UNLV Retrospective Theses & Dissertations

1-1-2007

Investigation of the robustness of star graph networks

Xiaolong Wu
University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/rtds

Repository Citation

Wu, Xiaolong, "Investigation of the robustness of star graph networks" (2007). *UNLV Retrospective Theses & Dissertations*. 2754.
http://dx.doi.org/10.25669/jm9c-w936

This Dissertation is protected by copyright and/or related rights. It has been brought to you by Digital Scholarship@UNLV with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Dissertation has been accepted for inclusion in UNLV Retrospective Theses & Dissertations by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.

INVESTIGATION OF THE ROBUSTNESS OF STAR GRAPH NETWORKS

by

Xiaolong Wu

Bachelor of Science in Mechanical Engineering
Nanjing University of Aeronautics & Astronautics
1998

Master of Science in Electrical Engineering
University of Nevada, Las Vegas
2004

A dissertation submitted in partial fulfillment
of the requirements for the

Doctor of Philosophy Degree in Electrical Engineering
Department of Electrical and Computer Engineering
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
August 2007

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

UMI Number: 3282017

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI®

UMI Microform 3282017

Copyright 2007 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

Dissertation Approval
The Graduate College
University of Nevada, Las Vegas

July 20, 20 07

The Dissertation prepared by

Xiaolong Wu

Entitled

Investigation of the Robustness of Star Graph Networks

is approved in partial fulfillment of the requirements for the degree of

Ph.D. in Electrical Engineering

S. Latifi
Examination Committee Chair

Dean of the Graduate College

Examination Committee Member

Examination Committee Member

Graduate College Faculty Representative

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ABSTRACT

Investigation of the Robustness of Star Graph Networks

by

Xiaolong Wu

Dr. Shahram Latifi, Examination Committee Chair
Professor of Electrical and Computer Engineering
University of Nevada, Las Vegas

The star interconnection network has been known as an attractive alternative to *n*-cube for interconnecting a large number of processors. It possesses many nice properties, such as vertex/edge symmetry, recursiveness, sublogarithmic degree and diameter, and maximal fault tolerance, which are all desirable when building an interconnection topology for a parallel and distributed system. Investigation of the robustness of the star network architecture is essential since the star network has the potential of use in critical applications. In this study, three different reliability measures are proposed to investigate the robustness of the star network. First, a constrained two-terminal reliability measure referred to as *Distance Reliability* (*DR*) between the source node u and the destination node I with the shortest distance, in an n-dimensional star network, S_n, is introduced to assess the robustness of the star network. A combinatorial analysis on *DR* especially for u having a single cycle is performed under different failure models (node, link, combined node/link failure). Lower bounds on the special case of the *DR*: antipode reliability, are

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

derived, compared with n-cube, and shown to be more fault-tolerant than n-cube. The degradation of a container in a S_n having at least one operational optimal path between u and I is also examined to measure the system effectiveness in the presence of failures under different failure models. The values of MTTF to each transition state are calculated and compared with similar size containers in n-cube. Meanwhile, an upper bound under the probability fault model and an approximation under the fixed partitioning approach on the (n-1)-star reliability are derived, and proved to be similarly accurate and close to the simulations results. Conservative comparisons between similar size star networks and n-cubes show that the star network is more robust than n-cube in terms of (n-1)-network reliability.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

TABLE OF CONTENTS

ABSTACT iii

LIST OF FIGURES vii

LIST OF TABLES ix

ACKNOWLEDGEMENTS x

CHAPTER 1 INTRODUCTION 1
- 1.1 Interconnection networks 2
- 1.2 Star interconnection network 8
  - 1.2.1 Cycle representation 10
  - 1.2.2 Transposition 11
  - 1.2.3 Properties of cycles 12
  - 1.2.4 Rouging 8
  - 1.2.5 Node-disjoint paths 19
  - 1.2.6 Antipode 20
  - 1.2.7 Decomposition 23
- References 24

CHAPTER 2 RELIABILITY AND EVALUATION 29
- 2.1 Fault, error, and failure 30
- 2.2 Quantitative evaluation methods 32
- 2.3 Performance measures 38
- 2.4 Reliability modeling methods 41
  - 2.4.1 Combinatorial method 42
  - 2.4.2 Markov model 52
- References 57

CHAPTER 3 DISTANCE RELIABILITY 59
- 3.1 Background 59
- 3.2 Node failure model 60
  - 3.2.1 Case I: $\pi(1)=1$ 61
  - 3.2.2 Case I: $\pi(i)\neq 1$ 62
  - 3.2.3 Special case: antipode reliability 67
- 3.3 Link failure model 73
- 3.4 Combined node and link failure model 75
  - 3.4.1 Case I: $\pi(1)=1$ 75

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.4.2 Case I: $\pi(i) \neq 1$ 77
3.4.3 Special case: antipode reliability 77
3.5 Conclusion 78
References 79

CHAPTER 4 MARKOV RELIABILITY MODELING 80
4.1 Background 81
4.2 Node failure model 83
4.2.1 Case I: $\pi(1) = 1$ 83
4.2.2 Case I: $\pi(i) \neq 1$ 92
4.3 Link failure model 94
4.4 Combined node and link failure model 97
4.5 Conclusion 100
References 100

CHAPTER 5 SUBSTAR RELIABILITY ANALYSIS 102
5.1 Background 103
5.2 Analysis of the $R_{n,n-1}(p)$ under the probability fault model 105
5.3 Approximation on $R_{n,n-1}(p)$ using the fixed partitioning 116
5.4 Results and analysis 119
5.5 Conclusion 124
References 125

CHAPTER 6 CONCLUSION AND DISCUSSION 127

APPENDIX MATLAB SIMULATION CODE 130

VITA 133

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF FIGURES

Figure 1.1 Hypercube of dimensions 1, 2, 3, and ... 6

Figure 1.2 A 3-dimensional butterfly network ... 8

Figure 1.3 Star graph of dimensions 2, 3, and 4 ... 9

Figure 1.4 Star transposition tree of dimension ... 13

Figure 1.5 Traveling of symbols on the leaves of a transposition tree after applying g_4g_5 ... 14

Figure 1.6 Positions of symbols after applying $g_2g_3...g_ng_2$ and $g_3g_4...g_ng_2g_3$ ... 14

Figure 2.1 Relationship between faults, errors, and failures ... 31

Figure 2.2 Relationship between the MTBF, MTBR, and MTTF ... 38

Figure 2.3 The reliability block diagram of a series system ... 43

Figure 2.4 The reliability block diagram of the parallel system ... 45

Figure 2.5 Example of a bridge-type network ... 48

Figure 2.6 Reliability diagram using the minimum cut set ... 51

Figure 2.7 Reliability diagram using the tie set method ... 52

Figure 2.8 Markov model of the TMR system ... 54

Figure 2.9 Reduced Markov model of the TMR system with a minimal number of states ... 55

Figure 3.1 Multiple ways of constructing four disjoint paths between $u(13254)$ and $I(12345)$ ... 65

Figure 3.2 *DR*s for star graph and hypercube when $|F|=2r-2$ ... 66

Figure 3.3 Hexagon construction of node-disjoint paths between u and its basic antipode I (Scenario 1) ... 70

Figure 3.4 Lower bounds on the antipode reliability for different node reliabilities. 71

Figure 3.5 Hexagon construction of node-disjoint paths between u and its basic antipode ... 72

Figure 3.6 Lower bounds on the antipode reliability for different values of the node reliability (Scenario 2) ... 73

Figure 3.7 Lower bounds on the antipode reliability for different node reliabilities. 78

Figure 4.1 Three node-disjoint optimal paths between $u(1342)$ and $I(1234)$ ... 84

Figure 4.2 State diagram for $S_{j,i}$ in a (u, I)-container of S_4 ... 85

Figure 4.3 Simple state diagram under the node-failure model for a (u, I)-container in a S_n ... 86

Figure 4.4 State diagram in the node failure model for a (u, I)-container in a S_n (Case 2) ... 92

Figure 4.5 State diagram in the link failure model for a (u, I)-container in a S_n ... 94

Figure 4.6 State diagram in the combined node and link failure model for a (u,I)-container in a S_n ... 98

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Figure 5.1 Scenarios where three (4-1)-stars intersect.......... 110
Figure 5.2 Four scenarios when three out of twelve (4-1)-stars in a S_4 are selected 114
Figure 5.3 The (n-1)-reliabilities of star graphs of sizes 5, 7, 8, and 10.......... 121
Figure 5.4 Approximate reliability comparisons between different size star graphs and hypercubes.......... 124

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF TABLES

Table 1.1 Comparisons between n-cube and n-star 9

Table 4.1 MTTF Values (Hours) for various size containers in star networks under the node failure model (Case 1), $\lambda_n = 10^{-5}$/hour. 90

Table 4.2 MTTF Values (Hours) for various size containers in hypercube networks under the node failure model (Case 1), $\lambda_n = 10^{-5}$/hour. 91

Table 4.3 MTTF Values (Hours) for various size containers in star networks under the node failure model (Case 1), $\lambda_n = 10^{-5}$/hour. 93

Table 4.4 MTTF Values (Hours) for various size containers in star networks under the node failure model (Case 1), $\lambda_l = 10^{-6}$/hour. 96

Table 4.5 MTTF Values (Hours) for various size containers in hypercube networks under the node failure model (Case 1), $\lambda_l = 10^{-6}$/hour. 97

Table 4.6 MTTF values (Hours) for various size containers in star networks under the combined node/link failure model, $\lambda_n = 10^{-5}$/hour, $\lambda_l = 10^{-6}$/hour. 99

Table 4.7 MTTF values (Hours) for various size containers in hypercube networks under the combined node/link failure model, $\lambda_n = 10^{-5}$/hour, $\lambda_l = 10^{-6}$/hour. 99

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ACKNOWLEDGEMENTS

This dissertation reflects what I have learned over the past three years. I have been extremely fortunate to work with my advisor Dr. Shahram Latifi. I am deeply grateful for his inspiration, kindness, and help.

I would like to thank Dr. Shahram Latifi for his valuable suggestions while working on my dissertation. I have learned from him through his insightful advice and suggestions during each discussion. I also thank Dr. Emma Regentova for knowledge that I learned from her class, and pleasant research experience under her supervision. I also thank Dr. Venkatesan Muthukumar and Dr. Laxmi Gewali for have kindly accepted to serve in my examination committee.

I sincerely thank all the people who have directly or indirectly helped me during my long journey study at this lovely university. I even cannot wait to miss everyone now. Last, and most importantly, I thank my family, my parents, my sister, and my uncle, especially my wife hong, for their support, help, and love. To them I dedicate this dissertation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER 1

INTRODUCTION

It has become generally accepted that an effective way to increase the system throughput is to gather a large set of processors to solve a single given complex engineering and science problem. There is a large class of problems that cannot be solved efficiently in the traditional sequential computers; however, many of them can be broken into smaller tasks and solved efficiently in the parallel fashion.

A parallel computer is one that consists of a collection of processing units, or processors, that cooperate to solve a problem by working simultaneously on different parts of that problem. The number of processors can vary from a few tens to several millions. Therefore, time required to solve the complicated engineering problem by a traditional uniprocessor computer can be significantly reduced using a parallel computer. This approach is attractive for a number of reasons. First, for many computational problems, the natural solution is a parallel one. Second, according to the latest technology development in the semiconductor industry [1], the cost and size of computer components have declined so sharply in recent years that parallel computers with a large number of processors have become feasible. Third, it is possible in parallel processing to select the parallel architecture that is best suited to solve the problem or class of problems under considerations.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Experiences of using parallel computers to solve various problems in the past decades by engineers and scientists in varieties of areas have indicated that the ultimate utilization of parallel computers is heavily dependent on the topology of the interconnection network that connects processors.

Many interconnection network topologies for parallel computers have been proposed in literatures. Among them, hypercube [8][10] is one of the most popular and has been studied extensively in different aspects. However, star networks [3][4] have been proposed as an alternative to the hypercube recently. The purpose of this dissertation is to investigate the robustness of the star graph by using the *combinatorial methods* and *Markov methods*. More specifically, we study the *distance reliability*, the *degradation of a container* between two arbitrary nodes in an n-dimensional star graph, and the *substar reliability* in the star interconnection network under the probability fault model and the fixed partitioning, and simulation, to investigate the robustness of the star graph. Meanwhile, conservative comparisons with the hypercube are performed for each above proposed merits. Throughout this dissertation, we use the terms "edge" and "link", "processor" and "node", "network" and "graph", and "S_{n-m}" and "$(n-m)$-star" interchangeably.

1.1 Interconnection networks

A processor/communication interconnection network is often modeled as an undirected graph, in which nodes correspond to processors and edges correspond to communication links between processors. Communications over a network is achieved by a message passing protocol, and the delay in the communication is usually measured

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

in terms of the edges traversed. Some of the key features of interest in an interconnection network are its degree, diameter, congestion, symmetry, connectivity, fault tolerance, routing algorithm, hierarchical structure, etc.

A number of topologies of interconnection networks have been proposed and some of them have been implemented in modern parallel computer systems [4][14][19][30][33][34]. These topologies range from simple graphs such as linear array, ring, binary tree, and complete graph, to more sophisticated graphs such as hypercube, butterfly network, and star network. Before we start to introduce interconnection networks, we need following terms which are frequently used to describe an interconnection network.

Degree of a node is the number of nodes that are adjacent to it. We say a network is *regular* if all nodes have the same degree. *Degree of a regular graph* is then the degree of any of its vertices. *Distance* between a pair of nodes is the smallest number of links that have been traversed to go from one to the other. *Diameter* of a network is the maximum distance between any pair of nodes. Clearly, the degree of a graph is a measure of the cost of interconnection networks and the diameter is a measure of the communication delay. Consequently, it is desirable to construct a large graph with small degree and small diameter.

A graph is *vertex symmetric* if the graph looks the same from each of its vertices. More formally, given any two vertices u and v there is an automorphism of the graph that maps u to v. Similarly, a graph is *edge symmetric* if every edge looks the same, i.e., given any two edges i and j there is an automorphism of the graph that maps i to j. Such symmetry properties of a graph are very important when viewed as an interconnection network. For example, a vertex symmetric graph allows for all the processors to be

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

identical. A vertex/edge symmetric graph has the desirable property that the communication load is uniformly distributed on all the nodes or over all the communication links, so that there is no congestion across the interconnection network. Moreover, this symmetry allows for identical nodes with identical routing algorithms. It is also very useful in designing algorithms that exploit the structure of the network. For a typical n-dimensional symmetric network, if it can be decomposed into a number of (n-1)-dimensional networks which has the same topological properties as the original one, and if this decomposition can be carried out recursively, we call this network has the *recursive decomposition property*.

A graph is said to be *f-fault tolerant* if whenever f or fewer vertices are removed from the graph the remaining graph always remains connected. The *fault tolerance* of a graph is defined to be the largest f for which it is f-fault tolerant. Thus, the *fault tolerance* of a graph can at most be d-1, where d is the degree of the graph. The *fault diameter* of a network is the maximum diameter of the network by removing d-1 nodes. A graph whose fault tolerance is exactly d-1 is said to be *maximally fault tolerant*.

Given a set of generators for a finite group G, a *Cayley graph* [2] is defined as a graph, in which nodes correspond to the elements of the group G and edges correspond to the action of the generators in G. That is, there is an edge from an element a to an element b if and only if there is a generator g such that $ag = b$ in G. For example, hypercubes and star graphs are two representatives of the Cayley graph. Each of the Cayley graphs contains the following properties, such as node/link symmetric properties, recursive decomposition property, maximally fault tolerant, etc. Before moving to the star network, we introduce some common interconnection topologies as follows.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Linear Array: The linear array [19] is the simplest connection topology. It is a one-dimensional network in which n nodes are connected by n-1 links in a line. Internal nodes have degree 2, and the terminal nodes have degree 1. The diameter of a linear array of n nodes is n-1. Its structure is not symmetric and thus poses a communication inefficiency when n becomes large.

Ring: A ring [19] is obtained by connecting the terminal nodes of a linear array with one extra link. A ring can be unidirectional or bidirectional. It is symmetric with a constant node degree 2. The diameter is $\lfloor n/2 \rfloor$ for a bidirectional ring, and n for a unidirectional ring.

Binary Tree: In this network [37], the nodes form a complete binary tree. A k-level complete binary tree has $n = 2^k - 1$ nodes. The maximum node degree is 3 and the diameter is $2(k-1)$.

Hypercube: The n-dimensional hypercube [14] shown in Fig. 1, n-cube or Q_n, has $N = 2^n$ nodes and $n2^{n-1}$ edges. Each node corresponds to an n-bit binary string, and two nodes are linked with a link if and only if their binary strings differ in precisely one bit. As a result, each node is incident to n other nodes. The edges of the hypercube can be naturally categorized according to the dimensions that they traverse. In particular, an edge is called a dimension i edge if it connected two nodes that differ in the i^{th} bit position. In addition the n-cube is a completely symmetric topology and, consequently, minimizes congestion problems. It also permits the use of identical processors since every vertex plays an identical role in the topology. The n-cube has a very simple and optimal routing algorithm that routes messages between processors along a shortest path [12]. This enables the design of low-cost routing hardwares. Other attractive features of

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the n-cube include our familiarity and understanding of the topology, particularly the recursive decomposition of the n-cube into successive cubes of smaller sizes.

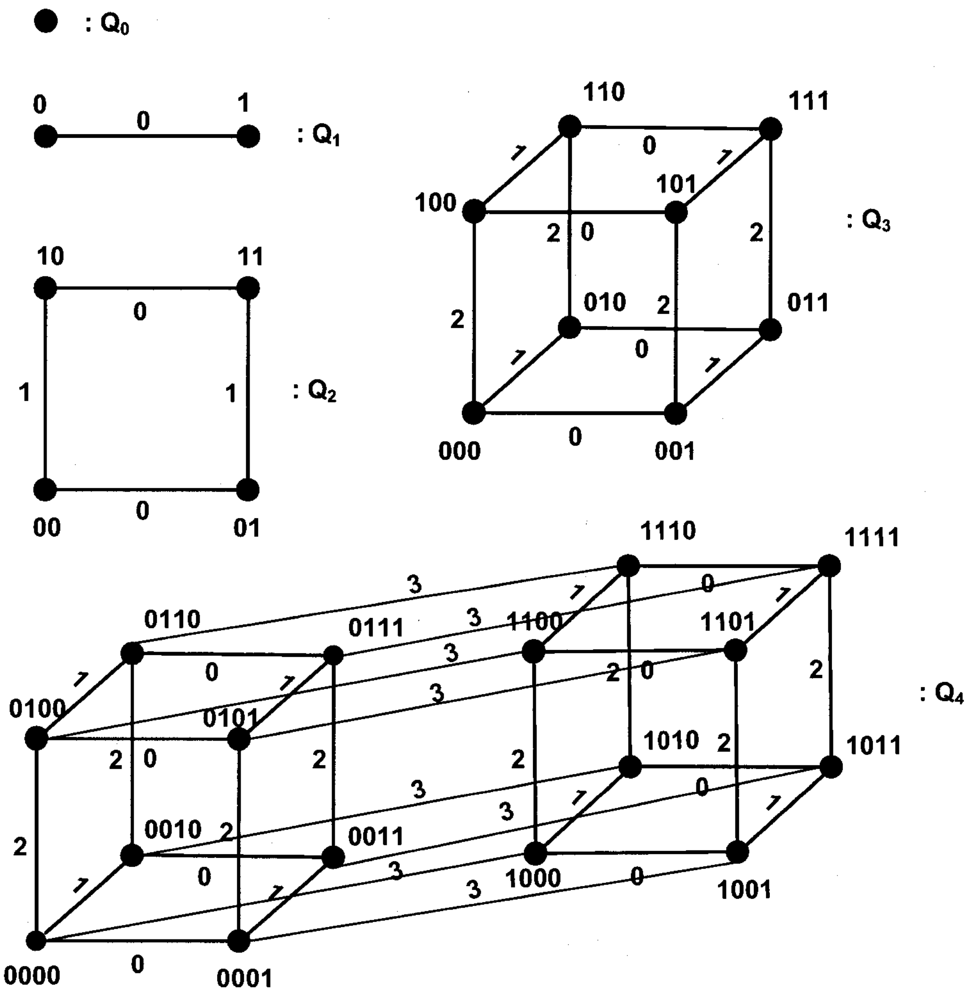


Figure 1.1. Hypercube of dimensions 1, 2, 3, and 4.

Each Q_n contains two disjoint Q_{n-1}'s. Partitioning a Q_n can be done in n different ways. This is implemented by removing the set of i-links, $0 \leq i < n$, every time. For example, partitioning a Q_4 along the 3-links results in two disjoint Q_3's, $XXX0$ and $XXX1$, respectively (Fig. 1.1); while partitioning a Q_4 along the 2-links results in two

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

disjoint Q_3's, $XX0X$ and $XX1X$, where X is either '0' or '1', and so on. It follows that the number of disjoint Q_{n-m}'s in a Q_n is 2^m, for $1 \leq m \leq n$, whereas the number of distinct Q_{n-m}'s is $\binom{n}{m} 2^m$.

The characterization of distinct and disjoint paths between two given nodes for the hypercube has been addressed in [33]. If u and v are two nodes in an n-cube with the Hamming Distance $r = H(u,v)$, then there are $r!$ distinct paths of length r between u and v, where there are r disjoint paths of length r, and $n - r$ disjoint paths of length $r + 2$ between u and v. Thus in total we can construct a family of n disjoint paths between u and v, which is the maximum allowable number of parallel paths between two nodes in the n-cube.

There are also a number of other fault tolerant properties of the n-cube that make it very attractive. Due to the multitude of paths between vertices the n-cube not only possesses optimal fault tolerance properties but provides little or no degradation of performance in the presence of faults. The fault tolerance of the n-cube is n-1 and therefore said to be maximally fault tolerant [8][35].

Butterfly Network: The r-dimensional butterfly [19] has $(r+1)2^r$ nodes and $r2^{r+1}$ edges. The nodes corresponds to pair $< w, i >, 0 \leq i \leq r$, where i is the level or dimension of the node and w is an r-bit binary number that denotes the row of the node. Two nodes $< w, i >$ and $< w', i' >$ are linked by an edge if and only if $i' = i + 1$ and either:

1. w and w' are identical, or
2. w and w' differ in precisely the $i'th$ bit.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

If w and w' are identical, the edge is said to be a straight edge. Otherwise, the edge is a cross edge. In addition, edges connecting nodes on levels i and $i+1$ are said to be level $i+1$ edges. A 3-dimensional butterfly network is shown in Fig. 1.2.

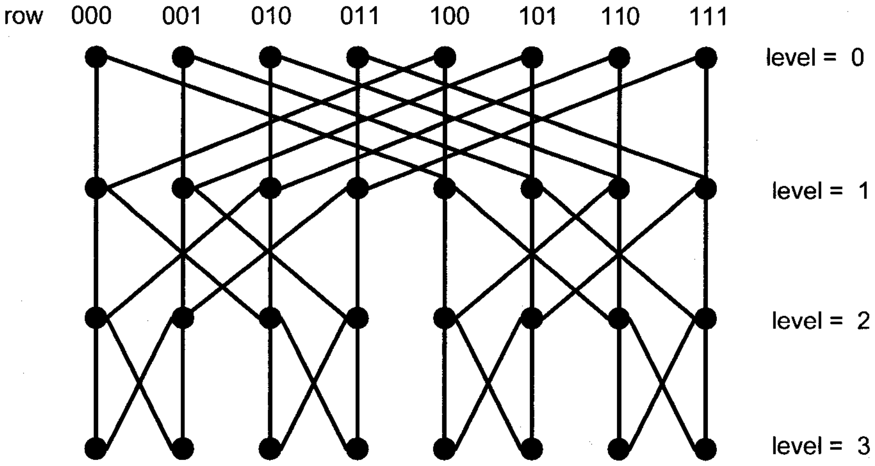


Figure 1.2. A 3-dimensional butterfly network.

1.2 Star interconnection network

An n-dimensional star network [4], S_n, is defined as an undirected graph $G = (V, E)$, where V is set of $n!$ nodes, and E is the set of $(n\text{-}1)n!/2$ links. Nodes are assigned with labels each of which is a distinct permutation on n different symbols (we use symbols 1, 2, 3,..., n). Two nodes, u and v, are connected with a link labeled with link i if and only if node v can be obtained through $ug_i = v$, where g_i, $2 \leq i \leq n$, is the generator swapping the first symbol with the i^{th} symbol in the permutation of node u. For example, in a 4-star containing 24 nodes, node 3214 can be obtained by applying g_3, swapping the first and third symbols in node 1234. Fig. 1.3 shows the star graph of dimensions 2, 3, and 4.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

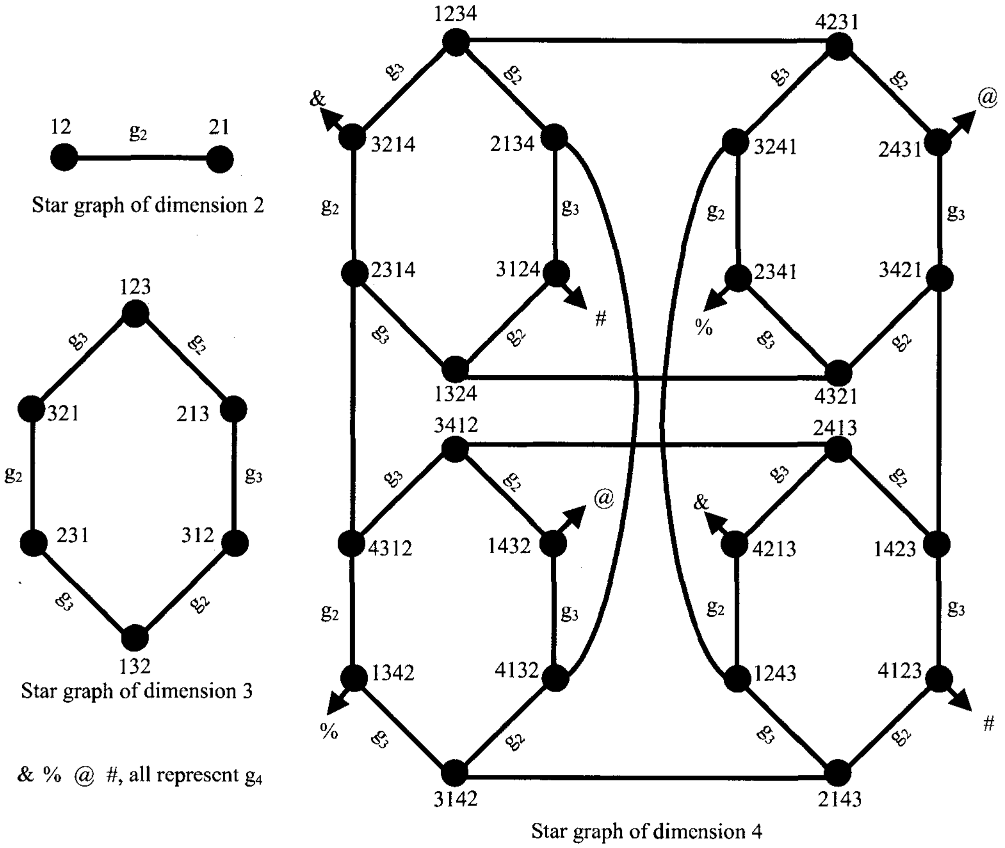


Figure 1.3. Star graph of dimensions 2, 3, and 4.

Table 1.1. Comparisons between *n*-cube and *n*-star.

n-cube					*n*-star graph				
n	node 2^n	degree *n*	diameter *n*	fault diameter *n+1*	*n*	node *n!*	degree *n-1*	diameter $\lfloor \frac{3}{2}(n-1) \rfloor$	fault diameter *Dia+1*
7	128	7	7	8	5	120	4	6	7
9	512	9	9	10	6	720	5	7	8
12	4096	12	12	13	7	5040	6	9	10
15	32768	15	15	16	8	40320	7	10	11
19	524288	19	19	20	9	362880	8	12	13

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Several researchers have investigated the algebraic properties of the star network using performance metrics such as number of nodes and links, connectivity, diameter, surface area, fault diameter, diagnosability, etc [2][4][15][20][23][36][42]. The star network can connect more processors with less interconnections and less communication delay than the popular hypercube [12][21]. More specifically, growth in the node degree and diameter is sub-logarithmic to the network size in star network, but logarithmic in the hypercube. Various networks have been mapped to the star network [6][24][29]. A treatment of communication aspects for this network is presented in [5][11][13][16][28][32]. Fault tolerance of the star network has also been investigated extensively in [3][9][17][18][25][26][27][31][38][39][40].

Like the hypercube, the stat network has rich structure and symmetric properties as well as maximally fault-tolerant characteristics. The star network has superior node degree and diameter compared to the hypercube of a comparable size. More specifically, growth in node degree and diameter is sub-logarithmic to the network size in the star network but logarithmic in the hypercube. This can be visualized in Table 1.1. Next we list some major properties of the star network.

1.2.1 Cycle representation

The node permutation can be decomposed into a sequence of one or more disjoint cycles, each of containing a set of distinct symbols. For example, the node permutation [42651387] in a 8-star, can be decomposed into cycles: (145)(2)(36)(78). In our case, cycles are built by identifying misplaced symbols in the node permutation, starting from the leftmost position. Note that, any digit already in the correct position appears in a cycle of length 1, or a 1-cycle. Depending on the symbol in the first position (leftmost) of

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the node permutation being '1' or not, cycles can be categorized into:

- Ordinary cycle that contains '1'
- None-ordinary cycle that does not contain '1'

In above-mentioned example, cycle (145) is an ordinary cycle, and cycles (36), (78) are none-ordinary cycles.

