Vol. 44 No. 5 IPSJ Journal May 2003

Technical Note

A Revised Fault-Tolerant Routing Algorithm for Faulty Hypercubes

Hiroshi Masuyama† and Toshiki Kawasaki††

A variety of routing algorithms have been proposed for hypercube networks. Recently, several routing algorithms have introduced the concept related to unsafe nodes and its extension, and achieved a successful routing. This paper focuses on one-to-one routing and shows that a considerable number of unexplored shortest paths can be found by introducing a preliminary knowledge into the routing.

1. Introduction

Hypercube architecture is one of the most widely used interconnection topologies in parallel and distributed processing due to its topological properties such as regularity, fault tolerance, or multitask capability [1)]. Therefore, much interest has been paid on hypercube. As the size of a system grows, the failing probability of some processors in the system increases. The condition that every node identifies the states of all nodes in the system leads to an optimal routing, but this is not realistic because of time complexities. One of the most important issues is to construct a fault-tolerant routing algorithm which tolerates faults while still being adaptable in the presence of faults. A variety of fault-tolerant routing strategies have been proposed. Random routing selection technique results in excessive message delay or rare routing failure [2),3)]. Depth-first search approach brings on an extra time overhead [4)]. In order to make these drawbacks light, the concepts of safety level and unsafe nodes [4)] are introduced for broadcasting and one-to-one routing, respectively. This paper first introduces a developed routing algorithm which adopts the concept of unsafe nodes and its extension, and next the revised algorithm which adopts a safe node with regard to distance. We will show that these algorithms are in danger of failing to take the shortest paths, so recommend adding the preliminary knowledge.

Hypercube network: Assume that $N = 2^n$. Let $i_{n-i}i_{n-2}\cdots i_1 i_0$ be the number of $i(0 \le i \le N-1)$ in the binary system, and let $i^{(j)}$ denote the number in binary system $i_{n-i}i_{n-2}\cdots i_{j+1}\overline{i_j}\cdots i_1 i_0$ where $0 \le j \le n-1$. In a hypercube network, computer i is connected to computer $i^{(j)}$, this is, nodes i and $i^{(j)}$ are adjacent.

† Information and Knowledge Engineering, Tottori University
†† Graduate School, Tottori University

2. Traditional Approaches

Consider a node, say i which currently has a message destined for a node t, and the Hamming distance is $H(i,t)$. The shortest distance between i and t is $H(i,t)$. Let two sets $N(i)$ and $D(i,t)$ be the set of neighbor nodes of the node i and the subset of neighbor nodes of i which are closer to the target node t than the node i, respectively, as follows:

$$N(i) = \{j | H(i,j) = 1\}$$

$$D(i,t) = \{j | H(j,t) = H(i,t) - 1, j \in N(i)\}$$

Definition 1 *A fault-free node is defined as an* **unsafe** *node if it has either two or more faulty adjacent nodes, or three or more (faulty or unsafe) adjacent nodes. A non-faulty node that is not unsafe is called a* **safe** *node.*

Theorem 1 *In a faulty hypercube, if node i is a safe node, for any non-faulty node j, there always exists a feasible minimum path of length equal to $H(i,j)$ between i and j.*

Definition 2 *An unsafe node is defined as* **strongly unsafe** *if none of its adjacent nodes is safe. An unsafe, but not strongly unsafe, node is called an* **ordinarily unsafe** *node.*

Theorem 2 *In a faulty hypercube, if node i is an ordinarily unsafe source node, for any unsafe terminal node j, there always exists a path of length no greater than two plus $H(i,j)$.*

Chiu and Wu [5)] developed the following routing algorithm based on the above definitions and theorems given by them, where s and t are source and terminal nodes, respectively.

Algorithm ROUTE(s,t):

begin
  $l := H(s,t)$; $N := N(s)$; $D := D(s,t)$; $S :=$ a set of safe nodes; $O :=$ a set of ordinarily unsafe nodes; $U :=$ a set of strongly unsafe nodes;
  if $l = 0$ then deliver the message to s and exit
  else if $\exists j \in D \cap S$ then take node j
  else if $\exists j \in D \cap O$ then take node j
  else if $\exists j \in D \cap U$ and ($s \in U$ or $l \leq 2$) then take node j
  else if $\exists j \in (N - D) \cap S$ then take node j
  else if $\exists j \in (N - D) \cap O$ then take node j
  else there exist no route
  ROUTE (j, t)
end

Definition 3 *Every non-faulty node is safe in connection with distance* 1. *A non-faulty node is defined as* **safe in connection with distance** l *if it has* $(n - l + 1)$ *or more adjacent nodes which are safe in connection with distance* $l - 1$.

