Computer Science
Technical Report

Colorado State University

Combinatorial Analysis of Star Graph Networks

Yordan Rouskov*, Shahram Latifi† and Pradip K Srimani‡

Technical Report CS-95-101

Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873

Phone: (303) 491-5792 Fax: (303) 491-2466
WWW: http://www.cs.colostate.edu

* Department of Computer Science, Colorado State University, Ft. Collins, CO 80523
† Department of Electrical Engineering, University of Nevada, Las Vegas, NV 89154
‡ Department of Computer Science, Colorado State University, Ft. Collins, CO 80523

Combinatorial Analysis of Star Graph Networks

Yordan Rouskov*, Shahram Latifi† and Pradip K Srimani‡

Abstract

It is well known that star graphs are strongly resilient like the n cubes in the sense that they are optimally fault tolerant and the fault diameter is increased only by one in the presence of maximum number of allowable faults [Lat93b, RS93]. We investigate star graphs under the conditions of *forbidden faulty sets* [Esf89] where all the neighbors of any node cannot be faulty simultaneously; we show that under these conditions star graphs can tolerate upto $(2n-5)$ faulty nodes and the fault diameter is increased only by 2 in the worst case in presence of maximum number of faults. Thus, star graphs enjoy the similar property of strong resilience under forbidden faulty sets like the n-cubes [Lat93a]. We have developed algorithms to trace the vertex disjoint paths under different conditions.

1 Introduction

The underlying topology of any multiple processor system is, in general, modeled as an undirected graph where the nodes represent the processing elements and the arcs (edges) represent the bidirectional communication channels. Design features for an efficient interconnection topology include properties like low degree, regularity, small diameter, high connectivity, efficient routing algorithms, high fault-tolerance, low fault diameter etc. Since more and more processors must work concurrently these days in a multiple processor environment, the criterion of high fault tolerance and strong resilience has become increasingly important. One of the most efficient interconnection network has been the well known binary n-cubes or hypercubes; they have been used to design various commercial multiprocessor machines [Sei85] and they have been extensively studied. Very recently another family of regular graphs, called the star graphs [AK89, AK87], are being extensively studied; star graphs seem to enjoy most of the desirable properties [QMA92, NSK90, QMA91, DT94] of the hypercubes at considerably less cost; they accommodate more nodes with less interconnection hardware and less communication delay. It has also been shown [QAM93, MS90, FA91, MS92] that many parallel algorithms can be efficiently mapped on these star graphs.

The focus of this paper is on fault tolerance of these star graph networks. The fault tolerance of an interconnection network is usually measured by the *vertex connectivity* of the underlying graph as well as the *fault diameter*. Vertex connectivity of an n-cube (which is a n-regular graph) is n and the corresponding fault diameter is $n+1$ (the fault-free diameter is n) [SS88]. Vertex connectivity of a star graph S_n of dimension n (which is a $(n-1)$ regular graph) is $n-1$ [AK89] and the corresponding fault diameter is $\lfloor 3(n-1)/2 \rfloor + 1$ [Lat93b, RS93] (the fault-free diameter is $\lfloor 3(n-1)/2 \rfloor$). Star graphs thus are comparable to hypercubes also in the sense that the fault diameter is increased only by one over the fault free diameter.

Although the measure of vertex connectivity correctly reflects the fault tolerance of systems with few processors, it underestimates the resilience of large networks. Esfahanian [Esf89] has emphasized and elaborated on the second point with respect to the n-cubes. Let us illustrate the point with respect to the star graphs. Note that each minimum cut in S_n is of size $n-1$ and also the fact that a subset of $(n-1)$ vertices can be a cut when, and only when, these vertices represent the entire adjacency set

*Department of Computer Science, Colorado State University, Ft. Collins, CO 80523
†Department of Electrical Engineering, University of Nevada, Las Vegas, NV 89154
‡Department of Computer Science, Colorado State University, Ft. Collins, CO 80523

of any vertex in S_n. Thus, out of all possible $\binom{n!}{n-1}$ vertex subsets of size $n-1$, only $n!$ are minimum cuts of the star graph S_n. Assuming that every fault set is equally probable, the probability that an arbitrary fault set of size $n-1$ will disconnect the remaining network is very small and gets smaller as n grows large. This motivates one to define the restricted connectivity or connectivity under *forbidden faulty sets* [Esf89]. In particular, for the n-cube, Esfahanian has defined each forbidden faulty set to be all n neighbors of one processor (thus there are 2^n forbidden faulty sets in an n-cube, each containing n processors). Esfahanian has shown that an n-cube, under the forbidden sets, can tolerate up to $(2n-3)$ processor failures without being disconnected; Latifi [Lat93a] has shown that the fault diameter of n-cube under forbidden faulty sets is $n+2$, i.e., the fault diameter is increased only by 2 over the fault free diameter.

In this paper, our purpose is to investigate the star graphs under forbidden faulty sets and to show that a star graph, under the forbidden sets, can tolerate up to $(2n-5)$ processor failures without being disconnected and that the fault diameter is increased only by 2 over the fault free diameter. Thus we show that the star graphs also enjoy the same strong resilience properties like n-cubes and hence are strong competitors of n-cubes for large multiprocessor design. We also design algorithms to trace vertex disjoint paths in star graphs under the forbidden faulty sets. The rest of the paper is organized as follows. In section 2 we introduce the notations and terminologies. Section 3 treats the restricted vertex connectivity of star graphs and its fault diameter. Section 4 concludes the paper.

2 Basic Concepts

In this section we briefly introduce the basic terminology about star graphs and identify certain structural properties of these graphs. Graph theoretic terms not defined here can be found in [Har72]. A detailed treatment of star graphs can be found in [AK89, AK87] and more details about forbidden faulty sets can be found in [Esf89].

2.1 Star Graphs

A star graph S_n, of order n, is defined to be a symmetric graph $G=(V,E)$ where V is the set of $n!$ vertices, each representing a distinct permutation of n elements and E is the set of symmetric edges such that two permutations (nodes) are connected by an edge iff one can be reached from the other by interchanging its first symbol with any other symbol. For example, in S_3, the node representing permutation ABC have edges to two other permutations (nodes) BAC and CBA. Throughout our discussion we denote the nodes by permutations of English alphabets. For example, the identity permutation is denoted by $I=(ABCD...Z)$ (Z is the last symbol, not necessarily the 26th).

Remarks:

- These star graphs are members of the family of Cayley group graphs. For a star graph S_n of dimension n, there are $n-1$ generators, $g_2, g_3, \cdots, g_n$, where g_i swaps the first symbol with the i-th symbol of any permutation. Each generator is its own inverse, i.e., the star graph is symmetric. Also, the star graph S_n is a $(n-1)$-regular graph with $n!$ nodes and $n!(n-1)/2$ edges.

- Since star graphs are vertex symmetric [AK89], we can always view the distance between any two arbitrary nodes as the distance between the source node and the identity permutation by suitably renaming the symbols representing the permutations. For example, let $BDECA$ be the source node and $ECDAB$ be the destination node. We can map the destination node to the identity node by renaming the symbols as $E \mapsto A$, $C \mapsto B$, $D \mapsto C$, $A \mapsto D$, and $B \mapsto E$. Under this mapping the source node becomes $ECABD$. Then the paths between the original source and destination nodes become isomorphic to the paths between the node $ECABD$ and the identity node $ABCDE$ in the renamed graph. Hence, in our subsequent discussion about a path from a source node to a destination node, the destination node is always assumed to be the identity node I without any loss of generality. Also, we use distance of a node (permutation) to indicate its distance to the identity node.

- It is easy to see that any permutation of n elements can also be specified in terms of its cycle structure with respect to the identity permutation I. For example, $CDEBAF = (ACE)(BD)(F)$. The maximum number of cycles in a permutation of n elements is n and the minimum number is 1. When a cycle has only one symbol, that symbol is in its correct position in the permutation with respect to the identity permutation. The singleton cycles may be omitted in the cycle representation of a permutation if the number of symbols in the permutation is understood from the context.

We use following notations throughout the paper:

n: Order of the star graph S_n.

N: Number of nodes in S_n, $N = n!$.

Δ_n: Diameter of S_n, $\Delta_n = \lfloor 3(n-1)/2 \rfloor$.

u: An arbitrary permutation.

$symbol_u(i)$: The i-th symbol in the permutation of the node u, $1 \leq i \leq n$ (the symbols are numbered 1 through n from left to right).

$position_u(\alpha)$: An integer denoting the position of the symbol α in the permutation of the node u.

$D(u)$: Distance of the node u (from the identity permutation).

$\Pi(u)$: Cycle representation of the node (permutation) u, $\Pi = \{\pi_1, \pi_2, \cdots, \pi_k\}$. We omit all singleton cycles in the representation except the one containing the symbol A (if any) and π_1 is the cycle containing A and all other cycles are in any arbitrary order. Furthermore, each cycle π_i is represented as a sequence of symbols, $\pi_i = (\pi_i^1, \pi_i^2, \cdots, \pi_i^{|\pi_i|})$; π_1 is always represented as $(\pi_1^1, \pi_1^2, \cdots, A)$, while other cycles may be represented in any arbitrary order. Thus, $CDEBAF$ is represented as $(CEA)(BD)$. Hence the symbol π_1^1 of any permutation is the first symbol of that permutation.

$\mu(u)$: Number of cycles of length at least 2 in any permutation u.

$m(u)$: Total number of symbols in these μ cycles of the permutation u.

$n - m(u)$ Total number of *invariant* symbols in a permutation u, i.e., the number of singleton cycles in u.

It has been shown in [AK89] that the distance $D(u)$ for a given node (permutation) u is given by:

$$D(u) = \begin{cases} \mu + m & \text{when A is the first symbol} \\ \mu + m - 2 & \text{when A is not the first symbol} \end{cases}$$

Lemma 1 *[RS93] For odd n, a node (permutation) u in S_n is at the maximum distance $\lfloor 3(n-1)/2 \rfloor$ from I, iff its cycle structure satisfies the following: (i) A is the first symbol in u (i.e. A forms a 1-cycle) (ii) All other cycles are 2-cycles.*

Lemma 2 *[RS93] For even n, any permutation u has the maximum distance Δ_n iff its cycle structure satisfies one of the following two: (i) A is the first symbol in u (i.e., A forms a 1-cycle), three other symbols form a 3-cycle and the rest form 2-cycles, (ii) A is not in the first position in u, and all cycles are of length 2 ($n/2$ of them).*

Remark 1 *[AK89] For an arbitrary node u in S_n if $D(u) \geq \Delta_n - 3$, the node (permutation) u can have at most two invariant symbols and if $D(u) \geq \Delta_n - 2$ then the node (permutation) u can have at most one invariant symbol.*

Shortest Routing in Star Graph

Given an arbitrary node $u = \{\pi_1, \pi_2, \cdots, \pi_k\}$, the algorithm to find the shortest path to the destination node (the identity permutation) is described in [AK87]; it has also been shown that in general this shortest path is not unique, i.e., there may be multiple shortest paths to the destination from a given node u. In order to generate a path from u to I, generators are applied in a particular sequence. In order that the path is minimal in length, the moves (generating the next vertex in the sequence) must be restricted to two possible choices : for any node $u = \{\pi_1, \pi_2, \cdots, \pi_k\}$, **either (1)** exchange π_1^1 with π_1^2 thereby reducing the length of the cycle π_1 by 1, **or (2)** exchange π_1^1 with π_j^ℓ (where $2 \leq j \leq k$ and $1 \leq \ell \leq |\pi_j|$ thereby merging the two cycles π_1 and π_j into one cycle $(\pi_j^\ell, \pi_j^{\ell+1}, \cdots, \pi_j^{|\pi_j|}, \pi_j^1, \cdots, \pi_j^{\ell-1}, \pi_1^1, \cdots, A))$. When a path from a node u to the destination is said to follow the *shortest routing scheme* if the moves are restricted to these two types. We make the following observations about this shortest routing scheme:

- For any node u at least one of these moves is always possible until the node I is reached.
- The path generated by using only these two moves from a node u to the destination node I has a length $D(u)$.
- Any other move leads to a non minimal path. More specifically, if a node u' is reached from u by any other move, then $D(u') = D(u) + 1$. Also, if we go to a node u'' from u' by a non-minimal move, then $D(u'') = D(u') + 1 = D(u) + 2$.