For a given source permutation, cycles can move according to the following two rules:

- Cycles can appear in any order
- Cycles without "1" inside can be executed to reach the same destination regardless of the number of cyclic shifts on the symbols;

After applying above rules, the following representations (145)(2)(36)(78), (36)(145)(2)(78), and (145)(2)(63)(78), all specify the same source permutation [42651387].

1.2.2 Transposition

A transposition is a permutation which exchanges two elements and keeps all others fixed [7]. For example, transposition $(1a_i)$ means permutation between 1 and a_i. Cycles can be written as a product of transpositions using the following two rules:

- $(a_1a_2a_3\cdots a_k)=(1a_1)(1a_2)(1a_3)\cdots(1a_k)(1a_1)$
- $(1a_1a_2a_3\cdots a_k)=(1a_1)(1a_2)(1a_3)\cdots(1a_k)$

where a_1 through a_k denote distinct symbols from 2 to n. For example, cycle (145) can be written as a product of transpositions (14)(15), and cycle (36) can be written as a product of transpositions (13)(16)(13). If the number of transpositions for a given node permutation is odd, we name this node permutation *odd*; otherwise we name it *even*. The

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

identity permutation is considered as even, since it involves non transposition. Since the star graph is bipartite symmetry, the number of the odd node permutations equals to the number of the even node permutations.

1.2.3 Properties of cycles

Let us consider the star transposition tree with the following correspondence between transpositions and generators: $g_2 = (12)$, $g_3 = (13)$, $g_4 = (14)$, $\cdots, g_n = (1n)$. Generators are partially ordered according to the decreasing indices: $g_n > g_{n-1} > \cdots > g_3 > g_2$. Vertices are originally labeled as the identity permutation I, with the symbol '1' sitting in the center, as shown in Fig. 1.4(a). Therefore, *a product of transpositions for any cycle can be represented by a product of the corresponding generators*, where, for simplicity, dots in products will be omitted. Based on the two different categories of cycles, the product of generators can also be divided into two cases [28]:

i): *Ordinary product of generators*: where the first generator is different from the last one in the product of generators. This corresponds to the ordinary cycle. The general form for the ordinary product of generators can be described as $g_{i_1} g_{i_2} g_{i_3} ... g_{i_k}$, where $g_2 \le g_{i_j} \le g_n$, and $2 \le i_j \le n$.

ii): *None-ordinary product of generators*: where the first generator is the same as the last one in the product of generators. This corresponds to the none-ordinary cycle. The general form for this product can be described as $g_{i_1} g_{i_2} g_{i_3} ... g_{i_k} g_{i_1}$.

For example, the ordinary product of generators corresponding to cycle (145) is $g_4 g_5$, and the none-ordinary product of generators corresponding to cycle (36) is $g_3 g_6 g_3$. Next give details about the properties for each type of the product of generators.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

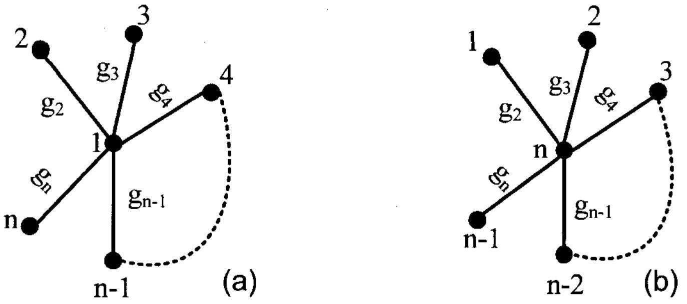


Figure 1.4. Star transposition tree of dimension *n*.

Case I: Properties of the ordinary product of generators

Lemma 1: For the star graph with *n*-1 generators, $(g_2\ g_3\ g_4 \cdots g_n)^n = I$ holds.

Proof: The product $g_2\ g_3\ g_4 \cdots g_n$ can be considered as a sequence of corresponding swapping of symbols on the transposition tree in Fig. 1.4(a). After applying the product, every symbol on the leaf will "travel" clockwise one step starting from the first symbol, and the last one moves to the center. The resulting placement is shown in Fig. 1.4(b). After *n* such operations on the product, symbols will return to the original positions. Q.E.D.

Each node permutations after applying the product every time is presented as follows:

$$12345 \cdots\cdots n-1n$$
$$n1234 \cdots\cdots n\text{-}3n\text{-}2n\text{-}1$$
$$\cdot$$
$$\cdot$$
$$\cdot$$
$$12345 \cdots\cdots n\text{-}2n\text{-}1n$$

Lemma 2: Lemma 1 holds for every ordinary product. In another word, $(g_{i_1} g_{i_2} g_{i_3} \cdots g_{i_k})^{k+1} = I$ holds.

Proof: We can extract a subtree with an arbitrary set of *k* generators and arrange its edges

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

corresponding to some permutation of generators in asset. After repetitive applying k+1 times of product $g_{i_1}g_{i_2}g_{i_3}...g_{i_k}$, symbols will reach their original positions by "traveling" on the certain leaves of the transposition tree. Q.E.D.

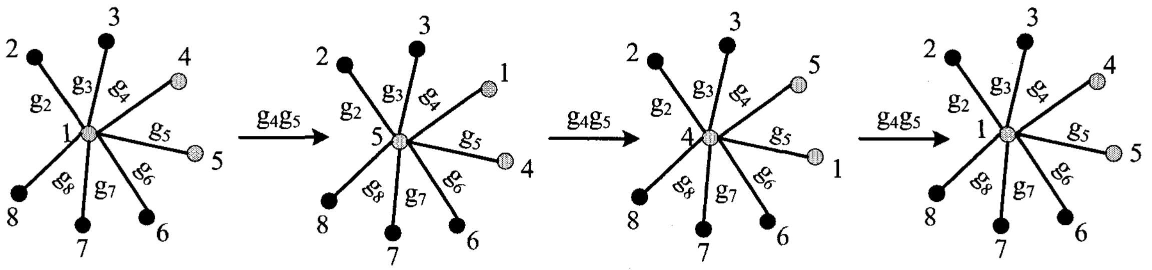


Figure 1.5. Traveling of symbols on the leaves of a transposition tree after applying g_4g_5.

For example, after applying the product g_4g_5 three times, the symbols in the vertices of tree (Fig. 1.5) will travel on the leaves (only grey ones) back to their original positions.

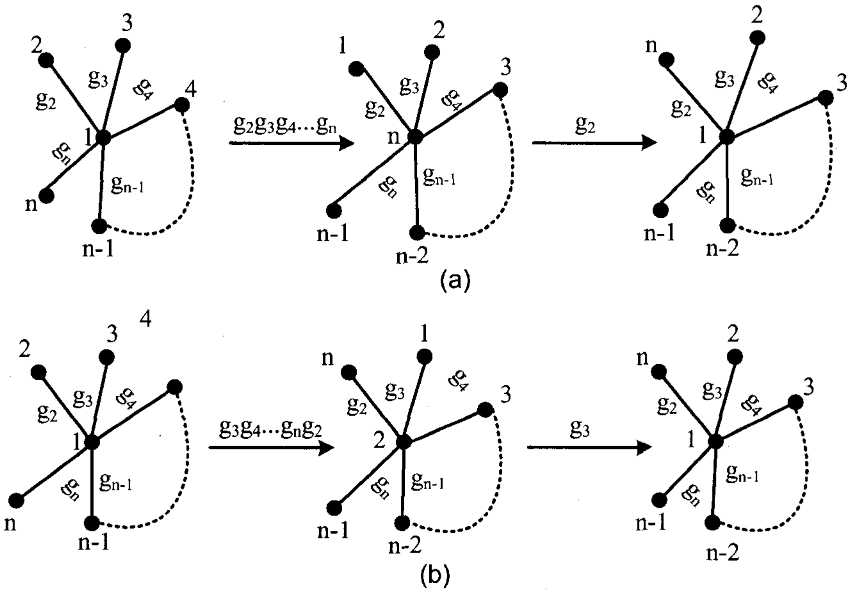


Figure 1.6. Positions of symbols after applying $g_2g_3 \cdots g_ng_2$ and $g_3g_4 \cdots g_ng_2g_3$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Case II: Properties of the none-ordinary product of generators

Lemma 3: For the star graph with n-1 generators, the following holds:

$$g_2g_3g_4\cdots g_{n-1}g_ng_2=g_3g_4g_5\cdots g_ng_2g_3=\cdots\cdots=g_ng_2g_3\cdots g_{n-2}g_{n-1}g_n$$

Proof: When the first n-1 generators are applied, all the symbols will shift one step clockwise on the corresponding leaf, as in Fig. 1.6 (only presenting the first two cases here, the remaining can be executed similarly). Then, when the last generator of the product is applied, the symbol 1 will return back to its original position (center), and we obtain the same permutation $1n23\cdots(n-2)(n-1)$ for all the products listed above. Q.E.D.

Corollary 3.1: For star graph with n-1 generators, the following does not hold:

$$g_2g_3g_4\cdots g_{n-1}g_n=g_3g_4g_5\cdots g_ng_2=\cdots\cdots=g_ng_2g_3\cdots g_{n-2}g_{n-1}$$

Part of the proving process for Lemma 3 has shown that symbols on the center vertex in the star transposition tree are different before applying the last generator.

Corollary 3.1.2: The position of each symbol in the ordinary cycle for the cycle representation of the source permutation is fixed, while cyclic shift operations on symbols in a none-ordinary cycle are allowed.

Proof: Proof follows Corollary 3.1 since each symbol in the ordinary cycle corresponds to a distinct generator in the star transposition tree. This concludes that the cyclic shift operations on the symbols in an ordinary cycle are not allowable. Proof for the second part is already shown in Lemma 3. Q.E.D.

Lemma 4: If some products of generators in a star graph consists of two subproducts of distinct sets of generators $\prod_o=\prod_a\prod_b$, and if either of them is of the none-ordinary type, then $\prod_o=\prod_a\prod_b=\prod_b\prod_a$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Proof: We can decompose the original star transposition tree into two substrees, corresponding to distinct sets of generators and connected through the center vertex. Let us assume that symbol x ($x \in \{1,2,3,\cdots,n\}$) was in the center vertex before applying the none-ordinary product. After applying the none-ordinary product, all symbols of its subtrees will be shifted one position clockwise on the leaves, and the symbol x will be returned to the center vertex. Since sets of generators in $\prod_a$ and $\prod_b$ are distinct, this cannot affect any shifting of symbols on the other subtree; therefore, the ordering of two subproducts (either ordinary or none-ordinary) is arbitrary. Q.E.D.

Lemma 5: The none-ordinary subproduct of generators can be nested in other subproduct consisting of a distinct set of generators, whether it is of ordinary or none-ordinary type.

Proof: Since the none-ordinary subproduct preserves the symbol in the center vertex of the star transposition tree and its distinct generators will not affect any shifting of symbols in the other subproduct, it can be nested in the other subproduct (whether it is of ordinary or none-ordinary type) without affecting the resulted final permutation. Q.E.D.

The following example will illustrate the nesting of subproducts and the corresponding cycles:

$$\prod_o = \underbrace{g_4g_5}\,\underbrace{g_3g_6g_3} = \underbrace{g_3g_6g_3}\,\underbrace{g_4g_5} \qquad C_o = (145)(36) = \underbrace{(14)(15)}\,\underbrace{(13)(16)(13)}$$

$$= \underbrace{g_4\,\underbrace{g_3g_6g_3}\,g_5} \qquad\qquad = \underbrace{(13)(16)(13)}\,\underbrace{(14)(15)} = \underbrace{(14)\,\underbrace{(13)(16)(13)}\,(15)}$$

The previous lemmas can be generalized to k subproducts of distinct sets of generators by the following lemma.

Lemma 6: If some product of generators in the star graph consists of k subproducts ($2 \le k < n$) of distinct sets of generators $\prod_o = \prod_1 \prod_2 \cdots \prod_k$ and if at most one of them is

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

an ordinary one, then the ordering of these subproducts is arbitrary, and the none-ordinary ones can be nested in the ordinary subproduct or/and other none-ordinary ones.

Proof: Again we can divide a star transposition tree into k distinct subtrees connected through the center vertex. Assume that symbol x ($x \in \{1,2,3,\cdots,n\}$) was in the center vertex before applying the none-ordinary products. Each none-ordinary subproduct will only shift the symbols of its subtree clockwisely on the corresponding leaves and return the symbol x to the center vertex. The ordinary subproduct will only shift the symbols of its subtree clockwisely on the corresponding leaves and replace the center vertex x with the last symbol form its subtree. Since the generators in the ordinary subproduct and none-ordinary ones are distinct, this cannot affect any shifting of symbols on other subtree, therefore the ordering of them is arbitrary. Due to the distinct sets of generators in none-ordinary subproducts and the preserving of the center symbol, they can be nested in the ordinary subproduct or/and other none-ordinary ones. Q.E.D.

The following example explains the above lemma 6:

$$\Pi_o = \underbrace{g_4g_5}\underbrace{g_3g_6g_3}\,\underbrace{g_7g_8g_7} = \underbrace{g_4g_5}\underbrace{g_7g_8g_7}\,\underbrace{g_3g_6g_3} = \cdots = \underbrace{g_7g_8g_7}\,\underbrace{g_3g_6g_3}\,\underbrace{g_4g_5}$$

$$= \underbrace{g_4\,\underbrace{g_3g_6g_3}\,g_5}\,\underbrace{g_7g_8g_7} == \underbrace{g_4g_5}\,\underbrace{g_3\,\underbrace{g_7g_8g_7}\,g_6g_3} = \cdots = \underbrace{g_4g_5}\,\underbrace{g_3g_6\,\underbrace{g_7g_8g_7}\,g_3}$$

Corollary 6.1: If the cycle representation of the source permutation is the product of k distinct cycles of distinct symbols $C_o = C_1C_2\cdots\cdots C_k$, and if at most one of them is an ordinary one, then the ordering of these cycles is arbitrary, and the none-ordinary cycles can be nested in the ordinary cycle or/and other none-ordinary ones.

Proof follows Lemma 6 and Corollary 4.1.

The following example explains the above Corollary 6.1:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\begin{aligned}C_o &= \underbrace{45}\underbrace{363}\underbrace{787}\underbrace{910119} = \underbrace{45}\underbrace{363}\underbrace{910119}\,\underbrace{787} = \underbrace{45}\underbrace{787}\underbrace{910119}\underbrace{363} = \underbrace{45}\underbrace{787}\underbrace{363}\underbrace{910119} = \\ &= \underbrace{4\underbrace{363}5}\,\underbrace{787}\underbrace{910119} = \underbrace{45}\,\underbrace{7\underbrace{363}87}\,\underbrace{910119} = \underbrace{45}\underbrace{78\underbrace{363}7}\,\underbrace{910119} = \underbrace{45}\underbrace{787}\underbrace{9\underbrace{363}10119} \\ &= \underbrace{4\underbrace{363}\underbrace{787}5}\underbrace{910119} = \cdots = \underbrace{45}\underbrace{9\underbrace{363}\underbrace{787}10119} \\ &= \underbrace{4\underbrace{363}\underbrace{787}\underbrace{910119}5}\end{aligned}$$

This example shows how large the number of alternative optimal paths between two arbitrary nodes can be in the star graph, even without taking into account all the cyclic shifts on each symbol in the none-ordinary cycle. If we do not consider the nesting of cycles and we assume all the cycles are of the none-ordinary type, the number of alternative optimal paths is $k!\prod_{i=1}^{k} p_i$, where k is number of cycles and p_i denotes the number of distinct symbols in each cycle C_k.

1.2.4 Routing

The routing between two nodes in the star graph is accomplished by sending the message from the current node to the next node until the destination node is reached. Since the star graph is vertex symmetric, with no loss of generality, the destination node is assumed to have been labeled as the Identity Permutation $I = 123\cdots n$, hence the routing is equivalent to sorting the source permutation to the destination permutation.

Given the label of the source node, there are two ways to specify the destination node. One way is to use the label of the destination node. The second way is to exploit the fact that in the star graph one label is simply a permutation on the digits of the other label. But, since any permutation can be viewed as a set of cycles, i.e., cyclically order sets of digits with the property that each digit's desired position is that occupied by the next digit

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

in the set. Hence, after having the label of the destination node fixed to *I*, to specify the source node is to exploit the fact that the label, *u*, the source permutation, is simply a permutation on the digits of *I*. Routing between two given nodes is accomplished based on the following two rules [4]:

i) If 1 is the leftmost digit, move it to any position not occupied by the correct digit;

ii) If *i* is the leftmost digit ($1 < i \leq n$), move it to its correct position.

It was shown that these two rules [4] ensure an optimal path of the minimum distance:

$$\text{distance} = c + m - \begin{cases} 0, & \text{if 1 is first in the source permutation} \\ 2, & \text{if 1 is not first in the source permutation} \end{cases}$$

where *c* denotes the number of cycles of length greater than 2, and *m* is the total number of symbols in these cycles, namely misplaced symbols. Next section continues to introduce the node-disjoint paths in the star graph.

1.2.5 Node-disjoint paths

A path originating from a node can be uniquely specified by the labels of links it traverses. The length of a path is the number of links it traversed. A path is operational if it passes through fault-free intermediary nodes and links. Two or more paths are node-disjoint if, except for the source and destination nodes, they do not have any node in common. It is important to have node-disjoint (or parallel) paths between two nodes in an interconnection network to speed up the transfer of a large amount of data and provide alternative routes in the case of node failures.

The maximum number of node-disjoint paths of the shortest length between a given pair of nodes in the star graph has been derived in [12] and is known to be "*n*-1". The notation $\pi(i)$ is used to refer to the i^{th} digit of the label of *u*. If the node permutation of *u*,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

expressed in its cyclic form, then it follows:

i) If $\pi(1) = 1$, there are n-1 node-disjoint paths between u and I as follows:

   a) m paths of the shortest length $c+m$

   b) n-m-1 paths of length $c+m+2$

ii) If $\pi(1) \neq 1$, there are n-1 node-disjoint paths between u and I as follows:

   a) c paths of the shortest length $c+m-2$

   b) m-c-1 paths of length $c+m$, and

   c) n-m paths of length $c+m+2$

The above results are very important as we shall make the extensive use of them in Chapters 3 and 4 of this dissertation. For the purpose of simplicity, we use r to denote the shortest distance between u and I in the rest of this paper.

1.2.6 Antipodes

A Dimension Permutation (*DP*) δ is a permutation on the set of dimensions of the network (or equivalently a permutation on the digits of the label of each node in the network). The *DP* is a bijection which assigns uniquely to any given node, a specific node in the network. The first node can be thought of as the source and the second node as the destination. A *DP* δ can be specified by a set of cycles which maps the source to the destination. This set of cycles will be equivalent to the cycle representation of the source node label if the destination node is I.

In a network, the farthest node(s) from a given node along the shortest path is called the node's antipode(s) [22]. The antipode of a node is apart from it by d_n (diameter of the star network) and can be specified by a maximum permutation ($\delta_{\max}$). More specifically, in $\delta_{\max}$ with digit 1 in place, all cycles must be of length 2 for odd n. For even n, there

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

may be two possibilities. There there may be one cycle of length 3 and the rest of length 2 (again, digit 1 must be in place); or $n/2$ 2-cycle with one cycle containing digit 1. Details are explained in [22] as follows:

a) When n is odd, all cycles of $\delta_{\max}$ must be of length 2 with digit 1 in place. Then we have $\delta_{\max} = (1)(i_2 i_3)(i_4 i_5)\cdots(i_{n-1} i_n)$, where $1 < i_2, i_3, \cdots, i_n \leq n$. And $m = n-1$ and $c = (n-1)/2$.

b) When n is even, there are two possibilities. First, there is one cycle of length 3 and the rest of length 2 (digit 1 must be in place), therefore we have $\delta_{\max} = (1)(i_2 i_3 i_4)(i_5 i_6)\cdots(i_{n-1} i_n)$, where $1 < i_2, i_3, \cdots, i_n \leq n$. And $m = n-1$ and $c = (n-2)/2$. Second, there are $n/2$ 2-cycles with one cycle containing digit 1, therefore we have $\delta_{\max} = (i_1 i_2)(i_3 i_4)(i_5 i_6)\cdots(i_{n-1} i_n)$, where $1 < i_2, i_3, \cdots, i_n \leq n$. And $m = n$ and $c = n/2$.

As an example, in a 5-star, one of the antipodes of a node can be specified by the following permutation: $\delta_{\max} = (1)(23)(45)$.

For a 6-star, there are two possibilities to specify the antipodes of a node. For example, $\delta_{\max}$ can be in forms of: $(1)(234)(56)$ or $(12)(34)(56)$.

In S_n, there exist more than one way to construct cycles so that a $\delta_{\max}$ can be obtained. Two cases are distinguished to find the number of antipodes for a given node in a S_n [22]:

Case (i): n is odd. Then $\delta_{\max} = (1)(i_2 i_3)(i_4 i_5)\cdots(i_{n-1} i_n)$, where $1 < i_2, i_3, \cdots, i_n \leq n$. Note that the order in which the cycles appear does not change the antipode. The number of distinct cycles of which 2 is a member (i.e., 23, 24, $\cdots$, 2n) is (n-2). After selecting the pair for

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the first 2-cycle, there are (n-3) digits left. The number of distinct 2-cycles of which a given digit, say 4, is a member is (n-4), and so on. It follows that the number of antipodes for a given node in this case is $N_{antipode} = (n-2)(n-4)(n-6)\cdots 1$.

Case (ii): n is even. Depending on $\delta_{\max}$ we consider two possibilities. First, let $\delta_{\max} = (1)(i_2 i_3 i_4)(i_5 i_6)\cdots(i_{n-1} i_n)$. Using a similar argument as in Case (i) the number of antipodes for a given node here is given by: $2\times\binom{n-1}{3}(n-5)(n-7)\cdots 1$ (The factor of 2 is due to the fact that 3 elements x, y, and z can form two distinct 2-cyclces). Second, let $\delta_{\max} = (i_1 i_2)(i_3 i_4)(i_5 i_6)\cdots(i_{n-1} i_n)$. The number of antipodes corresponding to this permutation is: $(n-1)(n-3)(n-5)\cdots 1$. Summing the number of antipodes gives

$$N_{antipode} = (n-1)(n-3)(n-5)\cdots 1 + 2\times\binom{n-1}{3}(n-5)(n-7)\cdots 1.$$

Since the number of antipodes is more than one (for n>3), we define the antipode corresponding to the following *DP*'s as the basic antipode.

$$\delta_{\max} = (1)(23)(45)\cdots(n-1,n), \qquad \text{for odd } n;$$

$$\begin{aligned}\delta_{\max} &= (1)(234)(56)\cdots(n-1,n), \qquad \text{for even } n\\ &\text{or } (12)(23)(45)\cdots(n-1,n)\end{aligned}$$

As an example, in a 5-star, there exists 3 antipodes for each node which can be reached from that node by applying the following permutations:

$$\begin{aligned}\delta_{\max 1} &= (1)(23)(45) \qquad \text{(basic antipode)}\\ \delta_{\max 2} &= (1)(24)(35)\\ \delta_{\max 3} &= (1)(25)(34)\end{aligned}$$

However, there is only one unique antipode for a given node in the hypercube network. This antipode is in distance n away from the given node, and its binary string

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

representation is the complement of the label of the given node. This brings new requirements that communications between a given node and its antipodes in the star graph needs to be investigated extensively before applications.

1.2.7 Decomposition

Each S_n contains *n* disjoint $S_{n\text{-}1}$'s. Partitioning a S_n can be done in (*n*-1) different ways. This is implemented by removing the set of *i*-links, $2 \le i \le n$, every time. For example, partitioning S_4 along the *4*-link will result in $\{XXX4, XXX3, XXX2, XXX1\}$, while partitioning S_4 along the *3*-link will result in $\{XX4X, XX3X, XX2X, XX1X\}$, partitioning S_4 along the *2*-link will result in $\{X4XX, X3XX, X2XX, X1XX\}$. Fig. 3 illustrates a S_4 where the mentioned substars S_3's are shown as hexagons. Partitioning S_n along dimension "1" will result in *n*! isolated nodes. This is the only case where partitioning does not produce "*n*" $S_{n\text{-}1}$'s. It follows that the number of disjoint $S_{n\text{-}m}$'s in a S_n is $\binom{n}{m} m!$, for $1 \le m < n$, whereas the number of distinct $S_{n\text{-}m}$'s is $\binom{n-1}{m}\left[\binom{n}{m} m!\right]$.

Each substar can be uniquely labeled as a string of symbols over the set $\{1, 2, 3, \cdots, n-1, n, X\}$, where *X* is a Don't Care symbol. A service of *X*'s in the label of a substar implies all permutations on the digits not appearing in the label. The first position (i.e. the leftmost one) of the label of any substar always equals to *X* because of the connectivity conditions of the star graph, unless the substar is a single node (i.e. S_0). Notably, the number of *X* symbols in the string determines the dimension of the substar. Specifically, an *m*-dimensional substar, S_m, has exactly *m* *X*'s in its symbol

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

representation, as it involves a group of $m!$ nodes. For example, $X3X1X$, represents a 3-star formed by the set of nodes $\{23415, 23514, 43512, 43215, 53214, 53412\}$.

References

[1] International Technology Roadmap for Semiconductors (ITRS) 2003 Documents, http://www.itrs.net/Links/2006Update/FinalToPost/00_ExecSum2006Update.pdf.

[2] S. B. Akers and B. Krishnamurthy, “A group-theoretic model for symmetric interconnection networks,” in *Proc. of International Conference on Parallel Processing*, 1986, pp.216-223.

[3] S. B. Akers and B. Krishnamurthy, “The fault tolerance of star networks,” in *Proc. of the 2nd International Conference on Supercomputing*, 1987, pp.270-276.

[4] S. B. Akers, D. Horel, and B. Krishnamurthy, “The star network: An attractive alternative to the n-cube,” in *Proc. of International Conference on Parallel Processing*, 1987, pp.393-400.

[5] N. Bagherzadeh, N. Nassif, and S. Latifi, “A routing and broadcasting scheme on faulty star graphs,” *IEEE Transaction on Computers*, vol.42, no.11, Nov. 1993, pp.1398-1403.

[6] N. Bagherzadeh, M. Dowd, and N. Nassif, “Embedding an arbitrary binary tree into star graph,” *IEEE Transactions on Computers*, vol.45, no.4, 1996, pp.475-481.

[7] R. Billinton and R. Allan, “Reliability evaluation of engineering systems,” 2nd Edition, Plenum Press, 1992.

[8] Y. Chang and L. Bhuyan, “A combinatorial analysis of subcube reliability in hypercube,” *IEEE Transactions on Computers*, vol.44, no.7, July 1995, pp.952-956.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[9] E. Cheng and L. Liptak, "Linearly many faults in Cayley graphs generated by transposition trees," in press, *Information Science*, 2007.

[10] C. Das and J. Kim, "A unified task-based dependability model for hypercube computers," *IEEE Transactions on Parallel and Distributed Systems*, vol.3, no.3, May 1992, pp.312-324.

[11] R.K. Das, "Adaptive fault tolerant routing in star graph," *Distributed Computing-IWDC 2004*, vol.3326, pp.385-390.

[12] K. Day and A. Tripathi, "A comparative study of topologies properties of hypercubes and star networks," *IEEE Transactions on Parallel and Distributed Systems*, vol.5, no.1, Jan. 1994, pp.31-38.

[13] K. Day and A.E. Al-Ayyoub, "Reliable communication in faulty star networks," in *Proc. of the International Parallel and Distributed Processing Symposium (IPDPS'02)*, pp.260-266.

[14] T.Y. Feng, "A survey of interconnection networks," *IEEE Transactions on Computers*, vol.14, no.12, Dec. 1981, pp.12-27.

[15] J.S. Fu, "Conditional Fault-Tolerant Hamiltonicity of Star Graphs, *in Proc. of 7th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT),* 2006, pp.11-16.

[16] S. Fujita, "A fault-tolerant broadcast scheme in the star graph under the single-port, half-duplex communication model," *IEEE Transactions on Computers*, vol.48, no.10, Oct. 1999, pp.1123-1126.

[17] S. Y. Hsieh, "Embedding longest fault-free paths onto star graphs with more vertex faults," *Theoretical Computer Science*, vol.337, 2005, pp.370-378.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[18] S. C. Hu and C. B. Yang, "Fault tolerance on star graphs," in *Proc. of First Aizu International Symposium on Parallel Algorithms/Architecture Synthesis*, March 1995, pp.176-182.

[19] K. Kwang and F.A. Briggs, "Computer architecture and parallel processing," *McGraw-Hill*, New York, 1984.

[20] N. Imani, H. S. Azad, and S. G. Akl, "On some combinatorial properties of star graph," in *Proc. of 8th International Symposium on Parallel Architectures, Algorithms and Networks*, 2005, pp.58-65.

[21] A. E. Kiasari and H. Sarbazi-Azad, "A comparative performance analysis of n-cubes and star graph," in *Proc. 20th International Parallel and Distributed Processing Symposium*, 2006, pp.25-29.

[22] S. Latifi, "Parallel dimension permutations on star network," *IFIP Trans. A: Comp. Sci. Tech.*, A23, 1993, pp.191-201.

[23] S. Latifi, "On the fault-diameter of the star network," *Information Processing Letters*, vol.46, no.3, June 1993, pp.143-150.

[24] S. Latifi, "On embedding rings into a star-related network," *Information Sciences*, vol.99, 1997, pp.21-35.

[25] S. Latifi, "A study of fault tolerance in star graph," *Information Processing Letters*, vol.102, no.5, May 2007, pp.196-200.

[26] T. K. Li, J. M. Tan, and L. Hsu, "Hyper Hamiltonian laceability on edge fault star graph," *Information Processing Letters*, vol.46, no.3, June 1993, pp.143-150.

