JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 16, 25-40 (2000)
DOI:10.1688/JISE.1999.16.1.2

A Fault-Tolerant Reconfiguration Scheme In the Faulty Star Graph*

YUH-SHYAN CHEN AND JANG-PING SHEU[+]
Department of Computer Science and Information Engineering
Chung-Hua University
Hsinchu, Taiwan 300, R.O.C.
E-mail: chenys@csie.chu.edu.tw
[+]*Department of Computer Science and Information Engineering*
National Central University
Chungli, Taiwan 320, R.O.C.
E-mail: sheujp@csie.ncu.edu.tw

In this paper, we propose a scheme to identify the maximal fault-free substar-ring. This is the first attempt to derive a reconfiguration scheme with high processor utilization in the faulty n-star graph. The maximal fault-free substar-ring is connected by a ring of fault-free *virtual substars* and the maximal length of the ring is $n(n-1)(n-2)$. Our proposed scheme can tolerate $n-3$ faults such that the processor utilization is $\frac{n^2-2n+3}{n^2-n}$. This is a near optimal result since the maximal fault-free substar-ring is constructed using all of the possible fault-free $(n-2)$-substars. Moreover, our algorithm can still work when the number of faults exceeds $n-3$. The simulation results also show that the processor utilization is more than 50% if the number of faults is less than $\frac{n^2-n-1}{2}$ in the n-star graph.

Keywords: fault tolerance, interconnection network, parallel processing, reconfiguration, star graph

1. INTRODUCTION

Recently, one new interconnection network that has attracted lots of attention is the star graph [5]. The star graph [5], being a member of the class of Cayley graphs, has been shown to possess appealing features, including low degree of the node, small diameter, partitionability, symmetry, and high degree of fault tolerance. Especially when the size of the star graph system increases, fault tolerance is an important issue for such a large system to continue operation after failure of one or more processors/links. In this paper, we study how algorithms that are originally designed for the fault-free star graph can be implemented on star graphs containing faults with high processor utilization. To measure the efficiency

Received October 23, 1997; revised March 27, 1998; accepted May 13, 1998.
Communicated by Shing-Tsaan Huang.
*A preliminary version of this paper appeared in *the Proceedings of the IEEE Second International Conference on Algorithms & Architectures for Parallel Processing (ICA3PP'96)*.
*This research was supported in part by the National Science Council, R.O.C., under Grant number NSC86-2213-E-216-022.

of the fault-tolerant approach, we use *processor utilization ratio* or PUR, which is the total number of fault-free processors used by our reconfiguration scheme divided by the total number of processors of an n-star graph.

Much research recently has been directed toward studying the aspects of fault-tolerant computing on the hypercube and star graph [9, 11, 12, 19, 22-25]. The hierarchy of both the hypercube and star graph allow assignment of their special subgraphs, subcubes and substars, which have the same topological features as the original graph. Most of the fault-tolerant strategies address the issue of *reconfiguration* once the faulty processors/links are identified. One effective approach used in the reconfiguration strategies in hypercubes is to identify the largest fault-free subcube and use the subcube to emulate the entire hypercube [11]. An n-star graph can be recursively decomposed into $n(n-1)$-substars. The largest fault-free substar can be easily identified and used to emulate the entire star graph. It is unreasonable to use the largest fault-free substar approach as a reconfiguration scheme since its PUR becomes at most $1/n$ even when an n-star graph contains only one fault. A different but related research topic is how to allocate tasks in the complete star graph [15]. Note that this approach can better accommodate multiple jobs on substars of different sizes. Therefore, the purpose of this paper is to provide a novel fault-tolerant reconfiguration scheme in a n-star graph so that higher PUR and lower diameter can be obtained.

The major contribution of this paper is that we propose an efficient reconfiguration scheme, which can tolerate $f \le n-3$ faults in n-star graph, such that PUR is $\frac{n^2-2n+3}{n^2-n}$ and the diameter is $O(n^3)$. Furthermore, our scheme with reasonable PUR can still work when number of faults exceeds $n-3$. Based on our simulation results, our algorithm keeps more than 50% PUR if the number of faults is less than $\frac{n^2-n-1}{2}$ in an n-star graph. In addition, a novel communication pattern with constant time cost will be presented in this paper for the sake of easily performing algorithm on the maximal fault-free substar-ring. In this paper, we only consider node faults, and in an edge fault, it is assumed that one of the nodes incident upon it is faulty. We also assume that faulty nodes can neither perform calculations nor route data.

The rest of this paper is organized as follows. The primary properties of the maximal fault-free substar-ring are introduced in section 2. A systematic technique for identifying the maximal fault-free substar-ring is addressed in section 3. The performance of our approach is analyzed in section 4. Finally, conclusions are drawn in section 5.

2. PRELIMINARIES

An n-dimensional star graph, also referred to as n-star or S_n, is an undirected graph consisting of $n!$ nodes (or vertices) and $(n-1)n!/2$ edges. Each node is uniquely assigned a label $x_1x_2 \ldots x_n$, which is the concatenation of a permutation of n distinct symbols $\{x_1, x_2, \ldots, x_n\}$. Without loss of generality, let these n symbols be $\{1, 2, \ldots, n\}$. Given any node label $x_1 \ldots x_i \ldots x_n$, let the permutation function g_i, $2 \le i \le n$, be such that $g_i(x_1 \ldots x_i \ldots x_n) = x_i \ldots x_1 \ldots x_n$(i.e., swap x_1 and x_i and keep the rest of the symbols unchanged). In S_n, for any node x, there is an edge joining x and node $g_i(x)$, and this edge is said the dimension i. Each node in

S_n is connected to $n - 1$ adjacent nodes by $n - 1$ edges. Let F denote a set of faulty nodes in a faulty S_n. An S_n with set F is denoted as S_n^F. For instance, a star graph S_5 with set F = {52341, 43152}, or $S_5^{\{52341,\ 43152\}}$, is shown in Fig. 1.