2.2 Forbidden Faulty Sets

The set $A(v)$ ($A(v)$ denotes the adjacency set of any vertex v) of any vertex v is named as the *forbidden faulty set*; it implies that all the neighbors of any vertex in S_n cannot be faulty simultaneously. The restricted vertex connectivity $\kappa^{\mathcal{R}}(S_n)$ of the star graph S_n is defined as the minimum cardinality $|Y|$ of vertices such that the subgraph $S_n - G$ is disconnected and Y is restricted to the set of vertices $R = \{Z \subset S_n | \forall v \in S_n,\ A(v) \notin Z\}$. Thus, star graph can tolerate up to $\kappa^{\mathcal{R}}(S_n) - 1$ processor failures provided that for any node in the graph all of its adjacent nodes do not fail simultaneously. We use $\Delta_n^{\mathcal{R}}$ to denote the diameter of the star graph in presence of an arbitrary set of $\kappa^{\mathcal{R}}(S_n) - 1$ node failures. Our purpose in the next section is to prove that $\kappa^{\mathcal{R}}(S_n) = 2n - 4$ and that $\Delta_n^{\mathcal{R}} = \Delta_n + 2$. We present two quick lemmas to establish the upper bounds.

Lemma 3 *$\kappa^{\mathcal{R}}(S_n)$ is upper bounded by $2n - 4$.*

Proof : Consider any arbitrary edge (u, v) of S_n. $|A(u, v)| = 2n - 4$ since S_n doesn't have a cycle of length less than 6. If these $2n - 4$ nodes fail, the particular edge (u, v) will be disconnected from the remaining graph. 2

Lemma 4 *$\Delta_n^{\mathcal{R}}$ is lower bounded by $\Delta_n + 2$.*

Proof : Consider the scenario in Figure 1; node u has a single fault free neighbor v, node v has just one more fault free neighbor w such that $D(w) = \Delta_n$. Total number of faulty nodes is $2n - 5$ and the fault set does not contain any forbidden faulty set. Distance of the node u from the identity node is $\Delta_n + 2$. 2

3 Fault Diameter with Forbidden Faulty Sets

3.1 Nodes for which $|\pi_1| = 1$

Here we consider the nodes where the symbol A forms a singleton cycle and A is the first symbol of the permutation denoting the arbitrary node u. Let S_{n-1}^i denote the subgraph of the star graph S_n consisting of nodes (permutations) having the symbol A in their i-th positions, $2 \leq i \leq n$; each of these subgraphs is a star graph of dimension $n - 1$ with $(n - 1)!$ nodes. The subgraph S_{n-1}^1 is a set of $(n - 1)!$ nodes (each with A as its first symbol) and no edges. Consider the $n - 1$ vertex disjoint paths (as computed in [RS93]) from a node u (with $|\pi_1| = 1$) to the identity node I; it is easy to observe that $u, I \in S_{n-1}^1$ and each

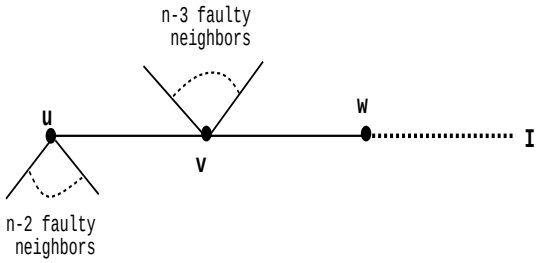

Figure 1: Lower bound for $\Delta_n^{\mathcal{R}}$

of the $n-1$ vertex disjoint paths is contained exclusively in one subgraph S_{n-1}^i, $2 \leq i \leq n$. Let u_i and I_i denote the adjacent nodes of u and I respectively with the symbol A in the i-th position.

Remark 2 *Since both u and I have A as first symbol, u_i and I_i are defined for all i, $2 \leq i \leq n$; also, $u_i, I_i \in S_{n-1}^i$ for $2 \leq i \leq n$.*

Remark 3 *Since not all neighbors of a node are allowed to be faulty simultaneously, there exists at least one non faulty neighbor u_i of node u and one non faulty neighbor I_j of node I; but i and j may not be equal, in general. In other words, the nodes u and I may not have non faulty neighbors in the same subgraph S_{n-1}^i, in general.*

Our objective is to show that in presence of $(2n-5)$ maximum faults, there exists a path from u to I and establish its length. Obviously, if any of the substars S_{n-1}^i does not have any faulty node, we are done since there exists a path from u to I via that subgraph [RS93]. We make the following observation:

Remark 4 *If any substar S_{n-1}^i has $\geq n-2$ faulty nodes, there exists at least one other substar S_{n-1}^j, $j \neq i$, that has no faulty nodes (since total number of faulty nodes cannot exceed $2n-5$ and number of substars is $n-1$).*

So, we assume that each substar S_{n-1}^i has no more than $(n-3)$ faulty nodes. Let α_i denote the symbol that is exchanged with the symbol A to go from u to u_i and β_i denote the symbol that is exchanged with the symbol A to go from I to I_i respectively for all i, $2 \leq i \leq n$. We consider two cases:

Case 1: There exists an i such that both u_i and I_i are non faulty

Here, we go from the node u to u_i (by exchanging the symbol α_i with A), trace a path from u_i to I_i in S_{n-1}^i, and go from I_i to I (by exchanging the symbol β_i with A). We show there are $(n-2)$ vertex disjoint paths in S_{n-1}^i between the nodes u_i and I_i.

Subcase A: $\alpha_i \neq \beta_i$. In this case, the symbol α_i is contained in a cycle of length ≥ 2 of the node u, i.e., the move from u to u_i belongs to the *shortest routing scheme*. By similar reasons, the move from I_i to I also belongs to the shortest routing scheme. Hence $D(u_i, I_i) = D(u) - 2$. The subgraph S_{n-1}^i is a star graph of dimension $n-1$ and hence there are $n-2$ vertex disjoint paths between two arbitrary nodes u_i and I_i each of which has a length $\leq D(u_i, I_i) + 4$. Hence, in presence of a maximum number of $n-3$ faulty nodes in S_{n-1}^i, there exists a path from node u to I (via nodes of S_{n-1}^i) of maximum length $D(u) + 4$.

Subcase B: $\alpha_i = \beta_i$. In this case, the symbol α_i is invariant in node u (i.e., is in its correct position) and hence the move from u to u_i does not belong to the *shortest routing scheme*. Hence, $D(u_i, I_i) = D(u)$. Now, S_{n-1}^i is a star graph of dimension $n-1$ and the source node u_i and destination node I_i have the same first symbol ($\alpha_i = \beta_i$) and hence there are $n-2$ vertex disjoint paths between u_i and I_i each of which has a length $\leq D(u_i, I_i) + 2$. Hence again, in presence of a maximum number of $n-3$ faulty nodes in S_{n-1}^i, there exists a path from node u to I (via nodes of S_{n-1}^i) of maximum length $D(u) + 4$.

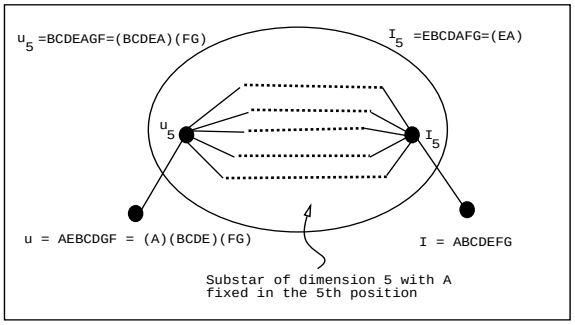

Figure 2: Case 1, Subcase A with $\alpha = A$ and $\beta = E$

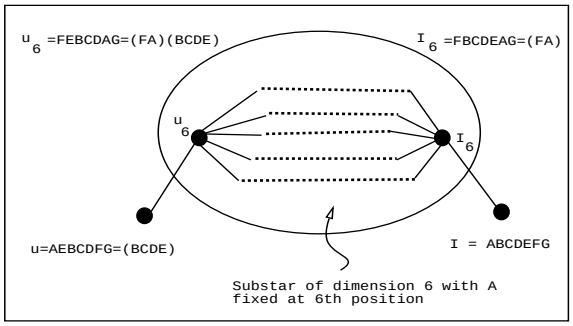

Figure 3: Case 1, Subcase B with $\alpha = F$ and $\beta = E$

Lemma 5 *For case 1, in presence of maximum $(2n-5)$ faulty nodes, there exists a path from node u to node I whose length is at most $\Delta_n + 2$.*

Proof : If $D(u) \leq \Delta_n - 2$, the claim easily follows from the above discussion of two subcases. If $D(u) \geq \Delta_n - 1$, node u has at most one invariant symbol (by Remark 1). In subcase A, each of the $n-2$ vertex disjoint paths in the subgraph has a length of $D(u_i, I_i) + 2$ (case of single invariant symbol, theorem 5 of [RS93]). In subcase B, the node u_i is reached by exchanging the only invariant α_i, nodes u_i and I_i have the same first symbol, and hence each of the $n-2$ vertex disjoint paths in the subgraph has a length of $D(u_i, I_i)$ (algorithm A in [RS93]). Hence the claim follows. 2

Example 1: Figure 2 and Figure 3 illustrate the two subcases with example nodes.

Case 2: There does not exist an i such that both u_i and I_i are non faulty

In this case, a total of $n-1$ faults have already been located among the neighbors of u and I. By the presumption of forbidden faulty sets, there exist at least one good neighbor of u, say u' and one good neighbor of I, say I'. Let α be the symbol that is exchanged with the symbol A to make the move $u \longrightarrow u'$; if α is not an invariant in u, then $D(u') = D(u) - 1$ and $D(u', I')$ is at most $D(u)$; otherwise, f α is an invariant in u, then $D(u') = D(u) + 1$ and $D(u', I')$ is at most $D(u) + 2$. There are $(n-1)$ vertex-disjoint paths between the nodes u' and I' in the star graph S_n; consider those $(n-3)$ of them that does not go through the nodes u or I; none of these paths can go through any immediate neighbor of the nodes u and I (since S_n does not have any cycle of length less than 6; see [RS93] or [Lat93b]). Since $n-1$ faults have already been located among the neighbors of u and I and the maximum number of faults is $(2n-5)$, all of these $(n-3)$ paths between u' and I' cannot be disconnected and hence the nodes u and I remain connected in presence of maximum number of faults.

Remark 5 *Each of those $(n-3)$ paths between u' and I' has a length of at most $\Delta_n + 1$ [RS93]; hence the length of the surviving path between nodes u and I is upper bounded by $\Delta_n + 3$.*

Remark 6 *An interesting special case is when either u or I (or both) has only one good neighbor. Suppose u' is the only good neighbor of u. Then there are $n-2$ vertex disjoint paths from u' to I (not to some I') that do not contain any of the other (faulty) neighbors of u and hence at least 1 cannot be disconnected. The same arguments apply if the only good neighbor of I is some I'. There are $n-2$ paths from u (not from u') to I' do not contain any other neighbor of I and at least 1 is guaranteed to be good. Also, the length of the surviving path(s) is upper bounded by $\Delta_n + 2$.*

Lemma 6 *The surviving path between u and I has a length of at most $D(u) + 8$.*

Proof : The distance between u' and I' is at most $D(u) + 2$. Any of the $n - 3$ vertex disjoint paths can require at most 2 non-optimal moves [RS93]; thus each of these paths has a length of at most $D(u) + 6$. Hence the surviving path between u and I has a length of at most $D(u) + 8$. 2

Lemma 7 *For Case 2, at least one path survives between u and I of length not more than $\Delta_n + 2$.*

Proof : We need to consider several cases separately.

