Fault-Tolerant Ring Embedding in Star Graphs*

Yu-Chee Tseng,
Dept. of Computer Science
Chung-Hua Polytechnic Institute
Hsin-Chu, 30067, Taiwan

Shu-Hui Chang, and Jang-Ping Sheu
Dept. of Computer Sci. and Information Engr.
National Central University
Chung-Li, 32054, Taiwan

Abstract

In this paper, we consider an injured star graph with some faulty links and nodes. We show that even with $f_e \leq n-3$ faulty links a Hamiltonian cycle still can be found in an n-star, and that with $f_v \leq n-3$ faulty nodes a ring containing at most $4f_v$ nodes less than that in a Hamiltonian cycle can be found (i.e., containing at least $n!-4f_v$ nodes). In general, in an n-star with f_e faulty links and f_v faulty nodes, where $f_e + f_v \leq n-3$, our embedding is able to establish a ring containing at least $n! - 4f_v$ nodes.

1 Introduction

One new interconnection network that has attracted a lot of attention recently is the star graph [1]. Large references can be found in studying the star graph's topological properties [2, 8], embedding capability [4, 6], and communication capability [3, 5, 7, 9].

The graph embedding problem has been heavily studied for various host graphs. With a star graph as the host graph, any ring of an even length has been shown to be embeddable [4]. Results regarding embedding multi-dimensional meshes into a star graph can be found in [4, 8]. The embedding of a Hamiltonian cycle and hypercubes is discussed in [6].

In this paper, we consider the problem of embedding a ring into an injured n-star graph which has some faulty links (or edges) and nodes (or vertices). Rings are common guest graphs with many applications. Fault tolerance is an important issue in a multicomputer network, especially when the network becomes large. If in a star graph some components fail, it is desirable that the injured components be isolated from the rest of the network so that the embedding is still possible. The embeddings achieved in this paper are summarized as follows: (1) with $f_e \leq n-3$ faulty links, the embedding of a Hamiltonian cycle, (2) with $f_v \leq n-3$ faulty nodes, the embedding of a ring containing at most $4f_v$ nodes less than that of a Hamiltonian cycle, and (3) with f_e faulty links and f_v faulty nodes, where $f_e + f_v \leq n-3$, the embedding of a ring containing at most $4f_v$ nodes less than that of a Hamiltonian cycle.

Preliminaries are given in Section 2. In Section 3 we develop a new scheme for finding a Hamiltonian cycle in a star graph. The embedding is then extended with fault-tolerant capability when only links and only nodes may fail in Section 4 and Section 5, respectively. The result to tolerate both link and node failure is presented in Section 6.

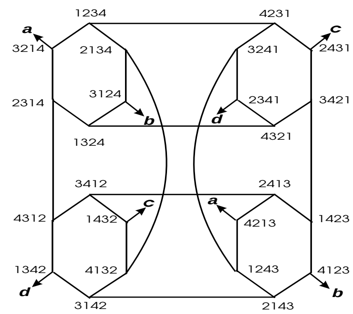

Figure 1. A 4-dimensional star graph S_4.

2 Preliminaries

An *n-dimensional star graph*, also referred to as *n-star* or S_n, is an undirected graph consisting of $n!$ nodes (vertices) and $(n-1)n!/2$ links (edges). Each node is uniquely assigned a label $x_1 x_2 \cdots x_n$ which is the concatenation of any permutation of n distinct symbols $\{x_1, x_2, \ldots, x_n\}$. Two nodes are joined by an edge *along* dimension d iff the label of one node can be obtained from the other by swapping the first symbol and the d-th symbol, $2 \leq d \leq n$. Without loss of generality, throughout we let these n symbols be $\{1, 2, \ldots, n\}$. A 4-dimensional star graph S_4 is shown in Fig. 1.