[27] T.K. Li, "Cycle embedding in star graphs with edge faults," *Applied Mathematics and Computation*, vol.167, 2005, pp.891-900.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[28] J. Misic and Z. Jovanovic, "Routing function and deadlock avoidance in a star graph interconnection network," *Journal of Parallel and Distributed Computing*, vol.22, 1994, pp.216-228.

[29] N. Nassif and N. Bagherzadeh, "A grid embedding into the star graph for image analysis solutions," *Information Processing Letters*, vol.60, no.5, Dec. 1996, pp.255-260.

[30] F.P. Preparata and J. Vuillemin, "The cube-connected cycles: a versatile network for parallel computation," *Communication of ACM*, vol.5, 1981, pp.300-309.

[31] A.A. Rescigno, "Vertex-disjoint spanning trees of the star network with applications to fault-tolerance and security," Information Science, vol.137, 2001, pp.259-276.

[32] S. M. Rezazad and H. Sabazi-Azad, "Fault-tolerant routing in the star graph," in *Proc. of the 18th International Conference on Advanced Information Networking and Application*, vol.2, 2004, pp.503-506.

[33] Y. Saad and M.H. Schultz, "Topological properties of hypercubes," *IEEE Transactions on Computers*, vol.37, July 1988, pp.867-872.

[34] H.S. Stone, "Parallel processing with the perfect shuffle," *IEEE Transactions on Computers*, vol.C-20, 1971, pp.153-161.

[35] S. Soh, S. Rai, and J.L. Trahan, "Improved lower bounds on the reliability of hypercube architectures," *IEEE Transactions on Parallel and Distributed Systems*, vol.5, no.4, April 1994, pp.364-378.

[36] S. Sur and P. K. Srimani, "Topological properties of star graph," *Computers & Mathematics with Applications*, vol.25, no.12, 1993, pp.87-98.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[37] J.D. Ullman, "Computational aspects of VLSI," *Computer Science Press*, Rockville, MD, 1984.

[38] X.L. Wu, S. Latifi, and Y. Jiang, "A combinatorial analysis of distance reliability in star network," in *Proc. of 21st IEEE International Parallel & Distributed Processing Symposium*, 2007.

[39] X.L. Wu, S. Latifi, and Y. Jiang, "Markov reliability modeling of star networks," *in Proc. of 2007 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'07).*

[40] X.L. Wu, S. Latifi, and Y. Jiang, "Robustness assessment of star graph," submitted to *Journal of Information Science and Engineering*," 2006.

[41] M. Xu, X.D. Hu, and Q. Zhu, "Edge-bipancyclicity of star graphs under edge-fault tolerant," *Applied Mathematics and Computation*, vol.183, 2006, pp.972-979.

[42] J. Zheng, S. Latifi, E. Regentova, K. Luo, and X.L. Wu, "Diagnosability of star graphs under the comparison diagnosis model," *Information Processing Letters*, vol.93, no.1, Jan.2005, pp.29-36.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER 2

RELIABILITY AND EVALUATION

A fault-tolerant network is one that can continue to correctly perform its specified tasks in the presence of failures which could be in forms of node failures, link failures, or combined node/link failures. The most prominent requirements to achieve a fault-tolerant network are reliability, availability, safety, performability, dependability, maintainability, and testability.

As the size of a system grows, the probability of a fault occurring in the network increases. It is important to quantify the effect of the faults, so the fault-tolerant network can be pursued. Normally, reliability is used to evaluate a multiprocessor interconnection network. The *reliability* $R(t)$ of a system [1] is a function of time, defined as the conditional probability that the system will perform correctly throughout $[t_0, t]$, given that the system was performing correctly at time t_0. In other words, the reliability is the probability that the system will operate correctly throughout a complete interval of time. The *unreliability* $R(t)$ of a system is a function of time, defined as the conditional probability that the system will perform incorrectly throughout $[t_0, t]$, given that the system was performing correctly at time t_0 [2]. The unreliability is often referred to as the probability of failure. Fault tolerance is a technique that can improve reliability, but a fault-tolerant system does not necessarily have a high reliability. In other words, the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

system can achieve a high reliability but not possess attributes of the fault tolerance. Before we introduce different reliability measures and reliability evaluation models, a brief description about some fundamental terms is given first.

2.1 Fault, error, and failure

Three fundamental terms in the fault-tolerant design are *fault*, *error*, and *failure* [6]. There is a cause-and-effect relationship between faults, errors, and failure. Specifically, faults are the causes of errors, and errors are the cause of failures.

A *fault* is a physical defect, imperfection, or flaw that occurs within some hardware or software components. Essentially, the definition of a fault, as used in the fault tolerance community, agrees with the definition found in the dictionary. A fault is a blemish, weakness, or shortcoming of a particular hardware or software component. Examples of faults include shorts between electrical conductors, open or breaks in conductors, or physical flaws or imperfections in semiconductor devices. Similarly, in software, an example of a fault is a program loop that when entered can never be exited.

An *error* is the manifestation of a fault. Specifically, an error is a deviation from accuracy or correctness. For example, suppose that a physical short results in a line within a circuit being permanently stuck at logic 1. The physical short is a fault within the circuit. If some condition occurs that requires the line to transition to logic 0, the value on the line will be in error. In other words, the correct value for the line will be logic 0, but the existence of the fault has caused the line to have an erroneous value. In other words, an error is the result of a fault.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Finally, if the error results in the system performing one of its functions incorrectly, a system *failure* has occurred. Essentially, a failure is the non-performance of some action that is due or expected. A failure is also the performance of some function in a subnormal quantity or quality. As an example, suppose that a line in a circuit is responsible for turning a value on or off: logic 1 turns the value on and logic 0 turns the value off. If the line is stuck at logic 1, the value is stuck on. As long as the user of the system wants the system the value one, the system will be functioning correctly. However, when the user wants to turn the value off, the system will experience a failure.

Fig. 2.1 illustrates the cause-and-effect relationship between faults, errors, and failures. Faults result in errors, and errors can lead to system failures. One way to think of Fig. 2,1 is as a hierarchy. At the bottom of the hierarchy are faults. Errors are the effect of faults, and finally, failures are the effect of errors.

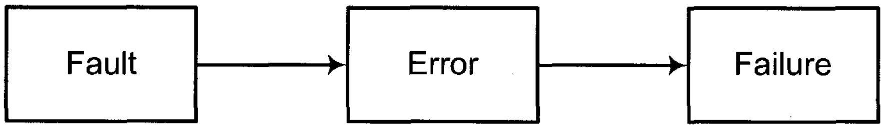


Figure 2.1. Relationship between faults, errors, and failures.

Faults can be the results of a variety of things that occur within electronic components, external to the components, or during the component or system design process. It is very important to understand all the possible causes of faults. Possible fault causes can be associated with problems in four basic areas: *specifications*, *implementation*, *components*, and *external factors*. The first cause of faults is the possibility of *specification* mistakes, including incorrect algorithms, architectures, or

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

hardware and software design specifications. The next cause of faults is *implementation* mistakes, defined as faults during the process of transforming hardware and software specifications into the physical hardware and the actual software. The next cause is *component defects*, defined as manufacturing imperfections, random device defects, and component wear-out, etc. The final cause of faults is the *external disturbance*, for example, radiation, electromagnetic interference, battle damage, operator mistakes, and environmental extremes.

To adequately describe faults, however, characteristics other than the causes are required. In addition to the causes, four major attributes are critical to the description of faults: *nature*, *duration*, *extent*, and *value*.

The fault nature specifies the type of faults. This can specified using terms such as hardware, software, analog, and digital. The fault duration specifies that length of time that a fault is active. And this could be permanent, transient, or intermittent. The fault extent whether the fault is localized to a given hardware or software or whether it globally affects the hardware, the software, or both. The fault value can either determinate or indeterminate. A determinate fault is one whose status remains unchanged throughout time unless externally acted upon. Meanwhile, an indeterminate fault is one whose status at some time *T* may be different from its status at some other time *T'*.

2.2 Quantitative evaluation methods

The methods for evaluating fault-tolerant networks can be divided into two major categories: *quantitative* and *qualitative*. *Qualitative* measures are typically subjective in nature and describe the benefits of one network over another. *Quantitative* evaluation

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

techniques produce numbers that can be used to compare two or more systems. Usually we discuss methods of evaluation that generate specific numbers to compare two or more systems. Often, we find that certain attributes of a system that enter the design process are extremely difficult to quantity. And in practice, major decisions are often using more qualitative information than quantitative. These qualitative comparisons are *flexibility*, *technology dependence*, *transparency to the user*, and *testability*. The flexibility of a system is referred as the ability to expand and improve as customer needs change and technological advance occur. Technology dependence is easy to understand as the ability of the system capable of adapting itself to the new developed technology. Easy to understand, to operate, to test the system are also important concerns for qualitatively evaluating a fault-tolerant network. However, these qualitative evaluations are extremely difficult to determine and out of the scope of this dissertation. Therefore, we only stick to the quantitative evaluation techniques throughout the dissertation.

The purpose of *quantitative* evaluation is to assign a number to some attribute of a system such that this attribute can be compared among systems. For example, the reliability of one system may be greater than that of another. Next we introduce several quantitative evaluation techniques [6] , including *the failure rate, reliability, mean time to failure (MTTF), mean time to repair (MTTR), mean time between failures (MTBF),*.

- Failure rate and reliability function

Intuitively, *failure rate is the expected number of failures of a type of device or system per a given time period.* For example, if a computer fails, on the average, once every 2000 hours, the computer has a failure rate of one failure per 2000 hours, or 1/2000 failures/hour. The failure rate is typically denotes as λ. The failure rate is one measure

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

that can be used to compare systems or components. In selecting a banking application, one would like to select a computer that fails as infrequently as possible.

The *reliability $R(t)$ of a system is a function of time, defined as the conditional probability that the system will perform correctly throughout the interval $[t_0,t]$, given that the system was performing correctly at time t_0*. Suppose that we test N identical components by placing all N components in operation at time t_0 and recording the number of failed and working components at time t. Let $N_f(t)$ be the number of components that have failed at time t and $N_o(t)$ be the number of components that are operating correctly at time t. It is assumed that once a component fails it remains failed indefinitely. The *reliability* of the components at time t is given by

$$R(t)=\frac{N_o(t)}{N}=\frac{N_o(t)}{N_o(t)+N_f(t)},$$

which is simply the probability that a component has survived the interval $[t_0,t]$. The probability that a component has not survived the time interval is called the *unreliability* and is given by

$$Q(t)=\frac{N_f(t)}{N}=\frac{N_f(t)}{N_o(t)+N_f(t)}.$$

Note that at any time *t*, $R(t)=1.0-Q(t)$ because $R(t)+Q(t)=\frac{N_o(t)+N_f(t)}{N_o(t)+N_f(t)}=1.0$.

If we rewrite the reliability function and differentiate $R(t)$ with respect to time, we obtain $\frac{dN_f(t)}{dt}=-N\frac{dR(t)}{dt}$. The derivative of $N_f(t)$, $dN_f(t)/dt$, is simply the instantaneous rate at which components are failing. At time *t*, there are still $N_o(t)$ components operational. Dividing $dN_f(t)/dt$ by $N_o(t)$ we obtain

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$z(t) = \frac{1}{N_o(t)} \frac{dN_f(t)}{dt}.$$

$z(t)$ is called the *hazard function, hazard rate,* or *failure rate function.* The failure units for the failure rate function are failures per unit of time. The failure rate function can be expressed in different ways. For example, $z(t)$ can be written strictly in terms of the reliability function $R(t)$ as

$$z(t) = \frac{1}{N_o(t)} \frac{dN_f(t)}{dt} = \frac{1}{N_o(t)} \left[-N \frac{dR(t)}{dt} \right] = -\frac{\frac{dR(t)}{dt}}{R(t)}.$$

Similarly, $z(t)$ can also be written in terms of the unreliability $Q(t)$ as

$$z(t) = -\frac{\frac{dR(t)}{dt}}{R(t)} = \frac{\frac{dQ(t)}{dt}}{1-Q(t)}.$$

The derivative of the unreliability $dQ(t)/dt$ is called the *failure density function.*

If we assume that the system has a given constant failure rate λ, the solution to the above differential equation is well known to be an exponential function given by $R(t) = e^{-\lambda t}$. This exponential relationship between the reliability and time is known as the *exponential failure law,* which states that for a constant failure rate function, the reliability varies exponentially as a function of time.

The exponential failure law is extremely valuable for the analysis of electronic components and is by far the most commonly used relationship between reliability and time. Many cases, however, cannot assume that the failure rate function is constant, so the exponential failure law cannot be used; other modeling schemes and representations must be employed. An example of the time-varying failure rate function is found in the analysis of software. Software failures are the result of design faults, and as a software package is used, design faults are discovered and corrected. Consequently, the reliability

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

of software should improve as a function of time, and the failure function should decrease.

A common modeling technique used to represent time-varying failure rate functions is the *Weilbull distribution.* The failure rate function associated with the Weibull distribution is given by $z(t) = \alpha\lambda(\lambda t)^{\alpha-1}$, where α and λ are constants that control the variation of the failure rate function with time. The failure rate function given by the Weibull distribution is intuitively appealing. For example, if the value of α is 1, $z(t)$ is simply the constant λ. If α is greater than 1, $z(t)$ increases as time increases; if α is less than 1, $z(t)$ decreases as time increases. Consequently, we can envision modeling software using the Weibull distribution with the constant α being less than 1. The reliability function that results from the Weibull distribution is given as $R(t) = e^{-(\lambda t)\alpha}$.

Although time-varying failure rate functions are important in the analysis of software and other systems, by far the most common analysis is performed assuming a constant failure rate function and the exponential failure law. Thus we continue to use the exponential failure distribution for the remaining of this dissertation.

- Mean time to failure

In addition to the failure rate, the *mean time to failure (MTTF)* is a useful parameter to specify the quality of a system. *The MTTF is the expected time that a system will operate before the first failure occurs.* For example, if we have *N* identical system placed into operation at time $t = 0$, and we measure the time that each system operates before failing, the average time is the MTTF. If each system *i* operates for a time t_i before encountering the first failure, the MTTF is given by $MTTF = \Sigma_{i=1}^{N} t_i / N$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The MTTF can be calculated by finding the expected value of the time of failure. From the probability theory, we know that the expected value of a random variable X is $E[X]=\int_{-\infty}^{\infty} xf(x)dx$, where $f(x)$ is the probability density function. In reliability analysis we are interested in the expected value of the time of failure (MTTF), so $MTTF=\int_{0}^{\infty} tf(t)dt$, where $f(t)$ is the failure density function, and the integral runs from 0 to ∞ because the failure density function is undefined for times less than 0. Using integration by parts and the fact that $f(t)=-dR(t)/dt$, we can show that

$$MTTF=-\int_{0}^{\infty} t\tfrac{dR(t)}{dt}dt=[-tR(t)+\int_{0}^{\infty} R(t)dt]_{0}^{\infty}=\int_{0}^{\infty} R(t)dt$$

The term $-tR(t)$ clearly disappears when $t=0$; but, it also disappears when $t=\infty$ because $R(\infty)=0$. Consequently, the MTTF is defined in terms of the reliability function as $MTTF=\int_{0}^{\infty} R(t)dt$, which is valid for any reliability function that satisfies $R(\infty)=0$. If the reliability function obeys the exponential failure law, the result of calculating the MTTF is given by $MTTF=\int_{0}^{\infty} e^{-\lambda t}dt=\frac{1}{\lambda}$. In other words, the MTTF of a system that obeys the exponential failure law is the inverse of the failure rate of the system.

- Mean time to repair

The *mean time to repair (MTTR) is simply the average time required to repair a system.* The MTTR is extremely difficult to estimate and is often determined experimentally by injecting a set of faults, one at a time, into a system and measuring the time required to repair the system in each case. The measured repair times are averaged to determine an average time to repair. The MTTR is normally specified in terms of a repair rate μ, which is the average number of repairs that occur per time period. The

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

units of the repair rate are normally the number of repairs per hour. The MTTR and the repair rate μ are related by $MTTR = \frac{1}{\mu}$.

- Mean time between failure

The *mean time between failures (MTBF) is the average time between failures of a system.* If we assume that all repairs to a system make the system perfect once again until it was when it was new, the relationship between the MTTF and the MTBF is as illustrated in Fig. 2.2. Once successfully placed into operation, a system operates, on the average, a time corresponding to the MTTF before encountering the first failure. The system then requires some time, MTTR, to repair the system and place it back into operation once again. The system then is perfect once again and will operate for a time corresponding to the MTTF before encountering its next failure. The time between the two failures is the sum of MTTF and MTTR and is the MTBF. Thus, the difference between the MTTF and the MTBF is the MTTR. Specifically, the MTBF is given by MTBT = MTTF + MTTR. In most practical applications the MTTR is a small fraction of the MTTF, so the approximation that the MTBF and MTTF are equal is often quite good.

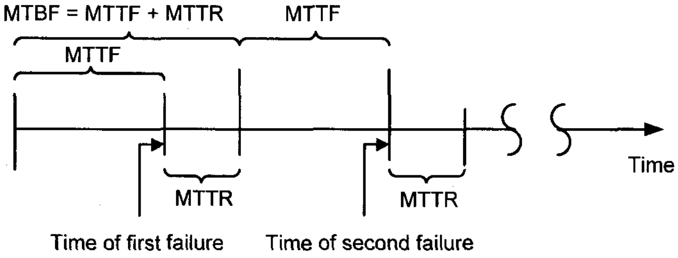


Figure 2.2. Relationship between the MTBF, MTBR, and MTTF.

2.3 Performance measures

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

An undirected network $G=(V,E)$ is usually modeled as either a deterministic network or a probabilistic network [8]. In *deterministic networks* it is considered that working elements can be successfully attached by an adversary, resulting in their failure or inactivation. The failure of an edge means that it is removed from the network, while the failure of a node means that the node and all its incident edges are removed from the network. In deterministic network models the focus is typically on evaluating the worst-case performance of the network, in which the adversary intelligently choose certain elements to inactivate, resulting in the maximum damage to the network. This type of model thus provides a conservative assessment of performance, and it would be particularly appropriate in the design of robust military systems.

By contrast, in *probabilistic networks* it is usually assumed that, at any instant, elements fail randomly and independently of one another, according to certain known probabilities. Specifically, each node *i* has an associated reliability p_i indicating the probability that it is operational, and each edge *l* has a reliability p_l, the probability that it is operational. Thus, at any instant the elements of the network fail independently with probabilities $q_i = 1 - p_i$ and $q_l = 1 - p_l$, respectively. In these circumstances, one would be interested in assessing the average performance of the network under the random failures. Thus, unless announced, most of our reliability analyses are based on the probabilistic network models.

For the case of probabilistic networks (in which nodes/links fail randomly and independently with known probabilities), a number of measures have been explored. Suppose *G* is directly, with *s* and *t* being two distinguished nodes of *G*. A traditional measure of the reliability evaluation is the *terminal reliability* [8][9][10], such as two-

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

terminal reliability, distance reliability, *k*-terminal reliability, and all terminal reliability. The *two-terminal reliability* $R_{st}(G)$ is the probability that *s* and *t* are connected by a path of operating edges and nodes in *G*. The *distance reliability* (*DR*) is the probability of having an operational path with the shortest distance between two given arbitrary nodes *s* and *t*. The *source-to-all-terminal reliability* $R(G)$ is the probability that there is an operative path from node *s* to all other nodes of the network. The all-terminal reliability would be an appropriate measure when all nodes are of equal importance in receiving a message sent from the source node, whereas the two-terminal reliability would apply when a critical message needs to be routed between specified sites in the network. The distance reliability is especially designed to achieve the efficient communication by only considering the message passing thought those paths of the shortest distance between two given arbitrary nodes *s* and *t*. A generalization of these concepts is embodied in the *source-to-K-terminal reliability* of the network, the probability $R_k(G)$ that there is an operative path from node *s* to all nodes in some specified set $K \subseteq N$. These probabilistic measures have analogous counterparts in the case where *G* is undirected. Notice that for undirected networks the all-terminal reliability simply expresses the probability that *G* remains connected.

One interesting measure is the *task-based reliability* [5], defined as the probability that some minimum number of connected nodes are available in the system for the specific task execution. This task-based reliability is based on the assumption that a system works as long as there is a group of connected working nodes for satisfying the task requirement. However, this measure is extremely difficult to model exactly.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

An alternative probabilistic measure takes into account the fact that different ways of disconnecting a network are of different severity. For example, if G is undirected and connected, then the failure of certain edges and nodes could separate G into several connected components, $G_1, G_2, \cdots, G_r$. All communication is then disrupted between nodes in different components, and the resulting communication capacity can be measured by the number of pairs of node able to communicate: $\sum_{i=1}^{r} \binom{n_i}{2}$, where n_i is the number of nodes in the component G_i. The average number of node pairs able to communicate, taken over all possible nodes and edge failures, thus provides a quite different type of the probabilistic measure, the *pair-connectivity* of G [1].

In Chapter 1, we mentioned some symmetric hierarchical networks have the *recursive decomposition property*, which means that an n-dimensional symmetric network can be recursively decomposed into smaller size networks with the same topological properties as the original one. Bhuyan [4] proposed a new idea of *subcube reliability*, defined as the probability that a subcube of a specific size is available in the system. Since the star network is strongly hierarchical as the hypercube, it is wise to explore the substar reliability in the star interconnection network. A similar idea with the aim of finding the minimum number of failed nodes or links to destroy all available substars has been studied and reported in [7]. Among these reliability measures, the subnetwork reliability is the most practical one because a user in the current multiprocessor network is given a specific subnetwork for the execution of his/her program.

2.4 Reliability modeling methods

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reliability is one of the most important attributes of systems. Almost all specifications for systems mandate that certain values for reliability be achieved and in some way proved. We have seen in Section 2.1 that reliability can be determined experimentally if a set of N system is operated over a period of time and the number of systems that fail during that time period is recorded. One problem with the experimental approach is the *number of systems* that would be required to achieve a level of confidence in the experimental results. This is particularly a problem when costs limit the number of systems that can be built. For example, the space shuttle program could not afford to build 1000 of its on-board processing systems that reliability could be experimentally verified. A second problem with the experimental approach is the *time* required to run such experiments. Many systems today are being designed to achieve reliability of 0.9_7, or higher, after ten hours of operation. Using the exponential failure law, a reliability of 0.9_7 corresponds to a failure rate of 10^{-8} failures per hour. Therefore, on the average, we would have to wait approximately 100 million hours, or approximately 11,416 years for the first failure to occur. Clearly, we need alternatives to the experimental approach.

The most popular reliability analysis techniques are the analytical approaches. Of the analytical technique, *combinatorial modeling* and *Markov modeling* [2][3][6][8] are the two most commonly used approaches.

2.4.1 Combinatorial model

Combinatorial models use probabilistic techniques that enumerate the different ways in which a system can remain operational. The probabilities of the events that lead to a system being operational are calculated to form an estimate of the system's reliability. The reliability of a system is generally derived in terms of reliabilities of the individual

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

components of the system. Two models that are most common in practice are the *series* and the *parallel.* In a series system, each element is required to operate correctly for the system to operate correctly. In a parallel system, on the other hard, only one of several elements must be operational for the system to perform its functions correctly. In practice, systems are typically combinations of series and parallel subsystems.

a. Series systems

The series system is best thought of as a system that contains no redundancy; that is, each element of the system is needed to make the system function correctly. One way of representing the series system is by the aid of *reliability block diagrams.* The reliability block diagram can be thought of a flow diagram from the input of the system to the output of the system. Each element of the system is a block in the diagram and, for the series system, the blocks are placed in series to indicate that a path from the input to the output is broken if one of elements fails.

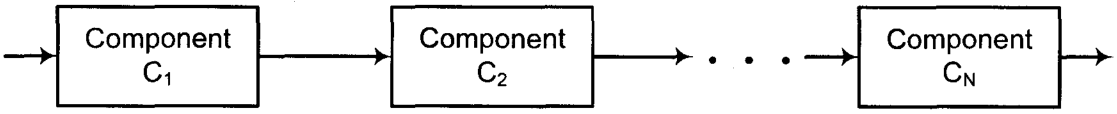


Figure 2.3. The reliability block diagram of a series system.

The generalized reliability block diagram of a series system that contains *N* elements is shown in Fig. 2.3. Each of the *N* elements is required for the system to function correctly. The reliability of the series system can be calculated as the probability that none of the elements will fail. Another way to look at this is that the reliability of the series system is the probability that all of the elements are working properly.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Suppose we let $C_{iw}(t)$ represent the event that component C_i is working properly at time t, $R_i(t)$ is the reliability of component C_i at time t, and $R_{series}(t)$ is the reliability of the series system. Further suppose that the series system contains N series components as shown in Fig. 9. The reliability at any time t is the probability that all N components are working properly. In mathematical terms,

$$R_{series}(t) = P(C_{1w}(t) \cap C_{2w}(t) \cap \cdots \cap C_{Nw}(t))$$

Assuming that each event, $C_{iw}(t)$, is independent, we have

$$R_{series}(t) = R_1(t)R_2(t)\cdots R_N(t) \text{ or } R_{series}(t) = \prod_{i=1}^{N} R_i(t)$$

An interesting relationship exists in a series system if each individual component satisfies the exponential failure law such that the reliability of each component is $R_i(t) = e^{-\lambda_i t}$. Suppose that we have a series system made up of N components, and each component i has a constant failure rate λ_i. The reliability of the series system is given by

$$R_{series}(t) = e^{-\lambda_1 t} e^{-\lambda_2 t} \cdots e^{-\lambda_N t} \text{ or } R_{series}(t) = e^{-\Sigma_{i=1}^{N} \lambda_i t} = e^{-\lambda_{system} t}$$

where $\lambda_{system} = \Sigma_{i=1}^{N} \lambda_i$ and corresponds to the failure rate of the system. In other words, the failure rate of a series system can be calculated by adding the failure rates of all the components that make up the series system. Thus, the value of the MTTF in the series system assuming that each component satisfies the exponential failure law can be obtained as $MTTF_{series} = \frac{1}{\Sigma_{i=1}^{N} \lambda_i}$.

b. Parallel systems

The basic feature of the parallel system is that only one of N elements is required for the system to function. The reliability block diagram of the basic parallel system that

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

contains N elements is shown in Fig.2.4. As can be seen, a path exists in the reliability block diagram from input to output as long as one of the N elements remains operational. The unreliability of the parallel system can be computed as the probability that all of the N elements fail at the same time. Suppose that we let $C_{if}(t)$ represent the event that component i has failed at time t, $Q_i(t)$ be the unreliability of the i^{th} element, and $Q_{parallel}(t)$ is the unreliability of the parallel system. $Q_{parallel}(t)$ can be computed as $Q_{parallel}(t) = P(C_{1f}(t) \cap C_{2f}(t) \cap \cdots \cap C_{Nf}(t))$, or if we assume that each event, $C_{if}(t)$, is independent, we have $Q_{parallel}(t) = Q_1(t)Q_2(t)\cdots Q_N(t) = \prod_{i=1}^{N} Q_i(t)$.

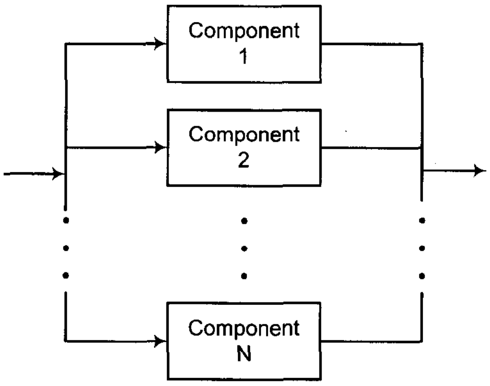


Figure 2.4. The reliability block diagram of the parallel system.