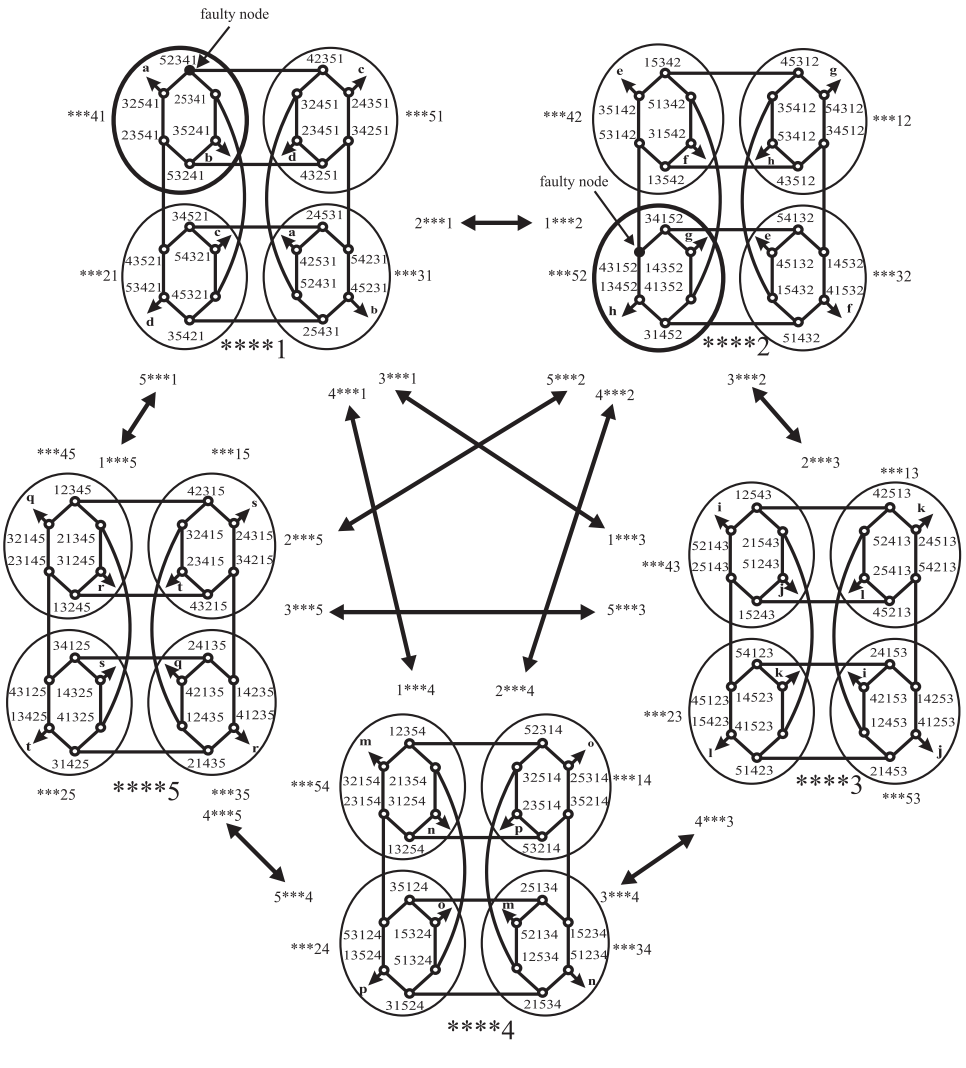

Fig. 1. A star graph S_5 with set F = {52341, 43152}.

Each S_n contains n disjoint S_{n-1}'s. Let $\Gamma = \{1, 2, ..., n, *\}$, where * denotes a *don't care* symbol. Every substar of S_n can be uniquely labeled by a string of symbols in Γ such that the only repeated symbol is *. Formally, a k-dimensional substar, S_k or k-substar, is denoted as a string $G = x_1x_2 ...x_n$, and number of * symbols in string G is k, where x_1 = * and $x_i \in \Gamma$,

$2 \le i \le n$. The substar represented by G is a subgraph of S_n containing all the vertices obtained from G by replacing each * with the digits {1, 2, ..., n}. These vertices are connected by the original links in S_n. For instance, the **3*1 is a 3-dimensional substar and contains the set of nodes {54321, 45321, 52341, 25341, 42351, 24351}. Throughtout this paper, a k-substar is said to be faulty if there exists at least one faulty node in the k-substar, where $1 \le k \le n$. Otherwise, the k-substar is said as fault-free. For example, in Fig. 1, the substars ***41 and ***52 are faulty substars.

Definition 1: ***j*-split** and ***D*-split**

Let $G = x_1x_2 \ldots x_j \ldots x_n$ be a k-substar with $x_j = *$. The j-split on G, $1 \le j \le n$, partitions G along the j-dimension into k number of $(k-1)$-substars, each obtained from G by replacing x_j with a legal non-$*$ symbol. Let $D = \{d_1, d_2, ..., d_m, 1\}$, $m \le k$, be a set of dimensions such that $x_{d_i} = *$, $i = 1..m$. Then, the D-split on G is used to first apply a d_1-split on G, whose result is then applied to a d_2-split, whose result is then applied to a d_3-split, etc., until there is $k(k-1) \ldots (k-m+1)$ number of $(k-m)$-substars. The final result of the D-split on G is obtained by applying a 1-split on each of the $(k-m)$-substars. □

In the above definition, if $j = 1$, then the partitioning result does not remain substars, which is defined as a *virtual substar* in the following. An S_n can be decomposed into $n(n-1) \ldots (k+1)$ copies of k-substar after applying D'-split on S_n, where $D' = \{d_1, d_2, ..., d_i, ..., d_{n-k}\}$, for all $d_i \ne 1$, $1 \le i \le n-k$, and $k \le n$. Given a k-substar S_k, 1-split on S_k is used to decompose S_k into k *virtual substars* represented as X_i, $1 \le i \le k$. The *virtual substar* $X_i = x_1x_2 \ldots x_n$ is a subgraph of S_k containing all the vertices obtained from S_k, where x_1 is a non-$*$ symbol and the number of * symbols of X_i is $k-1$. These vertices are connected as follows. Assume that $d'_1 < d'_2 < d'_3, \ldots, < d'_{k-1} \in \{1, 2, ..., n\} - \{d_1, d_2, ..., d_{n-k}\}$, where $x_{d'_1} = x_{d'_2} = \cdots x_{d'_{k-1}} = *$. For any node x in X_i, there is a *virtual edge* joining x and $g_{d'_1}(g_{d'_j}(g_{d'_1}(x)))$, $2 \le j \le k-1$, and this *virtual edge* is called the *virtual dimension* j. Each node in X_i is connected to $k-2$ *virtual adjacent nodes* by $k-2$ *virtual edges*. For a fixed *virtual dimension* j, $1 \le j \le k$, each disjoint node of X_i, in parallel, performs the same permutation functions $g_{d'_1}$, $g_{d'_j}$, and $g_{d'_1}$ simultaneously. Under the assumption of the bidirectional link, all paths from nodes of X_i, $1 \le i \le k$, to one of its *virtual adjacent nodes* are edge-disjoint. Moreover, each node of X_i, $1 \le i \le k$, sending data to one of its *virtual adjacent nodes* needs 3 time steps.

Consider any pair of *virtual substars* X_i and X_j obtained from 1-split on S_k, where $i \ne j$. Each node $rx_2x_3 \ldots x_n$ of X_i has a direct link to node $sy_2y_3 \ldots y_n$ of X_j along the dimension u such that $x_u = s$ and $y_u = r$, where r and s are non-$*$ symbols. That is, if we say the relation of the direct link is represented by lines connecting *virtual substar*, then, each *virtual substar* X_i of S_k is fully connected with every other X_j of S_k.

For instance, given an S_6, a 5-split on G is done to partition G into six 5-substars: *****1, *****2, *****3, *****4, *****5, and *****6. Consider that substar ****56 is one of substars after applying D'-split on G, where $D' = \{5, 6\}$. The 1-split on ****56 is done to decompose ****56 into *virtual substars*, 1***56, 2***56, 3***56, and 4***56, as shown in Fig. 2. Therefore, $d'_1 = 2, d'_2 = 3,$ and $d'_3 = 4$. Fig. 2 indicates that *virtual edges* joining x and $g_2(g_3(g_2(x)))$ along *virtual dimension* 2, where x in *virtual substars*, 1***56, 2***56, 3***56, and 4***56. After performing the permutation function, the number of nodes with different colors is always equal to 6, so all paths from x to $g_2(g_3(g_2(x)))$ are edge-disjoint. Node 123456 of *virtual substar* 1***56 has direct links to 213456, 321456, and 423156 of *virtual substars* 2***56, 3***56, and 4***56, respectively. The other nodes of

1***56 also have links to nodes in 2***56, 3***56, and 4***56. Therefore, *virtual substar* 1***56 is fully connected to 2***56, 3***56, and 4***56. Similarly, *virtual substars* 2***56, 3***56, and 4***56 are fully connected to each other.

