Fault-Tolerant Routing Algorithms for Hypercube Networks

Keiichi Kaneko Hideo Ito
Faculty of Engineering, Chiba University
Yayoi-cho 1–33, Inage-ku, Chiba-shi, Chiba 263-8522, Japan
kaneko@ics.tj.chiba-u.ac.jp

Abstract

For hypercube networks which have faulty nodes, a few efficient dynamic routing algorithms have been proposed by allowing each node to hold the status of neighbors. We propose two improved versions of the algorithm of Chiu and Wu by using the notion of full reachability. A fully reachable node means that the node can reach all nonfaulty nodes which have Hamming distance h from the node via a path of length h. The simulation shows that the algorithms give sufficient effect when they are applied to low-dimensional hypercubes.

1. Introduction

Recently, interests in massively parallel processing are spreading rapidly. However, as the number of processors augments, the probability of occurrences of faulty nodes also increases. Hence, it is relevant to construct communication paths which detour faulty processors. We adopt hypercube networks [5, 9, 8] as the target system on which there are many studies due to their good properties such as symmetric and regular structure and relatively small diameters [1, 6, 10]. We focus on a dynamic routing scheme on interprocessor message communications in a faulty hypercube (a hypercube network which has faulty nodes) to suppress degradation of performance.

In a parallel processing system which has faulty processors, it is very important to select a shortest path to establish an interprocess communication. If every processor in the system identifies the status of all processors, an optimal routing is possible. However, because of restrictions of space and time complexities, it is very hard to adopt this approach. For hypercube networks, several efficient dynamic routing algorithms have been proposed by allowing each node to hold the status information of neighbor nodes [4, 3, 7, 11]. One of the algorithms we propose is based on Chiu and Wu's one [4] and it uses the notion of full reachability to improve performance. In addition, we introduce the notion of unsafe nodes with respect to distance to improve the algorithm. Finally, we conduct simulation to measure the improvement ratio and to compare our algorithms with an algorithm by Chiu and Chen [3] which uses the equivalent notion to our full reachability.

2. Routing Algorithms

2.1. Hypercube Networks

Definition 1 (d-dimensional hypercubes or d-cubes) *A d-dimensional hypercube network consists of 2^d nodes whose addresses are represented d-bit binary numbers. A node n has d adjacent nodes whose addresses are obtained by reverting each bit of its address.*

Let $H(n_1, n_2)$ give the Hamming distance between two nodes n_1 and n_2 in a hypercube, then the length of the shortest path between two nodes n_1 and n_2 is equal to $H(n_1, n_2)$ if there is no faulty node.

Consider a message delivery from a source node s to a target one t. First, let $N(s)(= \{n \mid H(s,n) = 1\})$ represent the set of neighbor nodes of the node s. Next, let $D(s,t)(= \{n \mid H(n,t) = H(s,t) - 1, n \in N(s)\})$ represent the subset of neighbor nodes of s which are closer to the target node t than the node s. Then, the equation $|D(s,t)| = H(s,t)$ holds and we can send the message with the address of target node t to any node c in $D(s,t)$. Now, let us consider the node c as a new source one, and repeat the process above until the message reaches the target node.

2.2. Routing Problem

In a faulty hypercube, it is necessary for message delivery to find a path of nonfaulty nodes from the source to the target. For this purpose, each node can store some information about neighbors to select one of them for message sending by using it with the address of the target. In addition, a detour must be detected even if any shortest path can not be found by using the information. An algorithm to perform these operations is called a routing algorithm. In this situation, a good algorithm finds as many shortest paths as possible while it holds as simple information as possible.

Definition 2 (reachability and communicatability) *If there exists a path from the source node s to the target one t which includes no faulty nodes, t is said to be reachable from s. If a routing algorithm R finds the path, t is said to be communicatable from s by R.*

2.3. Algorithm by Chiu and Wu

In this section, we describe the algorithm proposed by Chiu and Wu [4] called as `route` in this paper. The algorithm first divides the nonfaulty nodes in a hypercube network into safe and unsafe nodes. It then classifies the unsafe nodes into ordinary and strongly unsafe nodes. Here are their main definitions, one is recursive, and theorems followed by the algorithm.

