From Lemma 6.1, it is demonstrated that an n-cube in the presence of at most $(n^2 + 2n - c) / 4$ faulty links can be partitioned into m-subcubes such that all m-subcubes consist of same dimensional links and contain at most $(m^2 + 2m - c) / 4$ faulty links, where $m > 2$. Therefore, there is an $(n - 4)$-partition of an n-cube such that each 4-subcube contains at most 5 faulty links.

The time complexity for determining whether each 4-subcube contains at most five faulty links is $O(n^3)$. There are $(n(n - 1)(n - 2)(n - 3) / 24)$ different $(n - 4)$-partitions. Therefore, for the worst case, the time complexity for finding a 4-partition such that each 4-subcube contains at most 5 faulty links is $O(n^7)$.

The time complexity for finding communication comcubes between fault-tolerant subcubes is $\Omega(n^{2\log(n)})$ since the largest comcube is $2\log(n)$ dimensional. The following corollary demonstrates how the time complexity for partitioning cubes and finding comcubes is reduced.

Corollary 6.1: Let an n-cube contain at most $(n^2 + 2n - c) / 4$ faulty links. Let the transferred cube of an n-cube be the n-cube which is obtained by transferring faulty links in the n-cube by use of the Transfer

Rule in Theorem 3.1. Suppose that an (n - k)-partition is constructed on the transferred cube so that a k-subcube containing the node(0) contains at most $(k^2 + 2k - c) / 4$ faulty links. When the (n - k)-partition is constructed on the n-cube, all k-subcubes contains at most $(k^2 + 2k - c) / 4$ faulty links.

Proof: Again, suppose that the transferred cube of the n-cube is partitioned into k-subcubes such that the k-subcube containing the node(0) contains at most $(k^2 + 2k - c) / 4$ faulty links. And suppose that the k-partition is performed for the n-cube along the same dimensions.

Suppose that the claim is not true. Then, there exists an k-subcube of the n-cube which contains more than $(k^2 + 2k - c) / 4$ faulty links. When the faulty links in that k-subcube are transferred using the Transfer Rule, the k-subcube of new transferred cube containing the node(0) contains more than $(k^2 + 2k - c) / 4$ faulty links. It is then obvious that the k-subcube of older transferred cube containing the node(0) contains all faulty links in the k-subcube of the new transferred cube containing the node(0). Therefore, the k-subcube of the older transferred cube containing the node(0) must contain more than $(k^2 + 2k - c) / 4$ faulty links.

This gives contradiction.

The time complexity for transferring n^2 faulty links is $O(n^3)$. Based upon Corollary 6.1, the time complexity for constructing a k-partition for an n-cube in the presence of $(n^2 + 2n - c) / 4$ faulty links such that each k-cube contains at most $(k^2 + 2k - c) / 4$ faulty links is $O(n^4)$. Therefore, the time complexity for constructing an (n - 4)-partition for an n-cube in the presence of $(n^2 + 2n - c) / 4$ faulty links such that each 4-cube contains at most five faulty links is reduced to $O(n^4)$. Also, the time complexity for finding comcubes for communications between fault-tolerant subcubes is $O(n^4)$.

Fault-tolerant partitioning method-C is as follows. First, all faulty links in an n-cube are transferred by using the Transfer Rule. Then, the transferred cube is partitioned into (2log(n))-subcubes such that an (2log(n))-subcube containing the node(0) contains at most $((2\log(n))^2 + 2(2\log(n)) - c) / 4$ faulty links. Then, the n-cube is partitioned into (2log(n))-subcubes along the dimensions that the transferred cube has been partitioned. A comcube is found by sorting and counting the faulty links. If some of (2log(n))-subcubes are not faulty, then the same steps are repeated until all of the subcubes are fault-tolerant.

The fault-tolerant broadcasting algorithm-C is identical to the fault-tolerant broadcasting algorithm-B. If the fault-tolerant broadcasting algorithm-B is implemented with the fault-tolerant partitioning method-C, then the fault-tolerant broadcasting algorithm-B will be called the fault-tolerant broadcasting algorithm-C.

Thereom 6.1: Given any n-cube in the presence of at most $(n^2 + 2r - c) / 4$ faulty links, the fault-tolerant broadcasting algorithm-C is implemented in $1.3n + 6\log(n) + 9$ time units.

