Fault-Tolerant Routing on the Star Graph with Safety Vectors[#]

Sheng-I Yeh[*], Chang-Biau Yang[**], and Hon-Chan Chen[***]

[*] *Department of Applied Mathematics*
National Sun Yat-sen University

[**] *Department of Computer Science and Engineering*
National Sun Yat-sen University
Kaohsiung, Taiwan 80424
cbyang@cse.nsysu.edu.tw

[***] *Department of Shipping Transportation and Management*
National Kaohsiung Institute of Marine Technology,
Kaohsiung, Taiwan

Abstract

The concept of safety vector can guide efficient fault-tolerant routing on interconnection networks. The safety vector on the hypercube is based on the distance of a pair of nodes. However, the distance measure cannot be applied on star graphs directly, since there are many routing path patterns when the distances of two pairs of nodes are the same. Thus, on star graphs, we define the safety vector based on the routing path patterns. Based on the concept of routing path patterns, we first define the undirected safety vector, which is a one-dimensional vector on each node. In addition, we propose some methods for solving some problems on the safety vectors of the star graph, such as the length of safety vectors and the ranking of the routing path patterns.

Key words: interconnection network, star graph, routing, fault tolerant, safety vector

1. Introduction

In the design of a multi-processor system, the topology of the communication structure is important. The topology can be represented as a graph, in which each node stands for a processor and each edge stands for a communication channel between a pair of processors. The Cayley graph [4] offers a group-theoretic model to study the symmetric interconnection networks such as the hypercube Q_n and the star graph S_n. The *star graph* [1,4] has some good properties of the Cayley graph, and it has some better properties such as the degree and the diameter, comparing with the hypercube.

When nodes or links fail in an interconnection network, some paths may not be available, and the routing may take more links or may not be routed at all. In such a situation, we need fault-tolerant routing to maintain the performance of the system [2, 3, 5, 6, 7, 9, 12, 14, 15, 16, 18, 21]. Based on the information about faults, fault-tolerant routing can be classified as follows [20]:

1. *Global information based method* [13]: Each node knows where all faults are, and the best possible path of any two nodes is known at first. One important issue in this method is how to get the global information.
2. *Local information based method* [17]: Each node knows only the states of its neighboring nodes and links. The routing path between any two nodes is not known in advance. In this method, we can apply trial-and-error method like the depth-first search to find the routing paths. For keeping tracks, the message overhead is needed.
3. *Limited global information based method*: The information about faults around the neighborhood is gathered and processed so that a guide of the fault-distribution in the system is obtained. This guide is helpful to the routing, and then the length of any path can be computed. The gathering work in this method is not as difficult as the global information based method. Moreover, there is no much message overhead. It is a compromised method of the above two.

There are two kinds of limited global information based method. They are *safety level* [19] and *safety vector* (or *routing capability*) [20]. They both have been

[#] This research work was partially supported by the National Scienc Council of the Republic of China under contract NSC-89-2213-E-110-046.

studied on the hypercube. The safety vector can describe the fault-distribution more precisely. The safety level has been applied in the star graph [8], but it does not perform well in fault tolerant routing. In this paper, we shall propose an efficient safety vector in the star graph.

The rest of this paper is organized as follows. In Section 2, we review some properties of the star graph interconnection network and the safety level on star graphs. We also introduce the previous result of the safety vector on hypercubes. In Section 3, we first propose the safety vector on star graphs. In section 4, we present the method for computing the length of our safety vector and giving the ranking of each element. And finally, the conclusion is given in Section 5.

2. Preliminaries

In an n-dimensional star graph, denoted as S_n, each node is identified by a distinct permutation of n symbols. For two nodes u and v in S_n, if the identifier of u can be obtained from v by swapping the first symbol with another symbol of v, then there is a link connecting these two nodes.