Definition 3 (safe and unsafe nodes) *A nonfaulty node n is unsafe if it is adjacent to two or more faulty nodes or it is adjacent to more than two faulty or unsafe nodes. A nonfaulty node is safe if it is not unsafe.*

Definition 4 (strongly and ordinary unsafe nodes) *An unsafe node n is strongly unsafe if every neighbor node of n is either unsafe or faulty. An unsafe node n is ordinary unsafe if it is not a strongly unsafe node.*

Figure 1 shows an example of node classification based on the definitions above. Moreover, we define full unsafeness as a property of hypercube networks.

Definition 5 (fully unsafe networks) *A hypercube network is fully unsafe if every nonfaulty node in the network is unsafe under a set of faulty nodes.*

Hereafter, let F, S, U, $\bar{U}$ and $\tilde{U}$ represent the set of faulty, safe, unsafe, ordinary unsafe and strongly unsafe nodes, respectively. Figure 2 shows the algorithm by Chiu and Wu. It is invoked with the addresses of the source and target nodes, initially. The following theorems hold due to properties of hypercube networks [4].

Theorem 1 *If the source $s \in S$ or the target $t \in S$, the algorithm `route` can communicate by one of the shortest paths of length $H(s,t)$.*

Theorem 2 *If the source $s \in \bar{U}$ and the target $t \in U$, the algorithm `route` can communicate by a path whose length is at most $H(s,t) + 2$.*

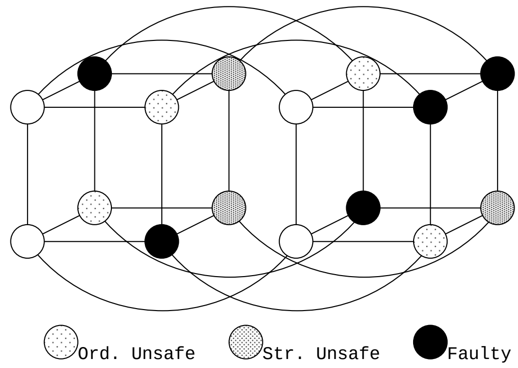

Figure 1. Node classification by Chiu and Wu.

```
procedure route(c, t)
begin
 h := H(c,t); N := N(c); D := D(c,t);
 if h = 0 then
  deliver the message to c and exit
 else if ∃n ∈ D ∩ S then nxt := n
 (* for future replacement *)
 else if ∃n ∈ D ∩ Ū then nxt := n
 else if ∃n ∈ D ∩ Ũ and (c ∈ Ũ or h ≤ 2) then
       nxt := n
 else if ∃n ∈ (N − D) ∩ S then nxt := n
 else if ∃n ∈ (N − D) ∩ Ū then nxt := n
 else error('unable to deliver');
 route(nxt, t)
end
```

Figure 2. Routing algorithm `route` by Chiu and Wu.

Theorem 3 *In a hypercube network which is not fully unsafe, every strongly unsafe node is adjacent to an ordinary unsafe node. Hence, if the source $s \in \tilde{U}$ and the target $t \in U$ in a hypercube which is not fully unsafe, the algorithm `route` can communicate by a path whose length is at most $H(s,t) + 4$.*

Each nonfaulty node in a hypercube network exchanges the information with its neighbors to classify itself into safe, ordinary unsafe, strongly unsafe nodes. According to this classification, if a d-cube network is not fully unsafe, an effective routing can be implemented based on the theorems 1, 2, 3. Even if the hypercube network is fully unsafe, the algorithm `route` is applicable. However, the target is not always communicatable by the algorithm even if it is reachable from the source. In this case, it is necessary to switch to other worse algorithms [2].

3. New Algorithms

3.1. Algorithm based on Full Reachability

3.1.1 Full Reachability and Safe Nodes w.r.t. Distance

This section describes the algorithm `FR` proposed by us which utilizes the information of Hamming distance to the target for selection of a neighbor node to send the message. For this purpose, we define a property that a node is reachable any nonfaulty node which is apart from the node by some fixed Hamming distance.