An S_n is a recursive structure that contains many smaller stars, or substars. Formally, a *k-dimensional substar*, or *k-substar*, is denoted as a string $X = x_1 x_2 \cdots x_n$, where $x_1 = *$ and $x_i \in \{*, 1, 2, \ldots, n\}$, $2 \leq i \leq n$. The symbol $*$ means a "don't care". In string X there are exactly k $*$'s. The substar represented by X is a subgraph of S_n containing all vertices obtained from X by replacing each $*$ with digits $\{1, 2, \ldots, n\}$. These vertices are connected by the original

*This research is supported by National Science Council of the R. O. C. under grants NSC85-2213-E-216-021 and NSC85-2213-E-008-030.
Authors' emails: yctseng@chpi.edu.tw, sheujp@mbox.ee.ncu.edu.tw.

links in S_n. For instance, **53* is a 3-substar containing six nodes 12534, 14532, 21534, 24531, 41532, and 42531.

Definition 1 Let $X = x_1 x_2 \cdots x_j \cdots x_n$ be a k-substar with $x_j = *$. The *j-cut on X*, $j \geq 2$, is to partition X along the j-th dimension into k number of $(k-1)$-substars, each obtained from X by replacing x_j with a legal non-$*$ symbol. Let $D = (d_1, d_2, \ldots, d_m), m \leq k$, be a sequence of dimensions such that the $x_{d_i} = *, i = 1..m$. Then the *D-cut on X* is to first apply a d_1-cut on X, whose result is then applied a d_2-cut, whose result is then applied a d_3-cut, etc. The final result is $k(k-1)\cdots(k-m+1)$ number of $(k-m)$-substars.

For instance, given a 4-substar $X =$ ***5*3 in an S_6, a 3-cut on X is to partition X into four 3-substars **15*3, **25*3, **45*3, and **65*3. If $D = (3, 5)$, a D-cut on X will apply a 3-cut and then a 5-cut on X. This generates the following 2-substars: {**1523, **1543, **1563}, {**2513, **2543, **2563}, {**4513, **4523, **4563}, and {**6513, **6523, **6543}.

Definition 2 Consider two k-substars X and Y in S_n. We define X and Y to be *adjacent* if their string representations differ in exactly one non-$*$ position. If X and Y are adjacent, the *difference from X to Y*, denoted as $dif(X, Y)$, is the symbol of X at the position where X and Y differ.

For instance, substar $X =$ **5*13* is adjacent to $Y =$ **5*23*, but not adjacent to $Y' =$ **4*23*. The difference from X to Y, or $dif(X, Y)$, is 1, whereas the expression $dif(Y, X)$ is 2.

The following discussion combines the notion of adjacency and cut. Consider two adjacent k-substars $X = x_1 \cdots x_j \cdots x_n$ and $Y = y_1 \cdots y_j \cdots y_n$ such that $x_j = y_j = *$. If we apply a j-cut on X and Y, we will obtain k substars (of dimension $k-1$) from each of X and Y. By the above definition, one easily sees that all k substars in X are adjacent to each other, and so are those in Y. Furthermore, among these substars, $k-1$ substars in X are adjacent to $k-1$ substars in Y in a one-to-one manner. Only the substar $x_1 \cdots x'_j \cdots x_n$ in X and the substar $y_1 \cdots y'_j \cdots y_n$ in Y are not adjacent, where $x'_j = dif(Y, X)$ and $y'_j = dif(X, Y)$. The idea is illustrated in Fig. 2, where the adjacency relation is represented by lines connecting substars. In particular, Fig. 2(a) shows three substars X, Y, Z, with X adjacent to Y and Y adjacent to Z. Within each of X, Y, Z, the 3-substars are fully connected, while between X and Y (and similarly Y and Z) there are three connections. Also note that the 3-substar ***256 in X, which is not connected to Y, satisfies $x'_j = 2 = dif(Y, X)$ (and similarly 3-substar ***526 in Y, which is not connected to X, satisfies $y'_j = 5 = dif(X, Y)$).

Definition 3 A sequence of k-substars $R = [X_0, X_1, \ldots, X_{r-1}]$ is called a *k-ring* if substar X_i is adjacent to its neighbors $X_{(i-1) \bmod r}$ and $X_{(i+1) \bmod r}$ for any $i = 0..r-1$.