Definition 2: Adjacent substars

Given any two k-substars, $G = x_1x_2...x_i...x_n$ and $H = y_1y_2...y_i...y_n$ are said to be adjacent if and only if the labels of G and H differ in exactly one dimension, where $1 \leq i \leq n$. □

For instance, the 3-substar G = ***31 is adjacent to H = ***41 but not adjacent to H' = ***13. In the following, we will describe the adjacent relation when the j-split operation is applied. For two given adjacent k-substars, $G = x_1x_2 \ldots x_j \ldots x_n$ and $H = y_1y_2 \ldots y_j \ldots y_n$ such that $x_j = y_j = *$. If we apply, the j-split on G and H, we will obtain k substars (dimension k – 1) from each of G and H. One can easily see that all the k substars in G are adjacent to each other, and so are those in H. If the adjacent relations are represented by of the lines connecting substars, each (k – 1)-substars of G(or in H) are fully connected to each of the other substars of G(or in H). Furthermore, among these substars, k – 1 substars in G are adjacent to k – 1 substars in H in a one-to-one manner. For example, if we apply 5-split on $G = S_5$ as shown in Fig. 1, ****1 is fully connected to ****2, ****3, ****4, and ****5. If G = ****3, H = ****4 and we apply 4-split on ****3 and ****4, we can see that each substar of ***13, ***23, ***43, and ***53 is fully connected to the others. Similarly, substars ***14, ***24, ***34, and ***54 are also fully connected. Moreover, 3 pairs of substars (***13, ****14), (***23, ****24), and (***53, ***54) are adjacent. Given a sequence of k-substars $[G_0, G_1, ..., G_{t-1}]$, a (k, t)-**ring** will be defined below before we construct our reconfiguration scheme.

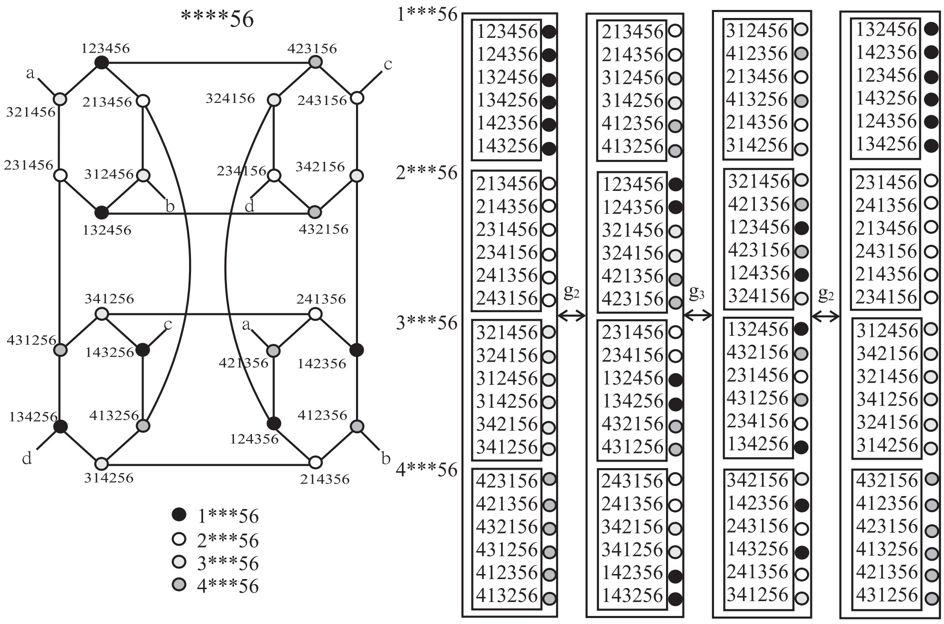

Fig. 2. The *virtual edges* joining x and $g_2(g_3(g_2(x)))$ along *virtual dimension* 2, where x in *virtual substars* 1***56, 2***56, 3***56, and 4***56.

Definition 3: (k, t)-**ring**

A sequence of k-substars $[G_0, G_1, ..., G_{t-1}]$ is denoted as an (k, t)-**ring** if substars G_i is adjacent to its neighboring $G_{(i-1)\ mod\ t}$ and $G_{(i+1)\ mod\ t}$ for any $i = 0..t - 1$. □

Before defining our final reconfiguration, we give the following lemma.

Lemma 1: Given a (k, t)-**ring** $= [G_0, G_1, ..., G_{t-1}]$, a feasible $(k - 1, kt)$-**ring** can be constructed from the (k, t)-**ring**.

***Proof*:** Let j be an integer such that the j-th symbol of all G_i, $0 \le i \le t - 1$, in the (k, t)-**ring** is $*$, each G_i is applied the j-split to obtain $k(k - 1)$-substars. All the $(k - 1)$-substars are fully connected by the adjacent relation, and there are $k - 1$ connections between G_i and G_{i-1} (and G_{i+1}). We can easily derive a feasible $(k - 1, kt)$-**ring** by visiting all $(k - 1)$-substars of G_i and one of the $k - 1$ connections between G_i and G_{i-1} (and G_{i+1}). □

The 1-split operation is performed on each G_i of a (k, t)-**ring** $= [G_0, G_1, ..., G_i, ..., G_{t-1}]$ to obtain a sequence of *virtual substars* $[X_0, X_1, ..., X_i, ..., X_{h-1}]$. As mentioned earlier, each X_i is a *virtual substar*. Based on Lemma 1, our final reconfiguration scheme, namely the *fault-free substar-ring* $R_s(k - 1, h)$, is defined as follows.

Definition 4: Fault-free substar-ring $R_s(k - 1, h)$

Let $R_s(k - 1, h) = [X_0, X_1, ..., X_i, ..., X_{h-1}]$ denote a feasible fault-free substar-ring, where $[X_0, X_1, ..., X_i, ..., X_{h-1}]$ is a sequence of disjoint fault-free *virtual substars* of dimension $k - 1$. $R_s(k - 1, h)$ is constructed by each node in X_i connected to a node in $X_{(i-1)\ mod\ h}$ and $X_{(i+1)\ mod\ h}$ with at most dilation 3, for all $0 \le i \le h - 1$. Therefore, $R_s(k - 1, h)$ is

$$X_0 \leftrightarrow X_1 \leftrightarrow X_2 \leftrightarrow ...\ X_{h-2} \leftrightarrow X_{h-1} \leftrightarrow X_0.$$ □

If the connection $X_{h-1} \leftrightarrow X_0$ does not exist, then a fault-free substar-chain, denoted by $C_s(k - 1, h)$, is constructed. Obviously, a fault-free substar-ring $R_s(k - 1, h)$ can be treated as a fault-free substar-chain $C_s(k - 1, h)$ with the same size of *virtual substars*. The processor utilization of $R_s(k - 1, h)$ and $C_s(k - 1, h)$ is $(k - 1)! \times h$. We can describe the diameter of $R_s(k - 1, h)$ and $C_s(k - 1, h)$ as follows. First, if each virtual substar is seen as a unit, then there are h virtual substars X which form a ring, so $\left\lfloor \frac{h}{2} \right\rfloor$ is the diameter of the ring of virtual substars. In each X_{k-1} of $R_s(k - 1, h)$, we can only use 3 steps to jump to the next virtual substar since all of the edges of G_i are nonfaulty if X_{k-1} is obtained by 1-split on G_i, so $\left\lfloor \frac{3h}{2} \right\rfloor$ is needed. When we arrive at the final virtual substar X_{k-1}, we still need at most $3 \times \left\lfloor \frac{3}{2}(k-2) \right\rfloor$ steps to arrive at any nodes in X_{k-1} since $\left\lfloor \frac{3}{2}(k-2) \right\rfloor$ is the diameter of S_{k-1}. Therefore, the diameters of $R_s(k - 1, h)$ and $C_s(k - 1, h)$ are at most $\left\lfloor \frac{9}{2}(k-2) \right\rfloor + \left\lfloor \frac{3h}{2} \right\rfloor$ and $\left\lfloor \frac{9}{2}(k-2) \right\rfloor + 3h$, where $h = n(n - 1) ... (k - 1)$. In this paper, we will only focus on constructing a feasible fault-free substar-ring $R_s(n - 3, h)$ in a faulty star graph.