Definition 6 (full reachability w.r.t. distance) *A nonfaulty node n is fully reachable with respect to (Hamming) distance h, if every nonfaulty node which is apart from the node n by Hamming distance exactly h is reachable from the node n via a path of distance h.*

Let R_h represent the set of nodes which are fully reachable w.r.t. distance h. When a node n which is apart from the target by Hamming distance $h+1$ receives a message and tries to find the node to hop, an unnecessary detour is avoidable if it can know the set of $N(n) \cap R_h$. However, it is difficult to identify R_h and find the appropriate route. Hence, we introduce an approximation of R_h.

Definition 7 (safe nodes w.r.t. distance) *Every nonfaulty node is a safe node with respect to (Hamming) distance 1. A nonfaulty node is a safe node w.r.t. distance h if it is adjacent to more than or equal to $d-h+1$ safe nodes w.r.t. distance $h-1$.*

In the rest of this paper, let S_h represent the set of safe nodes w.r.t. distance h. For example, consider a 4-cube network in which the set of the faulty nodes is $\{1, 4, 12, 13, 14\}$. Then following equations hold:

$S_1 = R_1 = \{0, 2, 3, 5, 6, 7, 8, 9, 10, 11, 15\}$,

$S_2 = \{2, 3, 7, 8, 10, 11\}$, $R_2 = S_2 \cup \{15\}$,

$S_3 = \{0, 2, 3, 6, 9, 10, 11, 15\}$, $R_3 = S_2 \cup \{7\}$.

The node 15 belongs to R_2 because every node of $3, 5, 6, 9, 10$ which are nonfaulty and apart from 15 by Hamming distance 2 is reachable from 15 on a path of length 2. However, it is not belongs to S_2. Similarly, the node $7 \in R_3$ though it does not belong to S_3.

3.1.2 Algorithm FR

About S_h and R_h, the following theorem holds.

Theorem 4 *For any distance h, $S_h \subset R_h$.*

(Proof) *It is proved by induction on h. Let $n \in S_1$, then $\forall n' \in N(n) - F$, n is reachable n' immediately. Hence, $S_1 \subset R_1$. Now, assume that $S_h \subset R_h$ holds for every $h < k$. Here, let $n \in S_k$, then from definition 7 $|N(n) \cap S_{k-1}| \geq d - k + 1$ (see figure 3). This means that $|N(n) - S_{k-1}| \leq k - 1$. $\forall n' \notin F$ s.t. $H(n, n') = k$, $|D(n, n')| = k$. Therefore, $D(n, n') \cap S_{k-1} \neq \emptyset$. From the hypothesis of induction, $S_{k-1} \subset R_{k-1}$. Then if we send a message to the node, it is possible to reach the target n' through a path of length k. Hence, $\forall n' \notin F$ s.t. $H(n, n') = k$, n is reachable n' using a path of length k and $n \in R_k$. From the above $S_k \subset R_k$, and for any distance h, $S_h \subset R_h$ holds.* □

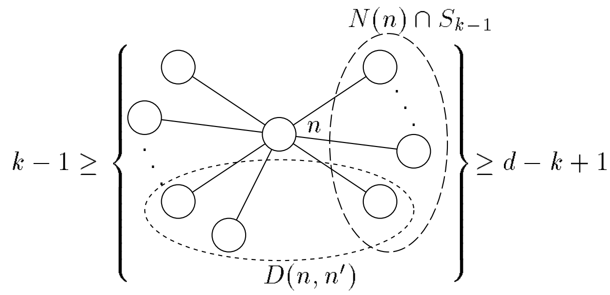

Figure 3. Relationship between node $n (\in S_k)$ and its neighbor nodes.

Generally, it is difficult to obtain $N(n) \cap R_h$ for any node n while S_h is easily detected because it can be determined only by exchanging some information with neighbors. By theorem 4, we may use S_h as a set of safe nodes to hop messages. If we calculate $S_2, \cdots, S_k$ in a preprocess, a new routing algorithm `FR` is obtained by changing the comment line `(*···*)` of figure 2 with `else if` $h \leq k{+}1$ `and` $\exists n \in D \cap S_{h-1}$ `then` nxt `:=` n (see figure 4). Note that we can stop the calculation of S_h $(h = 1, 2, \cdots)$ before S_{d-1} according to the available space of nodes. We assume in the rest of this paper that we caluculate from S_1 to S_k.

As shown in the following theorem 5, the algorithm `FR` has more nodes which can be selected to route safely than those in the algorithm `route`.