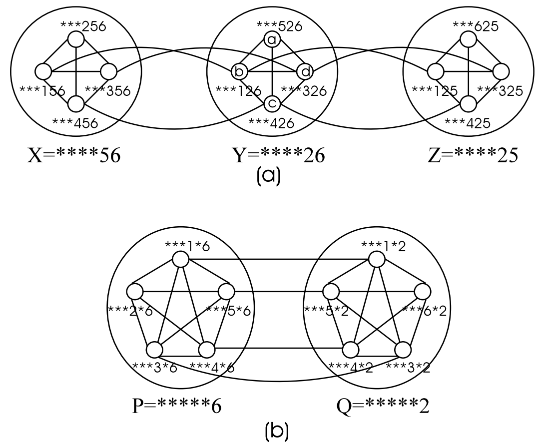

Figure 2. The substar adjacency relation.

For example, $R =$ [***3*2, ***1*2, ***4*2, ***4*5, ***3*5] is a 4-ring in an S_6.

Lemma 1 *Given a k-ring $R = [X_0, X_1, \ldots, X_{r-1}]$, $k \geq 4$, it is possible to construct a $(k-1)$-ring R' of length kr from R.*

Proof. We apply any legal j-cut on each X_i, $i = 0..r-1$, into $(k-1)$-substars (by "legal", the j-th symbol of X_i must be $*$). As mentioned earlier, in X_i all $(k-1)$-substars are fully connected (in terms of adjacency) and there are $k-1$ connections between X_i and its neighbors X_{i-1} and X_{i+1}. It is trivial to derive an R' which connects all $(k-1)$-substars by visiting X_i's along the direction of R. ■

3 Embedding of a Hamiltonian Cycle

It is known that a star graph contains a Hamiltonian cycle [4, 6]. Below we develop the equivalent result in a different way.

Given an S_n, our embedding works as follows. First, we construct from S_n an $(n-1)$-ring. Then, we apply Lemma 1 (possibly combined with some special techniques) to construct from the $(n-1)$-ring an $(n-2)$-ring. This will be repeated recursively until a 3-ring is obtained. In the end, we generate from the 3-ring a 1-ring, which is a Hamiltonian cycle.

In the following presentation, we will discuss the embedding backward from the last step. We first show how to construct a 1-ring from a 3-ring. Observe that there are 2 links between any two adjacent 3-substars. These connections have two properties.

P1: For any two adjacent 3-substars X and Y, the two nodes in X connecting to Y are located at anti-podal positions of the hexagon formed by X (i.e., the distance between these two nodes is 3).

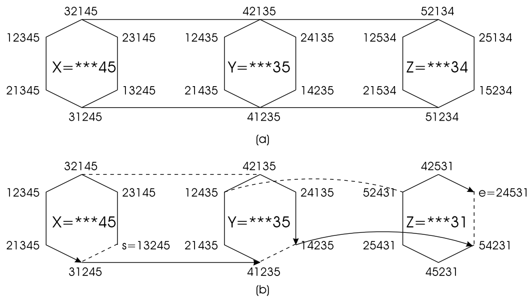

Figure 3. Three adjacent 3-substars X, Y, Z in an S_5. In (a), the graph is not Hamiltonian. In (b), a Hamiltonian path starting from node s to e can be found (shown in arrows).

P2: Consider any three 3-substars $X, Y,$ and Z such that (i) X is adjacent to Y, (ii) Y is adjacent to Z, and (iii) $dif(X,Y) \neq dif(Z,Y)$. The two nodes in Y connecting to X are disjoint from those two in Y connecting to Z.