Routing refers to sending a message from a *source* node to a *destination* node through some intermediate nodes in an interconnection network. The routing path should be as short as possible. According to the definition of a star graph, a routing path can be represented by the permutation from the source to the destination as a product of p-cycles, $p \geq 2$. Then, the routing path can be found by the cycles from right to left.

The *distance* of two nodes u and v in a graph is the length of the shortest path between u and v. The *diameter* of a graph is the maximum of all distances among all pairs of nodes in the graph. In S_n, the diameter has been proved to be $\left\lfloor \frac{3(n-1)}{2} \right\rfloor$ [1, 4].

Let π be the permutation from the source u to the destination v, and c be the number of cycles of size not smaller than 2. Denote 'A' to be the first symbol of the destination and 'X' to be any other symbol of the destination. Let m be the number of symbols of the source not at the correct positions of the destination. Moreover, let $u^{(i)}$ be the neighbor of node u got by swapping the first symbol with the ith symbol of u. Then, the distance $d(\pi)$ can be obtained as follows [1,4]:

$$d(\pi) = c + m - \begin{cases} 0, & \text{if } A \text{ is the same as the first} \\ & \text{symbol of the source,} \\ 2, & \text{otherwise.} \end{cases}$$

The neighbors on the shortest paths are called *preferred neighbors*, while the others are called *non-preferred neighbors*. Let Δc, Δm and Δd denote the changes of c, m and distance d, respectively. Table 1 shows the effect of moving a node to one of its neighbors, in which 'X' and 'X'' are any two symbols different from 'A'. We can use these *fault-free routing rules* to find the shortest paths. In Table 1, a neighbor in cases 1, 2, or 3 is a preferred neighbor, while a neighbor in cases 4 or 5 is a non-preferred neighbor.

Table 1. Fault-free routing rules.

Case	First Symbol	Method	New First Symbol	Δc	Δm	Δd
1	A	Swap A with any symbol X not at the correct position	X	0	+1	–1
2	X	Swap X to its correct position	A X'	–1 0	–2 –1	–1 –1
3	X	Swap X with any other symbol X' whose position is in another cycle	X'	–1	0	–1
4	X	Swap X with any other symbol X' or A whose position is in the same cycle	A X'	0 +1	–1 0	+1 +1
5	A X	Swap X or A with any other symbol X' already in the correct position	X'	+1 0	+2 +1	+1 +1

The *safety level* represents the ability to perform routing within some distance by shortest paths. In the star graph, the safety level of u contains the safety level $L_u[i]$ of each neighbor $u^{(i)}$, $2 \leq i \leq n$. The safety levels of S_n can be determined as follows [8]:

Step 1. For each node u, let $L_u[i] = 0$ for $i = 2, 3, \ldots, n$.

Step 2. Do the following computation for $\left\lfloor \frac{3(n-1)}{2} \right\rfloor$ times:

For each faultless node u, let

$$L_u[i] = \begin{cases} 0, & \text{if } u^{(i)} \text{ or } u \text{ is a faulty node, or} \\ & \text{if both } u \text{ and } u^{(i)} \text{ are two ends of a faulty link,} \\ \min(L_{u^{(i)}}[j] \mid j \in \{2,3,\ldots,n\} - \{i\}) + 1, & \text{otherwise,} \end{cases}$$

for $i = 2, 3, \ldots, n$.

Figure 1 shows an example of S_4 with faulty nodes 1324 and 3412.

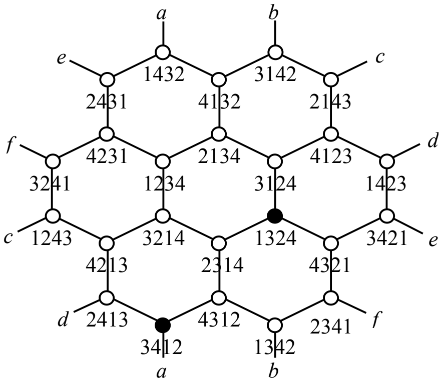

Figure 1. S_4 with faulty nodes 1324 and 3412.

