Technical Note

A Method of Fault-Tolerant All-to-All Personalized Communication in Hypercubes

Hiroshi Masuyama† and Etsuko Masuyama††

In this paper, for n-dimensional faulty hypercubes, we propose an all-to-all personalized communication algorithm obtained by extending an all-to-all broadcasting algorithm fault-tolerant in the presence of up to $\lfloor n/2 \rfloor$ faulty nodes. The proposed algorithm assumes a multiport communication and an availability of global fault information.

1. Introduction

For our experiment, we treat an all-to-all personalized communication algorithm for faulty hypercubes. Hypercube networks offer a cost-effective and practical approach to supercomputing by connecting a large number of processors using direct links. As a result the hypercubes have been studied extensively as an SIMD machine, and this has led to numerous experimental and commercial machines including the recent development of a system with more than 6,000 nodes by NCUBE [1].

Many algorithms have been proposed for hypercubes, most of them concentrating on routing, one-to-all, or all-to-all broadcasting. All-to-all personalized communication is also one of the most dense collective communication patterns and it occurs in many important applications in parallel computing. There have been many studies conducted for all-to-all personalized communication in various networks [2),3)]. Previous all-to-all personalized communication algorithms for hypercubes were developed for nonfaulty networks. Johnsson and Ho [2] proposed optimal all-to-all personalized communication algorithms on a nonfaulty n-dimensional hypercube with $O(2^n)$ time complexity (that is, required total amount of time units) for an all-port model. The algorithms given by Suh and Shin [3] have time complexity $O(N^{(k+1)/k})$ for a k-dimensional mesh and torus with N nodes. Since, the algorithms for a nonfaulty hypercube may achieve optimal time complexity, then the hypercubes are superior to mesh/torus networks which achieve higher time complexity. How about in the faulty networks?

Since there is a factor of 2^n difference between one-to-all personalized communication ($O(n)$) and all-to-all personalized communication ($O(2^n)$), inefficient algorithms may result in a very poor system performance. In this paper, we introduce a fault-tolerant all-to-all personalized communication algorithm for n-dimensional hypercube networks in the presence of up to $\lfloor n/2 \rfloor$ faulty nodes.

† Information and Knowledge Engineering, Tottori University
†† Health Science, Hiroshima Prefectural Women's University

2. Preliminaries

Assume that an n-dimensional hypercube Q_n has $N = 2^n$ nodes. The nodes are indexed 0 to $N-1$. Each index number $i(0\ leq\ i \leq N-1)$ is represented as $i_{n-1}\ i_{n-2} \cdots i_1\ i_0$ in the binary system. Let $i^{(j)}$ denote the number whose binary representation is $i_{n-1} \cdots i_{j+1}\ \bar{i_j}\ i_{j-1} \cdots i_0$ where $0 \leq j \leq n-1$. In a hypercube, node i is connected to node $i^{(j)}$, and they are called adjacent to each other. **Figure 1** (1-0) shows an N=16 hypercube network with 2 faulty nodes. In this paper, we assume that Q_n has n dimensions which are called as the 0-th, the 1-st, the 2-nd, $\cdots$, the $(n-1)$-th dimensions, respectively.

If a Q_n is divided along k dimensions $d_1, d_2, \cdots, d_k$, then there will be 2^k subcubes of size Q_{n-k}. A partner set (PS) denotes a set of nodes obtained by giving the same value for all $i_j \in \{i_{n-1}, i_{n-1}, \cdots, i_i, i_0\} - \{i_{d_1}, i_{d_2}, \cdots, i_{d_k}\}$ in $i_{n-1}\ i_{n-2} \cdots i_1\ i_0$. There are 2^{n-k} PSs, each of which contains 2^k nodes, and each PS forms a k-dimensional cube, that is, Q_k. Corresponding nodes in 2 l-dimensional cube Q_ls are a pair of nodes adjacent to each other along the dimension by which $2Q_l$s can be divided. Node i is the corresponding node to node j along dimension d, i f they are corresponding nodes and differ in the d-th bit.