- If $D(u) \leq \Delta_n - 6$, then the claim follows from Lemma 6
- $\Delta_n - 5 \leq D(u) < \Delta_n - 3$. Here, the node (permutation) u can have at most 3 invariant symbols, say δ, σ, ω. The distance $D(u', I')$ can be $D(u) + 2$ iff one of the invariants (say δ) is exchanged to reach u' from u and another one (say σ) is exchanged to reach I' from I. Then the nodes u' and I' are of the forms: $u' = (\delta, A)(\sigma)(\omega)(...)$ and $I' = (\sigma, A)$. Any vertex disjoint path from u' to I' requires at most one non-optimal move [see [RS93]]; hence any surviving path between u' and I' is of length at most $D(u) + 4$ and the overall path between u and I is of length at most $D(u) + 6 \leq \Delta_n + 2$.
- $\Delta_n - 3 \leq D(u) < \Delta_n - 1$. Node u can have at most 2 invariant symbols (say, σ and δ) and all other cycles of u contain 2 symbols, except possibly one that has 3. Then $u = (A)(\delta)(\sigma)(\alpha, \beta, \omega)(...)$. In order that a path between u and I has a length $\leq \Delta_n + 2$, or equivalently $D(u) + 4$, either (1) $D(u', I') = D(u) - 2$, or (2) $D(u', I') = D(u)$ and the path between u' and I' requires at most one non-optimal move. By Remark 6, both nodes u and I have at least two good neighbors, i.e., u must have a good neighbor ug_X where X is a symbol such that $X \notin \{\delta, \sigma\}$. Without loss of generality assume $ug_\alpha = u' = (\alpha, \beta, \omega, A)(\delta)(\sigma)(...)$ is a good neighbor if u. Since condition (1) is not satisfied, possible good neighbors of I are Ig_ω, Ig_β, Ig_σ, Ig_δ. Since condition (2) is not satisfied, Ig_σ or Ig_δ cannot be good neighbors of I. So the only possible good neighbors of I are Ig_ω, or Ig_β. Now, u has at least one other good neighbor. If ug_X is a good neighbor of u and $X \in \{\delta, \sigma\}$ (note that X cannot be ω or β), condition (1) is satisfied and if $X \in \{\delta, \sigma\}$, condition (2) is satisfied. Thus the surviving path between u and I has a length of at most $\Delta_n + 2$. If u has less than 2 invariant symbols, the claim follows using similar arguments.
- $\Delta_n - 1 \leq D(u) \leq \Delta_n$. Node u has a specific cycle structure: it has one invariant symbol and other symbols are contained in doubleton cycles, or it has no invariant symbol with at most one cycle of length 3 or 4 or 5. In order that a path between u and I has a length $\leq \Delta_n + 2$, , either (1) $D(u', I') = D(u) - 2$ and each of the paths between u' and I' requires at most one non-optimal move, or (2) $D(u', I') = D(u)$ and none of the paths between u' and I' requires any non-optimal move. Proceeding along similar lines as above and treating the two cases of presence and absence of the invariant symbol separately, it is easy to show that the surviving path between u and I has a length of at most $\Delta_n + 2$.

2

Theorem 1 *In presence of up to $2n - 5$ faults (subject to the restriction of the forbidden faulty sets) the distance of an arbitrary node u in S_n with $|\pi_1| = 1$ is $\leq \Delta_n + 2$.*

Proof : The proof readily follows from Lemma 5 and Lemma 7. 2

3.2 Nodes for which $|\pi_1| > 1$

Here we consider the nodes where the symbol A does not form a singleton cycle, i.e., A is not the first symbol of the permutation denoting an arbitrary node u. Let α be the *first symbol* of the permutation denoting node u. Also, let S^i, $2 \leq i \leq n$, denote the substar of $(n - 1)!$ nodes each of which has the symbol α fixed at position i.

Remark 7 *The substars S^i for each i are mutually disjoint and the given node u has exactly one neighbor in each substar S^i, $2 \leq i \leq n$.*

Now, consider the neighbors of the node I; each has the cycle structure $Ig_\gamma = (\gamma, A)$ where γ is any symbol other than A. Also, each of these I-neighbors has $(n-2)$ neighbors of the form $Ig_\gamma g_{\gamma'} = (\gamma', \gamma, A)$, $\gamma \neq \gamma'$ (also, neither γ nor γ' is the symbol A). By the assumption of forbidden faulty sets, there exists at least one good (non faulty) neighbor of node u and one good neighbor of node I. We need to consider 2 different cases:

Part 1: For a given node u (i.e., given symbol α), either the node (α, A) is non faulty or there exists at least one symbol γ (γ is neither α nor A) such that the node (γ, A) and node (α, γ, A) are non faulty.

Part 2: For a given node u (i.e., given symbol α), the node (α, A) is faulty and for every other symbol γ ($\neq \alpha$ or A) either the node (γ, A) is faulty or the node (α, γ, A) is faulty or both are faulty.

3.2.1 Part 1

Notation: Let Φ be the set of good neighbors of the node u and Ω denote the set of good neighbors of node I of the form (α, A) or of the form (γ, A) such that the node (α, γ, A) is also fault-free. By assumption of the present case, $|\Omega| \geq 1$ and by the nature of forbidden faulty sets, $|\Phi| \geq 1$. Let $|\Phi| = k_1$, $|\Omega| = k_2$ and $k = \min(k_1, k_2)$. Also, Let β_i be the symbol whose correct position (i.e., in node I) is i, $2 \leq i \leq n$ ($\beta_i \neq \alpha$). Consider an arbitrary substar S^i for some i, such that $\beta_i \neq \alpha$; call this β_i simply β, and this S^i simply S' (there are $(n-2)$ such substars S'). The node u has exactly one neighbor in each S'; call this neighbor u'.

Remark 8 *For any such choice of substar S', the symbol α (determined by the original source node u) is fixed for all nodes in S' at the position of the symbol β (determined by the choice of S').*

Remark 9

- *All nodes in Φ and Ω are fault free (good) by the conditions of this subsection and so are the nodes (α, γ, A) for each node $(\gamma, A) \in \Omega$ ($\gamma \neq \alpha$).*
- *The move from node (α, γ, A) to the node (γ, A) belongs to the shortest routing scheme.*

Definition 1 *There are $(n-3)$ nodes in S' (call this subset of nodes of S' to be $X[S']$) each with cycle structure $(\beta, \alpha, \gamma, A)$ (for $(n-3)$ different symbols γ), and one node with cycle structure (β, α, A) (call this node to be $Y[S']$).*

Remark 10 *Each of these nodes in $X[S']$ or $Y[S']$ can reach, by application of the generator g_β, a node of the form (α, γ, A) or (γ, A) and by hypothesis of this case (subsection 3.2.1) at least one of these nodes can reach the node I by at most two moves.*

Thus, for each choice of S' (and u') there are exactly $(n-3)$ nodes in $X[S']$ and one node in $Y[S']$. We design algorithms to trace $(n-2)$ vertex disjoint paths from the node u' to these $(n-2)$ nodes in $X[S'] \cup Y[S']$.

Algorithm A:

Input: A node $u' = ug_\alpha = \{\pi'_1, \pi'_2, \cdots, \pi'_k\}$ in S'; the symbol α is fixed in the correct position of the symbol β for all nodes in S'. There are $(n-3)$ other symbols in our symbol set; call any one of them γ.
Output: $(n-3)$ different paths from the node u' to the nodes in the set $X[S']$, i.e., one path each for each choice of γ from node u' to the node $(\beta, \alpha, \gamma, A)$.
Note: Name an arbitrary symbol from a cycle $\neq \pi'_1$ to be σ. The proposed algorithm is in fact a collection of three procedures: each one is used in a particular scenario as follows (see appendix for the details of the procedures):

If the cycle π'_1 of the node u' contains symbols α, β and A and β is the first symbol of π'_1
then call **Procedure A1**;

else if the cycle π'_1 of the node u' contains symbols α, β, but β is not the first symbol of π'_1
then call **Procedure A2**;
else if α and β are in some cycle π'_ℓ (of node u') not containing the symbol A
then call **Procedure A3**.

Remark 11

- *When procedure A1 is invoked, the move from node u to node u' does not belong to shortest routing scheme (invariant symbol β is exchanged with α).*
- *When procedure A2 is invoked, the move from node u to node u' belongs to shortest routing scheme (β was a symbol from another cycle).*
- *When procedure A3 is invoked, the move from node u to node u' does not belong to shortest routing scheme (the cycle containing A is split).*

Remark 12 *The following paths from from u' to $X[S']$ require* **at most** *one move that does not belong to shortest routing scheme:*

Procedure A1: Case 1 subcaste b, Case 2 subcaste a, Case 3 sub cases b, and c
Procedure A2: Case 2, Case 3
Procedure A3 : Case 1, Case 2, Case 4

Remark 13 *All other paths require at most two moves that do not belong to shortest routing scheme; however, for the following cases there is exactly one such move:*

Procedure A1 Case 1 sub case a and Case 3 sub case d,
Procedure A3 Case 3
Procedure A2 Case 5
when γ **is not** *an invariant in u'.*
Procedure A2 Case 1 if Step 1 happens to be an optimal move,
Procedure A2 Case 4 if $position_{u'}(A) = position_I(\gamma)$

Algorithm B:

Input: A node $u' = ug_\alpha = \{\pi'_1, \pi'_2, \cdots, \pi'_k\}$ in S'; the symbol α is fixed in the correct position of the symbol β for all nodes in S'.
Output: A path from the node u' to the node $Y[S'] = (\beta, \alpha, A)$.

Case 1: α, β, A **are all in the cycle** π'_1

Sub case a: β **is the first symbol of** u'
If π'_1 contains at least one other symbol, say δ, than α, β, A **then**
$(\beta, \alpha, \delta, .., A)(...)$ [Node u'] $\xrightarrow[\text{Step 1}]{\text{Exchange } A \text{ with } \beta}$ $(A)(\beta, \alpha, \delta, ..)(...)$

$\xrightarrow[\text{Step 2}]{\text{Exchange } A \text{ with } \delta}$ $(..\beta, \alpha, A)(...)$ [Node XX]

$\xrightarrow[\text{Resolve the last cycle}]{\text{Merge all cycles and}}$ (β, α, A) [target node]

Else $(\beta, \alpha, A)(...)$ [Node u', also Node XX] $\xrightarrow[\text{Resolve the last cycle}]{\text{Merge all cycles start by exchanging } \sigma}$
(β, α, A) [target node]

Sub case b: β **is not the first symbol of** u'

If π_1' contains at least one symbol, say δ, between α and A **then**

$(.., \beta, \alpha, \delta, .., A)(...)$ [Node u'] $\xrightarrow[\text{Step 1}]{\text{Bring symbol } A \text{ to first position}}$ $(A)(.., \beta, \alpha, \delta)(...)$

$\xrightarrow[\text{Step 2}]{\text{Exchange } A \text{ with } \delta}$ $(\delta, .., \beta, \alpha, A)(...)$ [Node XX]

$\xrightarrow{\text{Merge all cycles and Resolve the last cycle}}$ (β, α, A) [target node]

Else

$(\omega, .., \beta, \alpha, A)(...)$ [Node u'] $\xrightarrow[\text{Step 1}]{\text{Excahnge } \beta \text{ with } \omega}$ $(\beta, \alpha, A)(...)$ [Node XX]

$\xrightarrow{\text{Exchange } \beta \text{ with symbol other than } \omega}$ $\xrightarrow{\text{Merge all cycles and Resolve the last cycle}}$ (β, α, A) [target node]

Case 2: α, β **are in some cycle** π_ℓ' **not containing** A

$(...)(.., \beta, \alpha, \sigma)$ [Node u'] $\xrightarrow[\text{Step 1}]{\text{Bring } A \text{ to first position}}$ $(A)(.., \beta, \alpha, \sigma)(...)$