Theorem 5 *For any distance h, $S \subset S_h$.*

(Proof) *It is proved by induction on h. $S \subset S_1$ is trivial. If $n \in S$ then $|N(n) \cap F| \leq 1$ and $n \in S_2$ from definition 7. Hence $S \subset S_2$. Now, assume that $\forall h < k$, $S \subset S_h$ holds. Here, let $n \in S$ then $|N(n) - S| \leq 2$ from definition 3. Therefore, $|N(n) \cap S| \geq d - 2$. From the hypothesis of induction, $S \subset S_{k-1}$. Hence, $N(n) \cap S \subset N(n) \cap S_{k-1}$ holds and this means that $|N(n) \cap S_{k-1}| \geq d - 2 \geq d - k + 1$ because $k \geq 3$. This means $n \in S_k$ from definition 7. Hence $S \subset S_k$ and $\forall h$, $S \subset S_h$ holds.* □

```
procedure FR(c, t)
begin
 h := H(c,t); N := N(c); D := D(c,t);
 if h = 0 then
  deliver the message to c and exit
 else if ∃n ∈ D ∩ S then nxt := n
 else if h ≤ k + 1 and ∃n ∈ D ∩ S_{h-1} then
       nxt := n
 else if ∃n ∈ D ∩ Ū then nxt := n
 else if ∃n ∈ D ∩ Ũ and (c ∈ Ũ or h ≤ 2) then
       nxt := n
 else if ∃n ∈ (N − D) ∩ S then nxt := n
 else if ∃n ∈ (N − D) ∩ Ū then nxt := n
 else error('unable to deliver');
 FR(nxt, t)
end
```

Figure 4. Routing algorithm FR based on full reachability.

In an example shown in figure 5, since the source s is strongly unsafe, the algorithm `route` sends the message to a single ordinary unsafe node n_1 in $D(s,t)$, that is, the subset of neighbor nodes of s which are closer to the target t than the node s. However, nodes in $D(n_1,t)$ are all faulty ones and it must detour to a safe node n_3. While the algorithm `FR` finds that the node n_2 in $D(s,t)$ which is judged strongly unsafe in `route` is safe w.r.t. distance 2. Therefore, it can construct a path shown by arrows according to information about full reachability based on Hamming distance.

3.1.3 Initialization

Next, we show the algorithm for each nonfaulty node to accumulate the information of its neighbors in a faulty hypercube. We presume that every node has buffers, one for each link between a neighbor node and itself and constant time is required for sending/receiving a message and detecting faulty neighbor nodes.

Figure 6 shows the initialization procedure `init` for node c in `FR`. A variable $\sigma_{n,h}$ holds classification information about a node n w.r.t. distance h. A variable T_{h-1} represents a subset of neighbor nodes of c which belongs to S_{h-1}. The value $\sigma_{c,h}$ is determined according to the cardinality of T_{h-1}.

This procedure must be executed in addition to the initialization procedure for the algorithm `route` whose time complexity is $O(d^2)$ [4]. Because time complexity for the procedure `init` is $O(kd)$, it does not make worse the whole time complexity. Notice that the variable k is a parameter to which we calculate the S_h.

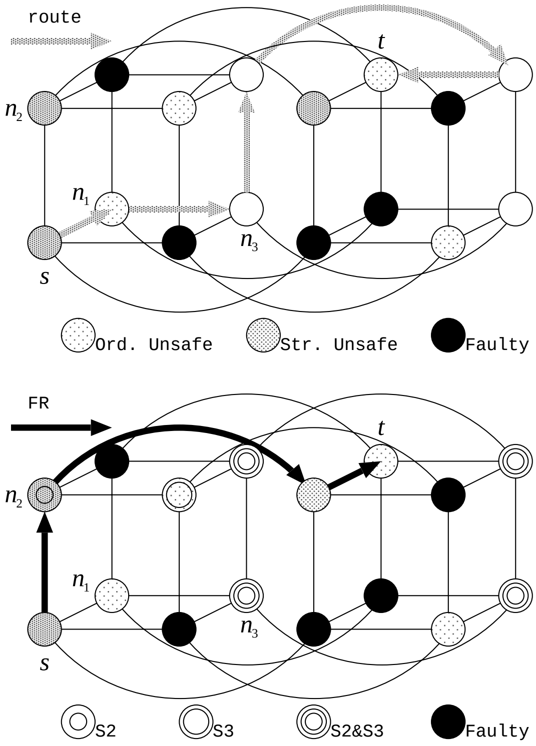

Figure 5. A routing example by `route` and `FR`.

