A Fault-tolerant Routing Algorithm using Directed Probabilities in Hypercube Networks

Manabu Myojin  Keiichi Kaneko
Department of Computer and Information Sciences
Graduate School of Engineering
Tokyo University of Agriculture and Technology
Koganei-shi, Tokyo, JAPAN

Abstract In this paper, we propose a new fault-tolerant routing algorithm for hypercube networks based on directed probabilities. The directed probabilities are obtained by improving the probabilities proposed by Al-Sadi et al. The probability represents ability of routing to any node at a specific distance. Each node selects one of its neighbor nodes to send a message by taking the probabilities into consideration. We also conducted a computer experiment to verify the effectiveness of our algorithms.

Keywords: multicomputer, interconnection network, parallel processing, fault-tolerant routing, hypercube, performance evaluation

1 Introduction

Recently, large-scaled computation is required in many fields, and interests in researches on parallel processing are increasing. Among them, studies on massively parallel systems where many processors are connected to execute computation are eagerly conducted. With a progressive increase in the number of processors in the system, probability of existence of faulty processors increases. Hence, it is necessary for massively parallel systems to establish a fault-free routing path. Such path construction is formulated to be a fault-tolerant routing problem in graph theory mapping processors and links between them to nodes and edges, respectively.

In this study, we focused on hypercubes [8, 9], which have simple and recursive structure and low diameter, and provide interconnection networks suitable for massively parallel systems. Figure 1 shows an example of a 4-dimensional hypercube $Q_4$. In an interconnection network, if each node can collect information of all the faulty nodes, shortest-path routing between any pair of nodes is possible. However, this method asks each node to store information of all the faulty nodes, and it requires too much memory space. In addition, collection of such information takes too much time complexity for communication.

To solve the problem, there are many routing approaches based on restricted global information where each node stores some restricted information of faulty nodes and executes quasi-optimal routing [1, 2, 3, 4, 5, 6, 7, 10, 11]. The approach by Al-Sadi et al. [1] is one such approach. In their approach, each node calculates probability of unreachability by minimal paths to destination nodes with each Hamming distance, exchanges these probabilities with its neighbors, and routes based on these probabilities. Although, their approach attains high reachability in a hypercube with faulty nodes, we believe that there remains some room for improvement. Therefore, in this study, we introduce a notion of directed routing probability, and try to improve the approach by Al-Sadi et al.

The rest of this paper is structured as follows. First, we survey related works in Section
2. Next, requisite terminology and notations are defined in Section 3. Then, in Section 4, we introduce the algorithm proposed by Al-Sadi et al. We proposed a fault-tolerant routing algorithm that is obtained by improving the algorithm by Al-Sadi et al. in Section 5. The algorithm is evaluated by a computer experiment in Section 6. Finally, we give a conclusion and future works in Section 7.

2 Related Works

For these two decades, there are many attempts in research for fault-tolerant routing in hypercube networks. Chiu and Wu have proposed an efficient fault-tolerant routing algorithm by recursively classifying non-faulty nodes into safe, ordinary unsafe, and strongly unsafe nodes depending on the classification of neighbor nodes [5]. Chiu and Chen have improved the algorithm by introducing the routing capabilities that are obtained by classifying the safety nodes with respect to the Hamming distance to the destination nodes [6]. Wu has also proposed a similar fault-tolerant routing algorithm independently by introducing the safety vectors [10]. Moreover, Kaneko and Ito have proposed a fault-tolerant routing algorithm based on classification of ordinary and strongly unsafe nodes with respect to the Hamming distance as well as an efficient method to obtain classification of them [7].

All of the above attempts are based on information if a message is surely routed to the destination node or not. On the other hand, Al-Sadi et al. have proposed a fault-tolerant routing algorithm that is based on probabilities that a message is sent from the source node to the destination node with a path of length of Hamming distance between them [1, 2]. In the algorithm, each non-faulty node exchanges information at most $O(n^2)$ times with its neighbor nodes to calculate the probabilities with respect to the Hamming distances to destinations. However, there are several cases where the algorithm fails to find a fault-free path even if there is a such path.

3 Preliminaries

In this section, we define a hypercube network and introduce requisite notations.

Definition 1 An $n$-dimensional hypercube $Q_n$ is an undirected graph, and $Q_n$ consists of $2^n$ nodes. Each node $a$ in $Q_n$ is an $n$-bit sequence $(a_1, a_2, \ldots, a_n)$ where $a_i \in \{0, 1\}$ ($1 \leq i \leq n$), and $a_i$ is called the bit of $i$-th dimension. For two nodes $a$ and $b$ in $Q_n$, there is an edge $(a, b)$ between them if and only if the Hamming distance between them $H(a, b)$ is equal to 1.

The neighbor node of $a$ that is obtained by changing the $i$-th bit ($1 \leq i \leq n$) is denoted by $a^{(i)}$ in the rest of paper.

In a hypercube $Q_n$ with a set of faulty nodes $F$, for a source node $s$ and a destination node $d$
that are both non-faulty, a fault-tolerant routing algorithm finds a fault-free path between \( s \) and \( d \).