$\xrightarrow[\text{Step 2}]{\text{Excahnge } A \text{ with } \sigma}$ $(\sigma, .., \beta, \alpha, A)(...)$ [Node XX]

$\xrightarrow{\text{Merge all cycles and Resolve the last cycle}}$ (β, α, A) [target node]

Remark 14 *In Algorithm B, (a) for Case 1, sub case a,* β *has to be an invariant symbol in the original source node* u*; (b) for Case 1, sub case b, the move from node* u *to* u' *belongs to the shortest routing scheme. (c) in case 2, the Step 1 may be a null move.*

Example 2: Let $u = BCDAFEG = (BCDA)(EF)$ with $\alpha = B$. (1) if $\beta = G$, algorithm B (case 1 subcase a) generates path from $u' = (GBCDA)(EF)$ to the node (GBA)

Remark 15 *For a given node* u*, algorithms A and B trace* $(n-2)^2$ *paths to the identity node* I*.*

Lemma 8 *For each choice of a neighbor* u' *of node* u *as done here, the different paths from* u' *to the nodes in* $X[S']$ *and* $Y[S']$ *(generated by algorithms A and B) are mutually vertex disjoint.*

Proof : Each node labeled XX in the executions of the routing algorithms has the property that a different symbol (for different paths) is fixed at $position_I(\alpha)$. That is true for all permutations on the paths until the target is reached. Also, the node (permutation) XX is reached in at most two moves from u' and the the starting move for all paths is different. This proves the vertex disjointness of the paths generated within the same S'. 2

Lemma 9 *The paths generated for different choices of* u' *are vertex disjoint.*

Proof : For each choice of u', all the $(n-2)$ paths are entirely contained in the substar S'. Since each S' is mutually disjoint from any other S', the claim follows. 2

Remark 16 *While routing from node* u *to the node* I*, there can be* **at most** *three moves that does not belong to the shortest routing scheme. One of these can be the move to reach* u' *from* u *and the other two are in the paths from* u' *to the designated nodes inside* S'*. Therefore the maximum length of any of these paths is* $D(u) + 6$

Lemma 10 *There are at least* $k_1 \times k_2 - k$ *paths from nodes in* Φ *to the nodes in* Ω*.*

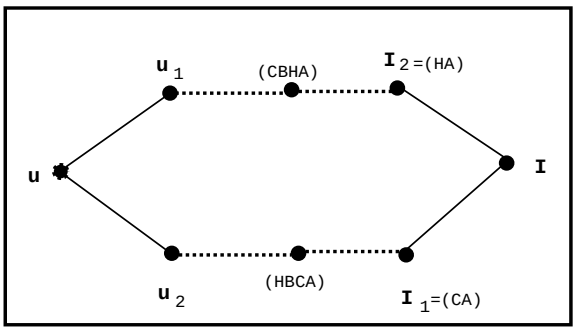

Figure 4: $k_1k_2 - k$ paths between Ψ and Ω

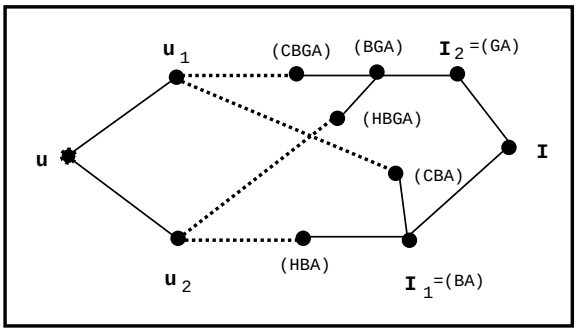

Figure 5: k_1k_2 paths between Ψ and Ω

Proof : Consider the node u' in an arbitrary substar S', where $position_{u'}(\alpha) = position_I(\beta)$. There is only one *possible* node in Ω, namely (β, A) (if indeed the node (β, A) belongs to Ω) to which a path may not be generated within S'. So there are at least $k_2 - 1$ nodes to which paths are generated. The same is true for each u' in Φ. Thus the number of paths that are generated by the algorithms A and B between the nodes u and I is $\geq k_1 \times k_2 - min(k_1, k_2)$. 2

Example 3: Consider the source node $u = BCDAFEHG = (BCDA)(EF)(GH)$ (here, $\alpha = B$). Let u has two good neighbors: $u_1 = (DA)(BC)(EF)(GH)$ (here $\beta = C$), and $u_2 = (GHBCDA)(EF)$ (here $\beta = H$). Let the identity node I also has two good neighbors: $I_1 = (CA)$ (here $\gamma = C$) and $I_2 = (HA)$ (here $\gamma = H$). Thus, $k_1 = k_2 = k = 2$. No path will be generated from node u_1 to I_1 (since $\beta = \gamma$) and similarly no path will be generated between u_2 and I_2. Thus, only $k_1k_2 - k = 2$ paths will be generated, as shown in Figure 4. Now, for the same choice of u, u_1 and u_2, let the node I has two good neighbors as $I_1 = (BA)$ (here $\gamma = B$), and $I_2 = (GA)$ (here $\gamma = G$). In this case, all k_1k_2 paths will be generated, as shown in Figure 5.

Remark 17 *The only nodes that can be common in any two of the paths mentioned in Lemma 10 are those in Φ, or in Ω, or are of the cycle structure (α, γ, A) which are all fault-free (see Remark 9).*

Theorem 2 *Under the conditions of this subsection, the node u remains connected to the node I when the number of faults does not exceed $2n - 5$ in the forbidden fault model; the surviving path(s) can have a maximum length of $D(u) + 6$.*

Proof : It is clear from Lemma 10 and the Remark 17 that in order to disconnect the nodes u and I we need *at least* $(k_1k_2 - k)$ additional faults (in addition to the $2n - 2 - |\Phi| - |\Omega| = 2n - 2 - (k_1 + k_2)$ faults already identified). So, the total number of faults needed to disconnect the nodes u and I is $2n - 2 - (k_1 + k_2) + (k_1k_2 - k)$. Now, for any arbitrary nonzero positive integers k_1 and k_2, it is true that $(k_1k_2 - k) \geq k_1 + k_2 - 2$ where $k = \min(k_1, k_2)$. Thus, total number of faults needed to disconnect the nodes u and $I \geq 2n - 4$, which is a contradiction. Hence the connectivity follows. The claim about the length of the surviving paths directly follows from Remark 16. 2

Remark 18 *For the previous theorem and the path generation algorithms we assumed $\alpha \neq \beta$. If it is the case that the neighbor of u which is reached by exchanging α with $symbol_u(position_I(\alpha))$ is good,then $\alpha = \beta$. Then we can generate paths to the nodes (γ, A) ($\gamma \neq \alpha$) inside the substar S^α as described in [RS93]. Again $n - 2$ paths are generated to all neighbors of I, but (β, A). Since any of these paths cannot be longer than $\Delta_{n-1} + 1$ the total path between u and I is of length at most $\Delta_n + 2$.*

Next, we want to prove that the node u remains connected to the node I by a path of length $\leq \Delta_n + 2$ when the number of faults does not exceed $2n - 5$.

Remark 19 *When $D(u) < \Delta_n - 3$, the claim readily follows from the previous theorem.*

Thus, we need to consider the scenarios when $D(u) \geq \Delta_n - 3$. We consider two distinct scenarios:

Scenario 1: $\Delta_n - 3 \leq D(u) \leq \Delta_n - 2$.

We investigate the case $D(u) = \Delta_n - 3$. All results are directly applicable to the case $D(u) = \Delta_n - 2$.

Remark 20 *If $D(u) = \Delta_n - 3$, then one of the following must be true for the permutation u:*

(A) There are two invariant symbols in u, say δ and σ. All other symbols form doubleton cycles, i.e., $u = (\sigma)(\delta)(\alpha, A)(...)$

(B) There is one invariant symbol in u, say σ and $|\pi_1'| \leq 5$, i.e., $u = (\sigma)(\alpha, x_1, x_2, x_3, A)(...)$

(C) There are no invariant symbols in u.

Our approach is to consider these three cases separately and to show that in each scenario there always exist at least one path between the nodes u and I of length $\leq \Delta_n + 2$, i.e., a path that requires at most two moves that do not belong to the shortest routing scheme. **Note** that in certain cases this particular path may not be generated by the algorithms A and B; we indicate alternate procedures for those situations.

Scenario 1A: There are two invariant symbols in u, i.e. $u = (\sigma)(\delta)(\alpha, A)(...)$.

Lemma 11 *If there are 2 invariant symbols in u and all other symbols form doubleton cycles, then exactly two out of $k_1k_2 - k$ paths (Lemma 10) are of length $D(u) + 6$ (3 non-optimal moves required).*

Proof : The only non-optimal moves that can be made from node u are the ones exchanging δ or σ with α. Suppose δ is exchanged with α. Then, $u \xrightarrow{\text{Exchange } \delta \text{ with } \alpha} (\sigma)(\delta, \alpha, A)(...)$ [Node u']. Routing from u' (to destination node I) is described in procedure A1 of algorithm A ($\beta = \delta$). The only path that requires two more non-optimal moves is when $\gamma = \sigma$ (see Remarks 9, 12 and 13). Same holds true for the symbol σ. 2

Corollary 1 *In order that a path is of length $D(u) + 6$, the set Ω must contain either the node (δ, A) or the node (σ, A) or both and the set Φ must contain ug_δ or ug_σ or both.*

Example 4: Consider the node $u = BACDFEGH = (BA)(EF)(GH)$ where $\alpha = B$, $\delta = D$ and $\sigma = C$. Hence, the two neighbors of node u (which would generate paths of length $D(u)+6$) are $u_1 = (DBA)(EF)(GH)$ and $u_2 = (CBA)(EF)(GH)$; also, $\Omega = \{((CA), (DA)\}$. The paths are shown in Figure 6.

Lemma 12 *If $k_1 = 1$ or $k_2 = 1$, there exists a path from u to I of length at most $\Delta_n + 1$ in presence of maximum number of faults.*

Proof : Same as Remark 6. 2

Lemma 13 *Suppose both $k_1, k_2 > 1$. If $k_1 = k_2 = 3$ or $k_2 > 3$ or $k_1 > 3$, then at least one path survives of length $\leq \Delta_n + 1$ between nodes u and I in presence of maximum number of faults.*

Proof : In the light of Lemma 11 we need only to prove that at least 3 out of $k_1k_2 - k$ paths survive. Similar to the arguments in the proof of Theorem 2, total number of faults needed to have less than 3 paths between u and I is $2n - 2 - (k_1 + k_2) + (k_1k_2 - k - 2)$. For the given values of k_1 and k_2, $(k_1k_2 - k - 2) \geq 2n - 4$, which is a contradiction. Hence, at least 3 paths survive between u and I and by Lemma 11 at least one of them has a length $\leq \Delta_n + 1$ (note $D(u) = \Delta_n - 3$). 2

Lemma 14 *If $k_2 = 2$ and $k_1 \leq 3$ or $k_1 = 2$ and $k_2 \leq 3$ and there does not exist a surviving path of length $D(u) + 6$, then there exists a symbol ξ, such that the node (ξ, A) is not in Ω, and $\xi \notin \{\alpha, \sigma, \delta\}$.*

Proof : We assume $n > 6$. At least one of the nodes (δ, A) or (σ, A) must belong to Ω (Corollary 1).
Suppose $k_2 = 2$. Thus, there is at most one node in Ω other than (δ, A) and (σ, A) ($k_2 = 2$). Since $n > 4$, there exists at least one other symbol ξ such that the node (ξ, A) does not belong to the set Ω.
Suppose $k_2 = 3$ and $k_1 = 2$. If one of (δ, A) or (σ, A) belong to Ω then there are at most two other nodes in Ω left, which establishes the existence of ξ if $n > 6$. 2

Lemma 15 *If $k_1 = k_2 = 2$, algorithm A can be slightly modified such that a good path always survives with less than 3 non-optimal moves from some node in Φ to some node in Ω in presence of maximum number of faults.*