3.2. Algorithm based on Classification of Unsafe Nodes

3.2.1 Classification of Unsafe Nodes w.r.t. Distance

If we can assign k equal to $d-1$ in the algorithm `FR`, it is not necessary to use classification information of safe nodes by Chiu and Wu for routing selection. Similarly, it is possible to make routing based only on node classification w.r.t. Hamming distance by classifying unsafe nodes w.r.t. distance.

Definition 8 (unsafe nodes w.r.t. distance) *A nonfaulty node n is unsafe w.r.t. (Hamming) distance h if the node n is not safe w.r.t. distance h.*

That is, if the number of neighbor nodes of a nonfaulty node n which are faulty or unsafe w.r.t. distance $h-1$ is greater than or equal to h, then the node n is unsafe w.r.t. distance h.

In addition, to detect a subset of unsafe nodes w.r.t. distance which gives guaranteed detours, we introduce its definition below.

Definition 9 (strongly and ordinary unsafe nodes w.r.t. distance) *For an unsafe node n w.r.t. distance h,*

```
procedure init(c, k)
begin
 σ_{c,1} := SAFE;
 Detect N(c) ∩ S_1;
 for h := 2 to k do
  begin
   send σ_{c,h-1} to N(c) ∩ S_1;
   for every n ∈ N(c) ∩ S_1 do
    receive σ_{n,h-1} from n;
   T_{h-1} := {n|n∈N(c) ∩ S_1, σ_{n,h-1} = SAFE}
   if |T_{h-1}| ≥ d - h + 1 then σ_{c,h} := SAFE
   else σ_{c,h} := UNSAFE
  end
end
```

Figure 6. Initialization procedure `init` for routing algorithm `FR`.

consider an arbitrary division of $N(n)$ into two disjoint subsets N_1 and N_2 where $|N_1| = h$ and $|N_2| = d - h$. The node n is ordinary unsafe w.r.t. distance h if, for all such divisions, $N_1 \cap S_{h-1} \neq \emptyset$ or $N_2 \cap S_{h+1} \neq \emptyset$. If a node n which is unsafe w.r.t. distance h is not ordinary unsafe w.r.t. distance h, it is a strongly unsafe node w.r.t. distance h.

In the rest of the paper, let U_h, $\bar{U}_h$ and $\tilde{U}_h$ represent the set of unsafe, ordinary unsafe, and strongly unsafe nodes w.r.t. distance h, respectively. A node $n \in \bar{U}_h$ has following property for a target t apart from n by Hamming distance $h(= H(n,t))$: $D(n,t) \cap S_{h-1} \neq \emptyset$ or $(N(n) - D(n,t)) \cap S_{h+1} \neq \emptyset$.

3.2.2 Initialization

It is difficult to detect ordinary unsafe nodes w.r.t. distance by using definition 9 directly. Hence we make use of the following theorem.

Theorem 6 *$\forall n \in U_h$, it is equivalent that $n \in \bar{U}_h$ and at least one of the following conditions holds:*

- $N(n) \cap S_{h+1} \cap S_{h-1} \neq \emptyset$.
- $|N(n) \cap S_{h+1}| \geq h + 1$.

(Proof) *Sufficiency: Let a node $n \in U_h$. Now, let us consider the case that $\exists n' \in N(n) \cap S_{h+1} \cap S_{h-1}$. In definition 9, if $n' \in N_1$ then $N_1 \cap S_{h-1} \neq \emptyset$; otherwise $n' \in N_2$ and $N_2 \cap S_{h+1} \neq \emptyset$. Next, consider the case that $|N(n) \cap S_{h+1}| \geq h + 1$. Then, since $|N_1| = h$, $N_2 \cap S_{h+1} \neq \emptyset$. Consequently, the node n is proved to belong to $\bar{U}_h$ in either case. Necessity: For a node $n \in \bar{U}_h$, we assume that $N(n) \cap S_{h+1} \cap S_{h-1} = \emptyset$, and $|N(n) \cap S_{h+1}| \leq h$. Then, we can divide $N(n)$ into N_1 and N_2 where $N(n) \cap S_{h+1} \subset N_1$ and $N(n) \cap S_{h-1} \subset N_2$. Hence, $n \notin \bar{U}_h$ which is a contradictory. The theorem is proved from above discussion.* □