**Definition 3** For two nodes \( a \) and \( b \) in \( Q_n \), the set of preferred neighbor nodes of \( a \) for \( b \) is denoted by \( N_0(a, b) \), and is defined by \( N_0(a, b) = \{ n \mid n \in N(a), H(n, b) = H(a, b) - 1 \} \). In addition, the set of spare neighbor nodes of \( a \) for \( b \) is denoted by \( N_1(a, b) \), and is defined by \( N_1(a, b) = \{ n \mid n \in N(a), H(n, b) = H(a, b) + 1 \} \).

Note that, in \( Q_n \), the number of nodes that are apart from a node \( a \) by Hamming distance \( h \) is equal to \( nC_h \). Note also that, for two nodes \( a \) and \( b \) in \( Q_n \), if \( H(a, b) = h \), then \( |N_0(a, b)| = h \) holds.

### 4 Algorithm by Al-Sadi et al.

In this section, we give the idea of the algorithm by Al-Sadi et al. In their algorithm, for an arbitrary non-faulty node \( a \) in an \( n \)-dimensional hypercube \( Q_n \) with a faulty node set \( F \), an estimate value of the probability that there is not any fault-free minimal path from \( a \) to an arbitrary node \( b \) such that \( H(a, b) = h \) is calculated, and routing is executed based on the estimated values.

For simplicity of explanation, we use an estimate value of the probability that there is some fault-free path instead of the probability that there is not any fault-free path in our paper. In addition, we assume that \( F \) represents a faulty node set in \( Q_n \) in the rest of the paper.

\( P_h(a) \) represents an estimate value of probability of existence of a fault-free minimal path from a fault-free node \( a \) to an arbitrary node at Hamming distance \( h \) from \( a \). \( P_h(a) \) is defined recursively:

\[
P_h(a) = \begin{cases} 
|N(a) \setminus F| / n & (h = 1) \\
1 - \prod_{i=1}^{n}(1 - R_h(a^{(i)})) & (2 \leq h \leq n, a \notin F) \\
0 & (b \in F)
\end{cases}
\]

The estimate values of probabilities can be calculated by the algorithm shown in Figure 2.

**function** \( EVP(a, h, F) \)

**begin**

\( h = 1 \) then \( P_h(a) := |N(a) \setminus F| / n \) else begin

\( P := 1 \);

for \( i := 1 \) to \( n \) do begin

if \( a^{(i)} \in F \) then \( R_h(a^{(i)}) := 0 \)

else \( R_h(a^{(i)}) := h \times P_{h-1}(a^{(i)}) / n \);

\( P := P \times (1 - R_h(a^{(i)})) \)

end;

\( P_h(a) := 1 - P \)

end;

return \( P_h(a) \)

**end**

![Figure 2: Function to calculate estimate values of probabilities](image-url)

Their routing algorithm route based on the estimate values of probabilities is shown in Figure 3. When a node \( a \) has to forward a message to its destination \( d \), the algorithm is used.

The algorithm first checks whether the current node \( a \) is the destination node \( d \) itself or not. If \( a = d \), the message is delivered to \( d \) immediately. Otherwise, if \( d \) is a neighbor node of \( a \) (\( d \in N(a) \)), the message is also delivered to \( d \) immediately. If \( H(a, d) \geq 2 \), the algorithm tries to find the preferred neighbor node of \( a \) that has the largest estimate value of probability. If the value is positive, then the message is forwarded to the node. Otherwise, if it is 0, then the spare neighbor nodes are checked to find one with the largest estimate value. If the value is not 0, the message is sent to the node while if it is 0, the delivery fails.

Their algorithm attains high reachability in \( Q_n \) with faulty nodes. However, there remains some room for improvement. That is, the value \( P_{h-1}(a) \) may have an effect on \( P_h(a^{(i)}) \). Hence, \( a \) may send a message to its neighbor \( a^{(i)} \) even if \( P_h(a^{(i)}) = 0 \) without \( P_{h-1}(a) \). Figure 4
function route\((a, d, F)\) begin 
\(h := H(a, d)\);
if \(h = 0\) then begin 
deliver the message to \(a\); exit
end;
if \(h = 1\) then begin 
send the message to \(d\); exit
end;
i_0 := \text{argmax}_i\{P_{h-1}(a^{(i)}) | a^{(i)} \in N_0(a, d)\};
if \(a^{(i_0)} \not\in F\) then begin 
deliver the message to \(a^{(i_0)}\); exit
end;
i_1 := \text{argmax}_i\{P_{h+1}(a^{(i)}) | a^{(i)} \in N_1(a, d)\};
if \(a^{(i_1)} \not\in F\) then begin 
deliver the message to \(a^{(i_1)}\); exit
end;
error(‘message delivery failed’) end
Figure 3: Routing algorithm based on estimate values of probabilities

shows an example of this case. In the figure, \(P_h(a^{(i)})\) may be positive if \(P_{h-1}(a)\) is also positive though there is no way out from \(a^{(i)}\). Therefore, in the next section, we introduce a notion of directed routing probability, and try to improve the algorithm by Al-Sadi et al.

