A Fault-tolerant Routing Algorithm based on Safety Levels in a Hyper-Star Graph

Yuko Sasaki Yuki Hirai Hironori Nakajo Keiichi Kaneko
Graduate School of Engineering
Tokyo University of Agriculture and Technology
Koganei-shi, Tokyo 184-8588, JAPAN
{50013646114@st,yhirai@cc,nakajo@cc,k1kaneko@cc}.tuat.ac.jp

Abstract A hyper-star graph $HS(n, k)$ provides a promising topology for interconnection networks of parallel processing systems because it combines the merits of a hypercube and a star graph. In this study, we propose a fault-tolerant routing algorithm that establishes a fault-free path between any pair of non-faulty nodes in an $HS(n, k)$ with faulty nodes by using limited global information called safety levels. In addition, we carried out a computer experiment to verify the effectiveness of the algorithm.

Keywords: multicomputer, interconnection network, parallel processing, hypercube, star graph, faulty node, performance evaluation

1 Introduction

With the development of research on parallel processing systems, many new topologies for interconnection networks have been proposed [1, 4, 5, 6, 8, 10, 12] instead of simple topologies such as a ring, a mesh, a torus, a hypercube [13], and so on. A hyper-star graph $HS(n, k)$ provides a such new topology, and it is promising because it combines the merits of a hypercube and a star graph [9]. Algorithms should be designed and developed presuming the existence of faulty elements in a large-scaled parallel system. Therefore, in this paper, we focus on a hyper-star graph $HS(n, k)$ that has faulty nodes, and propose an adaptive fault-tolerant routing algorithm between non-faulty nodes.

If each non-faulty node collects information of all faulty nodes as global information, optimal fault-tolerant routing is possible. However, this approach is impractical since it requires space and time complexities, whose orders are equal to the number of nodes in the graph. On the other hand, if each non-faulty node collects the status of its neighbor nodes only as local information for fault-tolerant routing, high reachability cannot be attained. Therefore, some approaches collect a part of the global information to attain high reachability. The information is called limited global information.

For a hypercube, there are several approaches based on the limited global information. By recursively classifying non-faulty nodes into safe, ordinary unsafe, and strongly unsafe nodes depending on the classification of neighbor nodes, Chiu and Wu have proposed an efficient fault-tolerant routing algorithm [2]. To improve the algorithm, Chiu and Chen introduced the routing capabilities that are obtained by classifying the safety nodes with respect to the Hamming distance to the destination nodes [3]. Wu has also proposed a similar fault-tolerant routing algorithm independently by introducing the safety vectors [14]. In addition, 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].

For a star graph, Yeh et al. have proposed an algorithm based on the safety vectors to attain
efficient fault-tolerant routing [15]. The routing in a star graph is more complicated than that in a hypercube. Hence, the safety vectors on a star graph are based on routing patterns while those on a hypercube are based on distances.

For a regular hyper-star graph $HS(2n, n)$, Nishiyama et al. have proposed an algorithm based on the safety levels, which represent limited global information [11]. In our approach, we introduce the safety levels for a generic non-regular hyper-star graph to attain high reachability.

The rest of this paper is structured as follows. In Section 2, a hyper-star graph, a safety level, and other requisite concepts are defined, and some properties are proved. In Section 3, we describe the fault-tolerant routing algorithm based on the safety levels. In Section 4, by a computer experiment, we verify the effectiveness of our algorithm. In Section 5, we give conclusions and a future work.

## 2 Preliminaries

In this section, we give a definition of a hyper-star graph and lemmas about its properties. We also introduce a definition of a safety level.

**Definition 1 (hyper-star graph $HS(n, k)$)** An $HS(n, k)$ is an undirected graph, which has $nC_k$ nodes. Each node $a$ consists of $n$ bits $(a_1, a_2, \ldots, a_n)$. Among these bits, $k$ bits are always equal to 1 while the remaining $(n-k)$ bits are always 0 ($a \in \{0, 1\}^n$, $\sum_{i=1}^{n} a_i = k$). For two nodes $a = (a_1, a_2, \ldots, a_n)$, $b = (b_1, b_2, \ldots, b_n)$, an edge $(a, b)$ exists if and only if there exists $j \in \{2, 3, \ldots, n\}$ such that $b_1 = a_1$, $b_j = a_j = a_1$, $b_i = a_i$ ($2 \leq i \neq j \leq n$). 