By using the theorem 6, an algorithm shown in figure 7 can classify the neighbors of each nonfaulty node c where as similar to the initialization procedure for the algorithm `FR`, we assume that each node has buffers each of which is situated at the link between its neighbor node and itself; message sending and receiving are performed in constant time; and faulty neighbor nodes are detectable in constant time. A variable $\sigma_{n,h}$ holds the classification information of node n w.r.t. distance h and a variable T_h represents the subset of neighbor nodes of c which are safe w.r.t. distance h. The value of $\sigma_{c,h}$ is determined according to the variable T_h.

```
procedure init2(c)
begin
 σ_{c,1} := SAFE; Detect N(c) ∩ S_1;
 for h := 2 to d do
  begin
   send σ_{c,h-1} to N(c) ∩ S_1;
   for every n ∈ N(c) ∩ S_1 do
    receive σ_{n,h-1} from n;
   T_{h-1} := {n|n∈N(c) ∩ S_1, σ_{n,h-1} = SAFE};
   if |T_{h-1}| ≥ d - h + 1 then σ_{c,h} := SAFE
   else σ_{c,h} := S_UNSAFE
  end;
 send σ_{c,d} to N(c) ∩ S_1;
 for every n ∈ N(c) ∩ S_1 do
  receive σ_{n,d} from n;
 T_d := {n|n ∈ N(c) ∩ S_1, σ_{n,d} = SAFE};
 for h := 2 to d - 1 do
  if σ_{c,h} = S_UNSAFE then
   if ∃n ∈ T_{h-1} ∩ T_{h+1} or |T_{h+1}| ≥ h + 1 then
    σ_{c,h} := O_UNSAFE;
end
```

Figure 7. Initialization procedure `init2` for routing algorithm `FR2`.

3.2.3 Algorithm FR2

Concerning with classification above, the following theorems 7 and 8 hold.

Theorem 7 *For any distance h, $S \cup \bar{U} \subset S_h \cup \bar{U}_h$.*
(Proof) *Because $S \subset S_h$, it is sufficient to show that the set $\bar{U} - S_h$*(the hatched part in figure 8) *is a subset of $\bar{U}_h$. $\forall n \in \bar{U} - S_h$, $n \in \bar{U}$. Hence, $\exists n' \in N(n) \cap S$ (S is the set of safe nodes by Chiu and Wu). Therefore,*

$n' \in S_{h-1} \cap S_{h+1}$. *From the theorem 6,* $n \in U_h$. *Consequently,* $\bar{U} - S_h \subset \bar{U}_h$. □

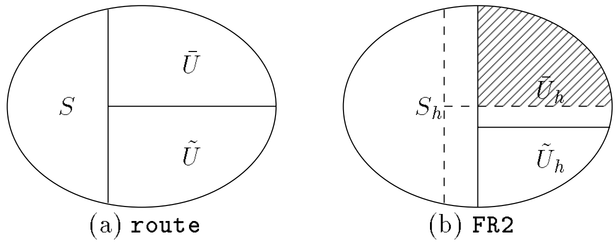

Figure 8. Node classification with respect to distance.

Theorem 8 *For any distance* h, $\tilde{U}_h \subset \tilde{U}$.
(Proof) *It is obvious from theorem 7.* □

From the theorem 8, if a hypercube network is not fully unsafe (by the definition 5), a strongly unsafe node w.r.t. distance h has an ordinary unsafe neighbor node. From this fact, we can construct a new algorithm `FR2` based on classification of unsafe nodes. See figure 9.

```
procedure FR2(c, t)
begin
 h := H(c,t); N := N(c); D := D(c,t);
 if h = 0 then
  deliver the message to c and exit
 else if ∃n ∈ D ∩ S_{h-1} then nxt := n
 else if ∃n ∈ D ∩ Ū_{h-1} then nxt := n
 else if ∃n ∈ D ∩ Ũ_{h-1}
         and (c ∈ Ũ_h or h ≤ 2) then nxt := n
 else if ∃n ∈ (N − D) ∩ S_{h+1} then nxt := n
 else if ∃n ∈ (N − D) ∩ Ū_{h+1} then nxt := n
 else error('unable to deliver');
 FR2(nxt, t)
end
```