Although the safety levels can keep the routing abilities within some distance via shortest paths, it cannot be sure whether one shortest path exists beyond that distance. The safety vectors, on the other hand, can keep the routing abilities more precisely at some exact distance. The safety vectors of a hypercube are introduced below, which is helpful to understand our safety vectors of a star graph.

In an n-dimensional hypercube, denoted as Q_n, each node is identified by a distinct binary string of length n, and there is a link connecting two nodes if their binary bit strings differ exactly at one position. For instance, there is a link connecting nodes 000 and 010. Let $u^{(i)}$ denote the neighbor of node u with a different bit at the ith position, $1 \leq i \leq n$. We say that node u is *k-Hamming distance* away from node v if there are k different bits between nodes u and v.

In computing the safety vectors for the hypercube, if a node is on one end of a faulty link, we consider the other end as a faulty node. The safety vector of node u in Q_n can be determined as follows [20]:

Step 1. If u is a faulty node, then let $u[k] = 0$ for $k = 1, 2, \ldots, n$, and the safety vector of u is determined completely.

Step 2. If u is not a faulty node, let

$$u[1] = \begin{cases} 0, \text{if } u \text{ is on one end of a faulty link,} \\ 1, \text{otherwise.} \end{cases}$$

Then, for $k = 2, 3, \ldots, n$, let

$$u[k] = \begin{cases} 0, \text{if } \sum_{1 \leq i \leq n} u^{(i)}[k\text{-}1] \leq n-k, \\ 1, \text{otherwise.} \end{cases}$$

For one source node, there are n different classes of destinations. In each class, the distances between the source and all destinations are the same. The safety vector of one node in the hypercube Q_n describes the routing abilities from it to n classes of destinations [10, 20].

3. Fault-tolerant routing by safety vectors in S_n

In the hypercube Q_n, we can find that each source node plays n kinds of *roles*, and each role corresponds to some destinations of a specific distance. For routing to one destination node, the source node has to route to one of its neighbors, and this neighbor becomes the new source node. The number of preferred neighbors for one role is different from that for another role. Therefore, the relations between all roles for all nodes describe all routing abilities. We can use a *path structure* to represent the relations of the hypercube Q_n, as shown in Figure 2.

$$(\ ) \xleftarrow{1} (1) \xleftarrow{2} \ldots \xleftarrow{k-1} (k-1) \xleftarrow{k} (k) \xleftarrow{k+1} \ldots \xleftarrow{n} (n)$$

Figure 2. The path structure of Q_n.

In the path structure, (k) represents the role of the source to the destinations of k-Hamming distance, and $(k-1) \xleftarrow{k} (k)$ means that (k) has k preferred neighbors of role $(k-1)$ with which the source can route to the same destination node. The other $n-k$ non-preferred neighbors are of role $(k+1)$. If one node has (k) routing ability, then it can surely route to any destination of k-Hamming distance via one shortest path. To compute the (k) routing ability of some node u, we have to check the $(k-1)$ routing abilities for all n neighbors of u. For all subsets of k neighbors of u, if there exists one subset having all $(k-1)$ routing ability, then u has (k) routing ability, and let $u[k] = 1$.