Proof : Consider application of algorithm A in some arbitrary substar S' in which $position_{u'}(\alpha) = position_I(\beta)$. Note that no path was generated from node u' to the neighbor (β, A) of node I; $(n-3)$ vertex disjoint paths are generated from u' to all other neighbors of I. In the present scenario, some neighbors of node I are known to be faulty and our approach is to replace a path to one such faulty neighbor with a path to the node (β, A). Let ξ be a symbol such that $(\xi, A) \notin \Omega$ and $\xi \neq \delta$ and $\xi \neq \sigma$; in the present scenario, ξ belongs to some doubleton cycle, say, (ξ, ξ_1). The path from u' to (ξ, A) is generated by Algorithm A, Procedure A1, Case 1, sub case a (for simplicity assume ξ is a symbol different from the one marked as σ in the algorithm). Consider the following path:

$(\beta, \alpha, A)(\ldots)$ [Node u']
$\xrightarrow[\text{Step 1}]{\text{Exchange } \xi \text{ and } \beta}$ $(\xi, \xi_1, \beta, \alpha, A)(\ldots)$
$\xrightarrow[\text{Step 2}]{\text{Exchange A and } \xi}$ $(\xi, \xi_1, \beta, \alpha)(A)(\ldots)$ [Node XX]
$\xrightarrow[\text{except the first cycle}]{\text{Merge \& Resolve all cycles}}$ $(\xi, \xi_1, \beta, \alpha)(A)$ [Node YY]
$\xrightarrow{\text{Exchange A and } \alpha}$ $(\alpha, \xi, \xi_1, \beta, A)$
$\xrightarrow[\text{and } g_\beta]{\text{Apply } g_\xi, g_{\xi_1}}$ (β, A) [Target Node]

This path is vertex disjoint to all others. The rout to the node marked YY is the same as for the path to (β, α, ξ, A) (which is not generated). The nodes on the path from YY to the target (β, A) lie entirely in the substar for which $\beta = \alpha$ and therefore is vertex disjoint with the paths in all other substars. Finally all such newly generated paths are vertex disjoint because A is fixed at $position_I(\beta)$ after the node YY and for different paths β's are different.

Thus, with the said modifications in algorithm A, we generate $k_1 \times k_2$ paths (the term $min(k_1, k_2)$ is gone, since paths are generated to **every** permutation of Ω from each node in Φ). For $k_1, k_2 = 2$ it is true that $k_1 \times k_2 > k_1 + k_2$, and total number of faults needed to disconnect all the paths is $k_1 k_2 + (2n-2) - (k_1 + k_2) \geq 2n - 2$. Thus in presence of at most $2n-5$ faults, at least three paths survive and at least one of them needs less than 3 non optimal moves (Lemma 11). 2

Example 5: Consider the previous example where $u = BACDFEGH = (BA)(EF)(GH)$, $\Phi = \{(DBA)(EF)(GH), (CBA)(EF)(GH)\}$, and $\Omega = \{(CA), (DA)\}$. Here, $k_1 = k_2 = 2$ and Algorithm A will generate only two paths from Φ to Ω which are shown in Figure 6. Now, by Lemma 14 there exists a symbol $\xi = G$ and the alternate procedure given in Lemma 15 generates two more vertex disjoint paths which are shown in Figure 7.

Scenario 1(B): There is one invariant symbol in u, i.e. $u = (\delta)(\alpha, x_1, x_2, x_3, A)(\ldots)$**.**

Lemma 16 *In scenario 1(B), there can be at most 3 possible paths out of possible $k_1 k_2 - k$ paths (Lemma 10) are of length $D(u) + 6$ (3 non-optimal moves required).*

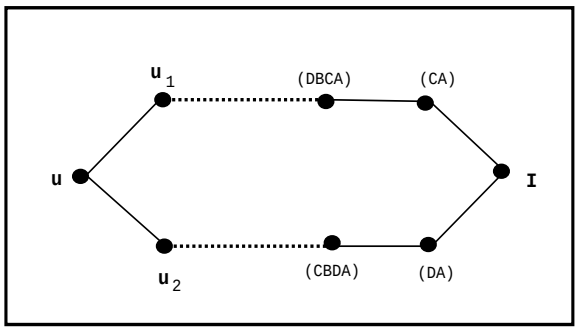

Figure 6: Example for Lemma 11

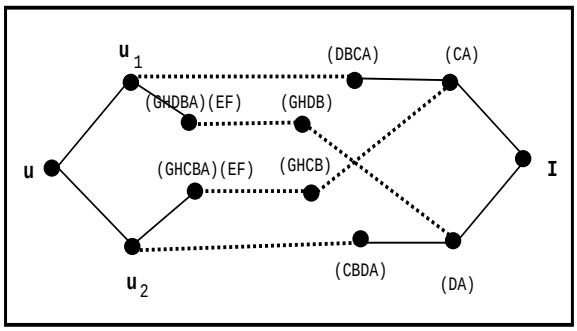

Figure 7: Example for Lemma 15

Proof : For any path from node u to node I requiring 3 non optimal moves, it must be true that the node u' is reached from u by exchanging α with a symbol in the set $\{\delta, x_2, x_3, A\}$. The corresponding u' is routed to node I using Procedure A1 of Algorithm A (if δ is exchanged with α as a first move) or Procedure A3 in Algorithm A (if any of the other listed symbols is exchanged with α). Further examination of the the paths reveal that the three paths that require 3 non-optimal moves are:

$$u \longrightarrow u'(ug_{\delta}) \xrightarrow[\text{Path 1}]{\beta = \delta \text{ and } \gamma = x_2} (\delta, \alpha, x_2, A) \longrightarrow (x_2, A) \longrightarrow \text{node } I$$

$$u \longrightarrow u'(ug_{x_2}) \xrightarrow[\text{Path 2}]{\beta = x_1 \text{ and } \gamma = \delta} (x_1, \alpha, \delta, A) \longrightarrow (\delta, A) \longrightarrow \text{node } I$$

$$u \longrightarrow u'(ug_{x_3}) \xrightarrow[\text{Path 3}]{\beta = x_2 \text{ and } \gamma = \delta} (x_2, \alpha, \delta, A) \longrightarrow (\delta, A) \longrightarrow \text{node } I$$

2

Remark 21 *Path 1 of Lemma 16 is generated by Procedure A1, Case 3, Sub case 1; Path 2 and Path 3 of Lemma 16 is generated by Procedure A3, Case 3.*

Corollary 2 *In scenario 1(B), no paths of length $D(u)+6$ from node u to node I can go via either the neighbor (x_3, A) or the neighbor (α, A) of I. In order that any of the paths of Lemma 16 is generated, Ω must have a node other than (x_3, A) or (α, A).*

Lemma 17 *If $k_1 = 1$ or $k_2 = 1$, there exists a path from u to I of length at most $\Delta + 1$ in presence of maximum number of faults.*

Proof : Same as the proof of Lemma 12. 2

Lemma 18 *If $k \geq 3$ and $\max(k_1, k_2) \geq 4$ or $k \geq 2$ and $\max(k_1, k_2) > 4$, at least one path generated by the proposed algorithm survives in presence of maximum number of faults that needs less than 3 non-optimal moves.*

Proof : For these values of k_1 and k_2 it is true that $k_1 \times k_2 - k > k_1 + k_2$. Thus, the total number of faults needed to disconnect nodes u and I is $k_1 k_2 - k + (2n-2) + (k_1 + k_2) > 2n - 2$. Hence, at least 4 of the generated paths are not disconnected. The claim follows from Lemma 16. 2

Lemma 19 *If the values of k_1 and k_2 do not satisfy the conditions of the previous lemma, but Ω includes* **both** *the nodes (x_3, A), and (α, A), then a good path always exists that requires less than 3 non-optimal moves.*

Proof : The proof is based on Corollary 2. If $k_2 = 2$, no path requiring 3 non-optimal moves is generated.

If $k_2 = 3$, then k_1 must be 2; at most two paths requiring 3 non-optimal moves can be generated. Then, $Number\_of\_paths(u, I) = k_1 \times k_2 = 6$, since for each generated path of length $D(u) + 6$ there are **two** generated paths of length $D(u) + 4$ to the nodes (x_3, A) and (α, A). Or, $k_1 + k_2 < Number\_of\_paths(u, I)$ and hence, at least 4 of the generated paths cannot be disconnected. Only 2 of the surviving paths require 3 non-optimal moves and hence the claim follows.

If $k_1 = 3$ and $k_2 = 3$, the proof is identical to that of Lemma 13 since there are at most two paths requiring 3 non-optimal moves.

If $k_2 = 4$ and $k_1 = 2$, again there are at most two paths requiring 3 non-optimal moves, since there are 2 nodes in Φ and each possible path requiring 3 non-optimal moves must start with different nodes of Φ. Then use the same arguments as in the proof of Lemma 13. 2

Remark 22 *We need to consider the cases where Ω does not contain either of the nodes (x_3, A) or (α, A) or both. Our approach is to modify parts of algorithm A to exploit the fact that either (x_3, A) or (α, A) or both are faulty.*

Lemma 20 *If $(\alpha, A) \notin \Omega$, then 2 of the three paths of Lemma 16, requiring 3 non-optimal moves can be replaced by paths, requiring just 2 non-optimal moves. Similarly if $(x_3, A) \notin \Omega$, then the third path requiring 3 non-optimal moves can also be replaced by one with just two.*

Proof : For each path that requires 3 non-optimal moves we show alternative paths that require just 2 such moves. These alternative paths start by exchanging at the first step the symbol A or the symbol x_3.

Alternative to Path 1:

$ug_\delta \xrightarrow[\text{Step 1}]{\text{Exchange } \delta \text{ and A}} (A)(\delta, \alpha, x_1, x_2, x_3)(...)$

$\xrightarrow[\text{Step 2}]{\text{Exchange A and } x_3} (x_3, \delta, \alpha, x_1, x_2, A)(...)$ [Node XX]

$\xrightarrow[\text{and resolve}]{\text{Merge all cycles}} (\delta, \alpha, x_1, x_2, A)$ [Node YY]

$\xrightarrow[\text{Final Moves}]{\text{Apply } g_\alpha, g_{x_1}, g_{x_2}} (x_2, A)$

Alternative to Path 2:

$ug_{x_3} = (\delta)(x_2, \alpha, x_1)(x_3, A)(...) \xrightarrow[\text{Step 1}]{\text{Exchange } A \text{ and } x_3} (\delta)(x_2, \alpha, x_1)(x_3)(A)(...)$

$\xrightarrow[\text{Step 2}]{\text{Exchange } A \text{ and } \delta} (\delta, A)(x_2, \alpha, x_1)(x_3)(...)$ [Node XX]

$\xrightarrow[\text{Step 3}]{\text{Exchange } x_1 \text{ and } \delta} (x_1, x_2, \alpha, \delta, A)(...)$

$\xrightarrow{\text{Merge \& Resolve}} (x_2, \alpha, \delta, A)$ [Target Node]

The first move of these two paths is the same as the first move of the paths to (α, A), but since that node is assumed to be faulty and the paths to (α, A) are not generated, we can use the same first move to generate other paths. The node marked with XX has the property that the symbol A is fixed at $position_I(\gamma)$ (γ is x_2 and δ for the two paths) and therefore from that point these paths are disjoint with all the others. The first alternative path reaches a node labeled YY which is different from the target node in the original path. (x_2, A) is reached with 3 moves from YY going through one permutation in which α is at first position and one in which α is at its correct position. Both of these permutations cannot be found on any other path.