P2 is important in finding a Hamiltonian cycle in our algorithm. To shed some light, Fig. 3(a) shows three adjacent substars X, Y, Z in an S_5 with $dif(X,Y) = dif(Z,Y) = 4$. Nodes 42135 and 41235 in Y are connecting to both X and Z. One easily sees that graph formed by X, Y, Z is not Hamiltonian. On the contrary, in Fig. 3(b), the condition $dif(X,Y) \neq dif(Z,Y)$ holds and the graph formed by X, Y, Z has a Hamiltonian path. In fact, by **P1** and **P2**, it is not hard to prove that as long as the condition $dif(X,Y) \neq dif(Z,Y)$ holds, we can construct a path starting from X, visiting all nodes in X, connecting to Y, visiting all nodes in Y, connecting to Z, and then visiting all nodes in Z.

Lemma 2 *Given a 3-ring $R = [X_0, X_1, \ldots, X_{r-1}]$ such that $dif(X_{(i-1)\text{mod}r}, X_i) \neq dif(X_{(i+1)\text{mod}r}, X_i)$ for any $i = 0..r-1$, we can find a 1-ring R' of length $6r$ from R.*

Proof. We traverse the 3-substars of R one after another. First, let x be any of the two nodes in X_0 that have a link connecting to X_1. We traverse starting from x, visiting every node in X_1, and stopping at a node in X_1 with a link to X_2 (see Fig. 4 for illustration). By **P1** and **P2**, it is easy to do so. Clearly, this can be repeated until X_{r-1} is reached.

Suppose we stop at a node in X_{r-1} with a link connecting to a node, say y, in X_0. By **P1** and **P2**, the distance between x and y is either 1 or 2 (see Fig. 4). Now we traverse nodes in X_0. In the former case, a ring of length $6r$ can be easily formed. In the latter case, a ring of length $6r - 1$ will be formed, which is impossible because a star graph is bipartite and a cycle must have an even length. Hence the lemma. ■

In earlier Lemma 1, we have shown how to construct a 3-ring from a given 4-ring. However, care must be taken

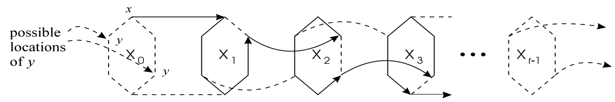

Figure 4. Proof of Lemma 2.

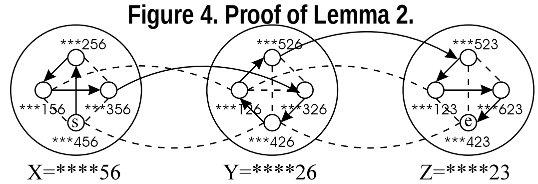

Figure 5. Three adjacent 4-substars X, Y, Z with $dif(X,Y) = 5 \neq dif(Z,Y) = 3$. The path from s to e satisfies P2.

to ensure that the 3-ring satisfies **P2** so as to be used by Lemma 2*. For instance, we can not find a 3-ring satisfying **P2** from the 4-ring in Fig. 2(a) for the following reasons.

1. Substar a can not be the first or last one visited in Y since it has no connection to X and Z.
2. Because $dif(X,Y) = 5$, the difference from any 3-substar in X to any adjacent 3-substar in Y is 5. However, the difference from substar a to any of b, c, d is also 5. So a can not be the second substar visited in Y.
3. Because $dif(Z,Y)$ is also 5, the difference from any 3-substar in Z to any adjacent 3-substar in Y is 5. So a can not be the third substar visited in Y, a contradiction.

As a counter-example, Fig. 5 shows three adjacent 4-substars X, Y, Z with $dif(X,Y) \neq dif(Z,Y)$. A path satisfying **P2** can be found. This is formally reasoned below.

In general, consider any two adjacent 4-substars X and Y. After applying an appropriate cut on X and Y, let x be the 3-substar in X that does not have a connection to Y, and similarly let y be the one in Y that does not have a connection to X. We propose two rules to visit 3-substars in X and Y:

R1: arrange x as the first or second substar traversed in X, and

R2: arrange y as the third or fourth substar traversed in Y.