In the star graph, routing can be represented as a permutation, and the permutation can be represented as the product of p-cycles, $p \geq 2$. We call a product structure a *pattern*. Let 'A' denote the first symbol of the destination and 'X' denote any other symbol of the destination. Then, a pattern is denoted by a product structure with one 'A' and some 'X's. The permutations with the same product structure belong to the same pattern. For example, for the source nodes 2143 and 3412 and the destination node 1234, their product structures are $\binom{1234}{2143} = (21)(34)$ and $\binom{1234}{3412} = (31)(24)$, respectively. They belong to the same pattern $(XA)(XX)$. A pattern represents a role of one source node. By fault-free routing rules, permutations with the same pattern have the similar shortest paths. However, the relations between all roles for all nodes cannot be described as the path structure of Q_n. For example, from nodes 2143, 3412, and 4321 to node 1234, the permutations are $\binom{1234}{2143} = (21)(34)$, $\binom{1234}{3412} = (31)(24)$, and $\binom{1234}{4321} = (41)(23)$, respectively. They all have the same pattern $(XA)(XX)$ and are of distance 4. Their shortest paths are similar, as shown in Figure 3. In addition, from nodes 1342 and 1423 to node 1234, the permutations are $\binom{1234}{1342} = (234)$ and $\binom{1234}{1423} = (243)$, respectively. They both have the same pattern (XXX) and are of distance 4, and their shortest paths are similar, as shown in Figure 4. We can observe that patterns on different shortest paths may be all different even when the distance is identical. Therefore, we need to find another path structure for the start graph S_n.

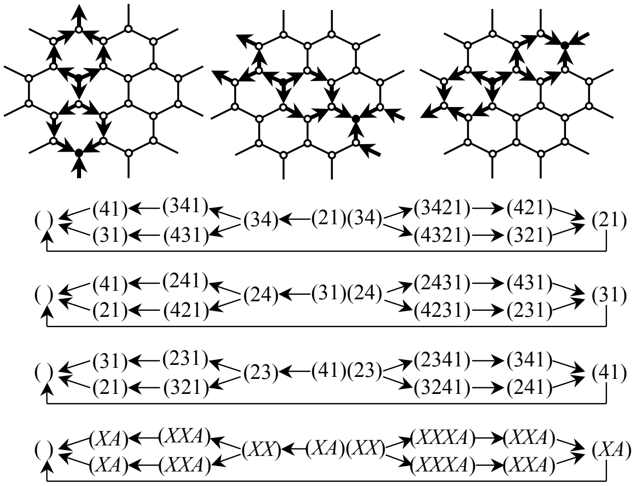

Figure 3. All shortest paths from the sources of pattern $(XA)(XX)$ in S_4.

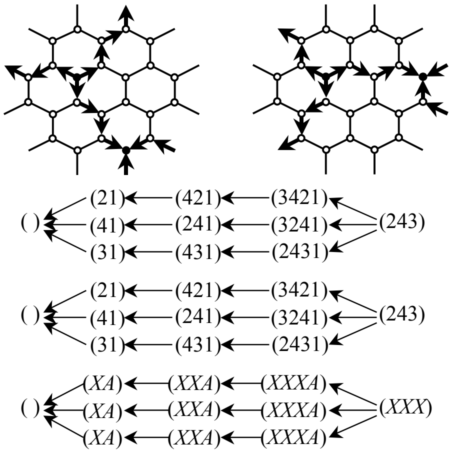

Figure 4. All shortest paths from the sources of pattern (XXX) in S_4.

Using the fault-free routing rules, we can find the relations between all patterns. Let α be a pattern. Given distance d and the first symbol of the source, we can compute α. For instance, in S_4, if $d(\alpha) = 3$ and the first symbol of the source is 'A', then $\alpha = (XX)$. The reason is that $m \geq 2c$ and $c + m = 3$, then we obtain $c = 1$, $m = 2$ and $\alpha = (XX)$. On the other hand, if $d(\alpha) = 3$ and the first symbol of the source is not 'A', then $\alpha = (XXXA)$ because $m \geq 2c$ and $c + m - 2 = 3$, then we get $c = 1$, $m = 4$ and $\alpha = (XXXA)$. With this computation, we can obtain all patterns in S_4, which are ( ), (XA), (XXA), $(XXXA)$, (XX), (XXX), and $(XA)(XX)$, where ( ) means that the source and the destination is the same node.