Figure 1 shows an example of $HS(6, 2)$. Table 1 shows comparison of a hyper-star graph $HS(n, k)$ ($n > 2k$) with a hypercube $Q_n$, a hierarchical hypercube $HCN_{2n+k}$, and a hierarchical cube network $HCN_n$.

In an $HS(n, k)$, for two nodes $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$, the distance between them $d(a, b)$ is given by $\sum_{i=2}^{n} a_i \oplus b_i$.

![Figure 1: An example of a hyper-star graph $HS(6, 2)$](image)

<table>
<thead>
<tr>
<th></th>
<th>#nodes</th>
<th>degree</th>
<th>connect</th>
<th>diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HS(n, k)$</td>
<td>$2^n$</td>
<td>$n-k$</td>
<td>$n$</td>
<td>$2n$</td>
</tr>
<tr>
<td>$Q_n$</td>
<td>$2^n$</td>
<td>$n$</td>
<td>$n$</td>
<td>$2n$</td>
</tr>
<tr>
<td>$HCN_{2n+k}$</td>
<td>$2^{2n+k}$</td>
<td>$n+1$</td>
<td>$n+1$</td>
<td>$2^{n+1}$</td>
</tr>
<tr>
<td>$HCN_n$</td>
<td>$2^n$</td>
<td>$n+1$</td>
<td>$n+1$</td>
<td>$\lfloor 4(n+1)/3 \rfloor$</td>
</tr>
</tbody>
</table>

Moreover, among the neighbor nodes of $a$, let $Pre(a, b) = \{n | n \in N(a), d(n, b) = d(a, b) - 1\}$ and $Spr(a, b) = \{n | n \in N(a), d(n, b) = d(a, b) + 1\}$ be the neighbor nodes that are on the shortest paths from $a$ to $b$ and those on the detour paths from $a$ to $b$, respectively.

**Lemma 1** For an $HS(n, k)$, its diameter $diam(HS(n, k))$ is given as follows:

$$diam(HS(n, k)) = \begin{cases} 2(n-k) & (n < 2k) \\ 2k-1 & (n = 2k) \\ 2k & (n > 2k) \end{cases}$$

(Proof) For two nodes $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$ in an $HS(n, k)$, if $n < 2k$, the diameter $2(n-k)$ is, for instance, given by $d(a, b)$ where $a_1 = a_2 = \cdots = a_k = 1$, $a_{k+1} = \cdots = a_n = 0$, $b_1 = 0$, $b_2 = 0$, $\cdots = b_{n-k+1} = 0$, $b_{n-k+2} = \cdots = b_n = 1$. If symmetric property, if $n > 2k$, the diameter is equal to $2k$. If $n = 2k$, the diameter $2k-1$ is, for instance, given by $d(a, b)$ where $a_1 = a_2 = \cdots = a_k = 1$, $a_{k+1} = \cdots = a_n = 0$, $b_1 = b_2 = \cdots = b_k = 0$, $b_{k+1} = \cdots = b_n = 1$. 

**Lemma 2** For a node $a = (a_1, a_2, \ldots, a_n)$ in an $HS(n, k)$, $|N(a)| = k$ if $a_1 = 0$, and $|N(a)| = n-k$ if $a_1 = 1$. 

Table 1: Comparison of a hyper-star graph with other topologies.
are equal to \( \bar{a} \) bits of \( \text{cap} \), and \( d \) and \( b \) are among \( \sum_{i=1}^{a} \). Thus, \( |\text{Pre}(a, b)| = d/2 \) and \( |\text{Spr}(a, b)| = |N(a)| - d/2 \). On the other hand, if \( d \) is odd, \( a_1 = b_1 \). Hence, among \( d \) bits of \( a_2, a_3, \ldots, a_n \) that are different from corresponding \( b_2, b_3, \ldots, b_n \), \( (d+1)/2 \) bits are equal to \( a_1 \). Thus, \( |\text{Pre}(a, b)| = (d+1)/2 \) and \( |\text{Spr}(a, b)| = |N(a)| - (d+1)/2 \). To recap, \( |\text{Pre}(a, b)| = [d/2] \) and \( |\text{Spr}(a, b)| = |N(a)| - [d/2] \).