This safe node in connection with distance l is the node which can be selected to route shortly. Kaneko and Ito [6),7)] presented a revised algorithm FR(s,t) by using the concept "safe in connection with distance l" defined above and by inserting the following statement between the 1^{st} and 2^{nd} "else if" statements in algorithm ROUTE(s,t):

  else if $l \leq k + 1$ and $\exists j \in D \cap S_{l-1}$ then take j

Where k is an integer which is greater than 2 but under n, and S_{l-1} is a set of safe nodes in connection with distance $l - 1$. It is reported that, by using FR(s,t) instead of ROUTE(s,t), 16–63% in total are saved as message paths in systems of scale $n = 4$–6.

3. Substitute Approach

Figure 1 shows a 4-dimensional hypercube with 4 faulty nodes where nodes 2 and 9 are source and terminal nodes, respectively. In Fig. 1, 2 paths (the shortest and non-shortest) are drawn where the former can't be obtained with algorithm ROUTE(s,t) or FR(s,t). **Figure 2** shows a 5-dimensional hypercube with 7 faulty nodes. In this figure, the shortest and not-shortest paths are also drawn, and it is proved that the former can't be obtained with algorithm ROUTE(s,t) or FR(s,t). These deserted shortest paths can be found with the following concept: If faulty nodes don't stand in the shortest way and so at least one shortest path is alive, it must be on the smallest sub-cube which contains two nodes s and t. If algorithm ROUTE(s,t) or FR(s,t) is applied only to the smallest sub-cube, since a certain number of nodes have a probability of changing from unsafe, ordinary, or strongly to more safety, the shortest path may be obtained. **Figures 3** (a) and (b) show the smallest sub-cubes which contain two node pairs $(2, 9)$ and $(2, 31)$, respectively. We can see that some strongly and ordinarily unsafe nodes changed into safe nodes. Apply ROUTE(s,t) or FR(s,t) to these sub-cubes, then the shortest paths are easily obtained. Therefore, new revised algorithm is represented as follows:

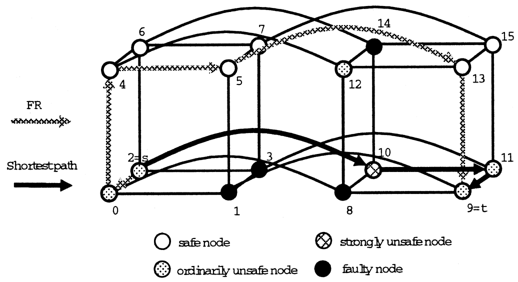


Fig. 1 Shortest and non-shortest paths in 4-dimensional hypercube with 4 faulty nodes.

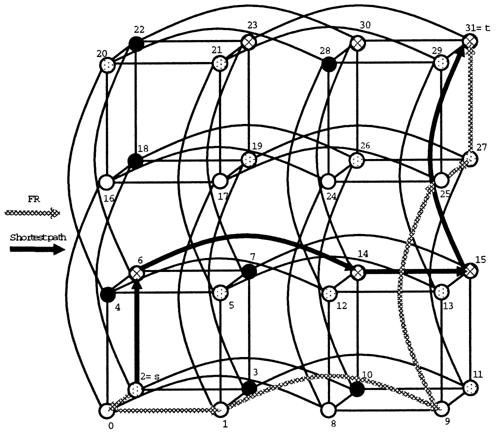
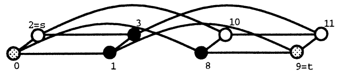


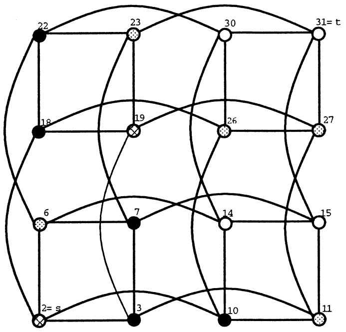
Fig. 2 Shortest and non-shortest paths in 5-dimensional hypercube with 7 faulty nodes.