Continue discussing the patterns of S_4. If we swap the first symbol 'A' with any symbol not in the correct position, then we have $(XXXA)\xleftarrow{3}(XXX)$, $\Delta c = 0$, and $\Delta m = 1$. If we swap the first symbol 'X' with some symbol in another cycle, then we have $(XXXA)\xleftarrow{2}(XA)(XX)$, $\Delta c = -1$, and $\Delta m = 0$. In addition, if we swap the first symbol 'X' to its correct position, then we obtain $(XX)\xleftarrow{1}(XA)(XX)$, $\Delta c = -1$, and $\Delta m = -2$. We can use the reasoning to get a *path structure* of S_4 as Figure 5 to describe the different roles and their relations.

$$\begin{tikzcd}
( ) & (XA) \arrow[l, "1"'] & (XXA) \arrow[l, "1"'] & (XXXA) \arrow[l, "1"'] & (XXX) \arrow[l, "3"'] \\
 & & & (XX) \arrow[ul, "2"] & (XA)(XX) \arrow[l, "1"'] \arrow[ul, "2"]
\end{tikzcd}$$

Figure 5. The path structure of S_4.

With the path structure of Figure 5, we can find the shortest paths in S_4. For example, $(XA)(XX)$ represents one role of a source u to a destination v. Since $(XXXA)\xleftarrow{2}(XA)(XX)$ and $(XX)\xleftarrow{1}(XA)(XX)$, u has three preferred neighbors routing to v, where two of them are of role $(XXXA)$, and the other is of (XX). We call patterns $(XXXA)$ and (XX) the *preferred patterns* of $(XA)(XX)$. In the star graph, for one source node u of pattern α, the preferred patterns of α are the patterns of the preferred neighbors of u. If a node has the routing ability of pattern α, then there is at least one shortest path to destinations through some preferred pattern of α.

Denote $d(\alpha)$ the distance from a source of pattern α to the destinations, and denote $p(\alpha)$ the set of the preferred patterns of α. That is, if $\alpha' \in p(\alpha)$, then $d(\alpha') = d(\alpha) - 1$. We also denote $n(\alpha, \alpha')$ the number of preferred neighbors (of pattern α') of a node (of pattern α). Let $|\Omega|$ be the number of elements in a set Ω. Moreover, if one node is on one end of a faulty link, then we consider the other end as a faulty node. The undirected safety vector of one node u of S_n can be defined as follows.

Definition 3.1. *Let α be a pattern and $\Omega_{u,\alpha} = \{\, i \mid 2 \leq i \leq n,\ u^{(i)}[\alpha] = 0\}$. If u is a faulty node, then let $u[\alpha] = 0$ for all α, and the undirected safety vector of u is determined. If u is not a faulty node, then let*

$$u[(XA)] = \begin{cases} 0, \textit{if } u \textit{ is on one end of a faulty link}, \\ 1, \textit{otherwise}. \end{cases}$$

Then, for all α, $\alpha \neq (XA)$, let

$$u[\alpha] = \begin{cases} 1, \textit{if } \exists\, \alpha' \in p(\alpha), |\Omega_{u,\alpha'}| < n(\alpha, \alpha'), \\ 1, \textit{if } (\forall\, \alpha' \in p(\alpha), |\Omega_{u,\alpha'}| \geq n(\alpha,\alpha')) \textit{ and} \\ \quad |\cup_{\alpha' \in p(\alpha)} \Omega_{u,\alpha'}| < \sum\limits_{\alpha' \in p(\alpha)} n(\alpha,\alpha'), \\ 0, \textit{otherwise}. \end{cases}$$

To obtain $u[\alpha]$ for each node u, we have to get $u^{(i)}[\alpha']$ and let $d(\alpha) = d(\alpha') + 1$ from all neighbors, where $2 \le i \le n$ and $\alpha' \in p(\alpha)$. The routing ability for distance 1 is computed at first, then for distance 2, 3, …, $\left\lfloor \frac{3(n-1)}{2} \right\rfloor$. Therefore, the safety vectors are computed simultaneously and iteratively. We have some results about the undirected safety vectors of S_n.