Alternative to Path 3:

$ug_{x_2} = (\delta)(x_1, \alpha)(x_2, x_3, A)(...) \xrightarrow[\text{Step 1}]{\text{Apply } g_{x_3}, g_A} (A)(\delta)(x_1, \alpha)(...)$

$\xrightarrow[\text{Step 2}]{\text{Exchange } A \text{ and } \delta} (\delta, A)(x_1, \alpha)(...)$[Node XX]

$\xrightarrow{\text{Apply } gx_1}$ $(x_1, \alpha, \delta, A)(...)$

$\xrightarrow[\text{all cycles}]{\text{Merge \& Resolve}}$ (x_1, α, δ, A)[Target Node]

To generate that path we have borrowed the first move of the path to (β, α, x_3, A). If (x_3, A) is faulty we can reuse that move to generate that path of. The generated path is vertex disjoint with all other paths that could be generated inside that substar (assuming a path to (β, α, x_3, A) is not generated). To prove that it is enough to observe that the first move is different and the third move reaches a permutation in which A is fixed at $position_I(\delta)$, which is not the case for any other path. Further, the symbol A stays at that position throughout the path. 2

Lemma 21 *If $(\alpha, A) \notin \Omega$, then there exists a good path from u to I that requires less than 3 non-optimal moves in the presence of maximum number of faults.*

Proof : By the previous lemma, if $(\alpha, A) \notin \Omega$, then at most one path (Path 1 of Lemma 16) is generated that requires 3 non-optimal moves. The only scenario, when only this path survives, is $k_1 = k_2 = 2$ and $\Phi = \{ug_{x_2}, ug_\delta\}$ and $\Omega = \{(\delta, A), (x_1, A)\}$ (in all other cases more than one path will survive). But since $(x_3, A) \notin \Omega$, we can replace the path with 3 non optimal moves with one with two non optimal moves using similar techniques as used for the alternate to Path 3 in the previous lemma. 2

Lemma 22 *If $(A, \alpha) \in \Omega$ and $(x_3, A) \notin \Omega$, then there exists a good path from u to I that requires less than 3 non-optimal moves in the presence of maximum number of faults.*

Proof : Only one of the 3 paths (Lemma 16), requiring 3 non-optimal moves can be replaced with a path that requires just 2. Also, a path is generated from **each** node in Φ to (α, A) (more precisely to (β, α, A) from which (α, A) is reached with one move).

If $k_1 = k_2 = 3$ the result from lemma 13 applies since there are just 2 possible paths of length $D(u) + 6$.

If $k_1 = k_2 = 2$ there are at least 3 path generated and at most one of these is of length $D(u) + 6$. There is one other permutation other than (α, A) in Ω. The only permutation that is the final node of 2 paths of length $D(u) + 6$ is (δ, A). But one of these paths is replaced by a path of length $D(u) + 4$, since (x_3, A) is known to be faulty. In that case at least 2 of the 3 generated paths cannot be disconnected and therefore there exist one good path of length $D(u) + 4$. If $k_1 = 3, k_2 = 2$ the same arguments apply.

If $k_1 = 2, k_2 = 3$ then there are 2 possible paths of length $D(u) + 6$. The worst case is when only these two paths are not disconnected. That can happen when $Number\_of\_paths(u, I) = 4 = k_1 \times k_2 - k_1$ and two paths requiring 3 non-optimal moves are among the four generated paths. For that specific case it should be true that $\Phi = \{ug_\delta, ug_{x_3}\}$ and $\Omega = \{(x_2, A)(\delta, A)(\alpha, A)\}$. In that case there are four paths generated only, namely: $ug_\delta \rightarrow (x_2, A)$,$ug_{x_3} \rightarrow (\delta, A)$, $ug_\delta \rightarrow (\alpha, A)$,$ug_{x_3} \rightarrow (\alpha, A)$. The first two are the two non-optimal paths. So there could be up to 2 faults outside the nodes in the set Φ and Ω and it is possible that the two remaining paths (of length $D(u) + 4$) are disconnected. It is enough to show one more path, which is vertex disjoint with the two paths of length $D(u) + 4$. We show the path $ug_\delta \rightarrow (\delta, A)$. Originally this path is not generated since $position_{u'}(\alpha) = position_I(\delta)$ and there is no path to (δ, A) inside that substar. However, for this specific case consider the path: $u' = (\delta, \alpha, x_1, x_2, x_3, A)(...)$ $\xrightarrow[\text{Step 1}]{\text{Apply } g_{x1}}$ $(\delta, \alpha)(x_1, x_2, x_3, A)$ [Node XX]

$\xrightarrow[\text{except } (\delta, \alpha)]{\text{Resolve all cycles}}$ $(A)(\delta, \alpha)$ [Node YY]

$\xrightarrow{\text{Exchange } A \text{ and } \alpha}$ (α, δ, A) $\xrightarrow{\text{Apply } g_\delta}$ (δ, A) [Target Node]

Note that until the node YY is reached the path is the same as the path to (x_3, A) (Procedure A1, Case 3, Sub case b). Since $(x_3, A) \notin \Omega$ (see Lemma 21) we can reuse that path. The target (δ, A) is reached from YY in two moves thru the node (α, δ, A),

which is not on any other path. Having generated that path now there are 5 paths, 3 of which are of length $D(u)+4$ and it is not possible to disconnect them with the 2 remaining faults. 2

Scenario 1(C): There is no invariant symbol in u, i.e. $u == (\alpha, x_1, x_2, x_3, x_3, x_4, A)(\ldots)$.

In this case, if the move from node u to u' belong to shortest routing, no path generated by the algorithms will need 3 non optimal moves. If the move from node u to u' is obtained by exchanging α with $x_i \in \pi_1^{'}$, the move does not belong to shortest routing, but the node u' will be of the form: $u' = (\ldots\beta, \alpha)(\ldots A)(\ldots)$. Path generation from u' to $n-2$ nodes inside S' is described in Procedure A3 of Algorithm A; None of them requires more than one non optimal move and consequently any surviving path in presence of maximum number of faults between nodes u and I has a length $\leq D(u)+4$.

Scenario 2: $\Delta_n - 1 \leq D(u) \leq \Delta_n$

In this scenario, the node (permutation) u has no invariant symbol. We assume $D(u) = \Delta_n - 1$ for brevity; the analysis and the results are directly applicable when $D(u) = \Delta_n$. We need to show that there exists at least one good path between u and I that requires at most one non-optimal move.

Remark 23 *If $D(u) = \Delta_n - 1$ and A is not an invariant, then u has no invariant symbol, and the maximum cycle length in u is 4.*

Lemma 23 *If $|\pi_1^{'}| = 2$, all paths generated by the proposed scheme require at most one non-optimal move, except one (which may need more than one non optimal move).*

Proof : Let $u = (\alpha, A)(\ldots)$. The maximum cycle length is 4 and the first move is always optimal (merging 2 cycles). Suppose there is a cycle π_ℓ with 4 symbols, $\pi_\ell = (x_1, x_2, x_3, x_4)$. If a symbol (say x_1) is exchanged with α as the first move, we get to the node $u \xrightarrow[\beta = x_4]{g_{x_1}} (x_1, x_2, x_3, \beta, \alpha, A)(\ldots)$ [Node u']. The paths, originating from u' are described in Procedure A2. The only path that requires more than one non-optimal move is the one, for which $\gamma = x_3$. If there is no cycle of length 4, then no path is generated with more than one non-optimal move. 2

Example 6: Consider the node $u = (BA)(CDEF)(GH)(IJ)(KL)$. Here $\pi_\ell = (CDEF)$ and hence $u' = (CDEFBA)(GH)(IJ)(K$ and the path from u' to the node (EA) (neighbor of I) makes two non-optimal moves (note $x_3 = E$).

Lemma 24 *If $|\pi_1^{'}| = 3$, then there exist $k_1 \times k_2 - k$ paths from u to I each of which requires at most one non-optimal move (some of these paths need be generated by the alternate procedure described in the proof).*

Proof : Let $u = (\alpha, \delta, A)(\ldots)$. In that case there is at most one cycle of length more than two, say $\pi_l = (x_1, x_2, x_3)$ If $u' \neq ug_\delta$, then the first move is optimal. Suppose a symbol from π_l is exchanged: $u \xrightarrow[\beta = x_3]{g_{x_1}} (x_1, x_2, \beta, \alpha, \delta, A)(..)$ [Node u']. There are no paths that require more than 1 non-optimal move, originating from u' (see Procedure A2).

Now suppose $u' = ug_\delta$: $u \xrightarrow[\beta = \delta]{g_{x_1}} (\alpha, \beta)(A)(\ldots)$ [Node u']

We use a modified routing scheme for this u'. First consider a path from u' to a node $(\beta, \alpha, \gamma, A)$. Since $\gamma \neq \alpha$, it must be true that γ belongs to some cycle of length 2, say $\pi_l = (\gamma, \delta)$

$u' = (A)(\alpha, \beta)(\gamma, \sigma)(\ldots) \xrightarrow[\text{Step 1}]{\text{Exchange } A \text{ and } \sigma} (\sigma, \gamma, A)(\alpha, \beta)(\ldots)$ [Node XX]

$\xrightarrow[\text{Step 2}]{\text{Apply } g_\gamma \text{ and } g_\beta} (\beta, \alpha, \gamma, A)(\ldots)$

$\xrightarrow[\text{and Resolve}]{\text{Merge All Cycles}} (\beta, \alpha, \gamma, A)$ [Target Node] 2

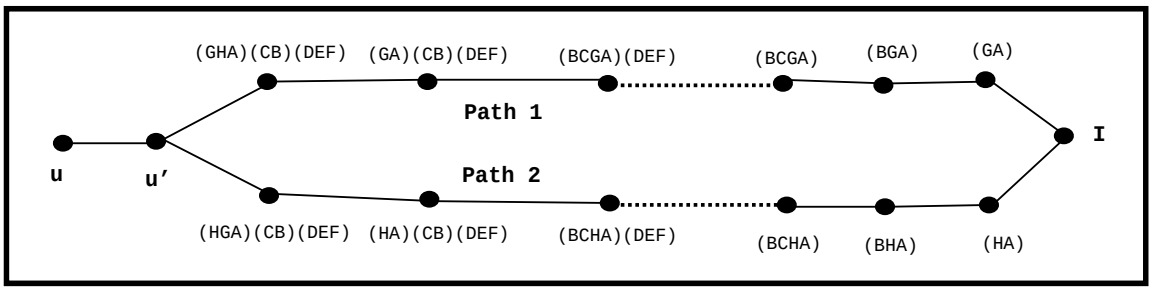

Figure 8: Example for Lemma 24

Example 7: Consider the node $u = (CBA)(DEF)(GH)$ and $u' = (CB)(DEF)(GH)$. Here, $\alpha = C,\ \beta = \delta = B$. Figure 8 shows two paths from node u' to I generated by the alternate procedure suggested in the above proof (note that $\gamma = H$ for Path 1 and $\gamma = G$ for Path 2.