For a node \( a \) in an \( HS(n, k) \) and a distance \( d \) (\( 1 \leq d \leq \text{diam}(HS(n, k)) \)), we introduce a safety level \( S_d(a) \) so that it indicates that for any non-faulty node which is located with distance \( d \) from the node \( a \), a fault-free path of length \( d \) from \( a \) to the node can be established.

**Definition 2** For a node \( a \) in an \( HS(n, k) \) with a set of faulty nodes \( F \), a safety level \( S_d(a) \) with respect to a distance \( d \) is defined as follows:

1. \( S_d(a) = 1 \) if \( a \not\in F \), \( d = 1 \),
2. \( S_d(a) = 1 \) if \( a \not\in F \), \( d \geq 2 \), and for any \( J(\subset N(a)) \) such that \( |J| = [d/2] \), there exists \( n(\in J) \) such that \( S_{d-1}(n) = 1 \),
3. \( S_d(a) = 0 \) otherwise.

Since it takes much time to calculate safety levels in each node based on Definition 2, we introduce a simple calculation method based on the following lemma.

**Lemma 4** For a node \( a(\not\in F) \) in an \( HS(n, k) \) with a set of faulty nodes \( F \) and a distance \( d(\geq 2) \), the following two conditions are equivalent:

1. For any \( J(\subset N(a)) \) such that \( |J| = [d/2] \), there exists a node \( n(\in J) \) such that \( S_{d-1}(n) = 1 \).
2. \( |\{n | n \in N(a), S_{d-1}(n) = 1\}| \geq |N(a)| - [d/2] + 1 \).

(Proof) Since \( |J| = [d/2] \), \( |N(a) \setminus J| = |N(a)| - [d/2] \) holds. Hence, Condition 2 implies Condition 1. Therefore, sufficiency is proved. For necessity, we assume that Condition 2 does not hold. Then, from \( |\{n | n \in N(a), S_{d-1}(n) = 1\}| \leq |N(a)| - [d/2] \), there exists \( J(\subset N(a)) \) such that \( |J| = [d/2] \) and \( \{n | n \in N(a), S_{d-1}(n) = 1\} \subset N(a) \setminus J \). Therefore, Condition 1 does not hold, either. Necessity is also proved. From above discussion, the lemma is proved.

### 3 Fault-tolerant routing algorithm

In an \( HS(n, k) \), from Lemma 4, for a non-faulty node \( a \) and a distance \( d \), we can compare \( \sum_{n \in N(a)} S_{d-1}(n) \) with \( |N(a)| - [d/2] + 1 \) to judge sufficiency of Condition 2 in Definition 2. Figure 2 shows an algorithm as Procedure \( SL \) to calculate safety levels \( S_d(a) \) (\( 1 \leq d \leq \text{diam}(HS(n, k)) \)) at a node \( a \). The procedure must be executed in synchronization at all nodes.

**Lemma 4**

**Theorem 1** At each node in an \( HS(n, k) \), the time complexity to calculate safety levels with respect to all distances \( d \) (\( 1 \leq d \leq \text{diam}(HS(n, k)) \)) is \( O(n^2) \).

(Proof) From Lemma 4, to calculate a safety level \( S_d(a) \) with respect to a distance \( d(\geq 2) \) at
a node \( a \), it is necessary to collect \( S_{d-1}(n) \) from each node \( n \) in neighbor nodes \( N(a) \) of \( a \) and sum up them. This process requires \( O(n) \) time complexity. Therefore, it takes \( O(n^2) \) time in total to calculate safety levels for all distances \( d \) \((2 \leq d \leq \text{diam}(HS(n,k)))\).

From Theorem 1, it takes \( O(n^2) \) time to calculate the safety levels with respect to all distances at each node. If all the nodes calculate the safety levels in synchronization, the total time complexity is \( O(n^2) \).