Property 3.2. *For one node u, if the nearest fault is k-distance away from u, then $u[\alpha] = 1$ for all patterns α with $d(\alpha) < k$.*

Theorem 3.3. *For one node u of pattern α with $d(\alpha) = k$ to each destination v, if $u^{(i)}$ is a preferred neighbor and $u^{(i)}[\alpha'] = 1$, $\alpha' \in p(\alpha)$, then there is a shortest path of length k from u to v through $u^{(i)}$.*

Corollary 3.4. *For one node u of pattern α' with $d(\alpha') = k$ to each destination v, if $u^{(i)}$ is a non-preferred neighbor and $u^{(i)}[\alpha] = 1$, $\alpha' \in p(\alpha)$, then there is a non-shortest path of length k+2 from u to v through $u^{(i)}$.*

Based on Theorem 3.3, we can develop a fault-tolerant routing algorithm of undirected safety vectors on star graphs. Let u and v be the source and the destination, respectively. Denote $PCP(u, v)$ the pattern from u to v, and denote $NDP(u,v)$ the number of shortest node-disjoint paths from u to v. Besides, let $MD(u, v)$ be $\max\{d(\alpha) \mid u[\alpha] = 1,\ d(\alpha) \le d(PCP(u, v))\}$. Our fault-tolerant routing algorithm is described below.

```
Algorithm A /* Fault-tolerant routing algorithm of undirected safety vectors */
if $u = v$, then stop routing
/* Optimal path guaranteed */
else if there exists one preferred neighbor $u^{(i)}$ such that
    $u^{(i)}[PCP(u^{(i)}, v)] = 1$, then send messages to $u^{(i)}$
/* Suboptimal path guaranteed */
    else if there exists one non-preferred neighbor $u^{(i)}$
        such that $u^{(i)}[PCP(u^{(i)}, v)] = 1$ and $PCP(u^{(i)}, v)$
        $\ne (X\ldots XA)$, then send messages to $u^{(i)}$
/* Nothing of certainty */
        else if there exists one faultless preferred
            neighbor $u^{(i)}$ such that $MD(u^{(i)}, v) \ge$
            $MD(u^{(j)}, v)$ and $NDP(u^{(i)}, v) \ge NDP(u^{(j)}, v)$,
            where $u^{(j)}$ is any other preferred
            neighbor, then send messages to $u^{(i)}$
            else if there exists one faultless
                non-preferred neighbor $u^{(i)}$
                such that $NDP(u^{(i)}, v) \ge$
                $NDP(u^{(j)}, v)$, where $u^{(j)}$ is any
                other non-preferred neighbor,
                then send messages to $u^{(i)}$.
$u = u^{(i)}$.
```

4. The Length and the Ranking

The length of a safety vector is the same as the number of all considered patterns. It can be obtained from the number of partitions of a positive integer [11]. We denote ϕ_k the number of partitions of a positive integer k. For instance, $\phi_4 = 5$ since 4 can be partitioned into (4), (1+3), (2+2), (1+1+2), and (1+1+1+1). Let P_n be the number of all possible patterns in S_n. Then, $P_n = \sum_{k=1}^{n} \phi_{n-k}$, where ϕ_{n-k} corresponds to all possible patterns beginning with $\underbrace{(X \cdots XA)}_{k}$.

When the length becomes larger, it is getting difficult to look up some routing ability in a safety vector. Thus, we need a ranking method for all patterns so that each pattern is associated with a distinct number with which we can find its position in one safety vector. Let $\phi_{k,q}$ be the number of partitions of a positive integer k with constraint q, in which each partition is of size 1 or less than q. We rank patterns in S_n by the following rule:

(1) Each pattern α is of the format $\alpha = \underbrace{(X \cdots XA)}_{k_1} \underbrace{(X \cdots X)}_{k_2} \ldots \underbrace{(X \cdots X)}_{k_{C_\alpha}}$, where $k_2 \ge k_3 \ge \ldots \ge k_{C_\alpha}$.