3. CENTRALIZED ALGORITHM FOR IDENTIFYING MAXIMAL FAULT-FREE SUBSTAR-RING

In section 3.1, an efficient algorithm will be proposed to identify the $R_s(n-3, h)$ that can tolerate at most $n-3$ faults. To show the applicability of this scheme, we will explain how to apply the ASCEND/DESCEND algorithms on $R_s(n-3, h)$. To tolerate more than $n-3$ faults, a modified algorithm is given in section 3.2.

3.1 Construction of $R_s(n-3, h)$ With $n-3$ Faults

In the following, we will describe a centralized algorithm to identify the maximal fault-free substar-ring $R_s(n-3, h)$ (IMSR) in order to tolerate $n-3$ faults in a faulty n-star graph. Furthmore, we also will explain how to apply the ASCEND/DESCEND algorithms on $R_s(n-3, h)$.

The IMSR algorithm is divided into three steps. First, we recognize all maximal fault-free S_{n-2} substars from an S_n^F. Second, a $(n-2, t)$-**ring** is constructed from the S_{n-2} substars, where $n(n-1)-(n-3) \le t \le n(n-1)-1$. Third, the maximal fault-free substar-ring $R_s(n-3, h)$, $h \le (n-2)t$, is constructed by applying 1-split on each S_{n-2} substar of $(n-2, t)$-**ring**. The steps in the IMSR algorithm are described in detail in the following.

First, we apply D-split on S_n^F to obtain $n(n-1)$ S_{n-2} substars based on the best selection of set D. Different values of set D will produce different sets of fault-free and faulty substars. If possible, all faulty nodes may be located in one S_{n-2} under the best selection of a set D. Then, at most n^2-n-1 fault-free S_{n-2} substars can be used. Finding the best set D is done by recognizing the maximal number of S_{n-2} substars. Our best set D produces the maximum number of fault-free S_{n-2}. This can be easily justified since all faulty nodes are possibly collected to the same substars by our selected set D, so the maximum number of fault-free S_{n-2} will be obtained. This task is carried out as follows. Given a set of faulty nodes F, $f = |F|$, in an n-star, consider a node or substar $x = x_1x_2 \ldots x_n$, $x_i \in \{*, 1, 2, \ldots, n\}$ and $i = 1..n$. An extraction function is defined by $e_i(x_1x_2\ldots x_i\ldots x_n) = x_i$. A *predicate function* [2] is defined by

$$P(x) = \begin{cases} 1 & \text{if } x = \text{TRUE} \\ 0 & \text{if } x = \text{FALSE}. \end{cases}$$

Let t_i^d be the occurrences of $e_d(y_j) = i$ under a fixed value d, $1 \le i \le n$ and $y_j \in F$. That is, $t_i^d = \sum_{j=1}^{f} P(e_d(y_j) = i)$, where "$e_d(y_j) = i$" is a boolean expression. Let m_d denote $\max_{i=1}^{n}(t_i^d)$. The best set D is obtained by finding the dimensions k and k' such that m_k and $m_{k'}$ are the first and second largest values among m_d, where $d = 1\ldots n$. For example, assume a faulty star S_5 with $F = \{12435, 32451, 52134\}$; since $e_2(12435) = e_2(32451) = e_2(52134) = 2$, we have $t_2^2 = 3$ and $t_1^2 = t_3^2 = t_4^2 = t_5^2 = 0$. Therefore, $m_2 = \max_{i=1}^{5}(t_i^2) = 3$. Similary, we can also obtain $m_1 = 1$, $m_3 = 2$, $m_4 = 2$, and $m_5 = 1$. Thus, set D is $\{2, 3\}$ or $\{2, 4\}$ since both m_3 and m_4 are equal to 2. If we choose $D = \{2, 3\}$, the minimal number of faulty substars is 2. That is, *24** and *21** are faulty substars, and *12**, *13**, *14**, *15**, *23**, *25**, *31**, *32**, *34**, *35**, *41**, *42**, *43**, and *45** are fault-free substars.

The next step is to construct a $(n-2, t)$-**ring** from S_n^F under set $D = \{k, k'\}$. Intuitively, k-split on the faulty S_n is performed to partition S_n^F into n copies of S_{n-1} substars so that we can construct a $(n-1, n)$-**ring**. Each $(n-1)$-substar of the $(n-1, n)$-**ring** is fault-free or not.

We then apply k'-split on all the S_{n-1} substars of the $(n-1, n)$-**ring** to obtain $n(n-1)$ S_{n-2} substars. In our scheme, we withdraw all the faulty substars from $n(n-1)$ S_{n-2} substars. Therefore, a $(n-2, t)$-**ring** is constructed from all the non-faulty S_{n-2} substars, where $t \le n(n-1) - f$.

Lemma 2: Given a $(n-1, n)$-**ring**, if $f \le n-3$, it is possible to construct a $(n-2, t)$-**ring** from the $(n-1, n)$-**ring**, where $n^2 - 2n + 3 \le t \le n^2 - n - 1$.

***Proof*:** Given a set $D = \{k, k'\}$, we apply the D-split on the S_n^F. A $(n-1, n)$-**ring** is first obtained by applying the k-split on the S_n^F. If the k'-th symbol of G_i, $0 \le i \le n-1$, in $(n-1, n)$-**ring** is *, then to each G_i is applied the k'-split operation to obtain $n-1$ S_{n-2} substars. As mentioned earlier, all the S_{n-2} substars are fully connected (by the adjacent relation), and there are $n-2$ connections between G_i and G_{i-1} and G_{i+1}. Since $n-3$ S_{n-2} substars of G_i at most are faulty, there exists at least one connection between G_{i-1} and G_{i+1}. For the similar reason given in Lemma 1, it is trivial to derive a feasible $(n-2, t)$-**ring** by visiting all non-faulty substars of G_i even when there are at most $(n-3)$ faulty substars in G_i, where $n(n-1) - (n-3) \le t \le n(n-1) - 1$. □

Therefore, a $(n-2, t)$-**ring**, is constructed, where $n^2 - 2n + 3 \le t \le n^2 - n - 1$. There are $n(n-1)$ S_{n-2} substars, and $n-3$ of them are faulty at most. In this case, the $(n-2, t)$-**ring** uses $n(n-1) - (n-3)$ S_{n-2} substars if $f \le n-3$; therefore, the PUR is $\frac{n^2-2n+3}{n^2-n}$, and the diameter is $\left\lfloor \frac{9}{2}(n-4) \right\rfloor + \left\lfloor \frac{3n(n-1)(n-3)}{2} \right\rfloor$. The reason is similar to that mentioned in section 2. For example, if set $D = \{4, 5\}$, then ****1 ↔ ****2 ↔ ****4 ↔ ****3 ↔ *****5 ↔ ****1 is a (4, 5)-**ring** as indicated in Fig. 3. Note that substars ***41 and ***52 are faulty. Therefore, a (3, 18)-**ring** results as shown in Fig. 3.