These two rules are sufficient to ensure finding a 3-ring satisfying **P2**. It will be helpful to first verify these rules using the example in Fig. 5. To prove **R1**, first observe that any path in X must satisfy **P2** even if we arbitrarily visit the substars in X. Secondly and apparently, we will not let x be the last substar visited in X, as there is no connection from x to Y. Thirdly, suppose x and x' are the third and forth substars, respectively, visited in X. Then $dif(x, x')$ must be equal to $dif(Y, X)$ for any x'. As $dif(Y, X)$ is the difference from any 3-substar in Y to any 3-substar in X, the

*By satisfying **P2**, we mean that every consecutive 3-substars in the ring has the property (iii) in **P2**.

path will violate **P2**. So x can not be the third or forth substar visited in X. Lastly, following rule **R1**, suppose $x'(\neq x)$ and $x''(\neq x)$ are the third and forth 3-substars, respectively, visited in X. One can easily show that $dif(x', x'')$ is not equal to $dif(Y, X)$, the difference from any 3-substar in Y to any 3-substar in X. So the path must satisfy **P2**. Similar argument can be extended to **R2**.

Lemma 3 *Given a 4-ring* $R = [X_0, X_1, \ldots, X_{r-1}]$ *such that* $dif(X_{(i-1)\mathrm{mod}r}, X_i) \neq dif(X_{(i+1)\mathrm{mod}r}, X_i)$ *for any* $i = 0..r-1$, *it is possible to construct a 3-ring* $R' = [X'_0, X'_1, \ldots, X'_{4r-1}]$ *from* R *such that* $dif(X'_{(i-1)\mathrm{mod}4r}, X'_i) \neq dif(X'_{(i+1)\mathrm{mod}4r}, X'_i)$ *for any* $i = 0..4r-1$.

Proof. First, we apply any (legal) cut on R. Let x be any of the three 3-substars in X_0 that have connections to X_1. Then, connect a path of 3-substars from x to X_1, X_2, etc., while in the process rules **R1** and **R2** must be followed. Note that there is no conflict in following both rules together because in any X_i the 3-substar that does not have a connection to X_{i-1} must be distinct from the 3-substar that does not have a connection to X_{i+1} (which is ensured by condition $dif(X_{i-1}, X_i) \neq dif(X_{i+1}, X_i)$).

When the path is built up to X_{r-1}, care must be taken to ensure that the last 3-substar visited in X_{r-1} is not adjacent to the starting 3-substar x. Then we can traverse X_0 and generate a 3-ring R' as desired. This step is possible because there are sufficient (three) connections from X_{r-1} to X_0. The proof is trivial and we leave it to the reader. ■

The next job is to construct a 4-ring as desired in Lemma 3 from a given 5-ring. The following lemma shows that any 5-ring can offer such possibility.

Lemma 4 *Given any 5-ring* $R = [X_0, X_1, \ldots, X_{r-1}]$, *it is possible to construct a 4-ring* $R' = [Y_0, Y_1, \ldots, Y_{5r-1}]$ *from* R *such that* $dif(Y_{(i-1)\mathrm{mod}5r}, Y_i) \neq dif(Y_{(i+1)\mathrm{mod}5r}, Y_i)$ *for any* $i = 0..5r-1$.

Proof. First, we apply any (legal) cut on R. For any two adjacent 5-substars X and Y, let x be the 4-substar in X that does not have a connection to Y, and y the one in Y that does not have a connection to X. Similar to **R1** and **R2**, we can derive two rules to construct a 4-ring:

R1': x is the first, second, or third 4-substar visited in X, and

R2': y is the third, forth, or fifth 4-substar visited in Y.

Using similar proving techniques as in Lemma 3, this lemma can be proved. We omit the details. However, as opposed to Lemma 3, note that this lemma does not rely on any relationship among X_{i-1}, X_i, X_{i+1} because if $dif(X_{i-1}, X_i) = dif(X_{i+1}, X_i)$, by **R1'** and **R2'**, the 4-substar in X_i that does not have a connection to both X_{i-1} and X_{i+1} still can be visited as the third one in X_i. ■

Below we put together the above lemmas into a complete algorithm. The algorithm finds a Hamiltonian cycle in any S_n with $n \geq 6$.