Figure 9. Algorithm `FR2` based on classification of unsafe nodes

4. Evaluation

To evaluate our algorithms `FR` and `FR2`, we repeat the following procedure for all combinations of addresses of faulty nodes and a target node. For the algorithm `FR`, we adopted the best parameter value $k = d - 1$.

1. In a d-cube, set f faulty nodes.
2. Classify all nodes into faulty, safe, ordinary unsafe, and strongly unsafe nodes. Moreover, calculate $S_h, \bar{U}_h$ and $\tilde{U}_h$ $(1 \le h \le d)$.
3. Due to symmetry, fix the node 0 as the source, and select a nonfaulty target node which is not the source and is reachable from the source.
4. Call the procedures for `route`, `FR` and `FR2` then count the numbers of unnecessary detours and failures of deliveries for each.

Figure 10 shows the results. The horizontal axis d/f represents pairs of the dimension d of the hypercube network and the number of faulty nodes f. For each pair of the dimension and the number of faulty nodes, the vertical axis represents the ratio of summation of the numbers of detours and failures of algorithms `FR` and `FR2` in comparison to that of `route`.

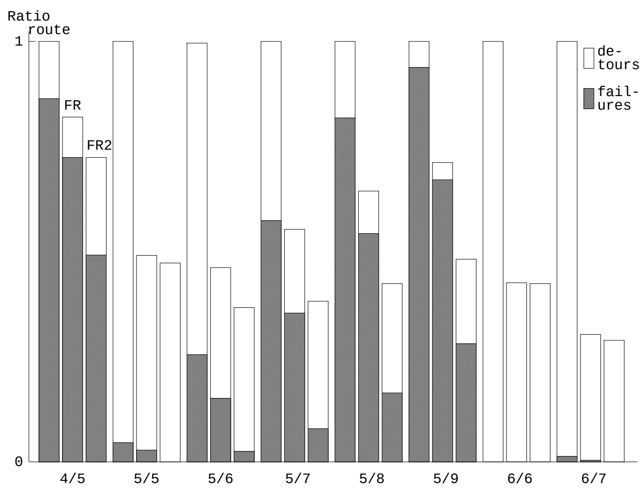

Figure 10. Improvement ratio by algorithm `FR2`.

In every case of simulation, `FR2` shows the best results and `FR` is better than `route`. Focusing on the number of failures, `FR2` further reduces it than `FR` does. We think that it is due to the new scheme based on the classification of unsafe nodes w.r.t. distance.

Moreover, we compare our algorithms `FR` and `FR2` with an algorithm `RC` by Chiu and Chen [3]. Their algorithm uses the notion of routing capability which is equivalent to our full reachability. Assume that the current and target nodes are c and t, respectively. Then the algorithm searches for a node to proceed or detour to in $D \cap S_{h-1}$, $D \cap S_{h+1}$, $(N - D) \cap S_{h+1}$, $D \cap S_{h+3}$,

$(N - D) \cap S_{h+3}$, ...in this order where $h = H(c,t)$, $N = N(c)$ and $D = D(c,t)$. Though the directed version of classification [3] is applicable to all algorithms including ours, it is ignored for simplicity. Simulation is performed by following the procedure mentioned above with one exception that the addresses of faulty nodes and a target node are randomly generated one million times. Figures 11 and 12 show the average percentage of the shortest path routing and the average reachability of the algorithms in a 6-cube, respectively. The reachability is the ratio of messages which managed to reach the target nodes. In either case, results show that the algorithms FR and FR2 are superior to RC by Chiu and Chen because RC uses only the routing capabilities while others use the notion of unsafeness.