Lemma 25 *Let $\pi'_1 = (\alpha, x_1, x_2, A)$ (since $|\pi'_1| = 4$, all other cycles of u are of length 2). Then, if $ug_{x_2} \notin \Phi$, the generated paths from u to I require at most 1 non-optimal move.*

Proof : There are two possible first moves from node u: (1) α is exchanged with some symbol δ from a doubleton cycle, say (δ, σ). We reach the node $u' = (\delta, \beta, \alpha, x_1, x_2, A)(..)$. Here $\beta = \sigma$ and all paths from u' (generated by Procedure A2) need at most one non-optimal move; (2) α is exchanged with A. We reach the node $u' = (A)(\beta, \alpha, x_1)(..)$. Here $\beta = x_2$ and using the alternative routing, described in the previous Lemma 24, the paths, generated from u' do not require non-optimal moves. 2

Lemma 26 *Let $\pi'_1 = (\alpha, x_1, x_2, A)$ (since $|\pi'_1| = 4$, all other cycles of u are of length 2). Then, if $u' = ug_{x_2} \in \Phi$, then there exist n-2 paths of optimal length from u' to all target nodes. This is true if $n > 8$.*

Proof : We have $u' = (\beta, \alpha)(x_2, A)(..)$ and in this case $D(u') \leq \Delta_n$ by Lemma 2 and Lemma 1. If $n > 6$, there exist $(n-1)$ vertex disjoint shortest paths [use algorithm D of [RS93]] from u' to I. Consider the substar (of dimension $n-1$) S' with β fixed at $position_I(\alpha)$; next consider consider the substar (of dimension $n'' = n-2$) S'', $S'' \subset S'$, with α fixed at $position_I(\beta)$. Obviously, $u' \in S''$; there are $n'' - 1 = n - 3$ vertex disjoint paths from u' to the node $I'' = (\beta, \alpha)$ within S'', assuming $n > 8$. For all symbols $\gamma_i \notin \{\beta, \alpha, x_2\}$ there is a (distinct) path with a node $(\beta, \alpha)(\gamma_i, A)$ (just prior to the node I''). Applying g_β from that node we reach the target $(\beta, \alpha, \gamma_i, A)$. If $\gamma_i = x_2$, then use the following path: $u' = (\beta, \alpha)(x_2, A)(..) \xrightarrow{\text{Exchange } \beta \text{ and } x_2} (\beta, \alpha, x_2, A)(..) \xrightarrow[\text{and resolve}]{\text{Merge all cycles}} (\beta, \alpha, x_2, A)$ [Target Node]. All nodes in this path are in S', but not in S'' since β is exchanged in the first move; hence it is vertex disjoint with all other paths. Finally we show the path to (β, α, A). Among the $n-3$ paths generated in S'', there is one that contains $(\beta, \alpha)(x_2, A)$. Applying consecutively g_A and g_β we reach the target. 2

Theorem 3 *There exist a path from u to I that requires at most 1 non-optimal move for the case $D(u) \geq \Delta_n - 1$ and $|\pi_1| > 1$.*

Proof : For the most part the proof is complete by the preceding lemmas. In particular it was shown that if $\pi'_1 = 3$ or $\pi'_1 = 4$ we can generate $k_1 \times k_2 - min(k_1, k_2)$ paths that require at most one non-optimal move. Case (1): $k_1 = 1$ or $k_2 = 1$: Use the results from Remark 6; Case (2): $k_1 = k_2 = 2$: Here, $(n-6)$ faults have been identified among the neighbors of u and I. If more than 2 paths are generated between Φ and Ω, we are done since only one path may need more than one non-optimal move. The worst case is that two paths are generated between Φ and Ω, and the path with two non-optimal moves survives. In that case it should be true that $ug_{x_1} \in \Phi$ and $(x_3, A) \in \Omega$, since the path requiring 2 non-optimal moves is between these nodes. Further,

it should be true that $(x_4, A) \in \Omega$ because there is no path generated between ug_{x_3} and (x_4, A). Finally $ug_{x_4} \in \Phi$, since there is no path between ug_{x_4} and (x_3, A). Hence, $\Phi = \{ug_{x_1}, ug_{x_4}\}$ and $\Omega = \{(x_3, A), (x_4, A)\}$. We add another path from ug_{x_4} to (x_3, A) in the same way we did in the previous case ($k_1 > 1, k_2 = 1$). That path is vertex disjoint with the path from ug_{x_4} to (x_4, A) in the same substar (see Procedure A2, Case 3). Now there are total of 3 paths generated and at most one of them can be disconnected and hence there exists a path that requires just one non-optimal move; Case (3): $k_1 > 2$ or $k_2 > 2$: The number of surviving paths generated by the algorithms is larger than one and since there can be only one path requiring 2 non-optimal moves, there always exists a path with at most one non-optimal move. 2

3.2.2 Part 2

In this part, for a given node u (i.e., given symbol α), the node (α, A) is faulty and for every other symbol γ ($\neq \alpha$ or A) either the node (γ, A) is faulty or the node (α, γ, A) is faulty or both are faulty. This is a relatively specific situation; at least $(n-1)$ faults are already located. By the assumption of forbidden faulty sets, the node I has at least one non faulty neighbor, say I'. Our approach is to show the existence of $(n-3)$ vertex-disjoint paths from the source node u to I'; thus, in the presence of at most $(2n-5)$ faults, the nodes u and I will remain connected.

Let $I' = (\sigma, A)$ be the good neighbor of I. Then, the nodes (α, σ, A) and (α, A) are faulty and for every other symbol γ_i ($\neq \alpha$ or A) either the node (γ_i, A) or the node (α, γ_i, A) is faulty; thus at least $(n-1)$ faulty nodes are identified. Consider the $(n-1)$ vertex disjoint paths from the node u to the node I' (each of length at most $\Delta_n + 1$ [RS93]); each of these paths reach I' via a different neighbor of I'; disregard the two paths that reach I' via the node i and the node (α, σ, A); none of the rest of the $(n-3)$ vertex disjoint paths can go through any of the already identified faulty nodes, since there is no cycle in star graphs of length less than 6. Under the assumption of forbidden faulty sets, the maximum number of faults is $2n-5$ ($< (n-1)+(n-3)$) and hence at least one path (of length at most $\Delta_n + 1$) survives between the node u and I' and thus there survives a path from u to I of length at most $\Delta_n + 2$.
All these results and discussions in this subsection 3.2 lead to the following theorem.

Theorem 4 *In presence of up to* $2n-5$ *faults (subject to the restriction of the forbidden faulty sets) the distance of an arbitrary node* u *in* S_n *with* $|\pi_1| > 1$ *is* $\leq \Delta_n + 2$.

Theorem 5 $\kappa^{\mathcal{R}}(S_n) = 2n-4$ *and* $\Delta_n^{\mathcal{R}} = \Delta_n + 2$.

Proof : The proof readily follows from Theorems 1 and 4. 2

4 Conclusion

We have established the restricted vertex connectivity and the fault diameter of the star graphs under the condition of forbidden faulty sets. It has been shown that the fault diameter of star graphs is increased only by 2 over its fault free diameter just like the n-cubes under similar conditions. Thus, the results add to the attractiveness of the star graphs as compared to n-cubes. Finally, it should be noted that the concept of forbidden faulty sets can be generalized where at most p ($p \geq 1$) neighbors of any node can fail; details can be found in [LhNP94] where the generalized concept has been applied to the n-cubes. It'd be interesting to investigate star graphs in this generalized setting. The derivation of the fualt-diameter may be generalized to account for any number of faults in the star graph. Let Y be the set of faulty vertices in S_n such that $|Y| = \beta(n-3)+1$ and the distribution of faults is such that the star remains connected. Consider two nodes u and I such that $D(u) = \Delta_n - \beta$ ($1 \leq \beta \leq \Delta_n - 2$). Now choose a fault distribution which leaves a unique path according to the shortest routing scheme between u and another node u' such that $D(u, u') = \beta$ and $D(u') = \Delta_n$ (Imagine one of the paths going from u' to I having visited u after β hops). What we are doing here is force the only possible path from u to I through u' and thus creating a fault-diamter of $\Delta_n + \beta$. It takes $\beta(n-3)+1$ faults to force this path as mentioned and it can be stated that: The fault-diamter of S_n with $\beta(n-3)+1$ faulty nodes

can be at most $\Delta_n + \beta$ for $n > 6$. We conjecture that this value is indeed the exact value of the fault-diamter. Note that the special cases of $\beta = 2$ and $\beta = 1$ have been treated in this paper and previous work respectively. One implication of this conjecture is that in a connected S_n, the diamter can at most be $2\Delta_n - 2$, if the number of faults is kept below: $(\Delta_n - 2)(n-3) + 1$.

References

[AK87] S. B. Akers and B. Krishnamurthy. The star graph: an attractive alternative to n-cube. In *Proceedings of International Conference on Parallel Processing (ICPP-87)*, pages 393–400, St. Charles, Illinois, August 1987.

[AK89] S. B. Akers and B. Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers*, 38(4):555–566, April 1989.

[DT94] K. Day and A. Tripathi. A comparative study of topological properties of hypercubes and star graphs. *IEEE Transactions on Parallel and Distributed Systems*, 5(1):31–38, January 1994.

[Esf89] A. H. Esfahanian. Generalized measures of fault tolerance with applications to n-cube networks. *IEEE Transactions on Computers*, 38(11):1586–1591, November 1989.

[FA91] P. Fragopoulou and S. G. Akl. Parallel algorithm for computing Fourier transforms on the star graph. In *Proceedings of the International Conference on Parallel Processing*, volume III, pages 100–106, St. Charles, Illinois, 1991.

[Har72] F. Harary. *Graph Theory*. Addison-Wesley, Reading, MA, 1972.

[Lat93a] S. Latifi. Combinatorial analysis of fault diameter of the n-cube. *IEEE Transactions on Computers*, 42(1):27–33, January 1993.

[Lat93b] S. Latifi. On fault diameter of star graphs. *Information Processing Letters*, 46:143–150, June 1993.

[LhNP94] S. Latifi, M. hedge, and M. Naraghi-Pour. Conditional connectivity measures for large multiprocessor systems. *IEEE Transactions on Computers*, 43(2):218–222, February 1994.

[MS90] A. Menn and A. K. Somani. An efficient sorting algorithm for the star graph interconnection network. In *Proceedings of the International Conference on Parallel Processing*, volume III, pages 1–8, St. Charles, Illinois, 1990.

[MS92] V. E. Mendia and D. Sarkar. Optimal broadcasting on the star graph. *IEEE Transactions on Parallel and Distributed Systems*, 3(4):389–396, July 1992.

[NSK90] M. Nigam, S. Sahni, and B. Krishnamurthy. Embedding Hamiltonians and hypercubes in star interconnection graphs. In *Proceedings of the International Conference on Parallel Processing*, pages 340–343, August 1990.

[QAM93] K. Qiu, S. G. Akl, and H. Meijer. On some properties and algorithms for the star and pancake interconnection networks. *Journal of Parallel and Distributed Computing*, 1993.

[QMA91] K. Qiu, H. Meijer, and S. G. Akl. Decomposing a star graph into disjoint cycles. *Information Processing Letters*, 39(3):125–129, 1991.

[QMA92] K. Qiu, H. Meijer, and S. G. Akl. On the cycle structure of star graphs. Technical Report 92-341, Department of Computer Science, Queen's University, Ontario, Canada, November 1992.

[RS93] Y. Rouskov and P. K. Srimani. Fault diameter of star graphs. *Information Processing Letters*, 48(5), December 1993.

[Sei85] C. L. Seitz. The cosmic cube. *Communications ACM*, 28(1):22–33, January 1985.

[SS88] Y. Saad and M. H. Shultz. Topological properties of hypercubes. *IEEE Transactions on Computers*, 37(7):867–872, July 1988.