Algorithm New ROUTE(s,t):
 begin
  ROUTE(s,t)/FR(s,t) for the smallest sub-cube which contains s and t
  if there exist no route, then ROUTE(s,t)/ FR(s,t) for the fully hypercube
 end

It is remarkable that 6 unexplored shortest paths in a 4-dimensional hypercube with 4



(a) Subcube of Fig. 1.



(b) Subcube of Fig. 2.

Fig. 3 Two smallest sub-cubes which contain source and terminal nodes.

faulty nodes can be saved by New ROUTE(s,t). Caution that s and t are fixed as 2 and 9, respectively, and $H(2,9) = 3$. From the property of hypercube, we can conclude that, if $H(s,t) = 3$ in a n-dimensional hypercube, then there exist 6 patterns of n-faults distribution which can be saved by New ROUTE(s,t). That is, in a 5-dimensional hypercube with 5 faulty nodes, New ROUTE$(2,30)$ can save 6 unexplored shortest paths. It is also verified, in a 5-dimensional hypercube with 7 faulty nodes, that New ROUTE$(2,31)$ can also save 24 unexplored shortest paths. Therefore, we can easily surmise that there are many fault patterns which can be saved by New ROUTE(s,t).

4. Conclusion

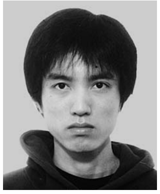
A revised fault-tolerant routing algorithm is proposed, and is based on the concept of unsafe node. In traditional approaches, a set of criteria is presented to identify the unsafe nodes that may cause routing difficulty. This paper showed that the stringency of the criteria reduces the number of unsafe nodes which are not only undesirable but also desirable in the routing, and presented the way to save the desirable nodes from forsaking. The time complexity required in New ROUTE(s,t) is the same in order as the one of ROUTE(s,t)/FR(s,t), because the extra procedure is necessary only for a sub-cube of given n-cube. Though this paper has treated the shortest paths with the Hamming distance, other unexplored shortest paths with longer distance than the Hamming distance may exist. These unexplored shortest paths can be found by treating not the smallest but the sub cube which contains two nodes s and t. However, this approach brings a great deal of time complexity.

References

1) Saad, Y. and Schultz, M.H.: Topological Properties of Hypercubes, *IEEE Trans. Comput.*, Vol.37, No.7, pp.867–872 (1988).
2) Gordon, J.M. and Stout, Q.F.: Hypercube Message Routing in the Presence of Faults, *Proc. Third Conf. on Hypercube Concurrent Computers and applications*, Vol.1, pp.318–327 (1988).
3) Gaughan, P.T. and Yalamanchili, S.: Adaptive Routing Protocols for Hypercube Interconnection Networks, *Computer*, Vol.26, No.5, pp.12–23 (1993).
4) Chen, M.S. and Shin, K.G.: Depth-First Search Approach for Fault-Tolerant Routing in Hypercube Multicomputers, *IEEE Trans. Parallel and Distributed Systems*, Vol.1, No.2, pp.152–159 (1990).
5) Chiu, G.M. and Wu, S.P.: A Fault-Tolerant Routing Strategy in Hypercube Multicomputers, *IEEE Trans.*, Vol.45, No.2, pp.143–155 (1996).
6) Kaneko, K. and Ito, H.: A Fault-Tolerant Routing Algorithm for Hypercube Systems Based on Full Reachability, *Trans. IEICE*, Vol.J81-D-I, No.8, pp.1024–1030 (1998).
7) Kaneko, K. and Ito, H.: A Fault-Tolerant Routing Algorithms for Hypercube Networks, *Proc. IPPS/SPDP 1999*, pp.218–224 (1999).

(Received November 27, 2002)
(Accepted February 4, 2003)

Hiroshi Masuyama was born in Japan on Sep.17, 1943. He is now a professor in the Department of Knowledge and Information Engineering of Tottori University. He has been a professor in Miyazaki and Osaka universities. He was also a visiting professor at Stanford Univ. between 1990 and 1991, and Boston Univ. in 1996. While at Hiroshima, Miyazaki, and Tottori Universities, Dr. Masuyama published papers on topics including fault tolerance of logical circuits, analysis and synthesis of parallel and distributed systems, and network algorithms.



Toshiki Kawasaki was born in 1979. He graduated from the Department of Information and Knowledge Engineering, Faculty of Engineering, Tottori University in 2002. He is now a graduate student of Tottori University. He is making a study of Hypercube networks.