(Example 1) If Q_4 is divided along 2 dimensions, 0-th and 2-nd, then there are 2^2Q_2, {0,1,4,5}, {2,3,6,7}, {8,9,12,13}, and {10,11,14,15}. Nodes 0 and 8 can be corre-

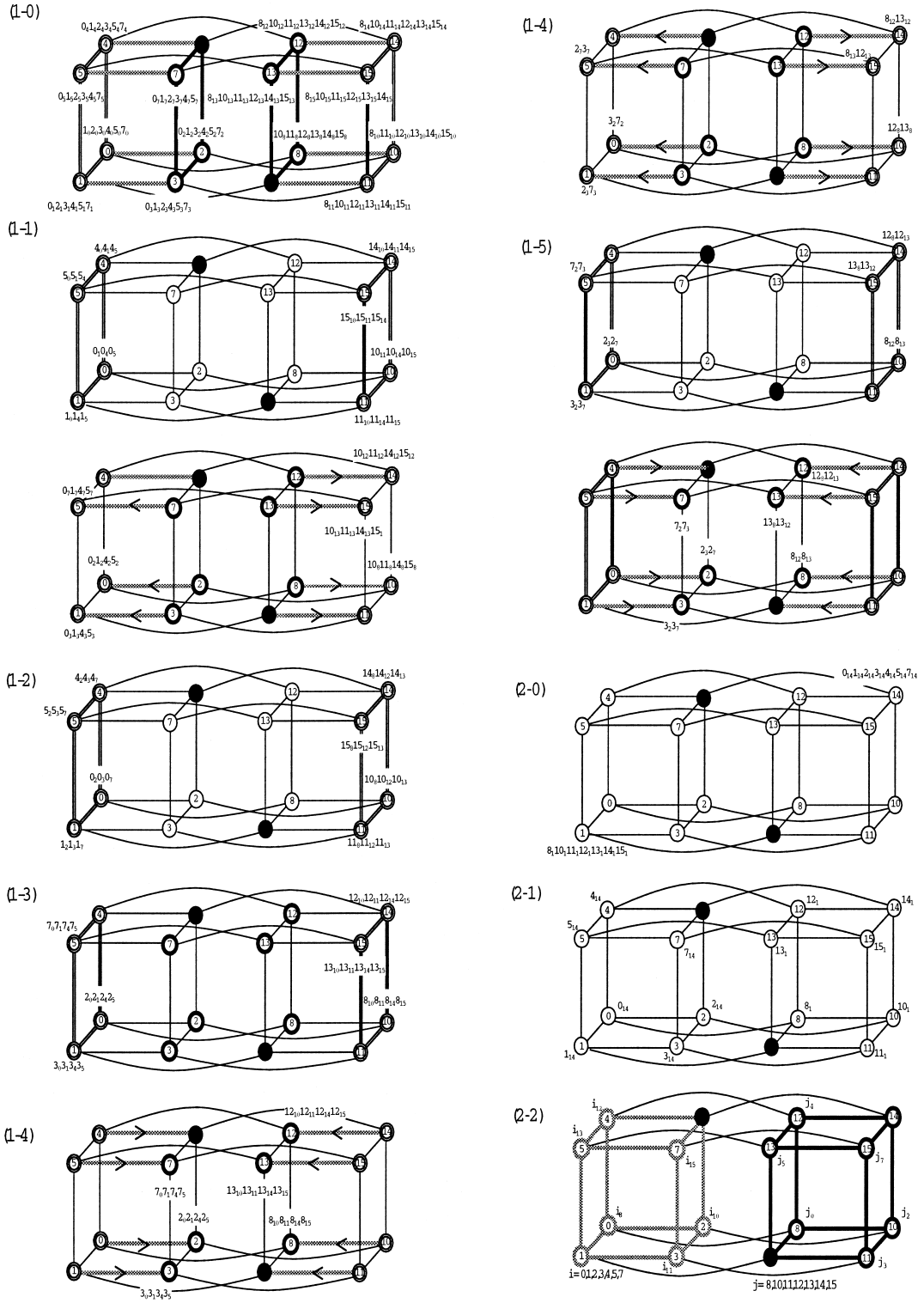

Fig. 1 AAPC algorithm applied to a 4-dimensional hypercube.

sponding nodes in two Q_3s.

A faulty (nonfaulty) subcube is one that contains some (no) faulty nodes.

The following 3 properties are inherent in the hypercubes and are given by S. Park and B. Bose [4]. Property 1 gives the f dimensions along which Q_n needs to be partitioned to have at most one faulty node in each PS.

[Property 1] A Q_n containing $f \leq \lfloor n/2 \rfloor$ faulty nodes can be partitioned into $(n-f+1)$-dimensional subcubes so that each PS contains at most one faulty node.

If the Q_n is partitioned along one more dimension that is not part of the distinguishing bits of the faulty node address bits, each PS still contains at most one faulty node. Properties 2 and 3 show how each of these faulty PS is coupled with a nonfaulty PS.