Algorithm *Ham*();

1) Apply an n-cut on S_n. Construct an $(n-1)$-ring (referred to as R_{n-1}) of length n from S_n.
2) **for** $k = n-1$ **downto** 6 **do**

   Apply a k-cut on R_k and then use Lemma 1 to construct from R_k a $(k-1)$-ring (referred to as R_{k-1}).
3) Apply a 5-cut on R_5 and construct from R_5 a 4-ring (referred to as R_4) using Lemma 4.
4) Apply a 4-cut on R_4 and construct from R_4 a 3-ring (referred to as R_3) using Lemma 3.
5) Construct from R_3 a 1-ring R_1 using Lemma 2. ■

When $n = 5$ (resp., 4), we can consider S_5 (S_4) as a trivial 5-ring R_5 (4-ring R_4) with a single node and directly run the algorithm from step 3 (step 4).

4 Ring Embedding When Links Fail

In this section, we enhance *Ham*() to tolerate at least $f_e \leq n-3$ faulty links. We first show how to tolerate one faulty link in Lemma 2.

Lemma 5 *In Lemma 2, if there exists a faulty link* e *which falls between two substars* X_i *and* X_{i+1}, *a 1-ring* R' *can still be constructed without using link* e.

Proof. Without loss of generality, we can assume that e falls between X_0 and X_1. Recall the proof of Lemma 2. We can traverse R from any of the two nodes in X_0 with a link connecting to X_1. Clearly link e can be avoided by choosing an appropriate x. ■

Note that in the above lemma, e may not be the only faulty link in the 3-ring. However, avoiding e already serves our need. The following lemma can be proved similarly.

Lemma 6 *In Lemma 1, Lemma 3, and Lemma 4, if there exists a faulty edge* e *which falls between two* k*-substars* X_i *and* X_{i+1}, *a* $(k-1)$*-ring* R' *still can be constructed without using link* e.

Using Lemma 5 and Lemma 6, we can tolerate at least one faulty link in each construction from R_{n-1} to R_{n-2}, from R_{n-2} to R_{n-3}, $\ldots$, from R_3 to R_1 (here we following the same notation as in *Ham*()). Thus we should be able to tolerate at least $n-3$ faulty links.

To use these two lemmas, we need to make sure that the faulty links are falling between two k-substars in R_k (observe that faulty links may be "encapsulated" within k-substars). This can be done by applying an appropriate cut on R_{k+1}. For instance, if a faulty link e along dimension j falls inside a $(k+1)$-substar in R_{k+1}, then we can apply a j-cut on R_{k+1} in the process of constructing R_k. Then two cases may happen: (a) e is not used in R_k at all (which is fine for us), or (b) e falls between two k-substars in R_k. Note that in the latter case e is ensured to be eliminated in the construction from R_k to R_{k-1} using the above lemmas.

The following embedding algorithm works for any S_n, $n \geq 6$, with $f_e \leq n-3$ faulty links.

Algorithm *Link-Failure*();

1) Let $D = (d_n, d_{n-1}, \ldots, d_4)$ be the sequence of dimensions such that numbers of faulty links falling on them are sorted in a descending order.
2) Execute steps 1 to 4 of algorithm *Ham*(), but apply a d_k-cut while constructing an R_{k-1} from R_k. Use Lemma 6 to avoid at least one (if any) faulty edge falling between two k-substars.
3) Construct from R_3 a fault-free 1-ring R_1 using Lemma 5. ■

Note that in step 1 we require the number of faulty links along dimension d_i be no less than that along dimension d_{i-1} so that faulty links may be avoided as early as possible. Also note that the above algorithm can be modified as we have done for *Ham*() in Section 3 to run for cases of $n = 4$ or 5.