The reliability of the parallel system can now be computed because we know that the reliability and the unreliability must add to 1.0. Mathematically, we must have $R(t) + Q(t) = 1.0$ for any system. Consequently, we can write

$$R_{parallel}(t) = 1 - Q_{parallel}(t) = 1.0 - \prod_{i=1}^{N} Q_i(t) = 1.0 - \prod_{i=1}^{N} (1.0 - R_i(t)).$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

If each component i in the parallel system has a constant failure rate of λ_i, also assume that each component satisfies the exponential failure law such that the reliability of each component is $R_i(t) = e^{-\lambda_i t}$. The reliability of the parallel system is given by

$$R_{parallel}(t) = 1 - Q_{parallel}(t) = 1.0 - \prod_{i=1}^{N} (1.0 - R_i(t)) = 1.0 - \prod_{i=1}^{N} (1.0 - e^{-\lambda_i t})$$

$$or\ \Sigma_{i=1}^{N} e^{-\lambda_i t} - \Sigma_{\substack{i,j=1 \\ i \neq j}}^{N} {}^{-\lambda_i t} e^{-\lambda_j t} + \Sigma_{\substack{i,j,k=1 \\ i \neq j \neq k}}^{N} {}^{-\lambda_i t} e^{-\lambda_j t} e^{-\lambda_k t} + \cdots + (-1)^{N-1} \Pi_{i=1}^{N} e^{-\lambda_i t} \cdot$$

where $\lambda_{system} = \Sigma_{i=1}^{N} \frac{1}{\lambda_i} - \Sigma_{\substack{i,j=1 \\ i \neq j}}^{N} \frac{1}{\lambda_i + \lambda_j} + \Sigma_{\substack{i,j,k=1 \\ i \neq j \neq k}}^{N} \frac{1}{\lambda_i + \lambda_j + \lambda_k} + \cdots + (-1)^{N-1} \Pi_{i=1}^{N} \frac{1}{\lambda_i}$ and corresponds to the failure rate of the parallel system. Thus, the value of the MTTF in the parallel system assuming that each component satisfies the exponential failure law can be obtained as $MTTF_{parallel} = 1 / \lambda_{system}$.

c. Series-parallel or parallel-series system

The series and parallel systems discussed in the previous sections form the basis for the analysis of more complex configurations. The general principle used is to reduce sequentially the complex configuration by combining appropriate series and parallel braches of the reliability model until a single equivalent element remains. This equivalent element then represents the reliability of the original configuration.

d. *r*-out-of-*N* systems

r-our-of-*N* systems are a generalization of the ideal parallel system. In the ideal parallel system, only one of N modules is required to work for the system to work. In the *r*-out-of-*N* system, however, at least r of the total of N identical modules are required to function for the system to function properly, and the system can tolerate at

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

most $N-r$ module failures. The expression for the reliability of an r-out-of-N system can be written as (assuming each module has the same reliability R)

$$R_{r-out-of-N}(t)=\sum_{i=r}^{N}\binom{N}{i}R^{i}(1-R)^{N-i}, \text{ where } \binom{N}{i}=\frac{N!}{(N-i)!i!}.$$

Reliability evaluation techniques described so far are limited in their applications to networks having a series or parallel type of structures. Many systems either do not have this simple type of structure or have complex operational logic. Additional modeling and evaluation techniques are necessary in order to determine the reliability of such systems. Most of these more advanced techniques are formalized methods for transforming the logic operation of the system, or the topology of the system, into a structure that consists only of series and parallel components, paths or braches. Next we continue to introduce techniques used for the reliability analysis in complex systems.

e. Conditional probability approach

One approach which can be used to evaluate the reliability of a complex system is to reduce sequentially the system into subsystems structures that are connected in series/parallel and then to recombine these subsystems using the *conditional probability method*. The reliability under this approach can use the following equation

$$\begin{aligned}P(\text{system success}) &= P(\text{system success if component X is good})\cdot P(\text{X is good})\\ &\quad + P(\text{system success if component X is bad})\cdot P(\text{X is bad})\end{aligned}$$

The conditional probability approach is efficient to solve the bridge-type network in Fig. 2.5. However, in some engineering systems further subdivision before a series/parallel structure is needed. This is only an extension of the technique being discussed since each time a subdivision is made; the two subdivisions must recombined using the conditional probability approach starting with the two most recent subdivided

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

subsystems, i.e., the lowest hierarchical level. After creating a set of subsystems in which all components are connected in series and parallel, the subsystems can be evaluated using the principles of series and parallel systems discussed before and the overall system reliability evaluated using the conditional reliability approach. The conditional probability approach is a useful tool for reliability evaluation and is frequently used in many applications.

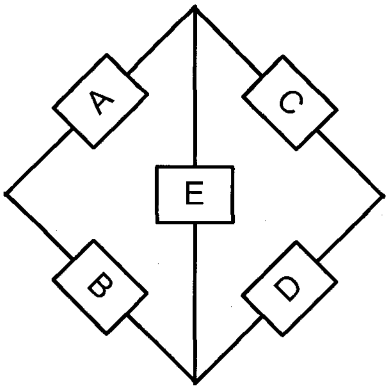


Figure 2.5. Example of a bridge-type network.

f. Minpaths: inclusion and exclusion

In the case of the two terminal reliability, minpaths [2] are paths with the minimum traveled links between the source node and the destination node. Suppose then that the minpaths $P_1, P_2, \cdots, P_h$ of a given graph G have been listed. Let E_i be the event that minpath P_i is operational, and let $\Pr[\ ]$ denotes the reliability of an event. Then the reliability is just the probability that one (or more) of the events $\{E_i\}$ occurs. Unfortunately, the $\{E_i\}$s are not disjoint events, and hence we cannot simply sum their

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

probabilities of occurrence. To be specific, $\Pr[E_1 \text{ or } E_2]$ is $\Pr[E_1]+\Pr[E_2]$ - $\Pr[E_1 \text{ and } E_2]$. Now $\mathrm{Re}l(G)=\Pr[E_1 \text{ or } E_2 \text{ or} \cdots \text{ or } E_h]$, and hence

$$\mathrm{Re}l(G) = \sum_{j=1}^{h} (-1)^{j+1} \sum_{\substack{I \subseteq \{1,\cdots,h\} \\ |I|=j}} [E_I].$$

where E_I is the event that all paths P_i with $i \in I$ are operational. This is a standard inclusion-exclusion expansion.

g. Using minpaths: disjoint products

Let us once again suppose that we have an enumeration $P_1, P_2, \cdots, P_h$ of the minpaths, and let E_i be the event that all nodes/edges in minpath P_i are operational. As we have remarked, the events $\{E_i\}$ are not disjoint; we pursue the strategy here of forming a set of disjoint events. Let $\overline{E_i}$ denotes the complement of event E_i ; now define the event $D_1 = E_1$, and in general, $D_i = \overline{E}_1 \cap \overline{E}_2 \cap \cdots \cap \overline{E}_{i-1} \cap E_i$. The events D_i are disjoint, and hence are often called *"disjoint product" events*. Moreover, we can get the reliability as a Boolean expression as follows (we use (+) to denote "or", and times (·) or simply concatenation to represent "and"):

$$\begin{aligned}\mathrm{Re}l(G) &= \sum_{i=1}^{h} \Pr[D_i] \\ &= \Pr[E_1] + \Pr[\overline{E}_1 E_2] + \Pr[\overline{E}_1 \overline{E}_2 E_3] + \cdots + \Pr[\overline{E}_1 \overline{E}_2 \cdots \overline{E}_{h-1} E_h]\end{aligned}$$

In employing this approach, one must obtain a formula for $\Pr[D_i]$ in terms of the states of the nodes/edges. Each event E_i can be written as a Boolean expression which is the product of the states of the nodes/edges in the minpath P_i. Hence D_i can also be

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

written as a Boolean expression. For this reason, methods using disjoint products are sometimes called "*Boolean algebra" methods*.

h. Cut set method

The cut set method is a powerful one for evaluating the reliability of a system for two main reasons:

(i) It can be easily programmed on a digital computer for the fast and efficient solution of any general network.

(ii) The cut sets are directly related to the modes of system failure and therefore identify the distinct and discrete ways in which a system may fail.

A cut set can be defined as follows: *A cut set is a set of system components which, when failed, causes failure of the system.* In terms of a reliability network or block diagram, the above definition can be interpreted as a set of components which must fail in order to disrupt all paths between the input and the output of the reliability network.

The minimum subset of any given set of components which causes the system failure is known as a *minimal cut set*, defined as follows: *A minimal cut set is a set of system components which, when failed, causes failure of the system but when any one component of the set has not failed, does not cause the system failure.* This definition means that *all* components of a minimal cut set must be in the failure state to cause the system failure.

In order to evaluate the system reliability (or unreliability), the minimal cut sets identified from the reliability network must be combined. From the definition of the minimal cut sets it is evident that all components of each cut must fail in order for the system to fail. Consequently, the components of the cut set are effectively connected in parallel and the failure probabilities of the components in the cut set may be combined

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

using the principles of parallel systems. In addition, the system fails if any one of the cut sets occurs and consequently each cut is effectively in series with all the other cuts.

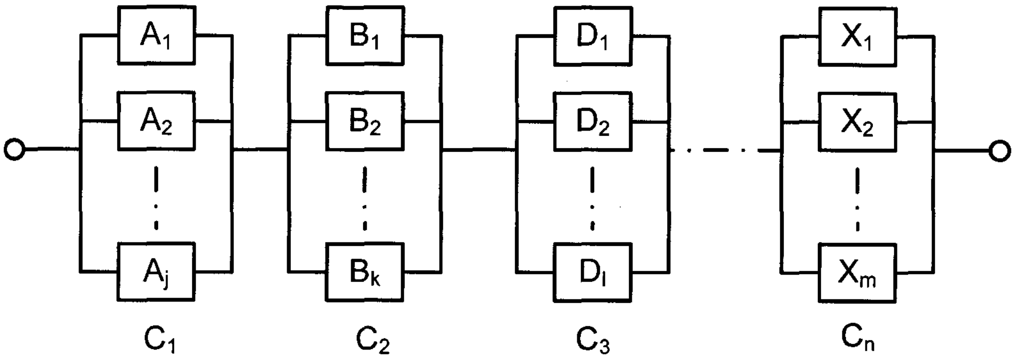


Figure 2.6. Reliability diagram using the minimum cut set.

The use of this principle gives the reliability diagram (Figure 2.6) for a general network. Although these cut sets are in series, the concept of series system cannot be used because the same component may appear in two or more of the cut sets. The concept of union does apply however and if the i^{th} cut is designed as C_i and its probability of occurrence is designated as $P(C_i)$, then the unreliability of the system is given by

$$
\begin{aligned}
Q_s &= P(C_1 \cup C_2 \cup C_3 \cup \cdots \cup C_i \cup \cdots \cup C_n) \\
&= \sum_{i=1}^{n} P(C_i) - \sum_{\substack{i,j=1 \\ i \neq j}}^{n} P(C_i \cap C_j) + \cdots + (-1)^{n-1} P(C_1 \cap C_2 \cap \cdots \cap C_n)
\end{aligned}
$$

i. Tie set method

The tie set method is essentially the complement of the cut set method. It is used less frequently, in practice, as it does not directly identify the failure modes of the system. *A tie set is a minimal path of the system and is therefore a set of system components connected in series.* Consequently, a tie set fails if any one of the components in it fails

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and this probability can be evaluated using the principle of series system. For the system to fail, however, all of the tie sets must fail and therefore all tie sets are effectively connected in parallel. A tie set diagram using these concepts is presented in Fig. 2.7.

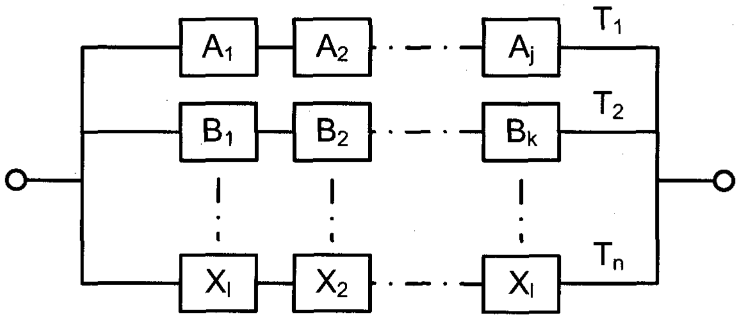


Figure 2.7. Reliability diagram using the tie set method.

Although the tie sets are in parallel, the concept of parallel systems cannot be used because the same component can appear in two or more of the tie sets. The concept of union does apply however in a similar manner to that discussed for minimal cut sets. Then the reliability of the system is given as follows:

$$R_s = P(T_1 \cup T_2 \cup T_3 \cup \cdots \cup T_i \cup \cdots \cup T_n)$$
$$= \sum_{i=1}^{n} P(T_i) - \sum_{\substack{i,j=1 \\ i \neq j}}^{n} P(T_i \cap T_j) + \cdots + (-1)^{n-1} P(T_1 \cap T_2 \cap \cdots \cap T_n) \cdot$$

2.3.2 Markov model

The primary difficulty with the combinatorial models is that many complex systems cannot be modeled easily in a combinatorial fashion. The reliability block diagrams can be extremely difficult to construct, and the resulting reliability expressions are often very

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

complex. In addition, the process of repair that occurs in many systems is very difficult to model in a combinatorial fashion. For these reasons, we use *Markov models*.

The two main concepts in the Markov model are *the system state* and *the state transition*. The *state of a system* represents all that must be known to describe the system at any given instant of time. For reliability models, each state of the Markov model represents a distinct combination of faulty and fault-free modules. For example, suppose we have a TMR system with three identical computers in a majority voting arrangement with a perfect voter. We can define the state of this system as $S = (S_1, S_2, S_3)$ where $S_i = 1$ if module i is fault free and $S_i = 0$ if module i is faulty. The TMR system has eight distinct states in which it can operate: $(000), (001), (010), (011), (100), (101), (110)$, and (111). Each state represents a unique combination of faulty and fault-free modules within the system. For TMR, we know that at least two of the modules must be fault free for the system to operate correctly. Therefore, the states $(000), (001), (010)$, and, (100) represent states in which the system has ceased to function correctly. The remaining states are those in which the system is functioning correctly.

The *state transitions* govern the changes of state that occur within a system. As time passes and failures and reconfigurations occur, the system goes from one state to another. For example, if the TMR system starts its operation in state (111) and at some time t module 1 fails, the system transitions to state (110). The state transitions are characterized by probabilities such as the probability of failure and repair.

We use TMR example to study the state transitions. We construct the TMR transitions using following assumptions. First, we assume that the system does not contain repair. In another word, once a module has failed, it remains failed permanently.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Second, we assume that that only one failure will occur at a time. In a TMR system, the single failure assumption implied that the system cannot go directly from the state corresponding to all modules operating correctly to a state that corresponds to the system having failed. Finally, we assume that the system starts in the perfect state (111) where all of the system's modules are operating correctly.

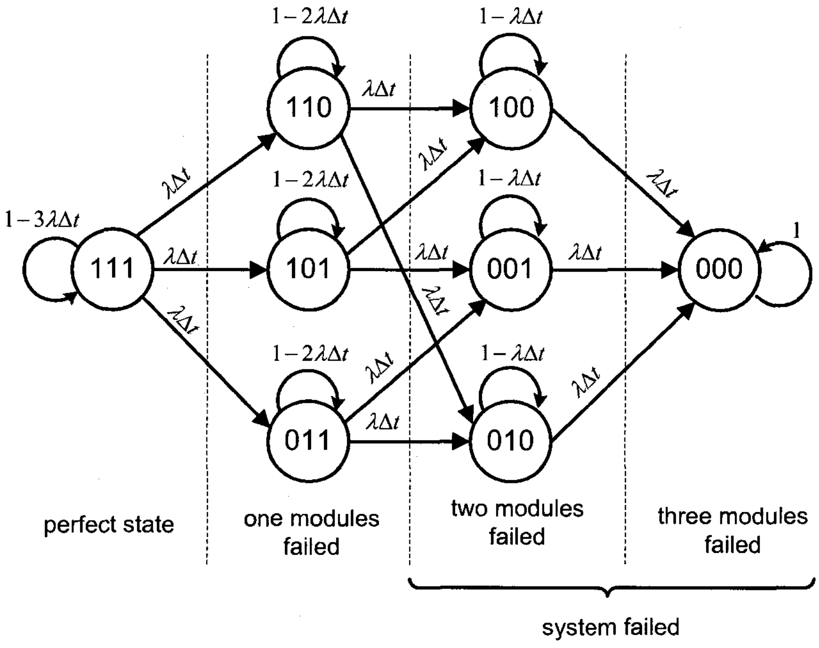


Figure 2.8. Markov model of the TMR system.

The state diagram is shown in Fig. 2.8. As can be seen the system begins in state (111) and, upon the first module failure, transitions to state (110), (101), or (011), depending on whether module 1, 2, or 3 is the module that fails. Note that the transition exists for the module to remain in a state if a module failure does not occur. The states can be partitioned into three categories: the perfect state (111) where all modules function correctly; the one-failed states (110),(101), and (011) where a single module

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

has failed, and the system-failed states (100), (001), (010), and (000), in which certain modules have failed to cause the system to fail.

Each state transition has associated with a transition probability that describes the probability of that state transition occurring within a specified period of time. If we assume that each module in the TMR system obeys the exponential failure law and has a constant failure rate of λ, the probability of a modular being failed at some time $t+\Delta t$, given that the module was operational at time t, is given by (for small value of Δt) $1-e^{-\lambda\Delta t} \approx \lambda\Delta t$. In other words, the probability that a module will fail within Δt is approximately $\lambda\Delta t$. The state transition probability can now be specified for each possible state transition.

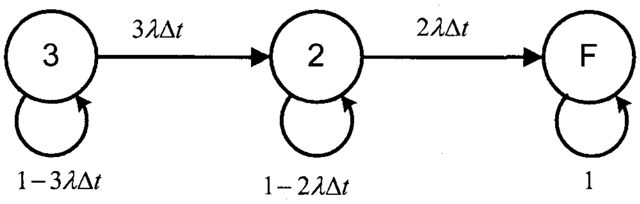


Figure 2.9. Reduced Markov model of the TMR system with a minimal number of states.

It is possible to reduce the Markov model of Fig. 2.8. If we appropriately define the state transition probabilities, several states within the TMR model can be combined. Suppose we let state 3 correspond to the state in which all three modules in the TMR system are functioning correctly; state 2 is the state in which two modules are working correctly; state *F* is the failed state in which two or more modules have failed. The resulting Markov model can be illustrated as shown in Fig. 2.9. The state transition probabilities shown in Fig. 2.9 have been derived to account for one of several failure

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

occurring. For example, the probability of transitioning from state 3 to state 2 depends on the probability of any one of three modules failing. Consequently, the transition probability assigned to the transition from state 3 to state 2 is $3\lambda\Delta t$.

Equations for the Markov model of the TMR system can be written easily from the state diagram Fig.2.9. The probability of the system being in any given state S at some time $t+\Delta t$ depends on the probability that the system was in a state from which it could transition to state S and the probability of that transition occurring. For example, the probability that the TMR system will be in state 3 at time $t+\Delta t$ depends on the probability that the system was in state 3 at time t (since the system can only transition to state 3 from state 3) and the probability of the system transitioning from state 3 back into itself. In mathematical form, the equations from the three states are

$$\begin{aligned}
p_3(t+\Delta t) &= (1-3\lambda\Delta t)p_3(t) \\
p_2(t+\Delta t) &= (3\lambda\Delta t)p_3(t)+(1-2\lambda\Delta t)p_2(t) \\
p_F(t+\Delta t) &= (2\lambda\Delta t)p_2(t)+p_F(t)
\end{aligned}$$

where $p_i(t)$ is the probability of being in state i at time t and $p_i(t+\Delta t)$ is the probability of being in state at time $t+\Delta t$.

The Markov models considered thus far have been discrete-time ones in which state transitions occur at fixed time interval Δt. It is possible to model systems using the continuous-time Markov model [6], in which state transitions can occur at any point in time. The continuous-time equations can be derived from the discrete-time equations by allowing the time interval Δt to approach zero. After simple algebraic manipulations and taking the limit as Δt approaches zero results in a set of differential equations given by

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\frac{dp_3(t)}{dt} = -3\lambda p_3(t)$$
$$\frac{dp_2(t)}{dt} = 3\lambda p_3(t) - 2\lambda p_2(t)$$
$$\frac{dp_F(t)}{dt} = 2\lambda p_2(t)$$

Using of the Laplace transforms (initial conditions: $p_3(0) = 1, p_2(0) = 0, p_F(0) = 0.$) results in the solution to the above differential equations

$$p_3(t) = e^{-3\lambda t}$$
$$p_2(t) = 3e^{-2\lambda t} - 3e^{-3\lambda t}$$
$$p_F(t) = 1 - 3e^{-2\lambda t} + 2e^{-3\lambda t}$$

Reliability of the TMR system is the probability of being in either state 3 or state 2, so

$$R_{TMR}(t) = p_3(t) + p_2(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}.$$

The TMR system has been used as an example to show how the Markov model can be used to model systems which would be difficult to be modeled in the combinatorial models. Meanwhile, Markov models can also be used to model systems with repair. This is out of the scope of our investigation for the robustness of the star graph and will not be covered in our study. The interested readers can refer to [2][6] for more details.

References:

[1] A.T. Amin, K.T. Siegrist, and P.J. Slater, “Pair-connected reliability of communication networks with vertex failures,” *Congressus Numerantium*, vol.67, 1988, pp.233-42.

[2] R. Billinton and R. Allan, “Reliability evaluation of engineering systems: concepts and techniques,” 2nd Edition, Plenum publishing corporation, 1992.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[3] C. Colbourn, “The combinatorics of network reliability,” Oxford University Press, 1987.

[4] Y. Chang and L. Bhuyan, “A combinatorial analysis of subcube reliability in hypercube,” *IEEE Transactions on Computers*, vol.44, no.7, July 1995, pp.952-956.

[5] C. Das and J. Kim, “A unified task-based dependability model for hypercube computers,” *IEEE Transactions on Parallel and Distributed Systems*, vol.3, no.3, May 1992, pp.312-324.

[6] B.W. Johnson, “Design and analysis of fault-tolerant digital system,” *Addison-Wesley Publishing Company*, 1989.

[7] S. Latifi, “A study of fault tolerance in star graph,” *Information Processing Letters* vol.102, no.5, May 2007, pp.196-200.

[8] D. Shier, “Network reliability and algebraic structures,” *Clarendon Press*, Oxford, 1991.

[9] S. Soh, S. Rai, and J.L. Trahan, “Improved lower bounds on the reliability of hypercube architectures,” *IEEE Transactions on Parallel and Distributed Systems*. vol.5, no.4, April 1994, pp.364-378.

[10] X.L. Wu, S. Latifi, and Y. Jiang, “A combinatorial analysis of distance reliability in star network,” in *Proc. of 21st IEEE International Parallel & Distributed Processing Symposium*, 2007.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER 3

DISTANCE RELIABILITY

In this chapter, we propose a two-terminal reliability measure referred to as Distance Reliability (*DR*) between any two given nodes *u* and *I* with the shortest distance, in an *n*-dimensional star graph, S_n, to assess the robustness of the star graph. Due to the fact that there exist numerous ways of constructing disjoint paths between *u* and *I* if *u*'s representation has more than one cycles, we only consider the special case of *u* having a single cycle under the node failure model, link failure model, and node/link failure models, respectively. For each failure model, two different cases depending on the relative positions of the source and destinations nodes are investigated to derive *DR*. This analysis gives us a basic understanding of *DR*. Conservative comparisons with the hypercube tell us that, *DR* of the star graph is expected to be closer to that of the hypercube when more cycles for the node representation and numerous ways of constructing disjoint paths are considered. Furthermore, *DR* for the antipodal communication is discussed as a special case. Lower bounds on the antipode reliability are derived under each failure model.

3.1 Background

Due to the high similarity between the hypercube and the star network [2][3], the star network is highly robust. It has been proved that the connectivity among nodes in this

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

topology can be preserved despite a substantial number of failures (in terms of node, link or node/link failures). This fact motivates us to look beyond the concept of connectivity, and demand more of this topology in terms of efficient communications. This poses stringent requirements on the connection of two nodes: i.e., not only two nodes have to be connected, but the distance between them must be the shortest. This idea consequently leads to a distance constrained reliability parameter which serves as a useful assessment to determine the communication delay, link-node utilization, and robustness in an interconnection network. Hence, we define Distance Reliability (*DR*) as the probability of having an operational path with the optimal distance between two given nodes u and I. A combinatorial approach has been used to evaluate *DR* especially for u having a single cycle in a S_n, under the node failure model, link failure model, and node/link failure models, respectively. From Chapter 1, we know that antipode(s) is the farthest node(s) from a given node with the distance of diameter (d_n) of the star graph [4]. There exists more than one antipode for a given node in the star graph. Thus, *DR* for the basic antipodal communication is discussed as a special case here. For the antipodal communication, different lower bounds on *DR* are also derived in this chapter.

3.2 Node failure model

Let F be the set of faulty nodes with $|F|$ denoted as the number of the faulty nodes. Here, we only focus on the case with at most $|F|$ node failures. Links are assumed to be perfect under this model. The objective in this model is to find *DR*, i.e. the probability of having at least one operational path with the shortest distance between u and I in the presence of node failures. A path is considered to be operational if it passes through fault-

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

free intermediary nodes between u and I. From Chapter 1, depending on the symbol in the leftmost position of the source permutation label being '1' or not, the node-disjoint paths between u and I are categorized into two different cases, and the shortest distance between them is either $c+m$ or $c+m-2$, where c is the number of cycles of length at least 2, and m is the number of misplaced symbols for the node permutation of u. For the purpose of simplicity, we use "r" to denote the shortest distance between u and I in the rest of this chapter.

3.2.1 Case I: $\pi(1)=1$

There are m optimal node-disjoint paths of the shortest distance $c+m$, and n-m-1 non-optimal paths of distance $c+m+2$ between u and I, where $m<n$, $c \le \lfloor (n-1)/2 \rfloor$, and $r=c+m$.

Definition 1. The union of all m optimal disjoint paths existing between u and I is referred to as a (u, I)-container. Total distinct nodes existing in this container are $m(r-1)+2$. In the container, u and I are always assumed to be fault-free.

Lemma 1. If $|F|<m$, there will be at least one optimal operational path between u and I.

Proof: In a star network there are m node-disjoint parallel paths of the shortest distance r between u and I. Each faulty node can at most belong to one path and since $|F|<m$, there will be at least one optimal path remaining operational between u and I.

Corollary 1. A star network is distance reliable for pairs of u and I, if $|F|<m$. Therefore it follows:

$$DR=1, \text{ when } |F|<m$$

Now we investigate DR when $|F|=m$. If all faults happen to locate in all but one of the optimal paths, one optimal fault-free path between u and I can be guaranteed. Here we

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

consider the worst case scenario: m faults distributions destroy all optimal paths. For a given u with only one cycle representation $(i_1 i_2 \cdots i_m)$, where $2 \le i_j \le n,\ 1 \le j \le m,\ r = m+1$, the union of all m optimal disjoint paths between u and I in the (u, I)-container can be listed as follows:

$$\begin{array}{l} i_1 i_2 i_3 \cdots i_{m-1} i_m i_1 \\ i_2 i_3 i_4 \cdots i_m i_1 i_2 \\ i_3 i_4 i_5 \cdots i_m i_1 i_2 i_3 \\ \quad\quad\quad \cdot \\ \quad\quad\quad \cdot \\ \quad\quad\quad \cdot \\ i_{m-1} i_m i_1 i_2 \cdots i_{m-2} i_{m-1} \\ i_m i_1 i_2 \cdots i_{m-2} i_{m-1} i_m \end{array}$$

All these disjoint optimal paths can be represented using a simple parallel reliability block diagram, where each path contains m distinct nodes. Scenarios that fail all optimal paths when $|F| = m$, happen only where each path exactly contains one fault, including cases where all neighbors of u or I are faulty. The probability associated with this event is

$$\Pr(\text{all } r\text{-pths destroyed when } |F| = m)$$
$$= \overbrace{\binom{m}{1}\binom{m}{1}\cdots\binom{m}{1}}^{m} / \binom{m^2}{m} = m^m / \binom{m^2}{m}$$

The above equation accounts for all scenarios that destroy all m optimal paths. Subtraction of the probability of occurrence of these scenarios from 1 will naturally give the probability of having at least one operational optimal path in the container and thus the *DR*.

Now let us determine the probability of having at least one operational optimal path when $m < |F| \le m^2 - m$ (note that if $|F| > m^2 - m$, then $DR_r = 0$). In practice, $|F|$ is

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

expected to be much less than this limit. We proceed to enumerate N_1, the number of faults distributions that render exactly one optimal fault-free path. For a container having m^2 nodes except u and I, any fault distributions that occur outside one optimal path qualifies in our enumeration. It follows that $N_1 = \binom{m^2 - m}{|F|}$. However, we cannot simply multiply N_1 by m to get the total faults distributions as some distributions will be double counted. To adjust this figure, we need to subtract by those occur outside any 2 optimal paths, and then add back those occur outside any 3 optimal paths, and so on (Principle of Inclusion and Exclusion). It follows that

$$N_i = \binom{m^2 - im}{|F|}, \text{ where } 1 \le i \le m-2$$

The total number of fault distributions N that will render at least one fault-free optimal path is given by: $N = \sum_{j=1}^{m-2} (-1)^{j+1} \binom{m}{j} N_j$. This can be interpreted as the sum of the number of fault distributions that will render one optimal path fault-free, subtracted by the number of fault distributions that will render two optimal paths fault-fee, added by the number of fault distributions that will render three optimal paths fault-free, and so on. Therefore, we have the following result.

Theorem 1: The probability of having at least one operational optimal path between u (with a single cycle representation $(i_1 i_2 \cdots i_m)$, where $2 \le i_j \le n,\ 1 \le j \le m$) and I when $m < |F| \le m^2 - m$ is given by:

$$DR = N / \binom{m^2}{|F|} = \sum_{j=1}^{m-2} (-1)^{j+1} \binom{m}{j} N_j / \binom{m^2}{|F|}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The result given in Theorem 1, however, does not consider the node cycle representation having cycles more than one. In [6], Misic proved that cycles for the node representation have such properties: cycles can appear in any order; cycles without "1" inside can be executed to reach the same destination regardless of the number of cyclic shifts on the symbols; and cycles without "1" inside can be nested in any other cycle. If we do not consider the nesting of cycles and we assume all the cycles are of none-ordinary type (without "1" inside), the number of optimal paths is $k!\prod_{i=1}^{k} p_i$, where k is number of cycles and p_i denotes the number of distinct symbols in each cycle C_k. Day [2] gave one possible way of constructing a set of node-disjoint paths between two given nodes. In fact there exist numerous ways of constructing disjoint paths between u and I when the number of cycles is greater than one.