Proof: In the worst case, the broadcasting time is the sum of 1), 2), and 3). 1) is the possible largest diameter of ICC, which is less than $2(n - 4) - (n - 2\log(n)) + 4\log(n) = n + 6\log(n) - 8$. 2) is the broadcasting time for two 4-cubes. It takes 14 time units in the worst case. 3) is the broadcasting time for isolated nodes, which is $0.3n + 2$ time units in the worst case.

For an n-cube in the presence of at most $(n^2 + 2n - c) / 4$ faulty links, the fault-tolerant broadcasting algorithm-C is implemented in $1.3n + 6\log(n) + 9$ time units.

To simply the fault-tolerant partitioning method-C, the largest comcube for communications between fault-tolerant subcubes is always $2\log(n)$ dimensional. In practice, the broadcasting time can be reduced by finding a larger comcube than the $2\log(n)$ dimensional comcube.

7. Conclusion

Three partitioning and broadcasting algorithms were proposed for n-cubes in the presence of up to $(n^2 + 2n - c) / 4$ faulty links.

The most efficient broadcasting algorithm is implemented in $1.3n + 6\log(n) + 9$ time units. The time complexity to partition an n-cube into fault-tolerant subcubes is $O(n^4)$.

The concept of partitioning and broadcasting strategies for n-cubes in the presence of faulty links could be extendable to partitioning and broadcasting algorithms for n-cubes in the presence of faulty nodes. Because an n-cube in the presence of a maximum of $(n^2 + 2n + z) / 4$ faulty nodes, where $z = 0$ or 1, if n is an even or odd number respectively, can be partitioned into 3-cubes such that each 3-cube contains at most four faulty nodes.

One open problem is to find a more efficient broadcasting algorithm for hypercubes in the presence of more than $n - 1$ faults.

References

[1] Y. Saad and M. H. Schultz, "Topological Properties of Hypercubes," *IEEE Trans. Comput.*, vol. 37, no. 7, pages 867 - 872, July 1988.

[2] S. Johnsson and C. T. Ho, "Optimum Broadcasting and Personalized Communication in Hypercubes," *IEEE Trans. Comput.*, vol. 38, no. 9, pages 1249 - 1268, Sept. 1989.

[3] J. Bruck, R. Cypher and D. Soroker, "Tolerating Faults in Hypercubes Using Subcube Partitioning," *IEEE Trans. Comput.*, vol. 41, no. 5, pages 599 - 604, May 1992.

[4] S. Park, B. Bose, "Broadcasting in Hypercubes with Links/Node Failures," *4th Symp. on the Frontiers of Massively Parallel Computation*, pages 286 - 290, 1992.

[5] T. C. Lee and J. P. Hayes, "Routing and Broadcasting in Faulty Hypercube Computers," In *Proc. 3rd Conf. on Hypercube Concurrent Computers and*

Applications., pages 625 - 630, 1988.

[6] B. S. Chlebus, K. Diks, and A. Pelc, "Optimal Broadcasting in Faulty Hypercubes," *Proc. 21th Inter. Symp. on Fault-Tolerant Computing*, pages 266 - 273, 1991.

[7] J. Bruck, R. Cypher and D. Soroker, "Runing Algorithms Efficiently on Faulty Hypercubes," In *Pro. 2nd Annu. ACM Symp. Parallel Algorithms Architectures*, pages 37 - 44, 1990.

[8] P. Ramanathan and K. G. Shin, "Reliable Broadcasting in Hypercube Multicomputers," *IEEE Trans. Comput.*, vol. 37, no. 12, pages 1654 - 1657, Dec. 1988.

[9] M. S. Chen and K. G. Shin, "Adaptive Fault-Tolerant Routing in Hypercube Multicomputers," *IEEE Trans. Comput.*, vol. 39, no. 12, pages 1406 - 1416, Dec. 1990.

[10] M. S. Chen and K. G. Shin, "Depth-First Search Approach for Fault-

Tolerant Routing in Hypercube Multicomputers," *IEEE Trans. Comput.*, vol. 1, no. 2, pages 152 - 159, Apr. 1990.