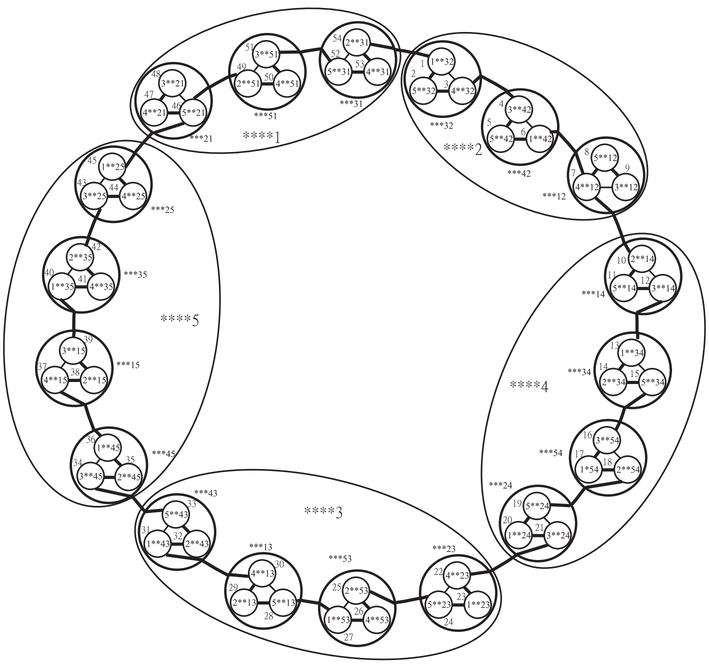

Fig. 3. A maximal fault-free substar-ring $R_s(2,54)$.

Given a $(n-2, t)$-**ring** = $[G_0, G_1, ..., G_{t-1}]$, all the $(n-2)!$ nodes of G_i exchange their contents with the corresponding nodes of adjacent substars $G_{(i-1)\ mod\ t}$ or $G_{(i+1)\ mod\ t}$. Using the GROUP-COPY procedure [20], this task is carried out in 3 time steps if there are no faults. But if faults exist, the GROUP-COPY procedure won't work, so the task costs $O(n-2)$ time using only $(n-3)!$ direct links. A scheme to reduce the communication cost is proposed here. Our scheme is to apply 1-split on each S_{n-2} substar G_i of a $(n-2, t)$-**ring** to obtain *virtual substars* X_j, where $0 \le j \le (n-2)t$. Each X_j is worked as a processing unit. These X_j can finally construct a ring with 3-dilation links as shown in Theorem 1.

Theorem 1: Given a $(n-2, t)$-**ring** = $[G_0, G_1, ..., G_l, ..., G_{t-1}]$, it is possible to construct a $R_s(n-3, h) = [X_0, X_1, ..., X_j, ..., X_{h-1}]$ such that each pair of neighboring X_j and X_{j+1} is connected at most 3-dilation links, where $h = (n-2)t$.

***Proof*:** First, if X_j and X_{j+1} are located in the same G_l of the $(n-2, t)$-**ring**, $1 \le l \le t$, then X_j is directly linked to X_{j+1}. Second, if X_j and X_{j+1} are, respectively, located in the neighboring G_l and G_{l+1} of the $(n-2, t)$-**ring**, then there exists a pair of X' (in G_l) and X'' (in G_{l+1}) such that X' is direct linked to X''. The 3-dilation links are $X_j \leftrightarrow X' \leftrightarrow X'' \leftrightarrow X_{j+1}$. Furthermore, if $X'' = X_{j+1}$, then there are only 2-dilation links between X_j and X_{j+1}. As a result, each pair of neighboring X_j and X_{j+1} of $R_s(n-3, h)$ is connected by most 3-dilation links. □

Recall the above example; Fig. 3 shows a feasible $R_s(2, 54)$ = [5**21, 4**21, 3**21, 2**51, 4**51, 3**51, ..., 1**35, 4**35, 2**35, 3**25, 4**25, 1**25]. Note that *virtual substars* 3**12 and 2**14 are, respectively, located in ***12 and ***14, and that the edges between 3**12 and 2**14 are 3**12 ↔ 4**12 ↔ 2**14.

The IMSR algorithm is outlined as follows.

Algorithm: Identifying maximal fault-free substar-ring (IMSR)

Input: An S_n with faulty node set F, where $1 \le f \le n-3$.
Output: Substar sequence $[X_0, X_1, ..., X_{h-1}]$ is obtained, where $(n-2)(n^2-2n+3) \le h \le (n-2)(n^2-n-1)$.
Step 1: Find the best set $D = \{k, k'\}$. The maximal number of fault-free substars is obtained by partitioning S_n^F into disjoint S_{n-2} substars along dimensions k and k'.
Step 2: Identify a $(n-2, t)$-**ring** based on Lemma 2 among all the fault-free S_{n-2} substars, where $n^2-2n+3 \le t \le n^2-n-1$,
Step 3: Construct the maximal fault-free substar-ring $R_s(n-3, h)$ according to Theorem 1, where $(n-2)(n^2-2n+3) \le h \le (n-2)(n^2-n-1)$

The total time cost T_{IMSR} of the IMSR algorithm is analyzed as follows. In step 1, the time cost $O(nf)$ can be obtained by using a linear-time integer sort [1] to determine the value of set D. When $f = n-3$, the time cost is $O(n^2)$. Step 2 only costs $O(n)$ time to construct the $(n-2, t)$-**ring**. In step 3, a time cost of $O(n^3)$ is needed to split each S_{n-2} of $(n-2, t)$-**ring** to obtain $R_s(n-3, h)$, where $h \le (n-2)t$. Consequently, the total time cost of T_{IMSR} can be obtained using the following equation:

$$T_{IMSR} = O(n^2) + O(n) + O(n^3) = O(n^3).$$

To demonstrate the application capability, we will show how to execute the sorting operation on $R_s(n-3, h) = [X_0, X_1, ..., X_j, ..., X_{h-1}]$. First, an efficient sorting algorithm for a star graph [17] can be performed on each *virtual substar* X_j of $R_s(n-3, h)$ such that the unsorted elements on each X_j are sorted. Second, if each *virtual substar* X_j of $R_s(n-3, h)$ is worked as a processing unit, then the Odd-Even Transposition Sort algorithm [1] can be performed on a ring of these processing units. After performing the above operations, data elements in all *virtual substars* will be sorted. The detail algorithm can refer the similar sorting operations on the maximal fault-free subcube-ring in faulty hypercubes [23] Furthermore, using the similar matrix-multiplication operations on the maximal fault-free subcube-ring [23], we can also perform the matrix-multiplication algorithm on $R_s(n-3, h)$. Similarly, many scientific algorithms on star graphs [8, 20] can be tailored onto our $R_s(n-3, h)$.