Based on above-mentioned properties of cycles, for example, 24 optimal paths exist between $u(13254)$ (decomposed into cycles (23)(45)) and $I(12345)$ through the following sequence:

$2-3-2-4-5-4$
$2-3-2-5-4-5$
$2-4-5-4-3-2$
$2-3-4-5-4-2$
$2-5-4-5-3-2$
$2-3-5-4-5-2$
$3-2-3-4-5-4$
$3-2-3-5-4-5$
$3-4-5-4-2-3$
$3-2-4-5-4-3$
$3-5-4-5-2-3$
$3-2-5-4-5-3$
$4-5-4-3-2-3$
$4-5-4-2-3-2$
$4-2-3-2-5-4$
$4-5-2-3-2-4$
$4-3-2-3-5-4$
$4-5-3-2-3-4$
$5-4-5-3-2-3$
$5-4-5-2-3-2$
$5-2-3-2-4-5$
$5-4-2-3-2-5$
$5-3-2-3-4-5$
$5-4-3-2-3-5$

Between these optimal paths, there exist more than one way of constructing four disjoint paths between u and I. Besides the set of disjoint paths in Fig. 3.1 (a) (explained in Day [2]), three more sets of node disjoint paths (shown in Fig. 3.1 (b)~(d) where nodes/links different from (a) are plotted in grey color) can also be constructed. The number of optimal paths between two given nodes when the node permutation is

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

decomposed into more than one cycle is going to be huge and difficult to determine. For instance, there exist more than 192 optimal paths between $u(1325476)$ and $I(1234567)$ even only considering the nesting of one cycle. Meanwhile the ways of constructing disjoint paths will be more and it appears to be impossible to derive a deterministic formula to find out all. Therefore, we only discuss the distance reliability for the source node having a single cycle representation. When the number of cycles for the node representation is greater than one, the problem of determining *DR* appears to be difficult in the star graph. A thorough investigation of *DR* for the general case is still under development and remains to be an open problem.

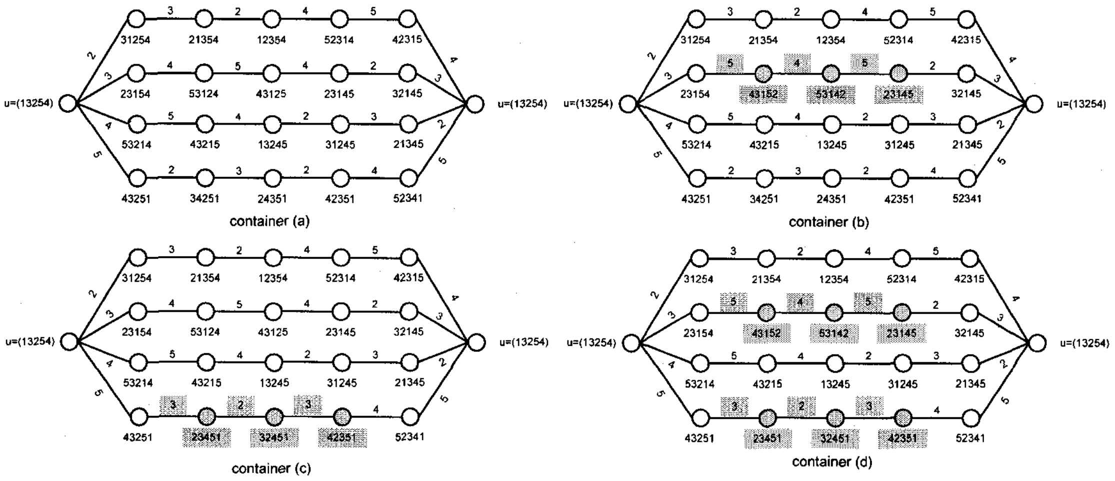


Fig. 3.1. Multiple ways of constructing four disjoint paths between $u(13254)$ and $I(12345)$.

To compare *DR* of the star graph with that of the hypercube, we use the same shortest distance between two given nodes u and I for both networks. However, the number of optimal paths and nodes in the container of the star graph is less than those of the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

hypercube. The container in the star graph has r-1 optimal paths and $(r\text{-}1)^2$ nodes, while the container in the hypercube has $r!$ optimal paths and 2^r nodes. This explains why the *DR* of the hypercube is higher than that of the star graph in Fig. 3.2. And the *DR* of the star graph in Fig. 3.2 shows an appreciable improvement with the increase of the shortest distance r. If more cycles for the node representation are considered, and more alternative ways of constructing node disjoint paths in the star graph are taken into account; hence *DR* of the star graph is expected to be closer to that of the hypercube.

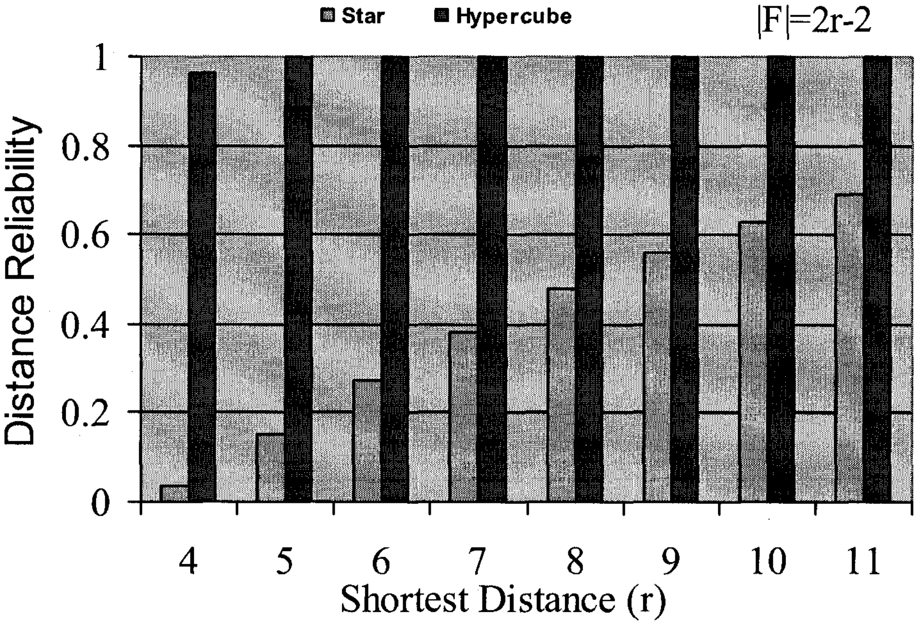


Fig. 3.2. *DRs* for star graph and hypercube when $|F|$=2r-2.

3.2.2 Case II: $\pi(1) \neq 1$

The container under this case has c optimal paths of the shortest distance $c+m-2$, $m-c-1$ paths of distance $c+m$, and $n-m$ paths of distance $c+m+2$ between u and I ($m \leq n$, $c \leq \lfloor n/2 \rfloor$, and $r = c+m-2$) with $c(r-1)+2$ distinct nodes. There will be at

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

least one operational path between u and I if $|F|<c$. For a given u with only one cycle presentation $(1i_1i_2\cdots i_m)$, where $2\le i_j\le n,\ 1\le j\le m,\ \ r=m-1$, there exists only one optimal path and hence one arbitrary node fault will destroy it. On the other hand if u's representation has more than one cycle, *DR* when $|F|\ge c$ is difficult to determine due to the numerous ways of constructing disjoint paths between u and I. Therefore, we will only consider Case I for the link failure model and the combined node/link failure model.

3.2.3 Special case: antipode reliability

Since the number of antipodes is more than one (for n>3) [4], the antipodal communication we are concerned about is the communication between u and its unique basic antipode (assuming to be I). Due to the symmetry property in the star network, similar analysis can be extended to the communication between the source node and other antipodes. The basic antipode can be reached from the source node by applying the following permutations:

$$\delta_{\max}=(1)(23)(45)\cdots(i,i+1)\cdots(n-1,n),\qquad \text{for odd } n;$$

$$\begin{aligned}\delta_{\max}&=(1)(234)(56)\cdots(i,i+1)\cdots(n-1,n)\\&\quad(12)(34)(56)\cdots(i,i+1)\cdots(n-1,n),\quad \text{for even } n\end{aligned}$$

As such, the antipodal communication can be performed concurrently according to the following sequences:

$$\begin{aligned}&2-3-2\cdots i-(i+1)-i\cdots(n-1)-n-(n-1), &&\text{for odd } n;\\&2-3-4-2\cdots i-(i+1)-i\cdots(n-1)-n-(n-1) &&\text{or}\\&2-3-4-3\cdots i-(i+1)-i\cdots(n-1)-n-(n-1), &&\text{for even } n.\end{aligned}$$

To have at least one operational optimal path between u and I, two different scenarios need to be considered.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Scenario 1. When n is odd, there are a total of $n-1$ disjoint paths of the shortest length d_n between u and I ($\pi(1)=1, m=n-1, c=(n-1)/2, r=c+m=\lfloor 3(n-1)/2 \rfloor = d_n$).

Paths distributions can be verified by the following example. For example, there are six node-disjoint paths between u and I when n=7. Based on the above discussion, a total six disjoint paths can be formulated based on the following permutations:

(1) $2-3-2-4-5-4-6-7-6$
(2) $3-2-4-5-4-6-7-6-3$
(3) $4-5-4-6-7-6-2-3-2$
(4) $5-4-6-7-6-2-3-2-5$
(5) $6-7-6-4-5-4-2-3-2$
(6) $7-6-4-5-4-2-3-2-7$

Antipode reliability is actually a special case of the distance reliability discussed in Section 3.2.1. In which, the source node can be decomposed into $(n-1)/2$ cycles, each of length 2. From the discussion in Section 3.2.1 we know that the determination of the antipode reliability using the combinatorial method when the number of cycles for the node permutation is more than one, is difficult due to the numerous ways of constructing the disjoint paths and many faults distributions. Under this circumstance, and for a given $|F|$ obtained from the network reliability data and its mission time, we can develop a reliability expression based on a stochastic model as follows.

In the stochastic graph model $G(V, E)$ for the star network S_n, the following assumptions are made:

- Source and destination nodes are always fault-free;
- The operational probabilities of all nodes (links) are the same and equal to p_n (p_l). For a given constant failure rate λ, using the exponential model one can compute p_i (p_n or p_l) as $p_i(t)=\exp(-\lambda t)$;

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

- Failures are independent and identically distributed.

A Boolean technique for the reliability evaluation starts with a sum of products expression for min-paths and converts it into an equivalent sum of disjoint products (SDP) expression [5]. In the SDP form, an UP or success (DOWN or failure) state of a node is replaced by its reliability p_n or $(1-p_n)$, and the Boolean sum (product) by the arithmetic sum (product). In other words, the SDP expression is interpreted directly as an equivalent probability expression of symbolic reliability. Let $P_1, P_2, \dots, P_h$ be all *r*-paths between *u* and *I*. Then the SDP expression is obtained as follows: $P_1 + P_2\overline{P}_1 + \dots + P_h\overline{P}_1 \dots \overline{P}_{h-1}$ where $\overline{P}_j$, denotes the DOWN event of the path P_j. The probability of UP (operational) for the i^{th} term $P_i\overline{P}_1\overline{P}_2 \cdots \overline{P}_{i-1}$ can be evaluated using the conditional probability and the standard Boolean operations, and is called the disjoint product event. It has been shown that the reliability evaluation for star networks with non-disjoint paths is NP-hard [1]. As mentioned in Chapter I, there are $(\frac{n-1}{2})!2^{(n-1)/2}$ distinct optimal paths between *u* and *I* even without considering the nesting of cycles if we consider the maximum value of *m* and *c*. Clearly, for the same reason, the determination of *DR* is also intractable. Thus, we will attempt to do the next best thing, i.e. derive bounds on *DR* of the node-disjoint paths.

A lower bound on *DR* can be obtained by considering only the set of *n*-1 node-disjoint paths between *u* and *I* with the shortest distance *r* as

$$DR_r \geq 1-\left(1-p_n^{r-1}\right)^{2r/3}$$

The above expression uses the principles for a simple parallel reliability block diagram. Note that this lower bound is quite pessimistic; even for small size star networks, it renders a large deviation. Next we present a tight lower bound by

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

constructing the hexagons between u and I since every cycle out of total $c=(n-1)/2=r/3$ has two distinct symbols inside, and hence provides two alternatives to construct the disjoint paths. Consider two nodes u and I, i.e. r = 3 when n is 3. There are only two node-disjoint *3*-paths (Fig. 3.3 a). The expression for DR_3 can be attributed to the following: $DR_3 \geq 1-\left(1-p_n^2\right)^2$.

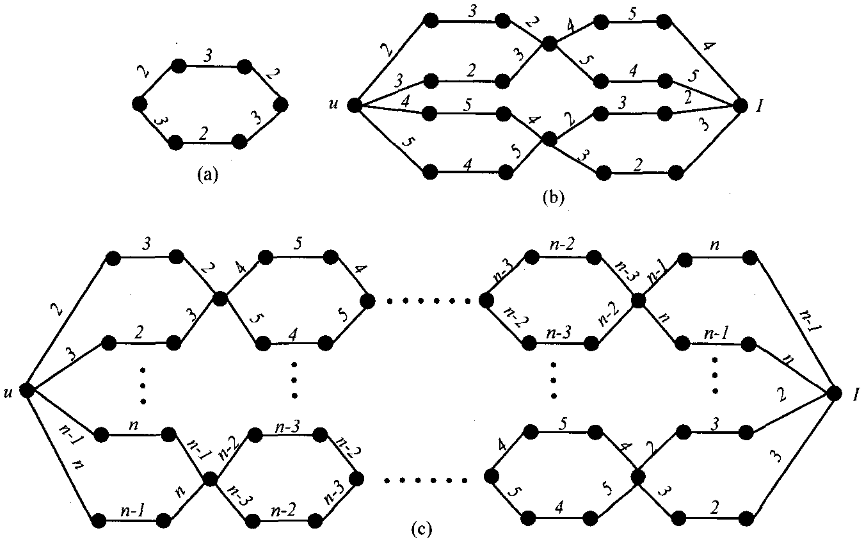


Figure 3.3. Hexagon construction of disjoint paths between u and its basic antipode I.

When n=5, there are two hexagons between u and I having four node-disjoint paths. Since each cycle provides two alternative choices to construct the node-disjoint optimal paths, thus, the corresponding DR can be obtained as follows shown in Fig. 3.3 b:

$$DR_6 \geq 1-\left(1-p_n^1\left(1-\left(1-p_n^2\right)^2\right)^2\right)^2$$

Similarly, extending the above concept to the general case where $r=2(n-1)/3$, the following equation holds for DR_r:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$DR_r \geq 1-\left(1-p_n^{c-1}\left(1-\left(1-p_n^2\right)^2\right)^c\right)^c = 1-\left(1-p_n^{3c-1}\left(2-p_n^2\right)^c\right)^c$$

$$= 1-\left(1-p_n^{r-1}\left(2-p_n^2\right)^{r/3}\right)^{r/3}, \text{ where } r = 3,6,9,\cdots,3(n-1)/2$$

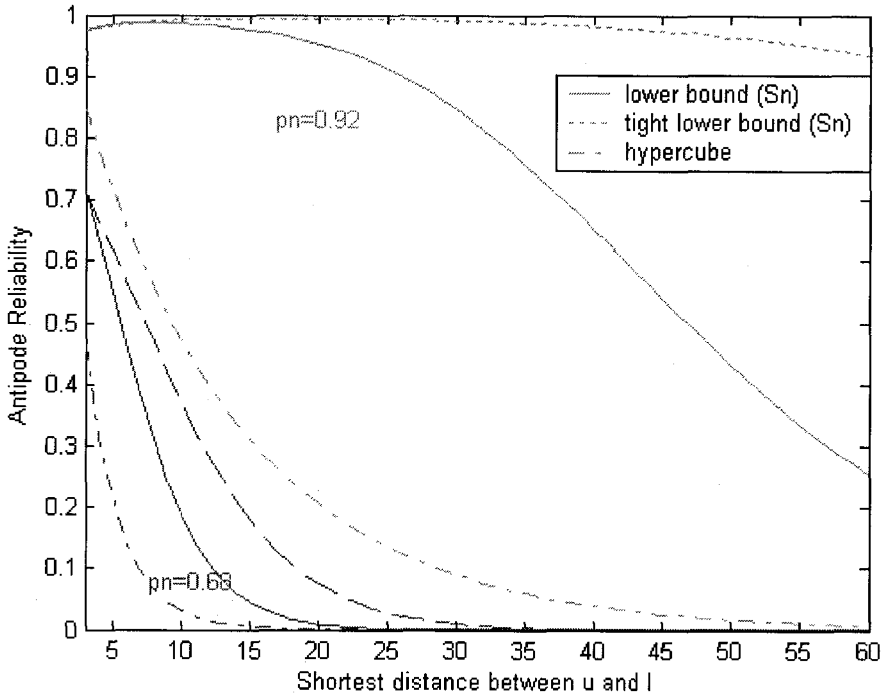


Figure 3.4. Lower bounds on the antipode reliability for different values of the node reliability (Scenario 1).

These two lower bounds are compared under different values of the node reliability $p_n = 0.91$ and 0.68, respectively. The gap between the tight lower bound on the antipode reliability with the help of constructing the hexagons and the lower bound applying the simple parallel reliability block diagram becomes larger with the increase of the node reliability. Furthermore, results in Figure 3.4 verify that the tight lower bound shows the appreciable improvement on the antipode reliability over the pessimistic lower bound

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

especially for the larger node reliability and large size of the star graph. The dashdot lines in Fig. 3.4 represent the antipode reliability in the hypercube. The large gap between the antipode reliabilities of the hypercube and the star graph is due to the fact that there is a single antipode for a given node in the hypercube. The antipode reliability in the star graph is expected to be even higher than that in the hypercube when numerous ways of constructing disjoint paths are considered.

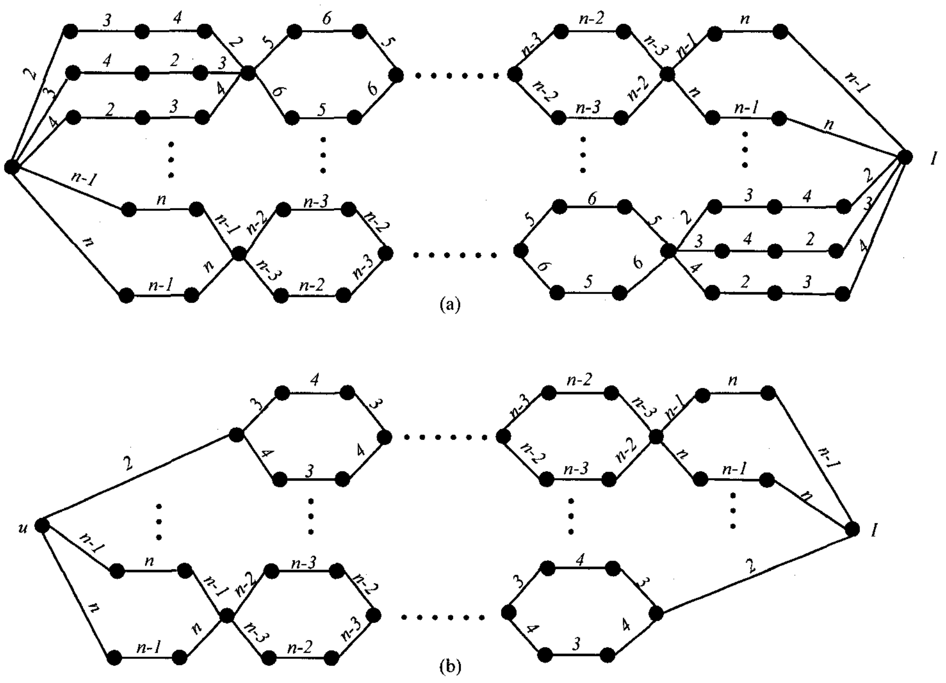


Figure 3.5. Hexagon construction of disjoint paths between *u* and its basic antipode *I*.

Scenario 2. When n is even, there are two choices of the permutations for the source node to reach the basic antipode, where $m = n-1$, $c = (n-2)/2$ or $m = n$, $c = n/2$, and $r = (3n-4)/2$. Paths distribution shown in Figure 3.5, are similar to Scenario 1 except the first cycle of length either 3 or 1. The tight lower bounds for *DR* under this scenario

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

are given as follows using the same strategy as Scenario 1. These two lower bounds are compared in Fig. 3.6 under different values of the node reliabilities $p_n = 0.91$ and 0.68, respectively. The antipode reliability between a given node and its single antipode reliability in the hypercube is also presented.

$$DR_r \geq 1-\left(1-\left(1-\left(1-p_n^3\right)^3\right)p_n^{3(c-1)}\left(2-p_n^2\right)^{c-1}\right)^c$$
$$=1-\left(1-\left(1-\left(1-p_n^3\right)^3\right)p_n^{(r-4)}\left(2-p_n^2\right)^{(r-4)/3}\right)^{(r-1)/3}, \text{ where } r=4,7,10,\cdots,(3n-4)/2$$

and

$$DR_r \geq 1-\left(1-p_n^{3(c-1)}\left(2-p_n^2\right)^{c-1}\right)^c = 1-\left(1-p_n^{(r-1)}\left(2-p_n^2\right)^{(r-1)/3}\right)^{(r+2)/3}, \text{where } r=1,4,\cdots,(3n-4)/2$$

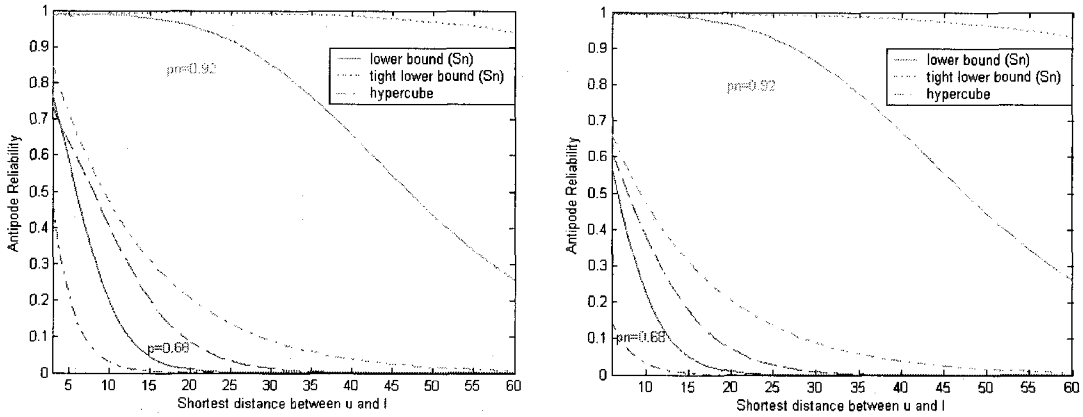


Figure 3.6. Lower bounds on the antipode reliability for different values of the node reliability (Scenario 2).

3.3 Link failure model

This section analyzes DR under the link failure model with at most $|F|$ failures, where F represents the set of faulty links. Nodes are assumed to be perfect under this model. As with the node failures, the interest is in the system configuration that has at

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

least one operational optimal path between u and I in the presence of link failures. Since the DR analysis and the antipode reliability under the link failure model are same as what have been performed in the node failure model, detailed analysis are not repeated here. Hence we give the results directly. The only difference is that considered network elements per optimal path here are r-1 links instead of r nodes per optimal path in the node failure model.

The probability of having at least one operational optimal path between u (with a cycle presentation $(i_1 i_2 \cdots i_m)$, where $2 \le i_j \le n,\ 1 \le j \le m$) and I when $m < |F| \le m^2 - 1$ is given by:

$$DR = \sum_{j=1}^{m-2} (-1)^{j+1} \binom{m}{j} N_j \Big/ \binom{m(m+1)}{|F|}, \text{ where } N_j = \binom{m(m+1) - j(m+1)}{|F|},\ 1 \le j \le m-2$$

The above result does not consider the node cycle representation having cycles more than one. Due to the fact that there exist numerous ways of constructing disjoint paths between u and I, the problem of determining DR when the number of cycles for the node representation under the link failure model is greater than one appears to be difficult.

The antipode reliability under this model can be analyzed similarly to the node failure mode described in Section 3.2.3. For the case scenario 1 where n is odd, the pessimistic lower bound using the simple parallel block diagram, and the tight lower bound with the help of the construction of hexagons, are given as follows:

$$DR_r \ge 1-\left(1-p_l^r\right)^{2r/3} \quad \text{and}$$

$$DR_r \ge 1-\left(1-\left(1-\left(1-p_l^3\right)^2\right)^{\frac{r}{3}}\right)^{\frac{r}{3}} = 1-\left(1-p_l^r\left(2-p_l^2\right)^{\frac{r}{3}}\right)^{\frac{r}{3}}, \text{ where } r = 3,6,9,\cdots,3(n-1)/2$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.4 Combined node and link failure model

In the previous two sections, we only assume either nodes or links could fail. However, all network components (nodes or links) can fail in real applications. A combined failure model is developed to analyze *DR* under the case with at most $|F|$ failures, where F represents the sum of faulty nodes and links. As with the node failure model, the interest is in the system that has at least one operational optimal path between u and I in the presence of node/link failures.

3.4.1 Case I: $\pi(1)=1$

There are m optimal node-disjoint paths of the shortest distance $c+m$, and $n-m-1$ non-optimal paths of distance $c+m+2$ between u and I $(m<n,\ c\le\lfloor(n-1)/2\rfloor$, and $r=c+m)$. Total distinct nodes and links in this container are $m(2r-1)+2$. The conclusion that there will be at least one operational path between u and I can be guaranteed if $|F|<m$.

Now we investigate *DR* when $|F|=m$. If all faults happen to reside in all but one of the optimal paths, one optimal fault-free path between u and I can be guaranteed. Here we consider the worst case scenario: m faults distributions destroy all optimal paths. For a given u with only one cycle representation $(i_1 i_2 \cdots i_m)$, where $2\le i_j\le n,\ 1\le j\le m,\ r=m+1$, scenarios that fail all optimal paths when $|F|=m$, happen only where each path exactly contains one fault (either node or link), including cases where all neighbors of u or I are faulty. The probability associated with this event can be obtained similarly as Case I in Section 3.2

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Pr(all r-pths destroyed when $|F| = m$)

$$= \overbrace{\binom{2m+1}{1}\binom{2m+1}{1}\cdots\binom{2m+1}{1}}^{m} / \binom{m(2m+1)}{m} = (2m+1)^m / \binom{m(2m+1)}{m}$$

The above equation accounts for all scenarios that destroy all m optimal paths. Subtraction of the probability of occurrence of these events from 1 will naturally give the probability of having at least one operational path in the container and thus the *DR*.

Now let us determine the probability of having at least one operational optimal path when $m < |F| \leq (m-1)(2m+1)$ (note that if $|F| > (m-1)(2m+1)$, then $DR_r = 0$). In practice, $|F|$ is expected to be much less than this limit. The analysis can be carried out similarly to Case I in Section 3.2. The total number of fault distributions N that will render at least one fault-free optimal path is given by: $N = \sum_{j=1}^{m-2} (-1)^{j+1}\binom{m}{j} N_j$, where $N_j = \binom{m(2m+1) - j(2m+1)}{|F|}$, and $1 \leq j \leq m-2$. Therefore, we have the following result.

Theorem 2: The probability of having at least one operational optimal path between u (with a cycle presentation $(i_1 i_2 \cdots i_m)$, where $2 \leq i_j \leq n,\ 1 \leq j \leq m$) and I when $m < |F| \leq (m-1)(2m+1)$ is given by:

$$DR = N / \binom{(m-1)(2m+1)}{|F|} = \sum_{j=1}^{m-2} (-1)^{j+1}\binom{m}{j} N_j / \binom{(m-1)(2m+1)}{|F|}.$$

Theorem 2, similar to Theorem 1, however, does not consider the node cycle representation having cycles more than one. Due to the fact that there exist numerous ways of constructing disjoint paths between u and I, the problem of determining *DR* when the number of cycles is greater than one appears to be difficult.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.4.2 Case II: $\pi(1) \neq 1$

There will be at least one operational path between u and I if $|F| < c$. Now we investigate DR when $|F|=c$. For a given u with only one cycle presentation $(1 i_1 i_2 \cdots i_m)$, where $2 \leq i_j \leq n,\ 1 \leq j \leq m,\ r = m-1$, there exists only one optimal path and hence one arbitrary node fault will destroy it. DR when $|F| \geq c$ is difficult to determine due to the numerous ways of constructing disjoint paths between u and I if the number of cycles for the node representation is greater than one.

3.4.3 Special case: antipode reliability

The antipode reliability under this model can be analyzed similarly to the node failure mode described in Section 3.2.3. For the case scenario 1 where n is odd, the pessimistic lower bound using the simple parallel block diagram, and the tight lower bound with the help of the construction of hexagons, are given as follows

$$DR_r \geq 1-\left(1-p_n^{r-1} p_l^{r}\right)^{2r/3} \quad \text{and}$$

$$DR_r \geq 1-\left(1-p_n^{c-1} p_l^{c}\left(1-\left(1-p_n^2 p_l^3\right)^2\right)^c\right)^c = 1-\left(1-p_n^{3c-1} p_l^{4c}\left(2-p_n^2 p_l^3\right)^c\right)^c$$
$$= 1-\left(1-p_n^{r/3-1} p_n^{(4r)/3}\left(2-p_n^2 p_l^3\right)^{r/3}\right)^{r/3}, \text{ where } r = 3,6,9 \cdots 3(n-1)/2$$

These two lower bounds shown in Fig. 3.7 are compared under different values of the node/link reliabilities, respectively. The gap between the tight lower bound on the antipode reliability using the hexagons and the lower bound applying the simple parallel reliability block diagram becomes even larger than the node failure model with the increase of the node/link reliability. This is due to the fact that both of nodes and links can fail is closer to the real applications. The dashdot lines in Fig. 3.7 represent the antipode reliability under different node/link reliabilities in the hypercube. The large gap

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

between the antipode reliabilities of the hypercube and the star graph is due to the fact that there is a single antipode for a given node in the hypercube.