(2) For two patterns α and β, where $\alpha = \underbrace{(X \cdots XA)}_{k_1} \underbrace{(X \cdots X)}_{k_2} \ldots \underbrace{(X \cdots X)}_{k_{C_\alpha}}$ and $\beta = \underbrace{(X \cdots XA)}_{h_1} \underbrace{(X \cdots X)}_{h_2} \ldots \underbrace{(X \cdots X)}_{h_{C_\beta}}$, the rank of α is larger than the rank of β if $k_1 k_2 \cdots k_{C_\alpha} < h_1 h_2 \cdots h_{C_\beta}$ in lexicographic order of strings.

Table 2 lists the ranks of all patterns in S_7.

Table 2. Ranking patterns of S_7.

Rank	Pattern	Rank	Pattern	Rank	Pattern
1	(*XXXXXXA*)	12	(*XXA*)	23	(*A*)(*XXXX*)
2	(*XXXXXA*)	13	(*XA*)(*XXXXX*)	24	(*A*)(*XXX*)(*XXX*)
3	(*XXXXA*)(*XX*)	14	(*XA*)(*XXXX*)	25	(*A*)(*XXX*)(*XX*)
4	(*XXXXA*)	15	(*XA*)(*XXX*)(*XX*)	26	(*A*)(*XXX*)
5	(*XXXA*)(*XXX*)	16	(*XA*)(*XXX*)	27	(*A*)(*XX*)(*XX*)(*XX*)
6	(*XXXA*)(*XX*)	17	(*XA*)(*XX*)(*XX*)	28	(*A*)(*XX*)(*XX*)
7	(*XXXA*)	18	(*XA*)(*XX*)	29	(*A*)(*XX*)
8	(*XXA*)(*XXXX*)	19	(*XA*)	30	(*A*)
9	(*XXA*)(*XXX*)	20	(*A*)(*XXXXXX*)		
10	(*XXA*)(*XX*)(*XX*)	21	(*A*)(*XXXXX*)		
11	(*XXA*)(*XX*)	22	(*A*)(*XXXX*)(*XX*)		

Lemma 4.1. *Let* $\alpha = \underbrace{(X\cdots XA)}_{k_1}\underbrace{(X\cdots X)}_{k_2}\ldots\underbrace{(X\cdots X)}_{k_C}$ *be a pattern, where* $k_2 \geq k_3 \geq \ldots \geq k_C$. *Then, the rank of* α *is* $\sum_{k=k_1+1}^{n}\phi_{n-k} + \sum_{i=2}^{C}\sum_{k=k_i+1}^{n-k_1-\cdots-k_{i-1}}\phi_{n-k_1-\cdots k_{i-1}-k,k} + 1$.

5. Conclusion

We proved that our safety vector is better in judging the routing paths than the safety level. But the safety vector requires more space. Although more information we get, the process is more complex than the safety level. It is a trade-off between the information and the complexity.

The basic property of safety vectors used for routing is if a routing ability is 1, it is sure that there exists one shortest path from the source node to all corresponding destinations. But if the routing ability is 0, it is not sure if there exists such a shortest path.

A good design of safety vectors is to increase the number of 1's in the vector so that it is near the optimal number. However, to get optimal safety vector is very difficult. One of the methods to get improvement is to design *directed safety vectors,* which is a two-dimensional vector on each node that stores the routing capability of each of its neighboring nodes. This method will increase the required space in each node.

Another approach to increase the accuracy is the concept of *statistical safety vectors*. With this concept, the value of each element of the vector on each node is not only 0 or 1. Instead, it is set between 0 and 1, which means the probability of successful routing ability from the node to all corresponding nodes. This will be worthy of further study.

References

[1] S. B. Akers, D. Horel, and B. Krishnamurthy, "The star graph: An attractive alternative to the *n*-cube," *Proceeding of the International Conference on Parallel Processing*, pp. 393-400, 1987.