In an \( HS(n,k) \) with a set of faulty nodes \( F \), a fault-tolerant routing algorithm based on safety levels is shown in Figure 3 as Procedure \( \text{FTS} \). To send a message from a non-faulty node \( s \) to a non-faulty node \( d \), we should call this procedure as \( \text{FTS}(s,d,F) \).

![Figure 3: Fault-tolerant routing algorithm based on safety levels.](image)

It takes \( O(n) \) time to identify \( \text{Pre}(c,d) \) and \( \text{Spr}(c,d) \), and to check \( S_{d-1}(n) \) to find the node \( n^* \). Note that Algorithm \( \text{FTS} \) may cause infinite loops.

**4 Computer experiment**

In this section, we give the detail of the results of a computer experiment conducted to compare our algorithm \( \text{FTS} \) and a simple algorithm \( \text{SMP} \) shown in Figure 4. Note that Algorithm \( \text{SMP} \) also has possibility to cause infinite loops. The computer experiment was carried out for an \( HS(n,k) \) where \( (n,k) = (9,2), (9,3), (9,4), (10,2), (10,3), \) and \( (10,4) \) changing the ratio of faulty nodes \( \alpha \) from 0.0 to 0.9, and we have measured the ratio of successful routings and their path lengths.

**Figure 4: A simple fault-tolerant routing algorithm.**

Concretely, first, in an \( HS(n,k) \), we selected faulty nodes randomly with the ratio \( \alpha \). Next, we selected the source node \( s \) and the destination node \( d \) from non-faulty nodes. Finally, after checking the connectivity of \( s \) and \( d \), we applied the fault-tolerant routing algorithms. If \( s \) and \( d \) are not connected, that is, there is no fault-free path between them, we start over from the selection of faulty nodes. For each pair of \( (n,k) \) and \( \alpha \), we executed at least 100,000 trials. Figures 5 to 10 show the ratios of successful routings by Algorithms \( \text{FTS} \) and \( \text{SMP} \), respectively. Also, Figures 11 to 16 show the average path lengths by Algorithms \( \text{FTS} \) and \( \text{SMP} \), respectively.

From these figures, we can see that Algorithm \( \text{FTS} \) shows better performance than Algorithm \( \text{SMP} \) in any pair of \( (n,k) \) with small amount of additional costs.

**5 Conclusions and future work**

In this paper, we have introduced the concept of safety levels in a hyper-star graph \( HS(n,k) \) and proposed a fault-tolerant routing algorithm. We have proved that the time complexity to calculate safety levels with respect to all the distances at each node is \( O(n^2) \). Moreover, we have carried out a computer experiment and verified high reachability to the destination nodes.

As a future work, it is interesting to introduce a stochastic framework into our method.
Acknowledgments

This study is partly supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science (JSPS) under Grant No. 25330079.

References


Figure 5: The ratios of successful routings by Algorithm FTS and SMP in an $HS(9, 2)$.

Figure 6: The ratios of successful routings by Algorithm FTS and SMP in an $HS(9, 3)$.

Figure 7: The ratios of successful routings by Algorithm FTS and SMP in an $HS(9, 4)$.

Figure 8: The ratios of successful routings by Algorithm FTS and SMP in an $HS(10, 2)$.

Figure 9: The ratios of successful routings by Algorithm FTS and SMP in an $HS(10, 3)$.

Figure 10: The ratios of successful routings by Algorithm FTS and SMP in an $HS(10, 4)$. 
Figure 11: The average path lengths by Algorithm **FTS** and **SMP** in an $HS(9, 2)$.

Figure 12: The average path lengths by Algorithm **FTS** and **SMP** in an $HS(9, 3)$.

Figure 13: The average path lengths by Algorithm **FTS** and **SMP** in an $HS(9, 4)$.

Figure 14: The average path lengths by Algorithm **FTS** and **SMP** in an $HS(10, 2)$.

Figure 15: The average path lengths by Algorithm **FTS** and **SMP** in an $HS(10, 3)$.

Figure 16: The average path lengths by Algorithm **FTS** and **SMP** in an $HS(10, 4)$.