5 Proposed Algorithm

To address the problem mentioned at the end of the previous section, we have excluded an effect by \(P_{h-1}(a)\) from calculation of \(P_h(a^{(k)})\). The new probability is called directed probability, and we calculate estimate values of directed probabilities \(\tilde{P}_h^k(a)\) where \(k\) represents the direction to which the estimate value is transmitted.

\[
\tilde{P}_h^k(a) = \begin{cases} 
\frac{|(N(a) \setminus \{a^{(k)}\}) \setminus F|}{n-1} & (h = 1) \\
1 - \prod_{i=1}^{n} \left(1 - \tilde{R}_h^i(a^{(i)})\right) & (2 \leq h \leq n, a \not\in F) \\
0 & (b \in F) \\
h\tilde{R}_{h-1}^i(b)/(n-1) & (b \not\in F)
\end{cases}
\]

The estimate values of directed probabilities can be calculated by the algorithm shown in Figure 5.

function EVDP\((a, h, k, F)\) begin 
if \(h = 1\) then 
\(\tilde{P}_h^k(a) := \frac{|(N(a) \setminus \{a^{(k)}\}) \setminus F|}{n-1}\)
else begin 
\(\tilde{P} := 1;\)
for \(i := 1\) to \(n\) do begin 
if \(i = k\) then continue;
if \(a^{(i)} \in F\) then 
\(\tilde{R}_h^i(a^{(i)}) := 0\)
else 
\(\tilde{R}_h^i(a^{(i)}) := h \times \tilde{R}_{h-1}^i(a^{(i)})/(n-1);\)
\(P := P \times \left(1 - \tilde{R}_h^i(a^{(i)})\right)\)
end;
\(\tilde{P}_h^k(a) := 1 - P\)
end;
return \(\tilde{P}_h^k(a)\) end
Figure 5: Function to calculate estimate values of directed probabilities

Our routing algorithm route2 based on the estimate values of directed probabilities is shown in Figure 6. It is similar to the algorithm by Al-Sadi et al., and it is used when a node \(a\) has to forward a message to its destination \(d\).

The time complexity to calculate estimate values of directed probabilities \(\tilde{P}_1^k(a), \tilde{P}_2^k(a), \ldots\) is linear in the number of nodes.
function route2(a, d, F)
    begin
        h := H(a,d);
        if h = 0 then begin
            deliver the message to a; exit
        end;
        if h = 1 then begin
            send the message to d; exit
        end;
        i0 := argmax_i{P_{h-1}^i(a)} \in N_0(a,d)};
        if a^{(i_0)} \not\in F then begin
            deliver the message to a^{(i_0)}; exit
        end;
        i1 := argmax_i{P_{h+1}^i(a)} \in N_1(a,d)};
        if a^{(i_1)} \not\in F then begin
            deliver the message to a^{(i_1)}; exit
        end;
        error('message delivery failed')
    end

Figure 6: Routing algorithm based on estimate values of directed probabilities

..., \bar{P}_k^i(a) (k = 1, 2, ..., n) is O(n^3), which is larger than the time complexity of the approach by Al-Sadi et al. The number of times that a node exchanges these values with its neighbor nodes is O(n^2). It is same as that by Al-Sadi et al.

6 Evaluation

To evaluate performance of our algorithm, we carried out an computer experiment to compare it with the algorithm by Al-Sadi et al. The experiment is conducted based on the following procedure:

1. In Q_n, for the ratio of faulty nodes \alpha = 0, 0.1, ..., 1.0, repeat Steps 2 to 4 for 10,000 times.
2. Set \lfloor \alpha 2^n \rfloor faulty nodes in Q_n.
3. Select two nodes s and d randomly such that there exists a fault-free path between them.
4. Apply the algorithms route and route2, and check if the message is delivered to the destination or not.

Figure 7 shows the result of the experiment with Q_{10}. According to the figure, we can see that our algorithm is superior to the algorithm by Al-Sadi et al. for \alpha = 0.6, 0.7, 0.8 and 0.9. The reason that the ratios of successful routings increased for \alpha = 0.9 is that the ratio of faulty nodes is so high that the pairs of s and d could be found with short Hamming distance only.

![Figure 7: Ratio of successful routings by algorithms route and route2 in Q_{10}](image)

7 Conclusion

In this paper, we have proposed a new fault-tolerant routing algorithm for hypercube networks based on directed probabilities, which are obtained by improving the probabilities proposed by Al-Sadi et al. The time complexity to calculate all of the estimate values of directed probabilities is O(n^3). The number of times for each node to exchange these values with its neighbors is O(n^2). We also conducted a computer experiment to verify the effectiveness of our algorithm, and we showed that our algorithm is superior to that by Al-Sadi et al. when the ratio of faulty nodes \alpha is relatively high (\alpha = 0.6, 0.7, 0.8, and 0.9).
Future works include reduction of the time complexity of the algorithm. Applying similar approach to other topologies is also included in future works.

Acknowledgement

This study was partly supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science under Grant No. 22500041.

References