[Property 2] Let the number of faulty nodes in a Q_n be $f \leq n$. Then there exists a dimension d such that if Q_n is divided along it, every faulty node has a unique nonfaulty adjacent node along d.

[Property 3] If a Q_n that contains $f \leq \lfloor n/2 \rfloor$ faulty nodes is divided along any f dimensions, then each faulty PS has a unique perfect PS adjacent to it.

By the guarantee of the above 2 properties, we can construct $2^{n-(f+1)}Q_{f+1}$s where $2^{(f+1)}Q_{f+1}$s are faulty and the remainders are nonfaulty. Since the above properties place a restriction only on the number of faulty nodes, the numbering to each node is free from these properties.

3. Personalized Communication Algorithm and Required Time Unit

In this section, we will develop the communication algorithms for single and double faults, and establish a communication algorithm for any m-multi fault.

3.1.1 Algorithm A_1 for a Single Node Fault

Q_{n-1} : An $(n-1)$-dimensional nonfaulty subcube which is partitioned along a dimension and has no faulty node.

Q^f_{n-1} : The other subcube which is partitioned along the dimension of Q_n.

Step 1: For data whose sources and destinations are in Q_{n-1}, perform all-to-all personalized communication (hereafter referred as AAPC) in Q_{n-1} in order that these data arrive at their destinations. Simultaneously, move data whose sources and destinations are in Q^f_{n-1} and Q_{n-1}, respectively, to their neighboring nodes in Q_{n-1}.

Step 2: For data whose sources and destinations are in Q^f_{n-1} and Q_{n-1}, respectively, perform AAPC in Q_{n-1} in order that these data arrive at their destinations.

Step 3: For data whose sources and destinations are in Q_{n-1} and Q^f_{n-1}, respectively, perform AAPC in Q_{n-1} in order that data arrive at neighboring nodes of their destinations.

Step 4: Move data that AAPC is performed in **Step 3** to their destinations.
Simultaneously, move data whose sources and destinations are in Q^f_{n-1} to their neighboring nodes in Q_{n-1}.

Step 5: For data moved from the neighboring nodes in Q^f_{n-1}, perform AAPC in Q_{n-1} in order that these data arrive at neighboring nodes of their destinations. Simultaneously, move the data to their destinations in the order of when they arrive at neighboring nodes of their destinations.

For example, in 3-dimensional hypercube applied this algorithm when Q_3 is divided along the 2-nd dimension, data which move along link between nodes 0 and 4 are 0_4, 1_4, 2_4, and 3_4 in Step 1, 4_0, 4_1, 4_2, and 4_3 in Step 4, 5_4 and 7_4 in Step 4, 4_5 and 4_7 in Step 5 [5], where mark i_j means a datum whose original source and destination are nodes j and i, respectively.

3.1.2 Required Time Units

Note that an optimal AAPC algorithm for a nonfaulty Q_n requires $2^n - 1$ time units. Then, the amount of time units required for A_1 in the case of a single fault is the sum of 2^{n-1} in Step 1, $2^{n-1}-1$ in Step 2, $2^{n-1}-1$ in Step 3, 2^{n-1} in Step 4, and 2^{n-1} in Step 5, then total $5 \cdot 2^{n-1}-2$.

3.2.1 Algorithm A_2 for a Double Node Fault

Q_n is divided along $f = 2$ dimensions where f faulty PSs have each corresponding disjoint nonfaulty PS of which nonfaulty Q_f is composed. Let two faulty Q_{f+1} which are composed of Q_f and the faulty PS, each, be Q^1_{f+1} and Q^2_{f+1}.

Q^1_{f+1} and Q^2_{f+1} have a single fault, each. Let two faulty disjoint Q_{n-1}s which contain Q^1_{f+1} and Q^2_{f+1} be Q^1_{n-1} and Q^2_{n-1}, respectively. A_2 consists of two phases.

(Phase 1) Using algorithm A_1, perform AAPC each for Q^1_{n-1} and Q^2_{n-1}.

(Phase 2) Let the dimension by which Q^1_{n-1} and Q^2_{n-1} are adjacent be d. Let two nodes whose adjacent nodes along d are faulty nodes be n_1 in Q^1_{n-1} and n_2 in Q^2_{n-1}.

Step 1: For data whose sources are n_1 and n_2, distribute to all nonfaulty nodes in Q_n.

Step 2: In dimension d, exchange data whose sources and destinations are in Q^1_{n-1} and Q^2_{n-1}, respectively, to data whose sources and destinations are in Q^2_{n-1} and Q^1_{n-1}, respectively, but except the data treated in **Step 1**.
Simultaneously, perform AAPC in each Q^1_{n-1} and Q^2_{n-1} in the order of when the data is exchanged and arrived at each Q^1_{n-1} and Q^2_{n-1}.