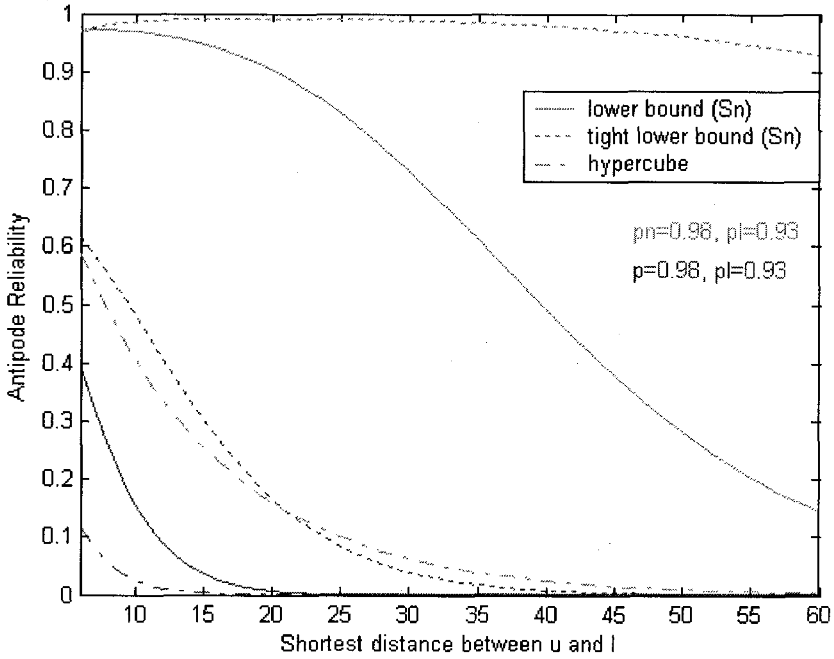


Figure 3.7. Lower bounds on the antipode reliability for different node reliabilities.

3.5 Conclusion

In this chapter, a figure of merit called distance reliability was introduced for the reliability analysis of star interconnection networks. This measure is appealing for the robust networks (such as star network) since it poses stringent requirements on the connection of two nodes; i.e. not only do two nodes have to be connected, but the distance between them must be the shortest. We presented a deterministic formulation of the distance reliability especially between u having a single cycle representation and I when the number of faults is bounded using the combinatorial method. For each of the node, link and node/link failure models, two different cases depending on the relative

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

positions of the source & destination, were analyzed to compute *DR*. The antipodal reliability was also considered as a special case to further demonstrate the fault tolerance of star networks. Lower bounds on the antipode reliability were derived and proven to be more tolerant that the hypercube through the comparisons between similar size star and hypercube networks.

References

[1] M. S. Chang, D. J. Chen, M. S. Lin, and K. L. Ku, “The distributed program reliability analysis on star topologies,” in *Proc. of 1998 International Conference on Parallel and Distributed Systems,* 1998, pp.100-106.

[2] K. Day and A. Tripathi, “A comparative study of topologies properties of hypercubes and star networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol.5, no.1, Jan. 1994, pp.31-38.

[3] A. E. Kiasari and H. Sarbazi-Azad, “A comparative performance analysis of n-cubes and star graph,” in *Proc. of 20th International Parallel and Distributed Processing Symposium*, 2006, pp.25-29.

[4] S. Latifi, “Parallel dimension permutations on star network,” *IFIP Trans. A: Comp. Sci. Tech.*, A23, 1993, pp.191-201.

[5] S. Latifi and S. Rai, “A robustness measure for hypercube networks,” in *Proc. of the 36th Midwest Symposium on Circuits and Systems*, Aug. 1993, vol.1, pp.546-549.

[6] J. Misic and Z. Jovanovic, “Routing function and deadlock avoidance in a star graph interconnection network,” *Journal of Parallel and Distributed Computing*, vol.22, 1994, pp.216-228.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER 4

MARKOV RELIABILITY MODELING

In this chapter, the degradation of a container between two given nodes u and I in a S_n having at least one operational shortest path is examined to measure the system effectiveness in the presence of failures. This measure is evaluated under the node failure, link failure, node/link combined failure models, respectively. Failure of the container is defined as being when no fault-free optimal path remains operational between u and I. States of the degradation of a container between u and I in a S_n are modeled by a Markov chain. The solution to transition state functions is derived and the MTTF (mean time to failure) for them under each failure model is also computed. For comparisons the results of similar size containers of the hypercube is presented.

Notation:

S_n	n-dimensional star network
S_0	system state with no failures
$S_{j,i}$	system state - j: number of failed optimal operational paths in the (u, I)-container; i: number of failures
λ_n	s-independent failure rate of system nodes
λ_l	s-independent failure rate of system links
t	time

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$P_j(t)$ $\Pr\{\text{system is in } S_j\}$

$\mathbf{P}_{0j}$ Laplace transform of P_j

$R_j(t)$ probability that j optimal paths in the (u, I)-container have failed

T_j MTTF estimate when there are j failed optimal communication paths between u and I.

4.1 Background

In the previous chapter, the distance reliability (*DR*) merit has been proposed to evaluate the reliability of the star graph. With this merit, two given nodes u and I in any (u, I)-container in a S_n are expected to not only be connected but to be apart by the optimal distance (i.e. the shortest distance between the source and destination nodes). Although the first fault (node or link) makes one of optimal paths between u and I in the (u, I)-container unavailable, it is important to know how many fault-free optimal paths are available in the damaged structure? The degradation of a container in a S_n having at least one operational optimal path with the shortest distance between two given nodes u and I is examined to measure the system effectiveness in the presence of failures. System is considered failed when there is no operational optimal path between u and I. The process of the degradation can be modeled by a Markov chain [3]. It is difficult to give an explicit reliability expression for a large system when there are many components and a diverse reliability structure. Therefore, we turn our attention to MTTF (mean time to failure) [2] used to describe the robustness of star networks.

To compare the reliability and degradation of the star graph to those of the hypercube, a similar analysis of impacts of node and link failures on the star graph is provided here;

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the number of nodes (and links) of a container between two given nodes *u* and *I* in the star graph is not the same as that for the hypercube interconnection topology. The star graph is based on a factorial growth in the number of nodes, while the hypercube is based on a power-of-2 growth in the number of nodes. For example, a star graph of dimension of 4 has 4! (24) nodes and a hypercube of dimension 4 has 2^4 (16) nodes. Consequently, a container in a S_4 has 9 (maximum) nodes, while a container in Q_4 has 12 (maximum) nodes. This makes a direct comparison difficult; however conservative comparisons are made where possible between the reliability and degradation of similar size containers in the star graph and the hypercube.

Here we only consider arbitrary and *s*-independent failure of nodes and links such that the probability of occurrences of a specific node or link failure is not affected by the previous failures. The method for characterizing the degradation of a container in a S_n is based on maintaining the maximum number of optimal paths between two given nodes *u* and *I* in the presence of tolerable number of failures. In some cases, a particular sequence of failures can present different degradation possibilities; however, specific sequences of failures are unlikely to occur and are not included in analysis here.

We can assume that whenever a failure renders an *r*-path faulty, that faulty *r*-path no longer belongs to the container: *r*-paths are isolated as soon as they fail. The system at S_j, has, in general, additional fault-free nodes/links belonging to an already faulty *r*-paths in the (*u, I*)-container; they do not need to be considered in their subsequent failure analysis since they do not affect the system failure anymore.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

We start to analyze the degradation of the container in a S_n under the node failure model first, then continue the analysis under the link failure model and the combined node/link failure model.

4.2 Node failure model

In this section, we focus on the scenario having at most F failures, where $|F|$ represents the set of faulty nodes. Links are perfect and considered to be negligible compared to the processor failures under the node failure model. The general objective is to keep at least one operational disjoint path of the shortest distance between two given nodes u and I in the presence of node failures. A path is operational if it passes through fault-free intermediary nodes. And a path is said to be optimal if it is of the shortest distance between u and I. From Chapter 1, we know that the shortest distance of paths between u and I is either $c+m$ or $c+m-2$ depending on the first symbol in the leftmost position of the permutation label of the source node being "1" or not.

4.2.1 Case 1 $\pi(1)=1$

There are m optimal disjoint paths of the shortest distance $c+m$, and $n-m-1$ non-optimal paths of distance $c+m+2$ between u and I $(m<n, c\le\lfloor(n-1)/2\rfloor,\ r=c+m)$. Total distinct nodes in a (u, I)-container are $k=m(r-1)$ except u and I.

Consider a (u, I)-container in a S_n, the first failure of an arbitrary node always leaves exactly one of these m optimal paths failed and keeps the rest undamaged. To damage all of the m paths, at least m node failures are necessary. On the other hand, although as few as m node failures (out of k nodes in the (u, I)-container) could possibly destroy all the r-

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

paths, this scenario is highly unlikely under the assumption that node failures are arbitrary and *s*-independently distributed.

Consider three disjoint parallel paths of the minimum distance 4 between the source node *u*(1342) and the destination node *I*(1234) in a S_4 shown in Fig. 4.1. Definition 1 tells that there are 9 distinct nodes except *u* and *I* in this (*u*, *I*)-container each of which containing 3 distinct nodes with the same possibility of failing (*u* and *I* are assumed to be always fault-free).

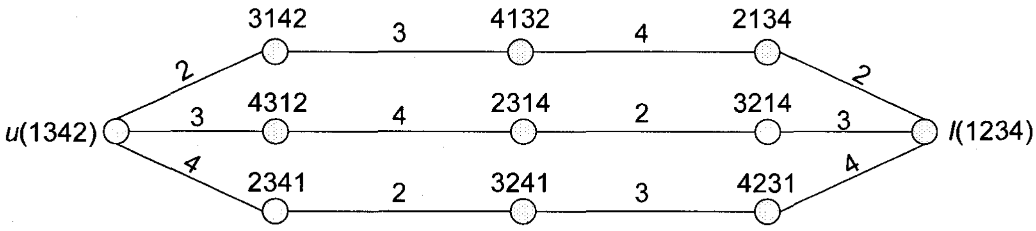


Figure 4.1. Three node-disjoint optimal paths between *u*(1342) and *I*(1234).

In Fig. 4.2, the first failure of an arbitrary node damages one of three *4*-paths, and the system enters state $S_{1,1}$; then the system has eight nodes which are equal likely to fail. At the next node failure, the system enters state $S_{2,2}$ with the probability 6/8, and enters state $S_{1,2}$ with the probability 2/8. Similarly, other state transitions can be explained. For example, in state $S_{2,2}$ any further node failure forces the system into state $S_{3,3}$ with the probability 3/7, etc. In each of the states $S_{3,i}$, $3 \leq i \leq 7$, there remains no operational *4*-path. These system states are important collectively because the system fails when no fault-free *4*-path is available. The states in Fig. 4.2 are arranged such that states that represent a common number of failed *4*-paths is arranged vertically and the degradation

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

of *4*-paths is arranged horizontally. This analysis can be extended to any container in star networks with different sizes. Similar state diagrams can be achieved with the different width and height, but the pattern remains essentially the same.

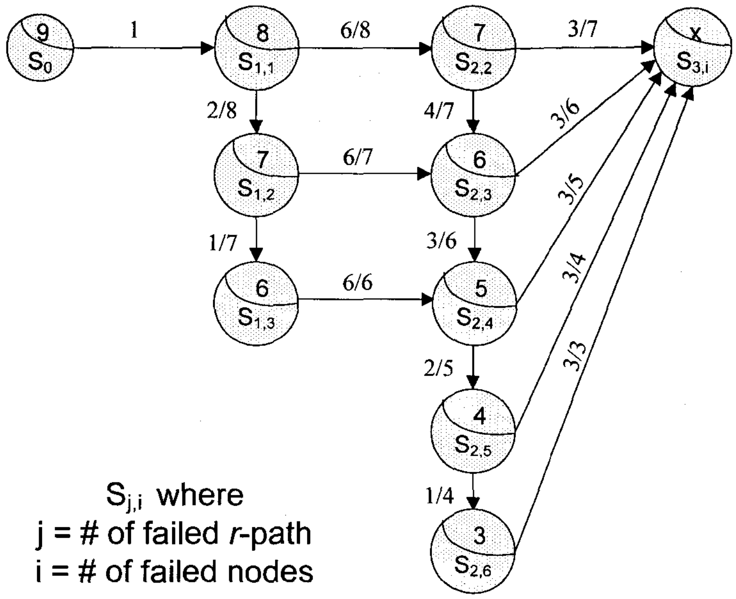


Figure 4.2. State diagram for $S_{j,i}$ in a *(u, I)*-container of S_4.

Consider $S_{j,i}$, for a given (*u*, *I*)-container, there are (*m-j*) fault-free disjoint parallel optimal paths between *u* and *I*, each of which has (*r*-1) fault-free nodes; there are (*k-i*) fault-free nodes in the container. Because failures are *s*-independent with the equal probability of failing, the probability that the next node-failure damages a fault-free optimal path is then:

$$\frac{(m-j)(r-1)}{k-i}.$$

Hence the probability that next node failure does not damage a fault-free optimal path is:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$1-\frac{(m-j)(r-1)}{k-i}.$$

Starting with a fault-free state S_0, the state diagram for a (u, I)-container under the node failure model is demonstrated in Fig. 4.3. The corresponding state transition rates are $r_j = \lambda_n(m-j)(r-1),\ 0 \le j \le m$.

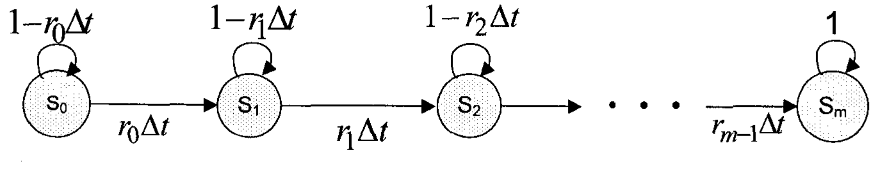


Figure 4.3. Simple state diagram under the node-failure model for

a (u, I)-container in a S_n.

The state transition expressions for the first three states are:

$$P_0(t+\Delta t) = (1-r_0\Delta t)P_0(t) \Rightarrow \frac{\partial P_0}{\partial t} = -r_0P_0$$

$$P_1(t+\Delta t) = r_0\Delta tP_0(t) + (1-r_1\Delta t)P_1(t) \Rightarrow \frac{\partial P_1}{\partial t} = r_0P_0 - r_1P_1$$

$$P_2(t+\Delta t) = r_1\Delta tP_1(t) + (1-r_2\Delta t)P_2(t) \Rightarrow \frac{\partial P_2}{\partial t} = r_1P_1 - r_2P_2$$

Similarly, we can derive the state transition function of the state j as follows:

$$P_j(t+\Delta t) = r_{j-1}\Delta tP_{j-1} + (1-r_j\Delta t)P_j \Rightarrow \frac{\partial P_j}{\partial t} = r_{j-1}P_{j-1} - r_jP_j, \text{ for } 1 < j \le m$$

The initial and final conditions for the states probabilities are:

$$P_0(0) = 1,\ P_j(0) = 0 \text{ for } 1 \le j \le m$$

$$P_j(\infty) = 0 \text{ for } 0 \le j < m,\ P_m(\infty) = 1$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

After the Laplace transformation,

$$s\mathbf{P_0} - 1 = -r_0\mathbf{P_0};\ \mathbf{P_0} = \frac{1}{s+r_0} \Rightarrow P_0(t) = \exp(-r_0 t)$$

$$\mathbf{P_1} = \frac{r_0}{s+r_1}\mathbf{P_0} = \frac{r_0}{r_0-r_1}\left(\frac{1}{s+r_1} - \frac{1}{s+r_0}\right) \Rightarrow P_1(t) = \frac{r_0}{r_1-r_0}\exp(-r_0 t) + \frac{r_0}{r_0-r_1}\exp(-r_1 t)$$

$$\mathbf{P_2} = \frac{r_1}{s+r_2}\mathbf{P_1} = \frac{r_1}{s+r_2}\frac{r_0}{s+r_1}\frac{1}{s+r_0} = \frac{A_0}{s+r_0} + \frac{A_1}{s+r_1} + \frac{A_2}{s+r_2})$$
$$\Rightarrow P_2(t) = A_0\exp(-r_0 t) + A_1\exp(-r_1 t) + A_2\exp(-r_2 t)$$

Next we are going to find these three parameters $A_0 \sim A_3$:

$$A_0 = \frac{r_0 r_1}{(s+r_0)(s+r_1)(s+r_2)} \times (s+r_0)\Big|_{s=-r_0} = \frac{r_0 r_1}{(r_1-r_0)(r_2-r_0)}$$

$$A_1 = \frac{r_0 r_1}{(s+r_0)(s+r_1)(s+r_2)} \times (s+r_1)\Big|_{s=-r_1} = \frac{r_0 r_1}{(r_0-r_1)(r_2-r_1)}$$

$$A_2 = \frac{r_0 r_1}{(s+r_0)(s+r_1)(s+r_2)} \times (s+r_2)\Big|_{s=-r_2} = \frac{r_0 r_1}{(r_0-r_2)(r_1-r_2)}$$

Now we have the solution to the $P_2(t)$:

$$P_2(t) = \frac{r_0 r_1}{(r_1-r_0)(r_2-r_0)}\exp(-r_0 t) + \frac{r_0 r_1}{(r_0-r_1)(r_2-r_1)}\exp(-r_1 t) + \frac{r_0 r_1}{(r_0-r_2)(r_1-r_2)}\exp(-r_2 t)$$

To compute P_j, for $2 < j \le m$, observe that

$$\mathbf{P_j} = \frac{r_{j-1}}{s+r_j}\mathbf{P_{j-1}} = \prod_{k=0}^{j} \frac{r_{k-1}}{s+r_k},\quad r_{-1} = 1$$

This suggests a solution of the form

$$P_j(t) = \sum_{k=0}^{j} A_k \exp(-r_k t),$$

where A_k, $0 \le k \le j$ are constants of integration determined by solving this equation:

$$\prod_{i=0}^{j}\left(\frac{r_{i-1}}{s+r_i}\right) = \sum_{i=0}^{j}\frac{A_i}{s+r_i},$$

which gives:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$A_k = \frac{\prod_{i=0}^{j} r_{i-1}}{\prod_{i=0,\, i\neq k}^{j} (r_i - r_k)}$$

This gives a general form solution to $P_j(t)$:

$$\begin{aligned} P_j(t) = (r_0 r_1 \cdots r_{j-1}) \Bigg\{ & \frac{1}{(r_1-r_0)(r_2-r_0)\cdots(r_j-r_0)} \exp(-r_0 t) \\ & + \frac{1}{(r_0-r_1)(r_2-r_1)\cdots(r_j-r_1)} \exp(-r_1 t) \\ & \vdots \\ & + \frac{1}{(r_0-r_j)(r_1-r_j)\cdots(r_{j-1}-r_j)} \exp(-r_j t) \Bigg\} \end{aligned}$$

$R_m(t)$ is the probability that there is no single fault-free optimal operational path in the (u, I)-container, i.e., the container has failed according to the failure definition. Hence, the probability that j optimal paths in the (u, I)-container has failed, $R_j(t)$ is given as follows

$$R_0(t) = P_0(t)$$

$$R_j(t) = R_{j-1}(t) + P_j(t), \text{ for } 1 \leq j \leq m$$

It is difficult to give an explicit reliability expression for a large system when there are many components and a diverse reliability structure. In our case, there exist numerous numbers of nodes in a container between two given nodes u and I with the increase of the size of the star graph. One method of simplifying this situation is to calculate the MTTF (mean time to failure), T_j, computed according to the following formula:

$$\begin{aligned} T_0 &= \int_0^\infty R_0(t)dt \\ &= \frac{1}{r_0} = \frac{1}{\lambda_n m(c+m-1)} \end{aligned}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$
\begin{aligned}
T_1 &= \int_0^\infty R_1(t)dt = \int_0^\infty [P_0(t)+P_1(t)]dt \\
&= T_0 + \int_0^\infty P_0(t)dt \\
&= \frac{1}{r_0} + [\frac{r_0}{r_1-r_0}\frac{1}{r_0} + \frac{r_1}{r_0-r_1}\frac{1}{r_1}] \\
&= \frac{1}{r_0} + \frac{1}{r_1} \\
&= \frac{1}{\lambda_n m(c+m-1)} + \frac{1}{\lambda_n (m-1)(c+m-1)}
\end{aligned}
$$

In the same way, we can derive the mean time to failure for the state j,

$$
\begin{aligned}
T_j &= \int_0^\infty R_j(t)dt \\
&= \int_0^\infty [\sum_{k=0}^{j} P_k(t)]dt \\
&= \int_0^\infty [\sum_{k=0}^{j-1} P_k(t)]dt + \int_0^\infty P_j(t)dt \\
&= T_{j-1} + \int_0^\infty P_j(t)dt \\
&= T_{j-1} + \int_0^\infty \sum_{k=0}^{j} A_k \exp(-r_k t)dt \\
&= T_{j-1} + \sum_{k=0}^{j} \frac{A_k}{r_k} \\
&= T_{j-1} + \frac{r_0 r_1 \cdots r_{j-1}}{(r_1-r_0)(r_2-r_0)\cdots(r_j-r_0)}\frac{1}{r_0} + \frac{r_0 r_1 \cdots r_{j-1}}{(r_0-r_1)(r_2-r_1)\cdots(r_j-r_1)}\frac{1}{r_1} + \cdots \\
&\quad + \frac{r_0 r_1 \cdots r_{j-1}}{(r_0-r_j)(r_2-r_j)\cdots(r_{j-1}-r_j)}\frac{1}{r_j}, \text{for } 1 < j \le m
\end{aligned}
$$

The resulting T_j for a (*u*, *l*)-containers of different sizes in a S_n are shown in Table 4.1 ($\lambda_n = 10^{-5}$ / hours as in [1]), where we only consider containers with the maximum values of *m* and *c*. In other words, the container we are considering here has the largest number of nodes for a given size of the star graph. For comparisons with other popular interconnection networks, Table 4.2 shows the MTTF estimates for similar size containers in the hypercube network, where T_j captures a scenario that all faulty nodes are confined into a container of a Q_n. (In this case, we are considering such situations

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where a (u, I)-container in Q_n having n node-disjoint parallel paths of the optimal distance n between u and I).

Table 4.1. MTTF Values (Hours) for various size containers in star networks under the node failure model (Case 1), $\lambda_n = 10^{-5}$/hour.

n	# of nodes in container	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
3	4	25000	75000						
4	9	11111	27778	61111					
5	20	5000	11667	21667	81667				
6	30	3333	7500	13056	104722	121389			
7	48	2083	4583	7708	116042	122292	134792		
8	63	1587	3439	5661	144550	148254	153810	164921	
9	88	1136	2435	3950	158496	160768	163799	168344	177435

Comparisons of the star network and the hypercube are approximate because the number of nodes involved does not match, and the degradation methods of two systems differ: a container in the star graph consists of m ($m = n-1$) optimal paths of the shortest distance, $r = n-1+\left\lfloor \frac{n-1}{2} \right\rfloor$, between u and I while a container in the hypercube consists of n optimal disjoint paths of the shortest distance n.

However, when a similar number of nodes exist for the containers in two different networks, the responses to the initial few failures can be used to compare the reliability and degradation of containers in the star graph to those in the hypercube. To be

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

conservative, comparisons are generally made between the hypercube with a slightly lesser number of nodes than the respective star graphs; e.g., 56 nodes in a container having 8 disjoint paths with the optimal length 8 in Q_8 compared to 63 nodes in a container having 7 disjoint paths with the optimal length 10 in S_8.

Table 4.2. MTTF Values (Hours) for various size containers in hypercube networks under the node failure model, $\lambda_n = 10^{-5}$ /hour.

n	# of nodes in container	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
3	6	16667	41667	91667						
4	12	8333	19444	36111	69444					
5	20	5000	11250	19583	32083	57083				
6	30	3333	7333	12333	19000	29000	49000			
7	42	2381	5159	8492	12659	18214	26548	43214		
8	56	1786	3827	6207	9065	12636	17398	24541	38827	
9	72	1389	2951	4737	6820	9320	12445	16612	22862	35362

The time of the first failure is a function of number of nodes, and is therefore equivalent to that of a similar size container in the hypercube, e.g., 3,333 hours for the 30 nodes in a container of S_6 versus 3,333 hours for the 30 nodes in a container of Q_6. However, the time of the second and subsequent failures for the 30 nodes in a container in a S_6 shows an appreciable improvement over the 30 nodes in a container in a Q_6. As the number of nodes increases, the improved reliability of the star network becomes

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

more apparent. For example, for a container (63 nodes) in S_8 versus a container (56 nodes) in Q_8, the MTTF of the container in Q_8 is still less than 24,541 hours for the first seven states while the MTTF of the container in S_8 continues to climb to nearly 164,921 hours even the MTTF of the first three states of the degradation of the container in Q_8 show a little advantages over the container in S_8.

4.2.2 Case 2 $\pi(1) \neq 1$

There are c parallel disjoint paths of the shortest distance $c+m-2$, $m-c-1$ paths of distance $c+m$, and $n-m$ paths of distance $c+m+2$ between u and I ($m \leq n, c \leq \left\lfloor \frac{n}{2} \right\rfloor$, and $r = c+m-2$). Total distinct nodes in a (u, I)-container are $k = c(r-1)$ except u and I.

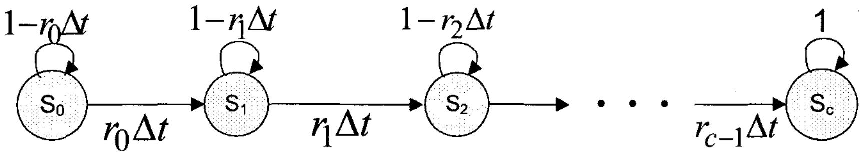


Figure 4.4. State diagram in the node failure model for a (u, I)-container in a S_n (Case 2).

This case is the same as Case 1 except the different numbers of disjoint paths and nodes in the container between u and I. Hence, the Markov analysis of the degradation of a container in S_n under this case can be implemented similarly to Case 1. The state diagram for this case under the node failure model is shown in Fig. 4.4. The state transition rates are given as

$$r_j = \lambda_n (c-j)(r-1), \ 0 \leq j < c.$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The corresponding MTTF to this case are derived based on the facet $R_j(t) = R_{j-1}(t) + P_j(t)$, for $1 \le j \le c$ as

$$T_0 = \int_0^\infty R_0(t)dt = \frac{1}{r_0} = \frac{1}{\lambda_n c(c+m-3)}$$

$$T_j = T_{j-1} + \sum_{k=0}^{j} \frac{A_k}{r_k}, \text{ for } 1 \le j \le c$$

Table 4.3. MTTF values (Hours) for various size containers in star networks under the node failure model (case 2), $\lambda_n = 10^{-5}$/hour.

n	# of nodes in container	T_0	T_1	T_2	T_3	T_4	T_5	T_6
4	6	16667	50000					
6	18	5556	13889	30556				
9	40	2500	5833	10833	20833			
10	60	1667	3750	6528	10694	19028		
12	90	1111	2444	4111	6333	9667	16333	
14	126	794	1720	2831	4220	6071	8849	14405

The resulting T_j for a (*u*, *I*)-container of different sizes are shown in Table 4.3 ($\lambda_n = 10^{-5}$ / hours as in [1]). Since the Markov analysis for this case is the same as Case 1 except different transition rates, the analysis for the link failure model and the combined node/link failure model will only consider the first case where $\pi(1) = 1$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

And for brevity, the definition of the container is not going to be defined in next two fault models.

4.3 Link failure model

This section considers the robustness of the star network under the link failure model with at most F link failures, where $|F|$ represents the set of faulty links. Nodes are assumed to be perfect under this model. As with link failures, the interest is in the system configuration that has at least one operational optimal path between u and I in the presence of link failures. Total distinct links existing in the container is $k = m(c+m)$ except u and I. Begin with a fault-free container in a S_n, where the links can fail s-independently at a constant rate; then find when the container does not have an operational optimal path.

Compared with Case 1 in the node failure model, the only difference is that we consider mr links here instead of $m(r-1)$ nodes in the node failure model while maintaining m disjoint optimal paths in a container between two given nodes u and I. Therefore, the Markov analysis of the degradation of a container in a S_n under the link failure model can be performed similarly as Case 1 in the node failure model. The state diagram under the link failure model is shown in Fig. 4.5.