[2] S. B. Akers and B. Krishnamurthy, "The fault tolerance of star graphs," *Proceedings of the 2nd International Conference on Supercomputing, San Franciso*, pp. 270-276, 1987.

[3] S. B. Akers and B. Krishnamurthy, "On group graphs and their fault-tolerance," *IEEE Transactions on Computers*, Vol. 36, No. 7, pp. 885-888, 1987.

[4] S. B. Akers and B. Krishnamurthy, "A group-theoretical model for symmetric interconnection networks," *IEEE Transactions on Computers*, Vol. 38, No. 4, pp. 555-566, Apr. 1989.

[5] N. Bagherzadeh and M. Nassif, "Computation in faulty stars [hypercube networks]," *IEEE Transactions on Reliability*, Vol. 44, No. 1, pp. 114-119, 1995.

[6] N. Bagherzadeh, M. Nassif, and S. Lati, "A routing and broadcasting scheme on faulty star graphs," *IEEE Transactions on Computers*, Vol. 42, No. 11, pp. 1398-1403, Nov. 1993.

[7] L. Q. Bai, H. Ebara, and H. Nakano, "An efficient adaptive routing algorithm for the faulty star graph," *Proceeding of the International Conference on Parallel and Distributed Systems*, pp. 82-87, 1997.

[8] C. H. Chang, "A study of fault-tolerant routing on star graph using limited-global-information," Master's thesis, National Taiwan University of Science and Technology, 1997.

[9] C. C. Chen and J. Chen, "Vertex-disjoint routings in star graphs," *IEEE First International Conference on Algorithms and Architectures for Parallel Processing*, Vol. 1, pp. 460-464, 1995.

[10] G. M. Chiu and K. S. Chen, "Fault-tolerant routing strategy using routing capability in hypercube multicomputers," *Proceeding of the International Conference on Parallel and Distributed Systems*, pp. 396-403, 1996.

[11] R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete mathematics : *a foundation for computer science*. Addison-Wesley, second ed., 1994.

[12] Q. P. Gu and S. Peng, "The shortest routing path in star graphs with faulty clusters," *Proceedings of Second Aizu International Symposium Parallel Algorithms/Architecture Synthesis*, pp. 91-96, 1997.

[13] Q. P. Gu and S. Peng, "Node-to-node cluster fault routing in star graph," *Information Processing Letters*, Vol. 56, pp. 29-35, 1995.

[14] S. C. Hu and C. B. Yang, "Fault tolerance on star graphs," *International Journal of Foundations of Computer Science*, Vol. 8, No. 2, pp. 127-142, 1997.

[15] C. Liang, S. Bhattacharya, and J. Tan, "Performance evaluation of fault-tolerant routing on star networks," *Proceedings of the Scalable High-Performance Computing Conference*, pp. 650-657, 1994.

[16] C. Liang, S. Bhattacharya, and J. Tan, "Design and analysis of fault-tolerant star networks," *Proceeding of the International Conference on Parallel Processing*, pp. 346-349, 1997.

[17] N. Nassif, N. Bagherzadeh, and S. Lati, "A routing and broadcasting scheme on faulty star graph," *IEEE Transactions on Computers*, Vol. 42, No. 11, pp. 1398-1403, 1993.

[18] C. P. Ravikumar and A. M. Goel, "Deadlock-free wormhole routing algorithms for star graph topology," *IEE Proceedings Computers and Digital Techniques*, pp. 395-400, 1995.

[19] J. Wu, "Unicasting in faulty hypercubes using safety levels," *IEEE Transactions on Computers,* Vol. 46, No. 2, pp. 241-244, Feb. 1997.

[20] J. Wu, "Adaptive fault-tolerant routing in cube-based multicomputers using safety vectors," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 9, No. 4, pp. 321-334, Apr. 1998.

[21] C. B. Yang and T. H. Liu, "Wormhole routing on the star graph interconnection network," *Proceeding of the 4th Australian Theory Symposium, CATS'98*, pp. 51-65, 1998.