(Example 2) Figure 1 shows an example of 4-dimensional hypercube applied this algorithm where Q_4 is divided along the 0-th and 2-nd dimensions. On and after the first step, only data which is moved and arrived at each node in this step is marked to the node, where mark i_j means a datum whose original source and destination are nodes j and i, respectively. d = 3, $n_1 = 1$, and $n_2 = 14$ in Phase 2.

3.2.2 Required Time Units

Note that total $5 \cdot 2^{n-1} - 2$ time units are given above to algorithm A_1 in the case of a single fault. Then, the amount of time units required in the case of double fault is the sum of $5 \cdot 2^{n-1} + n - 1$†

3.3.1 Algorithm A_f for an f-multi Node Fault

Let us generalize A_2 into the case of f faults. Q_n is divided along f dimensions where f faulty PSs have each disjoint nonfaulty PS adjacent to themselves. Let each $(f+1)$-dimensional subcubes be $Q^j_{f+1} (j = 1, 2, \cdots, 2^{n-(f+1)})$. f subcubes of these Q^j_{f+1}s have a single fault each which has a unique nonfaulty adjacent node in each different nonfaulty Q^j_{f+1}.

(Phase 1) Using algorithm A_1, perform AAPC each for f faulty Q^j_{f+1}s. Simultaneously, perform AAPC each for the rest of the nonfaulty Q^j_{f+1}s.

(Phase 2) Data exchange among all Q^j_{f+1}s are, first, performed, and then AAPC is performed in each Q^j_{f+1}.

Step 1: For data whose sources are nodes corresponding of faulty nodes (such as n_1 and n_2 in Phase 2 for a double fault), distribute to all nonfaulty nodes in Q_n.

Step 2: Exchange data among all Q^j_{f+1}s, but except the data treated in **Step 1**.
Simultaneously, perform AAPC in each Q^j_{f+1} in the order of when the data is exchanged and arrived.

3.3.2 Required Time Units

The time units required in Phase 1 is $5 \cdot 2^f - 2$. Since the time units required in Step 2 of Phase 2 is $(5 \cdot 2^f - 2) \cdot (2^{n-(f+1)} - 1)$, then the amount of time units required for A_f in the case of an f-multi fault is $(5 \cdot 2^f - 2) \cdot 2^{n-(f+1)} = 2^{n-1}(5 - 2^{1-f})$, that is about $5 \cdot 2^{n-1}$, where the time units required in Step 1 of Phase 2 ($= n$ in the case of a double fault) is ignored.

4. Conclusion

In this paper, we have presented an all-to-all personalized communication algorithm for faulty hypercubes. Though the algorithm proposed for Banyan networks has a 2^n time complexity, $3 \cdot 2^n$ time complexity can be evaluated at faulty Banyan networks [6]. A hypercube network might be a better choice for implementing all-to-all personalized communication due to its shorter communication latency.

References

1) Duzett, B. and Buck, R.: An Overview of the nCUBE3 Supercomputer, *Proc. Fourth Symp. Frontiers of Massively Parallel computation*, pp.458–464 (1992).
2) Gaughan, P.T. and Yalamanchili, S.: Adaptive routing protocols for hypercube interconnection networks, *Computer*, Vol.26, No.5, pp.12–23 (May 1993).
3) Suh, Y.J. and Shin, K.G.: Efficient All-to-All Personalized Exchange in Multidimensional Torus Networks, *Proc.1998 International Conference on Parallel Processing*, pp.468–475 (Aug. 1998).
4) Park, S. and Bose, B.: All-to-All Broadcasting in Faulty Hypercubes, *IEEE Trans. Comput.*, Vol.46, No.7, pp.749–755 (July 1997).
5) Masuyama, H., Kawamura, T. and Masuyama, E.: An All-to-All Personalized Communication Algorithm for Faulty Hypercubes, *Proc. SPECTS 2001*, pp.89–96 (July 2001).
6) Yaku, M. and Masuyama, H.: A Method of Fault-Tolerant All-to-All Personalized Communication in Banyan Networks, *IPSJ Journal*, Vol.42, No.10, pp.2476–2484 (Oct. 2001).

(Received December 12, 2002)
(Accepted March 4, 2003)

† Since we are discussing on a systematic manner, we will ignore the cunning idea by which somewhat of the total amount can be decreased.