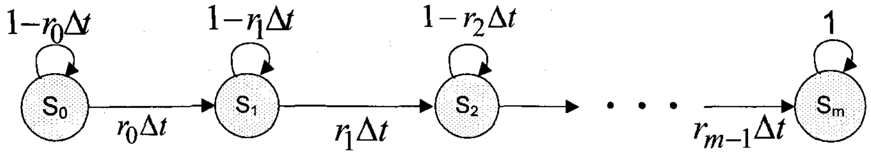


Figure 4.5. State diagram in the link failure model for a (u, I)-container in a S_n.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The state transition expressions are:

$$r_j = \lambda_l(m-j)r,\ 0 \le j \le m.$$

The solutions of the state equations are similar to those for the node-failure case. The initial and final conditions for the probabilities are:

$$P_0(0) = 1,\ P_j(0) = 0 \text{ for } 1 \le j \le m$$

$$P_j(\infty) = 0 \text{ for } 0 \le j < m,\ P_m(\infty) = 1$$

$$P_0(t) = \exp(-r_0 t),\ r_0 = \lambda_n m(c+m-1)$$

Once $P_j(t),\ 0 \le j \le m$ are determined, the expressions for T_j are derived similarly as that in the node failure model.

$$T_0 = \frac{1}{\lambda_l m(c+m-1)}$$

$$T_j = T_{j-1} + \sum_{k=0}^{j} \frac{A_k}{r_k},\ \text{for } 1 \le j \le m$$

The resulting T_j for (*u, I*)-containers of different sizes in star networks of various dimensions are shown in Table 4.4; ($\lambda_l = 10^{-6}$/hour as in [1]). Table 4.5 shows the reliability estimates (MTTF) for similar sized containers in hypercube. In this model, comparisons are made between hypercube with a slightly more number of links than the respective star graph; e.g., 36 links in a container having 6 disjoint paths with the optimal length 6 in Q_6 compared to 35 links in a container having 4 disjoint paths with the optimal length of 5 in S_6.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 4.4. MTTF values (Hours) for various size containers in star networks under the link failure model, $\lambda_l = 10^{-6}$/hour.

n	# of links in container	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
3	6	166667	500000						
4	12	83333	208333	458333					
5	24	41667	97222	180556	347222				
6	35	28571	64286	111905	183333	326190			
7	54	18519	40741	68519	105556	161111	272222		
8	70	14286	30952	50952	75952	109286	159286	259286	
9	96	10417	22321	36210	52877	73710	101488	143155	226488

The time of the first failure is a function of the number of links, and is therefore close to that of a similar size container in hypercube, e.g., 28,571 hours for the 35 links in a container of a S_8 versus 27,778 hours for the 36 links in a container of a Q_6. However, the time of the second and subsequent failures for the 35 links of a S_6 shows the appreciable improvement over the 36 links of a Q_6. As the number of links increases, the improved reliability of the star network becomes more apparent. For example, the mean time to the fourth link failure in the container (35 links) in a S_6 has climbed to 326,190 hours, while the mean time to the fourth link failure in the container (36 links) in a Q_6 just climbed to 241,667 hours. And this trend is believed to continue to improve for the MTTF of subsequent failures.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 4.5. MTTF values (Hours) for various size containers in hypercube networks under the link failure model, $\lambda_l = 10^{-6}$/hour.

n	# of links in container	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
3	9	111111	277778	611111						
4	16	62500	145833	270833	520833					
5	25	40000	90000	156667	256667	456667				
6	36	27778	61111	102778	158333	241667	408333			
7	49	20408	44218	72789	108503	156122	227551	370408		
8	64	15625	33482	54315	79315	110565	152232	214732	339732	
9	81	12346	26235	42108	60626	82848	110626	147663	203219	314330

4.4 Combined node and link failure model

So far cases were considered where either nodes or links could fail. Realistically, all network components can fail. A combined model is developed by assuming *s*-independent and different failure rates for nodes and links in the star networks. All nodes have the same probability of failing λ_n, while all links have the same probability failing λ_l. Total distinct components including nodes and links existing in a container is $k = m[(c+m-1)+(c+m)]$ except *u* and *I*.

Similar analysis as the node failure model in Section 4.2 is carried out here. The state diagram under the combined node/link failure model is shown in Fig. 4.6. The state transition rates are given as follows:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$r_j = \lambda_n(m-j)(c+m-1) + \lambda_l(m-j)(c+m), \quad 0 \le j \le m.$$

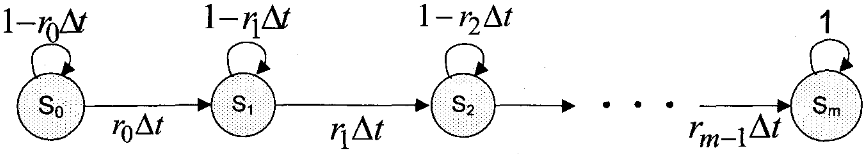


Figure 4.6. State diagram in the combined node and link failure model for a (*u*, *I*)-container in a S_n.

The state transition analysis will be same as the node failure model. The corresponding MTTF to each state are given as follows:

$$T_0 = \frac{1}{\lambda_n \cdot m \cdot (c+m-1) + \lambda_l \cdot m \cdot (c+m)}$$

$$T_j = T_{j-1} + \sum_{k=0}^{j} \frac{A_k}{r_k}, \text{ for } 1 \le j \le m$$

The resulting reliability estimates for different containers in star networks and hypercubes of various dimensions are shown in Tables 4.6 and 4.7. Comparisons of the star graph and the hypercube under the combined node/link failure model, is approximately done between similar size containers; e.g., 44 elements (including nodes and links) in a container having 4 disjoint paths with the shortest distance 6 in a S_5 compared with 45 elements in a container having 5 disjoint paths with the shortest distance 5 in a Q_5.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 4.6. MTTF values (Hours) for various size containers in star networks under the combined node/link failure model, $\lambda_n = 10^{-5}$/hour, $\lambda_l = 10^{-6}$/hour.

n	# of elements in container	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
3	10	21739	65217						
4	21	9804	24510	53922					
5	44	4464	10417	19345	37202				
6	65	2985	6716	11692	19154	34080			
7	102	1873	4120	6929	10674	16292	27528		
8	133	1429	3095	5095	7595	10929	15929	25929	
9	184	1025	2196	3562	5201	7250	9982	14081	22278

Table 4.7. MTTF values (Hours) for various size containers in hypercube networks under the combined node/link failure model, $\lambda_n = 10^{-5}$/hour, $\lambda_l = 10^{-6}$/hour.

n	# of elements in container	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
3	15	14493	36232	79710						
4	28	7353	17157	31863	61275					
5	45	4444	10000	17407	28519	50741				
6	66	2976	6548	11012	16964	25893	43750			
7	91	2132	4620	7605	11336	16311	23774	38699		
8	120	1603	3434	5571	8135	11340	15614	22024	34844	
9	153	1248	2653	4258	6131	8378	11187	14932	20550	31786

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The time of the first two failures in S_5 is slight higher than Q_5, but the MTTF of the subsequent failures for 44 elements in a container in a S_5 show the appreciable improvements over the 45 elements in a container in a Q_5. For example, the mean time to the third failures (either node or link) in a container in a S_5 (44 elements) has climbed to 37,202 hours, while the mean time to the third failures (either node or link) in a container (45 elements) in a Q_5 is still 28,519 hours.

4.5 Conclusion

In this chapter, the robustness of star networks was studied under the node, link, and combined node/line failure models. The degradation of a container in a S_n having at least one operational optimal path with the shortest distance between two given nodes u and I was examined to measure the system effectiveness in the presence of failures. For each failure model, two different cases depending on the relative positions of the source & destination nodes were considered to assess the star network. The states of the degradation of a container were modeled by a Markov chain. The solution to each transition state was derived, and values of MTTF (mean time to failure) for each failure model were computed and compared with similar size containers in the hypercube.

References

[1] S. Abraham and K. Padmanabhan, "Reliability of the hypercube," in *Proc. of International Conference on Parallel Processing*, 1988, pp.90-94.

[2] R. Billinton, R. Allan, Reliability evaluation of engineering systems, 2nd Edition, Plenum Press, 1992.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[3] K. Fitzgerald, S. Latifi, and P.K. Srimani, “Reliability modeling and assessment of the star-graph networks,” *IEEE Transactions on Reliability*, vol.51, no.1, March 2002, pp.49-57.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER 5

SUBSTAR RELIABILITY ANALYSIS

In this chapter, we derive an upper bound on the (n-1)-star reliability in a S_n using the probability fault model. We also compute an approximation on the (n-1)-star reliability by considering only disjoint (n-1) stars under the fixed partitioning. The numerical results show that the (n-1)-star reliabilities under the probability fault model and the approximation approach are in good agreement especially for the low value of the node reliability. And the numerical results are also shown to be consistent with and close to the simulation results. Conservative comparisons are made where possible between the (n-1)-network reliability of similar size star graphs and hypercubes.

Notation:

N: $N=n!$, the number of nodes in a S_n.

p: node reliability defined as the probability that the node is operational at time t.

$S_{n-1}(a_i)$: defined as $X^{i-1}a_iX^{n-i}$, which is a (n-1)-star in a S_n such that the i^{th} position of its label has the fixed value a_i, where $a_i \in \{1,2,3,\cdots,n\}$ and $i \in \{2,3,4,\cdots,n\}$, and all other positions are assigned with X's and superscripts are the repetition factors.

$R_{n-1}(a_i)$: defined as the reliability of the (n-1)-star $S_{n-1}(a_i)$.

$R_{n,n-m}(p)$: defined as the probability that there exists a fault-free S_{n-m} in a S_n.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$R_{n,n-1}(p,i)$: defined as the $R_{n,n-1}(p)$ under the fixed partitioning along the i^{th} dimension, where $2 \le i \le n$.

f: number of faulty nodes in a S_n.

$\binom{n}{m} = \frac{n(n-1)\cdots(n-m+1)}{m!}$: number of combinations of *m* components selected from a set of *n* components.

5.1 Background

As the size of a system grows, the probability of a fault occurring in the system increases. It is important to quantify the effect of the faults, so the fault-tolerant design can be pursued. Normally, reliability is used to evaluate the fault tolerance of a multiprocessor system. The *reliability* of a system as a function of time, *R(t)*, is defined as the probability that the system has survived the interval [t_0, t], given that it was operational at time t_0. A traditional measure of the reliability evaluation is *terminal reliability*, such as *all terminal reliability* and *distance reliability*, of a computer network [6][7]. Others are *task-based reliability*, defined as the probability that some minimum number of connected nodes are available in the system for the task execution [3], and *substar reliability* [2], defined as the probability that a fault-free subnetwork (a smaller dimension network with the same topological properties as the original one) is still available in the network in the presence of a tolerable number of faults. A similar idea with the aim of finding the minimum number of failed nodes or links to destroy all available substars has been studied and reported in [5]. In designing parallel processors using the star network as the interconnection topology as well as in designing real

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

applications on such processors, the estimates of these reliabilities are important in choosing algorithms and predicting their performance under different failure conditions. Among these reliability measures, the substar reliability is the most practical one because a user in the current star multiprocessors is given a specific substar for the execution of his/her program.

In this chapter, we adopt the probability fault model (originally proposed in [2] to derive the subcube reliability in the hypercube network) to study the substar reliability of the star graph. An upper bound for the $R_{n,n-1}(p)$ using the probability fault model is derived. The $R_{n,n-1}(p)$ is obtained by forming all available distinct S_{n-1}'s first, and then applying the principle of inclusion and exclusion [1] to get an upper bound while only considering the first three terms in the reliability formula since the fourth term is negative and dominate all the remaining terms. Meanwhile, an approximation on the $R_{n,n-1}(p)$ is obtained by considering only disjoint (n-1)-stars under the partitioning along the fixed dimension. Numerical results show that the probability fault model and the approximation approach are in good agreement especially for the low value of the node reliability. A search algorithm is developed to find the $R_{n,n-1}(p)$ in the star graph under a given number of node faults.

The reliability of the hypercube network is well known. Models exist to analyze the reliability of the hypercube under both node and link failure schemes [2][4][6]. To compare the reliability of the star graph with that of the hypercube, a similar analysis of impacts of node failures on the star graph is provided here; the number of nodes in the star graph is not the same as that for the hypercube interconnection network. The star graph is based on a factorial growth in the number of nodes, while the hypercube is based

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

on a power-of-2 growth in the number of nodes. This makes a direct comparison difficult; however conservative comparisons are made where possible between the reliability of similar size star graphs and hypercubes.

We assume that processors in the system have the homogenous reliability function. The failures of processors are assumed to be statistically independent and link failures are negligible compared to the processor failures. The reliability function of each node can have any failure distribution. The node reliability function can include the maintenance and repair capabilities of the node.

5.2 Analysis of the (*n*-1)-star reliability under the probability fault model

In this section, an upper bound on the (*n*-1)-star reliability under the probability fault model is derived first. In the probability fault model, the probability that a substar is operational is represented by the reliability of processors in the substar. The (*n*-*m*)-star reliability of an *n*-dimensional star graph, can be formulated as the union of the probabilistic events that all possible (*n*-*m*)-stars are operational. Since the terms in the (*n*-*m*)-star reliability obtained above may not be mutually disjoint, a technique to convert the reliability formula into one with only mutually disjoint terms is needed. The basic method used to compute the network reliability in the probability fault model is called the *Principle of Inclusion and Exclusion* (*PIE*). This principle is not efficient for calculating the reliability of the general networks. However, we show that it is useful for the star network reliability analysis.

To derive the (*n*-1)-star reliability in this model, all distinct $K_{n-1}^{n} = n(n-1)$ (*n*-1)-stars are formed first. Then the (*n*-1)-star reliability is expressed as the union of the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

reliability of these $n(n-1)$ (n-1)-stars. Since the probabilistic terms in the expression of the (n-1)-star reliability are not mutually disjoint, the key to calculate the (n-1)-star reliability is to convert the original reliability expression into one containing only disjoint terms. Let C_i denote the probability that one (n-1)-star is operational, then the (n-1)-star reliability can be represented according to the *PIE* as follows:

$$\begin{aligned} R_{n,n-1}(p) = & \sum_{i=0}^{n(n-1)-1} C_i + (-1) \sum_{i,j=0,1,\cdots,n(n-1)-1}^{i \neq j} C_i C_j \text{ (pair)} \\ & + (-1)^2 \sum_{i,j,k=0,1,\cdots,n(n-1)-1}^{i \neq j \neq k} C_i C_j C_k \text{ (triple)} \\ & + (-1)^3 \sum_{i,j,k,l=0,1,\cdots,n(n-1)-1}^{i \neq j \neq k \neq l} C_i C_j C_k C_l \text{(quadra)} + \cdots \\ & + (-1)^{n(n-1)-1} \prod_{i=0}^{n(n-1)-1} C_i \end{aligned} \tag{1}$$

Each C_i can be represented by the reliability of the $(n-1)!$ nodes in its corresponding S_{n-1}, $p^{(n-1)!}$. In the following, we derive an upper bound on the $R_{n,n-1}(p)$ in a S_n. We consider a simple case first. The S_1 reliability can be easily obtained as $R_{n,1}(p) = 1-(1-p)^N$ because the only instance where there is no fault-free S_1 in a S_n is when all the nodes are faulty. Before deriving the main result, we need the following lemmas.

Since terms in (1) are not mutually disjoint, we need to find the common nodes in any two or more S_{n-1}'s by considering the intersection between them to derive the $R_{n,n-1}(p)$. To find these common nodes between any S_{n-1}'s, first we study how distinct S_{n-1}'s pair up. In the following, three different ways in which S_{n-1}'s pair up are listed:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(a) paired S_{n-1}'s with the empty intersection where fixed digits in the (n-1)-stars are distinct and in the same position. For instance, the empty intersection between $XXX4$ and $XXX3$ in a S_4.

(b) paired S_{n-1}'s with the empty intersection where fixed digits in the (n-1)-stars are the same and in different positions. For instance, the empty intersection between $XXX4$ and $XX4X$ in a S_4.

(c) paired S_{n-1}'s with the non-empty intersection where fixed digits in the (n-1)-stars are distinct and in different positions. For instance, the non-empty intersection between $XXX4$ and $XX3X$ in a S_4.

Based on the construction conditions of the pairs between distinct S_{n-1}'s listed above, the number of total pairs for each case in a S_n are given as follows:

(a) $\binom{n}{2}(n-1)$ since two distinct fixed digits can be selected from n distinct digits, and the same position can be the i^{th} one, where $2 \le i \le n$.

(b) $n\binom{n-1}{2}$ since the fixed digit can be any of n distinct digits, and two different positions can be selected from these i positions, where $2 \le i \le n$.

(c) $2\binom{n}{2}\binom{n-1}{2}$ since two distinct fixed digits can be selected from n distinct digits, and two different positions can be selected from these i positions, where $2 \le i \le n$.

The total number of above-mentioned S_{n-1} pairs in a S_n equals to $\binom{n(n-1)}{2}$. For example, there are 15 pairs of S_2's in a S_3 as follows:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(a) $(XX3, XX2)$, $(XX3, XX1)$, $(XX2, XX1)$, $(X3X, X2X)$, $(X3X, X1X)$, $(X2X, X1X)$

(b) $(XX3, X3X)$, $(XX2, X2X)$, $(XX1, X1X)$

(c) $(XX3, X2X)$, $(XX3, X1X)$, $(XX2, X1X)$, $(XX2, X3X)$, $(XX1, X3X)$, $(XX1, X2X)$

Cases (a) and (b) can be combined and extended to the general scenario where the intersection between m $(2 \le m \le n)$ S_{n-1}'s is empty, while case (c) can be generalized to the scenario where m distinct S_{n-1}'s intersect into a S_{n-m}. Thus, the above results can be generalized to the following lemmas.

LEMMA 1. *There are* $(2n-m-1)\binom{n}{m}$ *ways that the intersection between m disjoint (n-1)-stars is empty.*

Proof: Cases (a) and (b) can be generalized to the intersection between m disjoint S_{n-1}'s: the numbers of these S_{n-1}'s of type (a) or (b) are $(n-1)\binom{n}{m}$ and $n\binom{n-1}{m}$ respectively. Adding them gives $(2n-m-1)\binom{n}{m}$. The intersection between these S_{n-1}'s is empty since they are disjoint to each other. Q.E.D.

LEMMA 2. *There are* $m!\binom{n}{m}\binom{n-1}{m}$ *ways that m distinct* S_{n-1}'s *in a n-star intersect into a (n-m)-star.*

Proof for this lemma can be obtained directly from the process of generalizing type (c) S_{n-1}'s, where m fixed digits for each of these S_{n-1}'s are distinct and in different m positions. This number happens to be the same as the number of distinct S_{n-m}'s in a S_n:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$\binom{n-1}{m}\left[\binom{n}{m}m!\right]$.

LEMMA 3. *The total number of nodes in m distinct* S_{n-1}'s *(type c) which intersect into a (n-m)-star equals to* $\sum_{i=1}^{m}(-1)^{i-1}\binom{m}{i}(n-i)!$.

Proof: We shall see that *m* type (c) S_{n-1}'s intersect into a (*n*-*m*)-star. Thus, by the *PIE*, the number of nodes in these S_{n-1}'s can be obtained as $\sum_{i=1}^{m}(-1)^{i-1}\binom{m}{i}(n-i)!$, i.e., the sum of the number of nodes in these S_{n-1}'s, subtracted by the number of nodes in the intersections between any two S_{n-1}'s, added by the number of nodes in the intersections between any three S_{n-1}'s, and so on. Q.E.D.

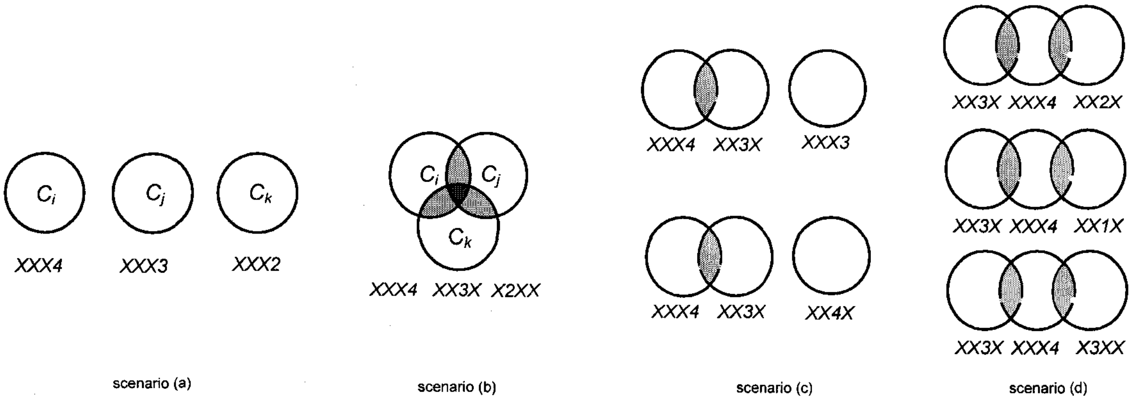
For example, the intersection between three (5-1)-stars (type c) *XXXX*5, *XXX*4*X*, and *XX*3*XX*, is a (5-3)-star *XX*234. It is easy to see that the total number of nodes in these three (5-1)-stars equals to $3\times4!-2\times3!+1\times2!=62$.

To derive the $R_{n,n-1}(p)$, we not only need to consider scenarios where (*n*-1)-stars overlap with either empty or non-empty intersections as stated in Lemmas 1~3, also scenarios where (*n*-1)-stars intersect in other numerous ways. For example, there are four scenarios where three distinct (*n*-1)-stars intersect, and more than eight scenarios where four distinct (*n*-1)-stars intersect, etc. To keep the problem tractable, we only consider scenarios where up to three (*n*-1)-stars are selected to derive the $R_{n,n-1}(p)$. Before deriving the general formula to find the total number of scenarios when three (*n*-1)-stars intersect in a S_n, we start with a simple example as the following.

Besides the two scenarios (a) and (b) shown in Fig. 5.1 as stated in Lemmas 1 and 2, there are two more scenarios where three out of twelve (4-1)-stars intersect in a S_4.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

- Scenario (c): there are 36 ways to select two (4-1)-stars such that they intersect into a (4-2)-star according to Lemma 2. For each of them, there are two choices to select the third (4-1)-star which dose not intersect with either of them. For instance, $XXX4, XX3X, XXX3$ and $XXX4, XX3X, XX4X$.
- Scenario (d): similar to (c), there are 36 ways to select two (4-1)-stars such that they intersect into a (4-2)-star. For each of them, there are three choices to select the third (4-1)-star which will only intersect with one of the two earlier selected (4-1)-stars. For instance, $XXX4, XX3X, XX2X$, $XXX4, XX3X, XX1X$, and $XXX4, XX3X, X3XX$.



scenario (a) scenario (b) scenario (c) scenario (d)

Figure 5.1. Scenarios where three (4-1)-stars intersect.

The results of the above example now can be generalized to the scenario where three S_{n-1}'s intersect in a S_n in the following lemma.

LEMMA 4. *There are* $(4n-6)\binom{n}{2}\binom{n-1}{2}$ *ways that three* S_{n-1}'s *intersect similarly to scenarios (c) and (d) in Fig. 5.1.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Proof: Lemma 2 gives $2!\binom{n}{2}\binom{n-1}{2}$ ways that two S_{n-1}'s intersect with each other. For each of these ways, there are two choices to select the third S_{n-1} which does not intersect with either of the earlier selected S_{n-1}'s similarly to scenario (c) in Fig. 5.1; while there are $\binom{n-2}{1}+\binom{n-3}{1}=2n-5$ choices to select the third S_{n-1} which intersect with only one of the earlier selected S_{n-1}'s similarly to scenario (d) in Fig. 5.1. Adding them gives the total number of ways that three S_{n-1}'s intersect similarly to scenarios (c) and (d) in Fig. 5.1: $(4n-6)\binom{n}{2}\binom{n-1}{2}$. Q.E.D.

Summing numbers in Lemmas 1, 3, and 4 when m equals to 3 gives the total ways that three (n-1)-stars intersect: $\binom{n(n-1)}{3}$. The number of nodes in these three (n-1)-stars under each scenario are $3(n-1)!$, $\sum_{i=1}^{3}(-1)^{i-1}\binom{3}{i}(n-i)!$, $3(n-1)!-(n-2)!$, and $3(n-1)!-2(n-2)!$. The corresponding probabilities of these three (n-1)-stars under each scenario are given as $(2n-4)\binom{n}{3}p^{3(n-1)!}$, $4\binom{n}{2}\binom{n-1}{2}p^{3(n-1)!-(n-2)!}$, $2\binom{n}{2}\binom{n-1}{2}(2n-5)\,p^{3(n-1)!-2(n-2)!}$, and $6\binom{n}{3}\binom{n-1}{3}p^{\sum_{i=1}^{3}(-1)^{i-1}\binom{3}{i}(n-i)!}$, respectively.

Before using these Lemmas to derive the $R_{n,n-1}(p)$, two simple examples are introduced as follows.

Example 1. For n=3, there are 6 distinct (3-1)-stars in a S_3. Based on the *PIE*, the $R_{3,3-1}(p)$ can be computed as:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$
\begin{aligned}
R_{3,3-1}(p) = {} & \sum_{i=0}^{5} C_i - \sum_{i,j=0,1,\cdots,5}^{i\neq j} C_iC_j + \sum_{i,j,k=0,1,\cdots,5}^{i\neq j\neq k} C_iC_jC_k - \sum_{i,j,k,l=0,1,\cdots,5}^{i\neq j\neq k\neq l} C_iC_jC_kC_l \\
& + \sum_{i,j,k,l,m=0,1,\cdots,5}^{i\neq j\neq k\neq l\neq m} C_iC_jC_kC_lC_m - \prod_{i=0}^{5} C_i
\end{aligned},
$$

the $R_{3,3-1}(p)$ can be interpreted as the sum of the reliability of (3-1)!=2 nodes in six distinct (3-1)-stars, subtracted by the reliability of nodes in the union of fifteen pairs of distinct (3-1)-stars, added by the reliability of nodes in the union of twenty triple of distinct (3-1)-stars, and so on. Each term in the $R_{3,3-1}(p)$ is listed as:

- First term: $C_0 + C_1 + C_2 + C_3 + C_4 + C_5 = 6p^2$
- Second term (pairs): $$\begin{aligned} & C_0C_1 + C_0C_2 + C_0C_3 + C_0C_4 + C_0C_5 + C_1C_2 + C_1C_3 + C_1C_4 + \\ & C_1C_5 + C_2C_3 + C_2C_4 + C_2C_5 + C_3C_4 + C_3C_5 + C_4C_5 \\ & = 9p^4 + 6p^3 \end{aligned}$$
- Third term (triple): $$\begin{aligned} & C_0C_1C_2 + C_0C_1C_3 + C_0C_1C_4 + C_0C_1C_5 + C_0C_2C_3 + C_0C_2C_4 + \\ & C_0C_2C_5 + C_0C_3C_4 + C_0C_3C_5 + C_0C_4C_5 + C_1C_2C_3 + \\ & C_1C_2C_4 + C_1C_2C_5 + C_1C_3C_4 + C_1C_3C_5 + C_1C_4C_5 + \\ & C_2C_3C_4 + C_2C_3C_5 + C_2C_4C_5 + C_3C_4C_5 \\ & = 2p^6 + 12p^5 + 6p^4 \end{aligned}$$
- Fourth term (quadra-): $$\begin{aligned} & C_0C_1C_2C_3 + C_0C_1C_2C_4 + C_0C_1C_2C_5 + C_0C_1C_3C_4 + \\ & C_0C_1C_3C_5 + C_0C_1C_4C_5 + C_0C_2C_3C_4 + C_0C_2C_3C_5 + \\ & C_0C_2C_4C_5 + C_0C_3C_4C_5 + C_1C_2C_3C_4 + C_1C_2C_3C_5 + \\ & C_1C_2C_4C_5 + C_1C_3C_4C_5 + C_2C_3C_4C_5 \\ & = 9p^6 + 6p^5 \end{aligned}$$
- Fifth term (five): $$\begin{aligned} & C_0C_1C_2C_3C_4 + C_0C_1C_2C_3C_5 + C_0C_1C_2C_4C_5 + C_0C_1C_3C_4C_5 + \\ & C_0C_2C_3C_4C_5 + C_1C_2C_3C_4C_5 = 6p^6 \end{aligned}$$
- Sixth term (six): $C_0C_1C_2C_3C_4C_5 = p^6$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Adding above six terms gives the exact value of the $R_{3,3-1}(p)$ as $R_{3,3-1}(p)=6p^2-6p^3-3p^4+6p^5-2p^6$.

Example 2. For n=4, there are 12 distinct S_{4-1}'s in a S_4. Each C_i can be represented by the reliability of (4-1)!=6 nodes in its corresponding S_{4-1}, for $0 \le i \le 11$. Based on the *PIE*, the $R_{4,4-1}(p)$ can be computed as

$$R_{4,4-1}(p)=\sum_{i=0}^{11} C_i - \sum_{i,j=0,1,\cdots,11}^{i\neq j} C_iC_j + \sum_{i,j,k=0,1,\cdots,11}^{i\neq j\neq k} C_iC_jC_k - \sum_{i,j,k,l=0,1,\cdots,11}^{i\neq j\neq k\neq l} C_iC_jC_kC_l + \cdots - \prod_{i=0}^{11} C_i ,$$

where each term is the reliability of the union of several S_{4-1}'s. The $R_{4,4-1}(p)$ can be interpreted as the sum of the reliability of (4-1)!=6 nodes in every S_{4-1}, subtracted by the reliability of nodes in the union of any pair S_{4-1}'s, added by the reliability of nodes in the union of any triple S_{4-1}'s, and so on. Following presents the first three terms for the $R_{4,4-1}(p)$:

- First term: $12p^6$ which corresponds to the scenario where each individual S_{4-1} is operational.
- Second term: two scenarios for any pair operational S_{4-1}'s.
    - Scenario (a): paired S_{4-1}'s with the empty intersection with the reliability $30p^{12}$.
    - Scenario (b): paired S_{4-1}'s which intersect into a S_{4-1} with the reliability $36p^{10}$.
- Third term: four union scenarios for three operational S_{4-1}'s shown in Fig. 5.2.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

- Scenarios (a): empty intersection between any pair of three $S_{4\text{-}1}$'s with the reliability $16p^{18}$.
- Scenario (b): three distinct $S_{4\text{-}1}$'s intersect with each other with the reliability $24p^{13}$.
- Scenario (c): two out of three (4-1)-stars intersect in a $S_{4\text{-}1}$ with the reliability $72p^{16}$.
- Scenario (d): one $S_{4\text{-}1}$ intersects with the other two respectively with the reliability $108p^{14}$.

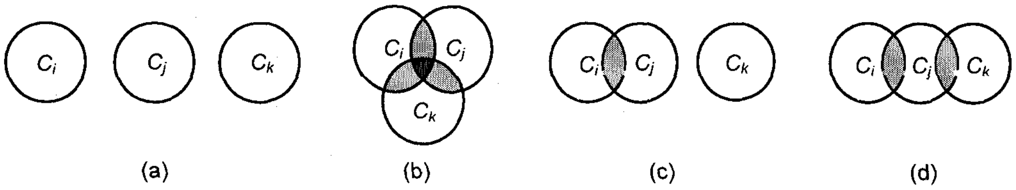


Figure 5.2. Four scenarios when three out of twelve (4-1)-stars intersect in a S_4.