Theorem 1 *Given an $S_n, n \geq 4$, with $f_e \leq n-3$ faulty links, algorithm Link-Failure() can find a fault-free Hamiltonian cycle in S_n.*

5 Ring Embedding When Nodes Fail

In this section we study the following problem: given an S_n with f_v faulty nodes, find a ring that is as large as possible without passing through any faulty node. Our main result shows that for any $f_v \leq n-3$ a ring of length at least $n! - 4f_v$ can be found.

We first consider the construction of a 1-ring from a 3-ring which has some faulty nodes. In Fig. 6(a), we show two adjacent 3-substars, through which a 1-ring passes (indicated by solid lines). Now suppose one node in the second 3-substar becomes faulty. In Fig. 6(b)–(g), we show how to "route around" the faulty node under six possible fault scenarios. Note that the routing is based on a simple greedy strategy by including as many nodes as possible. As one can observe, the number of nodes (both faulty and non-faulty) lost due to the failure is at most 4.

Lemma 7 *Given a 3-ring $R = [X_0, X_1, \ldots, X_{r-1}]$ in which (a) no two consecutive 3-substars both contain faulty nodes, (b) each 3-substar contains at most 1 faulty node, and (c) $dif(X_{(i-1) \bmod r}, X_i) \neq dif(X_{(i+1) \bmod r}, X_i)$ for any i, it is possible to construct a 1-ring R' of length at least $6r - 4f$ from R, where f is the number of faulty nodes in R.*

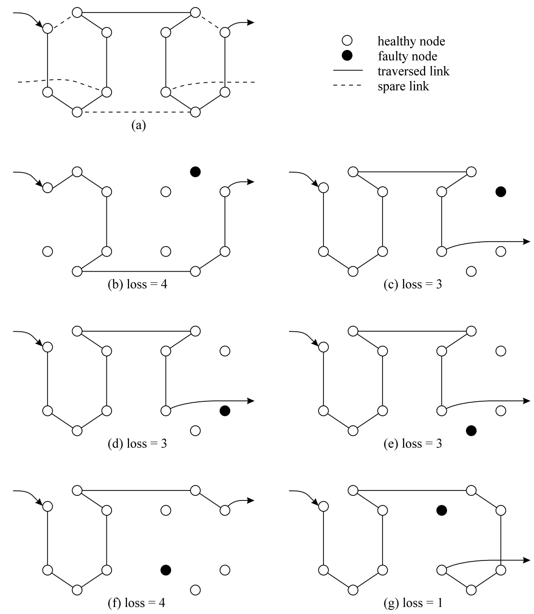

Figure 6. (a) The original routing on two adjacent healthy 3-substars, and (b)–(g) the fault-tolerant routing when one node in the second 3-substar becomes faulty.

Proof. By (a), let X_i be healthy and X_{i+1} contain a faulty node. By (b), let x be any node in X_i whose neighbor in X_{i+1} is healthy. We traverse R from x toward substars $X_{i+1}, \ldots, X_{r-1}, X_0, \ldots, X_{i-1}$, by applying the greedy strategy as indicated in Fig. 6. As discussed earlier, this will skip at most $4f$ nodes.

When the path returns back to X_i, the end node may be at a distance of 1 or 2 from x (the scenario is similar to that in Fig. 4). As X_i is fault-free, in the former case all nodes in X_i can be included in the ring, while in the latter case one more node will be excluded from the ring. Thus, the ring has a length $\geq 6r - 4f - 1$. As the ring length must be even, the lemma then follows. ■

Due to space limit, proof of the following lemmas is omitted and can be found in [10].

Lemma 8 *Given a 4-ring $R = [X_0, X_1, \ldots, X_{r-1}]$ in which (1) one of the 4-substars is fault-free, (2) each 4-substar contains at most 1 faulty node, and (3) $dif(X_{(i-1) \bmod r}, X_i) \neq dif(X_{(i+1) \bmod r}, X_i)$ for any i, it is possible to construct from R a 3-ring R' satisfying the conditions (a)–(c) in Lemma 7.*

Lemma 9 *In an $S_n, n \geq 4$, with $f_v \leq n-3$ faulty nodes, there always exists a D-cut, $|D| = n-4$, on S_n which results in 4-substars each containing at most one faulty node.*

Below we summarize the above discussion into an algorithm for ring embedding in an S_n, $n \geq 6$, with $f_v \leq n-3$

faulty nodes.