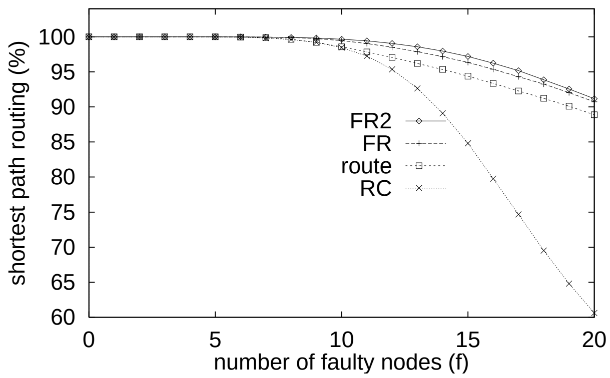

Figure 11. Shortest path routing percentage in 6-cube.

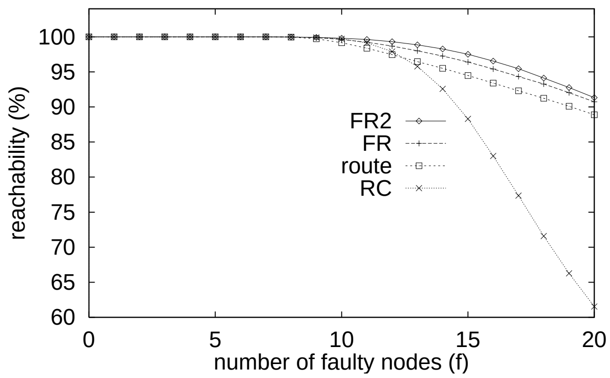

Figure 12. Reachability percentage in 6-cube.

5. Conclusioins

We first proposed a routing algorithm FR based on full reachability which is an extension of the algorithm by Chiu and Wu [4]. We proposed another routing algorithm FR2 which does not make use of classification information used in Chiu and Wu [4] by classifying unsafe nodes w.r.t. the Hamming distance. As a result of evaluation of the algorithms, it was shown that they can detect communication paths which do not include any faulty node and were not found by conventional algorithms route and RC. In addition, it is demonstrated by computer simulation that the FR and FR2 are effective for low-dimensional hypercubes and give good results.

In future, it is necessary to execute simulation to make sure the ability of the algorithms FR and FR2 for high-dimensional hypercube networks. It is also a future work to develop an efficient reinitilization method in case of occurences of faulty nodes in operation.

References

[1] J. Bruck, R. Cypher, and D. Soroker. Embedding cube-connected cycles graphs into faulty hypercubes. *IEEE Trans. Comput.*, 43(10):1210–1220, October 1994.

[2] M.-S. Chen and K. G. Shin. Depth-first search approach for fault-tolerant routing in hypercube multicomputers. *IEEE Transactions on Parallel and Distributed Systems*, 1(2):152–159, April 1990.

[3] G.-M. Chiu and K.-S. Chen. Use of routing capability for fault-tolerant routing in hypercube multicomputers. *IEEE Trans. Comput.*, 46(8):953–958, August 1997.

[4] G.-M. Chiu and S.-P. Wu. A fault-tolerant routing strategy in hypercube multicomputers. *IEEE Trans. Comput.*, 45(2):143–155, February 1996.

[5] W. J. Dally. Performance analysis of k-ary n-cube interconnection networks. *IEEE Trans. Comput.*, 39(6):775–785, June 1990.

[6] J. Kim and C. R. Das. Hypercube communication delay with wormhole routing. *IEEE Trans. Comput.*, 43(7):806–814, July 1994.

[7] T. C. Lee and J. P. Hayes. A fault-tolerant communication scheme for hypercube computers. *IEEE Trans. Comput.*, 41(10):1242–1256, October 1992.

[8] Y. Saad and M. H. Schultz. Topological property of hypercubes. *IEEE Trans. Comput.*, 37(7):867–872, July 1988.

[9] C. L. Seitz. The cosmic cube. *Commun. ACM*, 28(7):22–23, July 1985.

[10] N.-F. Tzeng and H.-L. Chen. Structural and tree embedding aspects of incomplete hypercubes. *IEEE Trans. Comput.*, 43(12):1434–1439, December 1994.

[11] J. Wu. Adaptive fault-tolerant routing in cube-based multicomputers using safety vectors. *IEEE Transactions on Parallel and Distributed Systems*, 9(4):322–334, April 1998.