3.2 Construction $R_s(n-3, h')$ With More Than $n-3$ Faults

In section 3.1, it was shown that a $(n-2, t)$-**ring** and its $R_s(n-3, h)$ can not be constructed when $f > n-3$. In this subsection, we will describe how to construct $R_s(n-3, h')$ when $n-3 < f < n^2 - n$. It is known that an N-node ring can be one-to-one embedded with dilation 3 in any connected N-node network [4]. Given a S_n^F and its best set $D = \{k, k'\}$, initially, we construct a tree, namely *substar-tree T*, among n S_{n-1}'s, which are obtained by performing k-split on S_n^F. Each node of *substar-tree T* is an S_{n-1}. Then, all possible fault-free *virtual substars* can be obtained by performing k'-split and 1-split operations on each node of *substar-tree T*. These fault-free *virtual substars* can still form a connected network. Therefore, a ring of the *virtual substars* with 3-dilation links is obtained. The ring is denoted as $R_s(n-3, h')$. In the following, we will describe the modified IMSR' algorithm used to construct *substar-tree T* and to then obtain the final $R_s(n-3, h')$.

The detail algorithm of modified IMSR' algorithm is described below. We apply k-split on S_n^F to produce n S_{n-1}'s and collect them into set Ψ, where $D = \{k, k'\}$. The *substar-tree T* is constructed based on set Ψ. As we stated earlier, each node of *substar-tree T* is an S_{n-1} substar. The total number of nodes of *substar-tree T* is at most n. Before we describe how to construct *substar-tree T*, we will define a function AD(G, G') to represent the adjacent relation of a pair of S_{n-1}'s G and G', where G and G' belong to set Ψ. k'-split is performed on G and G' to decompose G and G' into $2(n-1)$ S_{n-2}'s, each of which is fault-free or not. Let function AD(G, G') denote the number of pairs of adjacent fault-free S_{n-2} substars x and y, where x and y are located in G and G', respectively. If AD(G, G') > 0, then there exists at least one pair of adjacent fault-free S_{n-2}'s between G and G'. Otherwise, no fault-free S_{n-2}'s are adjacent if AD(G, G') = 0. For example, if set $D = \{4, 5\}$, then substars ***41, ****51, and ***52 are faulty substars as shown in Fig. 4, and there is no fault-free adjacent substar S_3's between ****1 and ****2, so AD(****1, ****2) = 0. Equation AD (****1, ****3) = 1 holds because ***21 and ***23 are the only pair of fault-free adjacent substars. Similarly, equations AD(****1, ****4) = 2, AD(****1, ****5) = 2, and AD (****3, ****2) = 2 can be obtained.

Continually, *substar-tree T* can be constructed as follows. First, the root of tree T is a substar selected from set Ψ randomly. Using the Breadth-First-Searching method, we can expand the branches of *substar-tree T* as follows. Branches of tree T represent the possible connecting substars. Each node u of *substar-tree T* probes each remaining substar v from

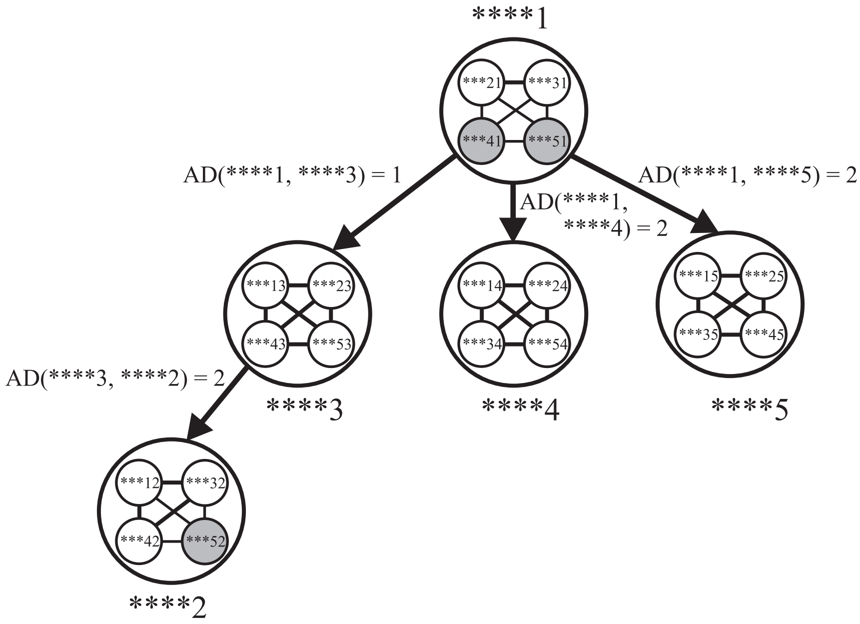

Fig. 4. Constructing a *substar-tree T*.

set Ψ. If the condition AD(u, v) > 0 exists, then node u is connected to substar v, and we can eliminate v from set Ψ. By repeatedly performing the above probing operations until set Ψ is empty or no further connecting substar can be found, *substar-tree T* is constructed. Since there exists at least one branch of each node of *substar-tree T*, *substar-tree T* is a connected network. The 1-split oppration is performed on each S_{n-2} substar of *substar-tree T* to obtain all possible *virtual substars*. Each *virtual substar* is treated as a processing unit, and these *virtual substars* still form a connected network. Therefore, a maximal fault-free substar-ring $R_s(n-3, h')$ with 3-dilation links can be obtained [4], where $h' \leq n(n-1)(n-2) - 1$. Recalling the above example, let the root of *substar-tree T* be ****1, the branches of root be ****3, ****4, and ****5, and the branch of ****3 be ****2. The *substar-tree T* is shown in Fig. 4. After applying 1-split on all the S_3's of *substar-tree T*, an $R_s(2, 17 * 3) = R_s(2, 51)$ with 3-dilation links is obtained.

Finally, we will analyze the total time cost $T_{IMSR'}$ of the modified IMSR algorithm. In step 1 of identifying all S_{n-2}, a time cost of O(nf) is needed if we use a linear-time integer sort [1]. When $f = n^2 - n - 1$, the time cost is $O(n^3)$. During construction of *substar-tree T*, the AD operation is performed $O(n^2)$ times, and each time, $O(n)$ time is needed. It takes $O(n^3)$ time to construct *substar-tree T*. A time cost of $O(n^3)$ is needed to split $n(n-1)$ S_{n-2}'s into $n(n-1)(n-2)$ *virtual substars*. Consequently, the total time complexity of $T_{IMSR'}$ can be measured by the following equation:

$$T_{IMSR'} = O(n^3) + O(n^3) + O(n^3) = O(n^3).$$

4. PERFORMANCE ANALYSIS

In this section, we will analyze the distributed percentage of the processor utilization of the maximal fault-free substar-ring under two cases in which the number of faulty processors is assumed. First, if f is not larger than $n-3$, then the PUR is at least $\frac{n^2-2n+3}{n^2-n}$. Second, we will find the PUR of our reconfiguration scheme even when $n-3 < f \le n^2 - n - 1$.