Algorithm *Node-Failure*();

1) Use Lemma 9 to find a sequence of dimensions $D = (d_n, d_{n-1}, \ldots, d_4)$.
2) Execute steps 1 to 3 of algorithm *Ham*(), but apply a d_k-cut in the construction from R_k to R_{k-1}.
3) Construct from R_4 a 3-ring R_3 using Lemma 8.
4) Construct from R_3 a 1-ring R_1 using Lemma 7. ■

After steps 1 to 3, a 4-ring R_4 is obtained. Note how the conditions (1)–(3) in Lemma 8 are satisfied. Condition (3) is guaranteed by algorithm *Ham*(). Condition (2) is ensured by Lemma 9. Condition (1) holds because the number of 4-substars $(n-4)! > n-3 \geq f_v$ for any $n \geq 6$. The correctness then follows directly from Lemma 8 and Lemma 7. Note again that the above algorithm can be easily modified to run for cases of $n = 4$ or 5.

Theorem 2 *Given an $S_n, n \geq 4$, with $f_v \leq n-3$ faulty nodes, algorithm Node-Failure() can find a fault-free ring of length $\geq n! - 4f_v$.*

6 When Both Links and Nodes Fail

By combining algorithms *Link-Failure*() and *Node-Failure*(), we can tolerate both link and node failure. Due to space limit, we only summarize the result. For details, refer to [10].

Theorem 3 *Given an S_n with f_e faulty links and f_v faulty nodes, where $f_e + f_v \leq n-3$, algorithm Link-Node-Failure() can find a fault-free ring of length $\geq n! - 4f_v$.*

References

[1] S. B. Akers, D. Harel, and B. Krishnameurthy. The star graph: An attractive alternative to the n-cube. In *Int'l Conf. on Parallel Processing*, pages 393–400, 1987.

[2] K. Day and A. Tripathi. A comparative study of topological properties of hypercubes and star graphs. *IEEE Trans. on Paral. and Distrib. Sys.*, 5(1):31–38, Jan. 1994.

[3] P. Fragopoulou and S. G. Akl. Optimal communication algorithms on star graphs using spanning tree constructions. *J. of Parallel and Distrib. Comput.*, 24:55–71, 1995.

[4] J.-S. Jwo, S. Lakshmivarahan, and S. K. Dhall. Embeddings of cycles and grids in star graphs. In *Symp. on Parallel and Distrib. Processing*, pages 540–547, 1990.

[5] V. E. Mendia and D. Sarkar. Optimal broadcasting on the star graph. *IEEE Trans. on Paral. and Distrib. Sys.*, 3(4):389–396, July 1992.

[6] M. Nigam, S. Sahni, and B. Krishnamurthy. Embedding hamiltonians and hypercubes in star interconnection graphs. In *Int'l Conf. on Parallel Processing*, pages III–340–343, 1990.

[7] K. Qiu. Broadcasting on the star and pancake interconnection networks. In *Int'l Parallel Processing Symp.*, pages 660–665, 1995.

[8] K. Qiu, S. G. Akl, and H. Meijer. On some properties and algorithms for the star and pancake interconnection networks. *J. of Parallel and Distrib. Comput.*, 22:16–25, 1994.

[9] J.-P. Sheu, C.-T. Wu, and T.-S. Chen. An optimal broadcasting algorithm without message redundancy in star graphs. *IEEE Trans. on Paral. and Distrib. Sys.*, 6(6):653–658, June 1995.

[10] Y.-C. Tseng, S.-H. Chang, and J.-P. Sheu. Fault-tolerant ring embedding in star graphs. Technical report, Chung-Hua Polytechnic Inst., Computer Sci. Dept., 1995.