Procedure A1

Case 1: $\pi_1' = (\beta, \alpha, A)$ [i.e., π_1' does not have any other symbol]

Subcase a: $\gamma \neq \sigma$

$(\beta, \alpha, A)(...)$ [Node u'] $\xrightarrow[\text{Step 1}]{\text{Excahnge } \gamma \text{ with } \beta}$ $(\gamma, ..., \beta, \alpha, A)(...)$

$\xrightarrow[\text{Step 2}]{\text{Excahnge } \gamma \text{ with } A}$ $(A)(\gamma, ..., \beta, \alpha)(...)$ [Node XX]

$\xrightarrow[\text{Step 3}]{\text{Merge \& Resolve cycles other than } (\gamma, ..., \beta, \alpha)}$ $(A)(\gamma, ..., \beta, \alpha)$

$\xrightarrow[\text{Step 4}]{\text{Exchange the symbol following } \gamma \text{ with } A}$ $(..., \beta, \alpha, \gamma, A)$

$\xrightarrow[\text{Step 5}]{\text{Resolve if necessary}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Subcase b: $\gamma = \sigma$

$(\beta, \alpha, A)(...)$ [Node u'] $\xrightarrow[\text{Step 1}]{\text{Exchange } \beta \text{ with } A}$ $(A)(\beta, \alpha)(...)$ [Node XX]

$\xrightarrow[\text{Step 2}]{\text{Exchange } A \text{ with } symbol_{XX}(position_I(\gamma))}$ $(.., \gamma, A)(\beta, \alpha)(...)$

$\xrightarrow[\text{Step 3}]{\text{Execute cycle } (.., \gamma, A) \text{ until } (\gamma, A)}$ $(\gamma, A)(\beta, \alpha)(...)$

$\xrightarrow[\text{Step 4}]{\text{Exchange } \gamma \text{ with } \beta}$ $(\beta, \alpha, \gamma, A)(...)$ $\xrightarrow[\text{and Resolve}]{\text{Merge other cycles}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Case 2: $\pi_1' = (\beta, \alpha, \delta, A)$ [i.e., π_1' contains exactly one other symbol]

Subcase a: $\gamma = \delta$

$(\beta, \alpha, \gamma, A)(...)$ [Node u' = Node XX]

$\xrightarrow[\text{Resolve the last cycle}]{\text{Merge cycles first by exchanging } \beta \text{ with } \sigma}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Subcase b: $\gamma \neq \delta$

If $\gamma = \sigma$ **then**

$(\beta, \alpha, \delta, A)(...)$ [Node u']

$\xrightarrow[\text{Step 1}]{\text{Exchange } \beta \text{ with } \delta}$ $(\delta, A)(\beta, \alpha)(...)$ [Node XX]

$\xrightarrow[\text{Step 2}]{\text{Exchange } \delta \text{ with } A}$ $(A)(\beta, \alpha)(...)$ $\xrightarrow[\text{Step 3}]{\text{Exchange } A \text{ with } symbol_{XX}(position_I(\gamma))}$ $(.., \gamma, A)(\beta, \alpha)(...)$

$\xrightarrow[\text{Resolve the cycle}]{\text{Merge cycles except } (\beta, \alpha)}$ $(\gamma, A)(\beta, \alpha)$ $\xrightarrow[\text{with } \beta]{\text{Exchange } \gamma}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Else [$\gamma \neq \sigma$]

$(\beta, \alpha, \delta, A)(...)$ [Node u']

$\xrightarrow[\text{Step 1}]{\text{Exchange } \beta \text{ with } \gamma}$ $(\gamma, .., \beta, \alpha, \delta, A)(...)$ $\xrightarrow[\text{Step 2}]{\text{Exchange } \gamma \text{ with } \delta}$ $(\delta, A)(\gamma, .., \beta, \alpha)(...)$ [Node XX]

$\xrightarrow[\text{Except the cycle of } \gamma]{\text{Merge cycles and resolve}}$ $(A)(\gamma, .., \beta, \alpha)$ $\xrightarrow{\text{Exchange } A \text{ with } symbol_{XX}(position_I(\gamma))}$ $(.., \beta, \alpha, \gamma, A)$

$\xrightarrow{\text{Resolve the cycle}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Case 3: $\pi'_1 = (\beta, \alpha, \delta, \omega, .., \sigma, A)$ [i.e., π'_1 contains σ and at least two other symbols in addition to β, α and A].

Subcase a: $\gamma \in \pi'_1$ and $\gamma \neq \delta$ and $\gamma \neq \sigma$.

$(\beta, \alpha, \delta, \omega, .., \sigma, A)(...)$ [Node u']

$\xrightarrow[\text{Step 1}]{\text{Exchange } \beta \text{ with } \gamma \text{ and then exchange } \gamma \text{ with } \delta}$ $(\delta, .., \beta, \alpha, \gamma, .., A)(...)$ [Node XX]

$\xrightarrow[\text{Step 2}]{\text{Exchange } \delta \text{ with } symbol_{XX}(position_I(\gamma))}$ $(.., A)(...)(\delta, .., \beta, \alpha, \gamma)$

$\xrightarrow[\text{except the last cycle}]{\text{Merge cycle \& Resolve}}$ $(A)((\delta, .., \beta, \alpha, \gamma)$ $\xrightarrow[\text{Step 4}]{\text{Exchange } A \text{ with } \delta}$ $(\delta, .., \beta, \alpha, \gamma, A)$

$\xrightarrow[\text{Step 5}]{\text{Resolve the cycle}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Subcase b: $\gamma = \sigma$.

$(\beta, \alpha, \delta, \omega, .., \gamma, A)(...)$ [Node u']

$\xrightarrow[\text{Step 1}]{\text{Exchange } \beta \text{ with } \delta}$ $(\delta, .., \gamma, A)(...)(\beta, \alpha)$ [Node XX]

$\xrightarrow[\text{Step 2}]{\text{Merge cycles except } (\beta, \alpha)}$ $(\gamma, A)(\beta, \alpha)$ $\xrightarrow[\text{Step 3}]{\text{Exchange } \gamma \text{ with } \beta}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Subcase c: $\gamma = \delta$.

$(\beta, \alpha, \gamma, \omega, .., \sigma, A)(...)$ [Node u'**,** also Node XX]

$\xrightarrow[\text{Step 1}]{\text{Exchange } \beta \text{ with } \sigma}$ $(\sigma, A)(...)(\beta, \alpha, \gamma, \omega, ..)$ $\xrightarrow[\text{Step 2}]{\text{Exchange } \sigma \text{ with } A}$ $(A)(...)(\beta, \alpha, \gamma, \omega, ..)$

$\xrightarrow[\text{Step 3}]{\text{Excahnge } A \text{ with } \omega}$ $(\beta, \alpha, \gamma, A, ..)(...)$ $\xrightarrow[\text{\& Resolve}]{\text{Merge cycles}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Subcase d: $\gamma \notin \pi'_1$.

$(\beta, \alpha, \delta, \omega, .., \sigma, A)(...)$ [Node u']

$\xrightarrow[\text{Step 1}]{\text{Exchange } \beta \text{ with } \gamma}$ $(\gamma, .., \beta, \alpha, \delta, \omega, .., \sigma, A)(...)$ $\xrightarrow[\text{Step 2}]{\text{Exchange } \gamma \text{ with } \delta}$ $(\delta, .., \sigma, A)(...)(\gamma, .., \beta, \alpha)$ [Node XX]

$\xrightarrow[\text{Except the last}]{\text{Merge Cycles \& Resolve}}$ $(A)(\gamma, .., \beta, \alpha)$ $\xrightarrow[\text{Step 4}]{\text{Exchange } A \text{ with } symbol_{XX}(position_I(\gamma))}$ $(.., \beta, \alpha, \gamma, A)$

$\xrightarrow[\text{Step 5}]{\text{Resolve the cycle}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Procedure A2

Case 1: γ follows α in π'_1 and there is at least one other symbol between γ and A.

$(.., \beta, \alpha, \gamma, \delta, .., \sigma, A)(...)$ [Node u', also node XX]

$\xrightarrow[\text{Step 1}]{\text{Step } \beta \text{ to the first position}}$ $(\beta, \alpha, \gamma, .., \sigma, A)(...)$

Follow the routing of Procedure A1, Case 3, Subcase c

Case 2: γ is the only symbol between α and A in π'_1.

$(\omega, .., \beta, \alpha, \gamma, A)(...)$ [Node u',also node XX] $\xrightarrow[\text{Step 1}]{\text{Exchange } \omega \text{ with } \beta}$ $(\beta, \alpha, \gamma, A)(\omega, ..)(...)$

$\xrightarrow[\text{Merge cycles \& Resolve}]{\text{Exchange } \beta \text{ with any symbol other than } \omega}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Case 3: γ is the first symbol in π'_1.

$(\gamma, .., \beta, \alpha, \omega, .., A)(...)$ [Node u']

$\xrightarrow[\text{Step 1}]{\text{Exchange } \gamma \text{ with } \omega}$ $(\omega, .., A)(...)(\gamma, .., \beta, \alpha)$ [Node XX]

$\xrightarrow[\text{Step 2}]{\text{Merge cycles except the last \& Resolve}}$ $(A)(\gamma, .., \beta, \alpha)$ $\xrightarrow[\text{Step 3}]{\text{Exchange } A \text{ with the symbol following } \gamma}$

$(.., \beta, \alpha, \gamma, A)$ $\xrightarrow[\text{Step 4}]{\text{Resolve the cycle}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Case 4: γ is a symbol between α and A in in π'_1.

$(\omega, .., \beta, \alpha, \delta, .., \gamma, .., A)(...)$ [Nodeu']

$\xrightarrow[\text{Step 1}]{\text{Exchange } \omega \text{ with } \gamma \text{ and then exchange } \gamma \text{ with } \delta}$ $(\delta, .., \beta, \alpha, \gamma, .., A)(...)$ [Node XX]

if $A \neq symbol_{XX}(position_I(\gamma))$

$\xrightarrow[\text{and then Exchange it with } symbol_{XX}(position_I(\gamma))]{\text{Put A at first position}}$ $(...\beta, \alpha, \gamma, A)(...)$

$\xrightarrow{\text{Merge all cycles and Resolve}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Else $XX = (\delta, .., \beta, \alpha, \gamma, A)(...)$

$\xrightarrow[\text{And Resolve}]{\text{Merge All Cycles}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Case 5: γ is any other symbol.

Bring γ to the first position and then follow the routing of Case 3.

Procedure A3

Let $\pi_l' = (.., \beta, \alpha, \delta)$ and $\pi_1' = (\omega, .., A)$

Case 1: $\gamma = \omega$

$(\gamma, .., A)(.., \beta, \alpha, \delta)(...)$ [node u'] $\xrightarrow[\text{Step 1}]{\text{Exchange } \gamma \text{ with } \delta}$ $(\delta, .., \beta, \alpha, \gamma, .., A)(..)$ [Node XX]

If $position_{XX}(A) = position_I(\gamma)$

$\xrightarrow{\text{Merge and Resolve}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Else

$\xrightarrow[\text{Step 2}]{\text{Exchange } \delta \text{ with } symbol_{XX}(position_I(\gamma))}$ $(\delta, .., \beta, \alpha, \gamma)(.., A)(..)$

$\xrightarrow[\text{Step 3}]{\text{Merge \& Resolve all cycles but the the first}}$ $(A)(\delta, .., \beta, \alpha, \gamma)$

$\xrightarrow[\text{Step 4}]{\text{Exchange A with } \delta \text{ and Resolve}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Case 2: $\gamma = \delta$

$(\omega, .., A)(.., \beta, \alpha, \gamma)$ [Node u' also Node XX]

$\xrightarrow[\text{Step 1}]{\text{Put } \beta \text{ at first position}}$ $(\beta, \alpha, \gamma, .., A)$ Node X

If $A \neq symbol_X(position_I(\gamma))$ **then** $\xrightarrow{\text{Exchange } A \text{ with } \beta \text{ and then } A \text{ with } symbol_X(position_I(\gamma))}$;

$(...\beta, \alpha, \gamma, A)(...)$ $\xrightarrow{\text{Merge and Resolve}}$ $(\beta, \alpha, \gamma, A)$ [Target Node]

Case 3: $\gamma \notin \pi_l'$

[Node u'] $\xrightarrow[\text{Step 1}]{\text{Exchange } \gamma \text{ and } \omega}$ $(\gamma, .., A)(.., \beta, \alpha, \delta)$

Route as descibed in Case 1

Case 4: $\gamma \in \pi_l'$ and $\gamma \neq \delta$

$(\omega, .., A)(.., \gamma, .., \beta, \alpha, \delta..)(...)$ [Node u']

$\xrightarrow[\text{Step 1}]{\text{Exchange } \gamma \text{ and } \omega}$ $(\gamma, .., \beta, \alpha, \delta, .., A)$

$\xrightarrow[\text{Step 2}]{\text{Exchange } \gamma \text{ and } \delta}$ $(\gamma, .., \beta, \alpha)(\delta, .., A)(...)$ [Node XX]

$\xrightarrow[\text{Step 3}]{\text{Merge \& Resolve all cycles but the the first}}$ $(A)(\gamma, .., \beta, \alpha)$ [Node X]

$$\xrightarrow[\text{Step 4}]{\text{Exchange A with } symbol_X(position_I(\gamma))} (..., \beta, \alpha, \gamma, A)$$

$$\xrightarrow[\text{Step 5}]{\text{Resolve}} (\beta, \alpha, \gamma, A) \text{ [Target Node]}$$