The percentage of the processor utilization of $R_s(n-3, h)$ is analyzed as follows. In our simulation, the addresses of faulty processors are randomly generated in each of 10000 simulations for fixed n and f. To illustrate the fault tolerance capability, we will consider the worst case of simulating the PUR. An S_n is partitioned into $n(n-1)$ S_{n-2} by step 1 of the IMSR algorithm. Here, we denote the number of faulty S_{n-2} by r. The factor of the value of r presents the degree of occurring faults. The larger the value of r is, the more faults there will be. The factor of the value of r is used to analyze the PUR. If the percentage of the processor utilization of $R_s(n-3, h)$ is larger than 50%, then the slowdown factor of computation has a better chance of reducing to less than 2. In the case of $f \le n-3$, the PUR is at least $\frac{n^2-2n+3}{n^2-n}$ and is always larger than 50%. All possible maximal fault-free substar-rings $R_s(n-3, h)$ and the percentage distribution of processor utilization in a faulty S_5, where $1 \le r \le 5^2 - 5 - 1$ (=19) and $3 \le h \le 57$, are shown in Table 1. For instance, when $n = 5$ and $r = 1$, 2, and 3, 100% of the cases of S_5 can be identified as $R_s(2, 57)$ with 95% processor utilization, 100% of the cases can be identified as $R_s(2, 54)$ with 90% processor utilization, and 100% of the cases can be identified as $R_s(2, 51)$ with 85% processor utilization, respectively. When $n = 5$ and $r = 4$, 99.57% of the cases of S_5 can be identified as $R_s(2, 48)$ with 80% processor utilization, and 0.43% of the cases can be identified into $R_s(2, 45)$ with 75% processor utilization. This indicates that the smaller the value of r is, the maximal fault-free substar-ring with high PUR generally be determined. As shown in Table 1, all more than 73% cases to exploit the more than 50% processor utilization in an faulty S_5 when $1 \le r \le 8$. This shows that the percentage of the processor utilization of maximal fault-free substar-ring $R_s(n-3, h)$ is always greater than 50% when $r < n(n-1)/2$.

The average PUR is discussed in the following. The average PUR is defined as the sum of the percentage of the processor utilization of each $R_s(n-3, h)$ * PUR of $R_s(n-3, h)$. For instance, in Table 1, when $n = 5$ and $r = 4$, 99.57% of $R_s(2, 48)$ with PUR = 80% and 0. 43% of $R_s(2, 45)$ with PUR = 75% are identified, so the average PUR is 99.57% * 80% + 0. 43% * 75% = 79.9785%. In our simulation, we estimate the average PUR under the case of $5 \le n \le 12$. The simulation results of the average PUR with different value of r are depicted in Fig. 5. The average PUR with a value of r larger than $n(n-1)/2$ is always larger than 50% as depicted in Fig. 5. The average PUR is inversely proportional to the value of r; i.e., the larger the value of r is, the lower the average PUR will be. For instance, when the number of faulty S_{n-2} substars is $n(n-1)/10$, $2n(n-1)/10$, and $3n(n-1)/10$, the average PUR is about 90%, 80%, and 70%, respectively. Consequently, the smaller the number of faulty S_{n-2} substars is, the high the average PUR is.

In a conclusion, when the number of faulty S_{n-2} is smaller than $n(n-1)/2$, more than 50% of average PUR are obtained by our simulation results. This indicates that our scheme can obtain a reasonable average PUR, so our scheme is a truly effective reconfiguration method.

Table 1. Percentage distribution of the processor utilization of maximal fault-free substar-ring $R_s(2, h)$ in an S_5, where the number of faulty 3-substars is from 1 to 19 and $3 \leq h \leq 57$.

PUR	h	The number of faulty 3-substars																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
5%	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.33	26.5	55.01	84.15	100
10%	6	0	0	0	0	0	0	0	0	0	3.21	11.45	26.40	45.68	65.38	74.96	66.72	43.31	15.85	0
15%	9	0	0	0	0	.24	.65	2.73	7.24	15.05	23.79	33.35	37.03	32.85	24.55	15.03	6.14	1.68	0	0
20%	12	0	0	0	0	0	0	0	.65	1.23	2.63	4.67	5.61	5.18	3.51	2.14	.64	0	0	0
25%	15	0	0	0	0	0	0	.4	1.9	4.35	8.36	9.47	8.7	6.65	4.19	1.54	0	0	0	0
30%	18	0	0	0	0	0	.37	.6	2.12	5.69	10.29	11.62	10.20	6.68	2.37	0	0	0	0	0
35%	21	0	0	0	0	0	0	1.10	3.31	8.07	11.09	12.12	7.82	2.96	0	0	0	0	0	0
40%	24	0	0	0	0	0	1.58	2.12	5.26	8.19	12.39	9.61	4.24	0	0	0	0	0	0	0
45%	27	0	0	0	0	..40	.12	1.26	5.57	12.17	12.89	7.71	0	0	0	0	0	0	0	0
50%	30	0	0	0	0	0	.06	6.21	10.02	16.29	15.35	0	0	0	0	0	0	0	0	0
55%	33	0	0	0	0	0	4.70	4.96	15.14	28.96	0	0	0	0	0	0	0	0	0	0
60%	36	0	0	0	0	1.52	.87	9.88	48.79	0	0	0	0	0	0	0	0	0	0	0
65%	39	0	0	0	0	0.45	5.44	70.73	0	0	0	0	0	0	0	0	0	0	0	0
70%	42	0	0	0	0	2.07	86.21	0	0	0	0	0	0	0	0	0	0	0	0	0
75%	45	0	0	0	.43	95.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80%	48	0	0	0	99.57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85%	51	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90%	54	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
95%	57	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

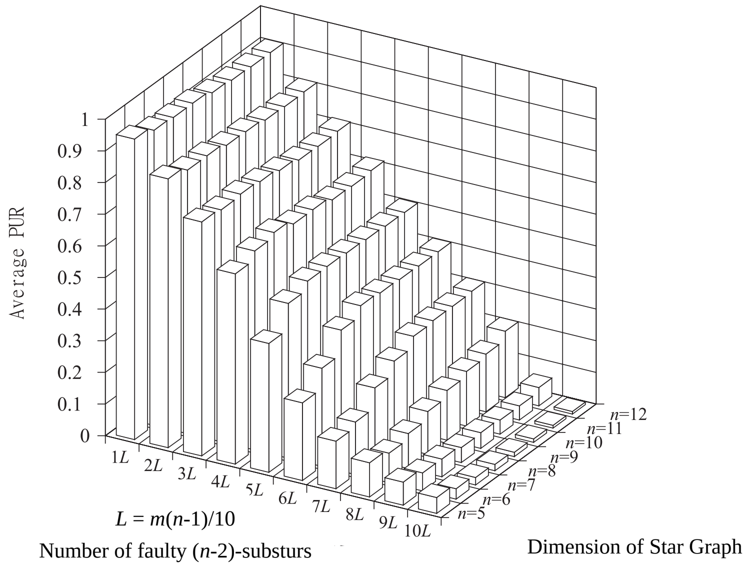

Fig. 5. The average PUR of $R_s(n\text{-}3, h)$ of S_n, for $5 \leq n \leq 12$.

5. CONCLUSIONS

In this paper, we have proposed a reconfiguration scheme to identify the maximal fault-free substar-ring for tolerating faults in faulty n-dimensional star graphs. The fault-free substar-ring is connected by a ring of fault-free *virtual substars* with dilation 3. This is the first result to propose a reconfiguration scheme in the faulty star graph. Our proposed scheme can tolerate $n - 3$ faults so that the processor utilization is $\frac{n^2 - 2n + 3}{n^2 - n}$ and the diameter is $\left\lfloor \frac{9}{2}(n-4) \right\rfloor + \left\lfloor \frac{3n(n-1)(n-3)}{2} \right\rfloor$. This is a near optimal result since the maximal fault-free substar-ring is constructed by using all of the possible fault-free $(n - 2)$-substars. To demonstrate the applicability of our scheme, we have described how to apply a sorting algorithm to our reconfiguration scheme. Moreover, our reconfiguration scheme can work when the number of faults exceeds $n - 3$. We have also simulated the algorithm to show that the reconfiguration scheme has high processor utilization.

In order to preserve a low diameter and obtain better processor utilization, identifying the maximal fault-free substar-ring $R_s(k - 1, h)$ has been the main objective of this study. Determining the maximal fault-free substar-ring $R_s(k - 1, h)$ is controlled by what values of k and h being are the best selection. It is possible to construct a $R_s(k - 1, h)$, $k \leq n - 2$, to obtain a large diameter and greater processor utilization. If $k - 1 = 1$, our scheme becomes a simple problem of ring embedding on a faulty star graph [25]. However, in this paper, we have only focused on identifying the maximal fault-free substar-ring $R_s(n - 3, h)$ to keep a smaller diameter and obtain reasonable processor utilization.

ACKNOWLEDGMENTS

We appreciate the constructive comments received from reviewers, which have helped to improve the correctness and presentation of this article.

REFERENCES

1. S. G. Akl, *Parallel Sorting Algorithms*, Academic Press, Inc, 1985.
2. D. Davis and E. J. Weyuker, *Computability, Complexity, and Languages*, Academic Press, Inc, 1983.
3. R. P. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*, Addison-Wesley Publishing Company, 1985.
4. F. T. Leighton, *Introduction to Parallel Algorithms and Architecture: Array · Tree · Hypercube*, Morgan Kaufmann Publishers, 1992.
5. S. Akers, D. Harel and B. Krishnamurthy, "The star graph: an attractive alternative to the n-Cube," in *Proceedings of International Conference on Parallel Processing* 1987, pp. 393-400.
6. S. B. Akers and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Transactions on Computers*, Vol. 38, No. 4, 1989, pp. 555-565.

7. S. G. Akl, K. Qiu and I. Stojmenovic, "Data communication and computational geometry on the star and pancake interconnection network," in *Proceedings of the Third IEEE Symposium on Parallel and Distributed Processing*, 1991, pp. 415-422.
8. S. G. Akl, K. Qiu and I. Stojmenovic, "Fundamental algorithms for the star and pancake interconnection networks with applications to computational geometry," *Networks*, Vol. 23, No. 4, 1993, pp. 215-225
9. N. Bagherzadeh, N. Nassif and S. Latifi, "A routing and broadcasting scheme on faulty star graphs," *IEEE Transactions on Computers*, Vol. 41, No. 11, 1993, pp. 1398-1403.
10. M. Y. Chan and S. J. Lee, "Distributed fault-tolerant embeddings of rings in hypercubes" *Journal of Parallel and Distributing Computing*, Vol. 11, No. 1, 1991, pp. 63-71.
11. H. L. Chen and N. F. Tzeng, "Quick determination of subcubes in a faulty hypercube," in *Proceedings of 1992 International Conference on Parallel Processing*, Vol. III, 1992, pp. 338-345.
12. Y. S. Chen and J. P. Sheu, "A reliable sorting algorithm on hypercube multicomputers," *Journal of Parallel Algorithms and Applications*, Vol. 5, No. 2, 1995, pp. 165-186.
13. K. Day and A. Tripathi, "A comparative study of topological properties of hypercubes and star Graphs," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 5, No. 1, 1994, pp. 31-38.
14. J. S. Jwo, S. Lakshmivarahan and S. K. Dhall, "Embedding of cycles and grids in star graphs," in *Proceedings of 2nd IEEE Symposium on Parallel and Distributed Processing*, 1990, pp. 540-547.
15. S. Latifi, "Task allocation in the star graph," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 5, No. 11, 1994, pp. 1220-1224.
16. J. Misic and Z. Jovanovic, "Communication aspects of the star graph interconnection network," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 5, No. 7, 1994, pp. 678-687.
17. A. Menn and A. K. Somani, "An efficient sorting algorithm for the star graph interčonnection network," in *Proceedings of the 19th International Conference on Parallel Processing*, Vol. III, 1990, pp. 1-8.
18. D. Nassimi, "Parallel algorithms for the classes for $\pm 2^b$ DESCEND and ASCEND computations on a SIMD hypercube," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 4, No. 12, 1993, pp. 1372-1381.
19. Y. C. Tseng, Y. S. Chen, T. Y. Juang and C. J. Chang, "Congestion-free, dilation-2 embedding of complete binary trees in star graphs," *Networks*, Vol. 33, No. 3, 1999, pp. 221-231.
20. K. Qiu, S. G. A. and H. Meijer, "On some properties and algorithms for the star and pancake interconnection networks," *Journal of Parallel and Distributing Computing*, Vol. 22, No. 1, 1994, pp. 16-25.
21. K. Qiu, H. Meijer and S. Akl, "Decomposing a star graph into disjoint cycles," *Information Processing Letters*, No. 39, 1991, pp. 125-129.
22. J. P. Sheu, Y. S. Chen and C. Y. Chang, "Fault-tolerant sorting algorithm on hypercube multicomputers," *Journal of Parallel and Distributed Computing*, Vol. 16, No. 2, 1992, pp. 185-197.
23. Y. S. Chen and J. P. Sheu, "Tolerating faults in injured hypercubes using maximal fault-free subcube-ring," *Parallel Computing*, Vol. 23, No. 3, 1997, pp. 311-331.
24. S. Sur and P. K. Srimani, "A fault tolerant routing algorithm in star graph interconnection

network," in *Proceedings of the 20th International Conference on Parallel Processing*, Vol. III, 1991, pp. 267-270.

25. Y. C. Tseng, S. H. Chang and J. P. Sheu, "Fault-tolerant ring embedding in star graphs," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 8, No. 12, 1997, pp. 1185-1195.

Yuh-Shyan Chen (陳裕賢) received the B.S. degree in computer science from Tamkang University, Taiwan, Republic of China, in June 1988, and the M.S. and Ph.D. degrees in Computer Science and Information Engineering from National Central University, Taiwan, Republic of China, in June 1991 and January 1996, respectively. He joined the faculty of the Department of Computer Science and Information Engineering at Chung-Hua University, Taiwan, Republic of China, as an associate professor in February 1996. His current research interests include parallel and distributed processing, interconnection network, wireless communication and mobile computing.

Dr. Chen is a member of the IEEE Computer Society and the Phi Tau Phi Society.

Jang-Ping Sheu (許健平) received the B.S. degree in computer science from Tamkang University, Taiwan, Republic of China, in 1981, and the M.S. and Ph.D. degrees in computer science from National Tsing Hua University, Taiwan, Republic of China, in 1983 and 1987, respectively.

He joined the faculty of the Department of Electrical Engineering, National Central University, Taiwan, Republic of China, as an Associate Professor in 1987. He is currently a Professor and Chair of the Department of Computer Science and Information Engineering, National Central University. From March to June of 1995, he was a visiting researcher at the IBM Thomas J. Watson Research Center, New York. His current research interests include parallelizing compilers, interconnection networks, and mobile computing.

Dr. Sheu is a member of IEEE, ACM, and the Phi Tau Phi Society. He is an associate editor of the Journal of Information Science and Engineering and the Journal of the Chinese Institute of Electrical Engineering. He received the Distinguished Research Awards of the National Science Council of the Republic of China in 1993-1994, 1995-1996, and 1997-1998.