As the number of the $S_{4\text{-}1}$'s in each term for the $R_{4,4-1}(p)$ increased, scenarios where (n-1)-stars intersect become more complicated, making it difficult to enumerate all. Thus, we give an upper bound on the $R_{4,4-1}(p)$ here since the next term is negative and dominate all the remaining ones as $R_{4,4-1}(p) \leq 12p^6 - \ (36p^{10} + 30p^{12}) + (24p^{13} + 108p^{14} + 72p^{16} + 16p^{18})$.

Theorem 1. *Given a homogeneous node reliability p in a n-star, an upper bound on the* $R_{n,n-1}(p)$ *is given as*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$R_{n,n-1}(p) \le n(n-1)p^{(n-1)!} - \binom{n}{2}\left\{(2n-3)p^{2(n-1)!} + 2\binom{n-1}{2}p^{\sum_{i=1}^{2}(-1)^{i-1}\binom{2}{i}(n-i)!}\right\}$$
$$+\binom{n}{3}\left\{2(n-2)p^{3(n-1)!} + 6(n-1)p^{(3n-4)(n-2)!} + 3(2n-5)(n-1)p^{(3n-5)(n-2)!} \right. \quad (2)$$
$$\left. +6\binom{n-1}{3}p^{\sum_{i=1}^{3}(-1)^{i-1}\binom{3}{i}(n-i)!}\right\}$$

Proof: According to the *PIE*, the probability of having a fault-free (n-1)-star is obtained as

$$\begin{aligned} R_{n,n-1}(p) = {} & \sum_{i=0}^{n(n-1)-1} R_{n-1}(a_i) + (-1) \sum_{i,j=0,1,\cdots,n(n-1)-1}^{i\neq j} R_{n-1}(a_i)R_{n-1}(a_j) + \\ & (-1)^2 \sum_{i,j,k=0,1,\cdots,n(n-1)-1}^{i\neq j\neq k} R_{n-1}(a_i)R_{n-1}(a_j)R_{n-1}(a_k) + \cdots \\ & + (-1)^{n(n-1)-1} \prod_{i=0}^{n(n-1)-1} R_{n-1}(a_i) \end{aligned} \quad (3)$$

where $a_i \in \{1,2,3,\cdots,n\}$, and $i \in \{2,3,4,\cdots,n\}$. Terms in (3) can be interpreted as, the sum of the probability that each (n-1)-star is operational, subtracted by the probabilities that any two (n-1)-stars are operational, added by the probabilities that any three (n-1)-stars are operational, and so on.

Since it is difficult to enumerate all scenarios in (3) when four or more S_{n-1}'s are selected as the size of the star graph increases, thus we give an upper bound on the $R_{n,n-1}(p)$ by only considering the first three terms in (3) as

$$\begin{aligned} R_{n,n-1}(p) \le {} & \sum_{i=0}^{n(n-1)-1} R_{n-1}(a_i) - \sum_{i,j=0,1,\cdots,n(n-1)-1}^{i\neq j} R_{n-1}(a_i)R_{n-1}(a_j) + \\ & \sum_{i,j,k=0,1,\cdots,n(n-1)-1}^{i\neq j\neq k} R_{n-1}(a_i)R_{n-1}(a_j)R_{n-1}(a_k) \end{aligned}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

because the fourth term is negative and dominate all the remaining ones. Next we derive the expressions for the first three terms in terms of p:

First term is the sum of probabilities that each (n-1)-star is operational as $n(n-1)p^{(n-1)!}$.

Second term is the union of probabilities that two S_{n-1}'s are operational. There are two possibilities: one is the S_{n-1}'s pair with the empty intersection (Lemma 1); the other is the S_{n-1}'s pair that intersects into a S_{n-2} (Lemma 2). The corresponding probabilities are $(2n-3)\binom{n}{2}p^{2(n-1)!}$ and $2\binom{n}{2}\binom{n-1}{2}p^{\sum_{i=1}^{2}(-1)^{i-1}\binom{2}{i}(n-i)!}$.

Third term is the union of the probabilities that three S_{n-1}'s are operational. There are four different possibilities same as those in Fig. 5.2: the first is that the intersection between three S_{n-1}'s is empty; the second is that three S_{n-1}'s intersect into a S_{n-3}; the third is that only two out of three S_{n-1}'s intersect into a S_{n-2}; the last is that the middle S_{n-1} intersects with the other two respectively. The corresponding probabilities under each scenario are given as $(2n-4)\binom{n}{3}p^{3(n-1)!}$ (Lemma 2), $4\binom{n}{2}\binom{n-1}{2}p^{3(n-1)!-(n-2)!}$ (Lemma 4), $2\binom{n}{2}\binom{n-1}{2}\times(2n-5)\,p^{3(n-1)!-2(n-2)!}$ (Lemma 4), and $6\binom{n}{3}\binom{n-1}{3}p^{\sum_{i=1}^{3}(-1)^{i-1}\binom{3}{i}(n-i)!}$ (Lemmas 2 and 3).

Adding the above three terms, we obtain the proof for Theorem 1. Q.E.D.

5.3 Approximation on $R_{n,n-1}(p)$ using the fixed partitioning

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The basic idea is to determine $R_{n,n-1}(p)$, the probability of having at least one operational S_{n-1} in a S_n, given that p is the reliability of each node. It is well known that the star graph is a highly robust network containing increasingly numerous substars as the size of substars decreases. Therefore, we proceed to derive $R_{n,n-1}(p)$ by a combinatorial approach.

In this section, the (n-1)-reliability is approximated by considering the probabilities of disjoint S_{n-1}'s under partitioning along the fixed dimensions. To do so, we first give a pessimistic lower bound on the $R_{n,n-1}(p)$ by only considering the set of disjoint S_{n-1}'s along one fixed dimension. However, this approximation does not take into account any disjoint S_{n-1}'s that may work correctly if we considered the partitioning along fixed dimensions other than one. Thus, a tighter approximation on the $R_{n,n-1}(p)$ is derived later by ignoring the fact that $R_{n,n-1}(p,i)$ values along two or more dimensions are not independent.

There are total $n(n-1)$ distinct S_{n-1}'s in a S_n. Partitioning a S_n along any dimension i, $2 \leq i \leq n$, will render a set of n disjoint S_{n-1}'s. The labels of these S_{n-1}'s are obtained by assigning an integer k, $1 \leq k \leq n$, to dimension i, and X to all other dimensions. Clearly if all nodes in at least one of these S_{n-1}'s are working properly with the node reliability p, then $R_{n,n-1}(p) = 1$. Unfortunately, nodes in these distinct S_{n-1}'s are not disjoint and therefore we use a method to derive an approximate figure for the $R_{n,n-1}(p)$. In doing so, we first consider only the set of disjoint S_{n-1}'s by fixing one dimension (without the loss of the generality we select the n^{th} dimension). For a given S_{n-1}, the reliability of this S_{n-1} is represented by the reliability of the (n-1)! nodes in it. For a given node reliability p,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

this can be $p^{(n-1)!}$. However, we cannot simply multiply this reliability by n to get the $R_{n,n-1}(p)$ along the n^{th} dimension. To adjust this figure, we need to subtract the probabilities that two S_{n-1}'s along the n^{th} dimension are working correctly, and then add back the probabilities that three S_{n-1}'s along the n^{th} dimension are working correctly, and so on according to *PIE*. This is equivalent to the scenario where the probability that each (n-1)-star is working correctly is disjoint to each other. Hence, a lower bound on the $R_{n,n-1}(p)$ by considering disjoint (n-1)-stars partitioned along the n^{th} dimension only, is given as:

$$R_{n,n-1}(p,n) = 1 - \left(1 - p^{(n-1)!}\right)^n \tag{4}$$

The approximation figure given in (4), however, does not take into account any disjoint S_{n-1}'s that may work correctly if we considered the partitioning along dimensions other than the n^{th} one. To improve this result, we could consider two different approaches. One is to employ a second level of *PIE* by finding the probability of at least one fault-free S_{n-1} along (i) one dimension, (ii) two dimensions, etc. This approach becomes quickly complicated as we consider several dimensions at a time due to the numerous ways that two or more S_{n-1}'s can intersect, making it very difficult to enumerate the common patterns. The second approach which yields an approximation is by ignoring the fact that the $R_{n,n-1}(p,i)$ values along two or more dimensions are not independent. After this relaxation, applying the *PIE* will give a more accurate approximation on the $R_{n,n-1}(p)$ as the following:

$$R_{n,n-1}(p) \approx \sum_{i=1}^{n-1} (-1)^{(i-1)} \binom{n-1}{i} R_{n,n-1}^{i}(p,n) \tag{5}$$

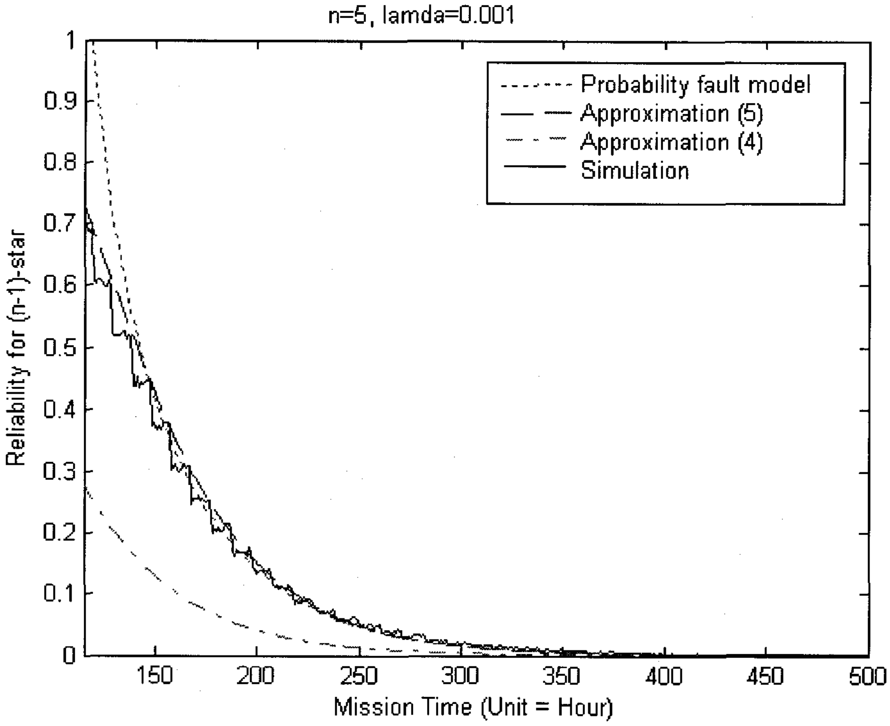
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

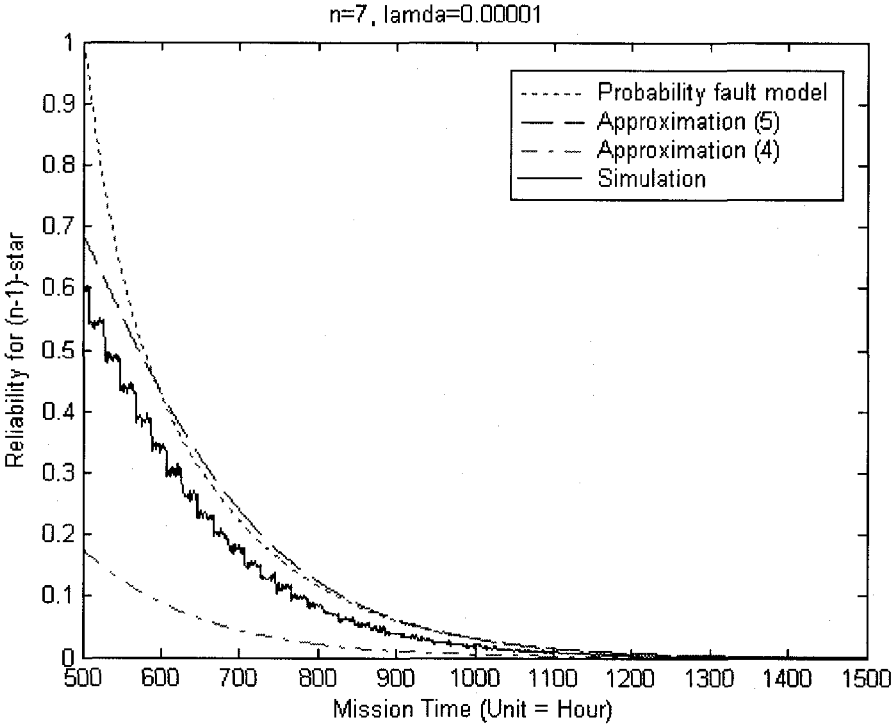
5.4 Results and analysis

In this section, we plot and compare the numerical results for the $R_{n,n-1}(p)$ of star graphs of different sizes with the simulation. The node reliability is assumed to be homogeneous and follows an exponential distribution with a constant failure rate λ (failures/hour). A search algorithm was developed to find the (n-1)-reliability in the star graph under a given number of faulty nodes. The number of nodes considered to be faulty is determined using equations $f = n![1-\exp(-\lambda t)]$. The faults were generated randomly using a random permutation generator. The ranges for n and f are chosen to be: $5 \le n \le 10$ and $0 \le f \le 100$, respectively. For each scenario, the simulation was carried out for 10,000 iterations and the results are compared with the numerical results under the probability fault model and the fixed partitioning.

Fig. 5.3 depicts the $R_{n,n-1}(p)$ using the results from the probability fault model in (2), the approximation in (4) and (5), and the simulation. The results from (4) are far below those from (5) and the simulation since only disjoint S_{n-1}'s partitioned along the n^{th} dimension are considered. It is seen that the numerical results under both the probability fault model and the approximation approach (5) are getting overlapped as the node reliability decreases, and close to the simulation results. This further verifies that the $R_{n,n-1}(p)$ under these two approaches are similarly accurate. The large gap between the probability fault model and the simulation results, especially under the large value of the node reliability, is due to the fact that the fourth term in (2) is negative and large when the size of the star graph increases.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.





Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

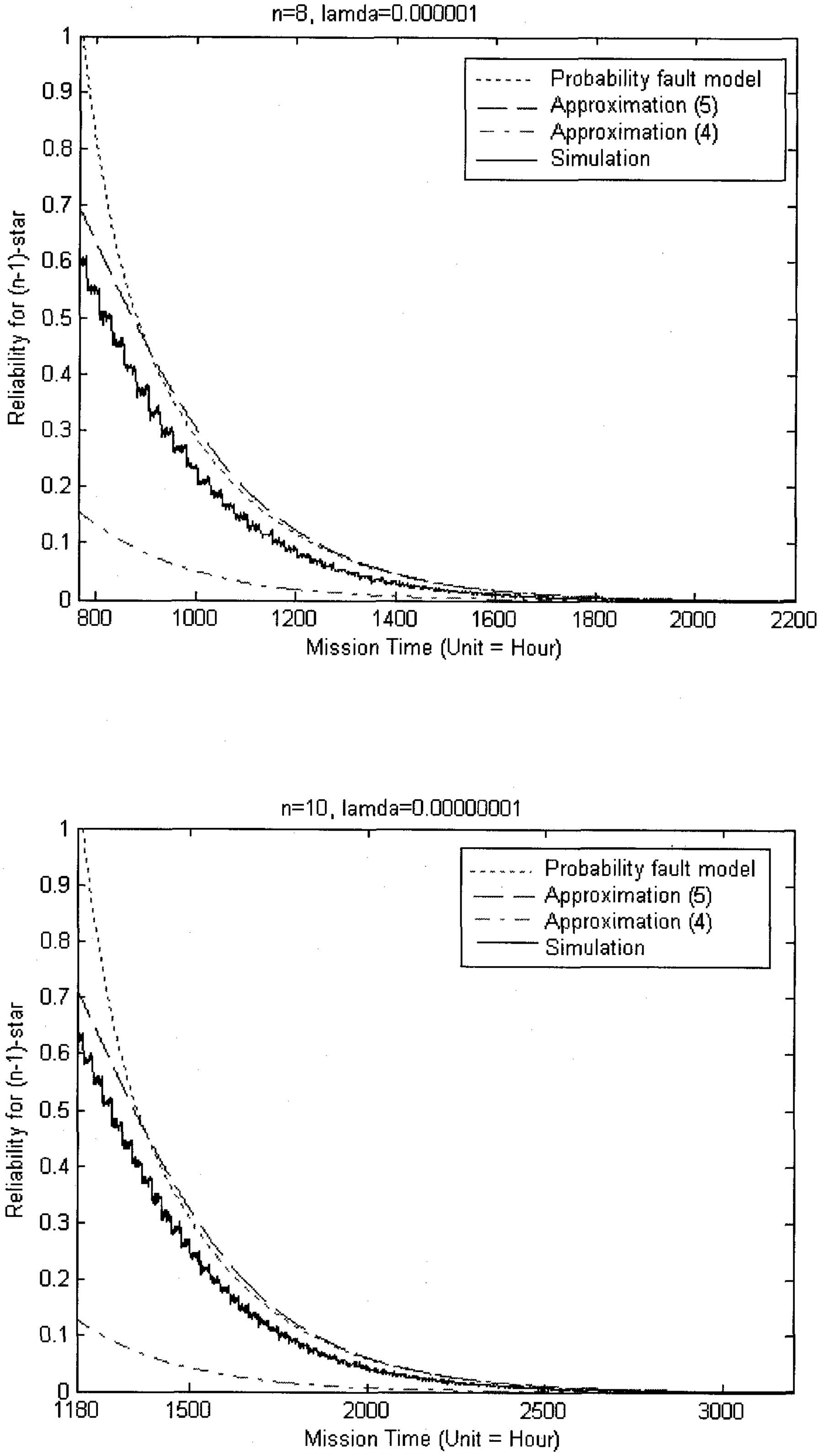


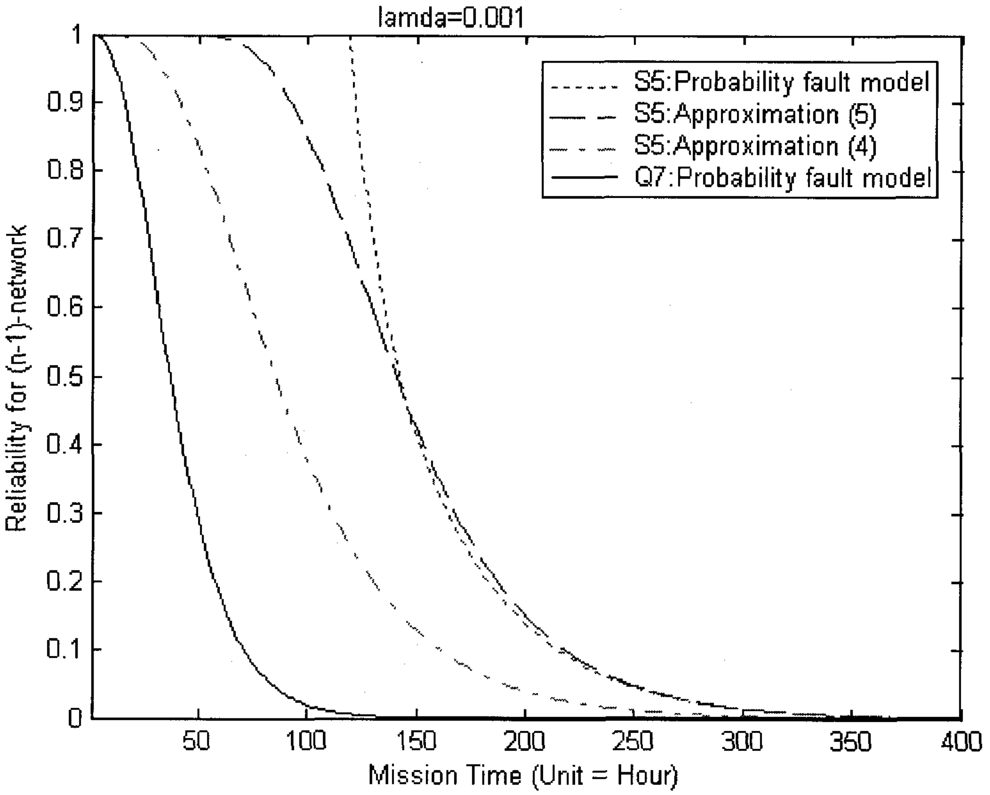
Figure 5.3. The (n-1)-reliabilities of star graphs of sizes 5, 7, 8, and 10.

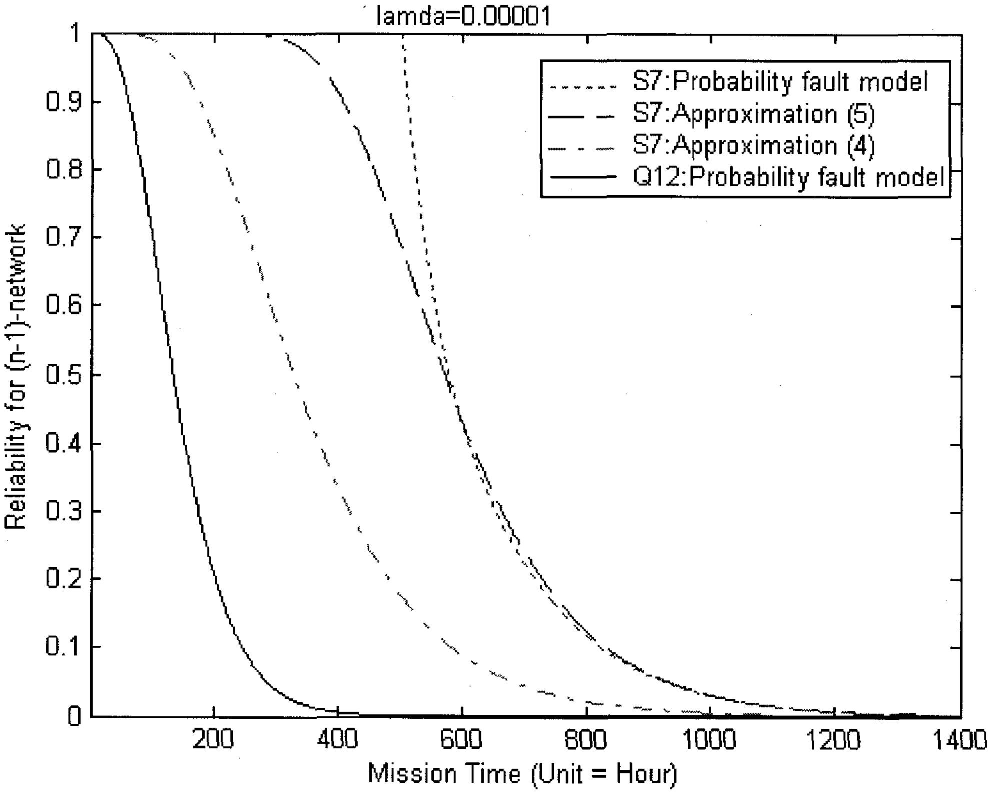
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Comparisons of the star networks and hypercubes are approximate because the number of nodes involved in both networks does not match, and the partitioning of an n-dimensional network into (n-1)-dimensional networks differs: a star graph S_n can be partitioned into $n(n\text{-}1)$ distinct (n-1)-stars in n-1 different ways, while a hypercube Q_n can be partitioned into $2n$ distinct (n-1)-cubes in n different ways. However, when a similar number of nodes exist for the two networks, the $R_{n,n-1}(p)$ can be used to compare the reliability between the star graph and the hypercube. To be conservative, comparisons are generally made between hypercube with a slightly lesser number of nodes than the respective star graphs; e.g., 32,768 nodes in a Q_{15} compared to 40,320 nodes in a S_8.

The dropping rate of the (n-1)-network reliability shown in Fig. 5.4 is faster for the hypercube than that for the star graph, e.g., the dropping rate for a Q_7 (128 nodes) versus that for a S_5 (120 nodes). This is due to the fact that there are 20 distinct $S_{5\text{-}1}$'s in a S_5, while 14 distinct $Q_{7\text{-}1}$'s in a Q_7. Furthermore, this trend continues to be even more appreciable as the size of the networks increases. For example, for a S_8 (40,320 nodes) versus a Q_{15} (32,768 nodes), the (n-1)-network reliability in Q_{15} quickly drops to zero around 600 hours, while the (n-1)-network reliability in S_8 is almost one at the instance t=600 hours and slowly decreases to zero until 2,000 hours. This is due the fact that there are $8\times(8-1)=56$ distinct $S_{8\text{-}1}$'s in a 8-dimensional star graph S_8, while $2\times 15=30$ distinct $Q_{15\text{-}1}$'s in a 15-dimensional hypercube network Q_{15}.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.





Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

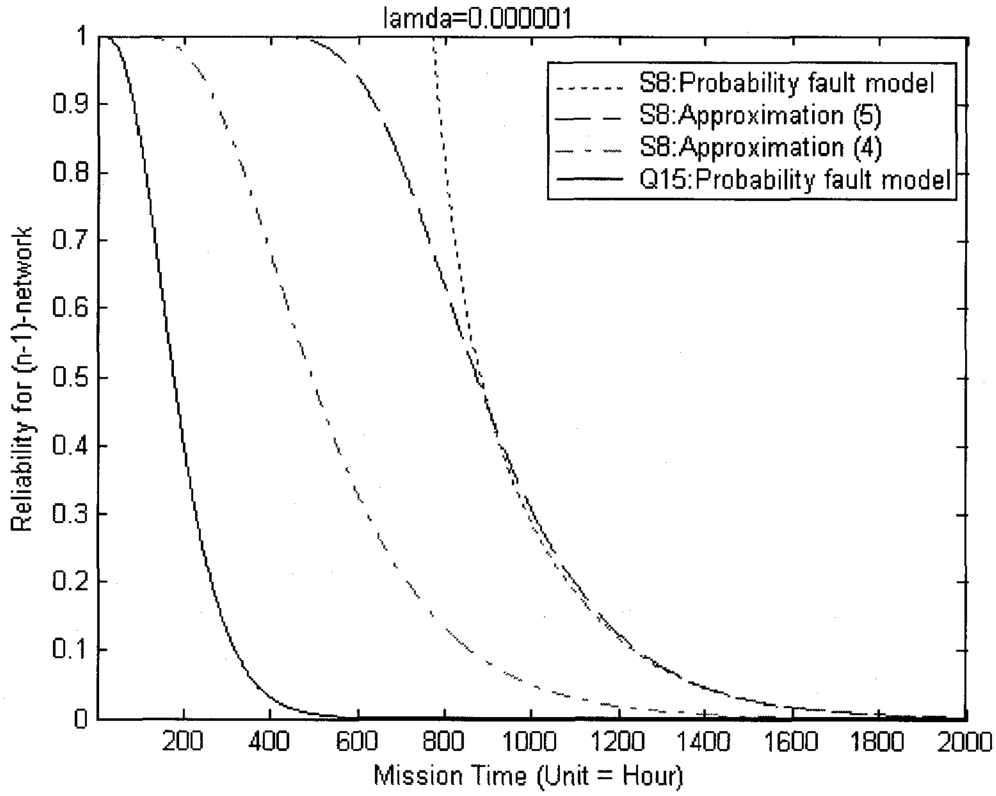


Figure 5.4. Approximate reliability comparisons between different size star graphs and hypercubes.

5.5 Conclusion

Two different methods, the probability fault model and the approximation approach, have been used to predict the substar reliability of the star interconnection network. An upper bound on the (n-1)-star reliability in a S_n using the probability fault model was derived, while an approximation on the (n-1)-star reliability by considering only disjoint (n-1)-stars under the fixed partitioning was computed. A search algorithm was developed to find the (n-1)-star reliability in the star graph for a given number of node faults. The numerical results under the probability fault model and the approximation approach were

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

shown to be in good agreement especially for the low value of the node reliability, also consistent with and close to the simulation results. Conservative comparisons between similar size star graphs and hypercubes proved that the star graph is more tolerant than the hypercube in terms of the (n-1)- network reliability.

Derivation of the (n-k)-star reliability can be represented by the reliability of (n-k)! nodes in each (n-k)-star under the probability fault model by the *PIE*, and also can be performed using the same approximation approach by considering disjoint (n-k)-stars along a fixed set of dimensions and applying the *PIE*. This task, however, is not simple (computationally) and becomes complicated as the size of the star graph and the value of k become large due to the numerous ways that (n-k)-stars intersect. A more detailed analysis using these approaches is currently under development to obtain a reasonably accurate approximation on the actual (n-k)-star reliability for k>1.

References

[1] R. Billinton and R. Allan (2nd Edition), “Reliability evaluation of engineering systems”, *Plenum Press*, 1992.

[2] Y. Chang and L. Bhuyan, “A combinatorial analysis of subcube reliability in hypercube,” *IEEE Transactions on Computers*, vol.44, no.7, July 1995, pp.952-956.

[3] C. Das and J. Kim, “A unified task-based dependability model for hypercube computers, *IEEE Transactions on Parallel and Distributed Systems*, vol.3, no.3, May 1992, pp.312-324.

[4] S. Latifi and S. Rai, “A robustness measure for hypercube networks,” in *Proc. of the 36th Midwest Symposium on Circuits and Systems*, vol.1, Aug. 1993, pp.546- 549.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[5] S. Latifi, “A study of fault tolerance in star graph,” *Information Processing Letters*, vol.102, no.5, May 2007, pp.196-200.

[6] S. Soh, S. Rai, and J.L. Trahan, “Improved lower bounds on the reliability of hypercube architectures,” *IEEE Transactions on Parallel and Distributed Systems*, vol.5, no.4, April 1994, pp.364-378.

[7] X.L. Wu, S. Latifi, and Y. Jiang, “A combinatorial analysis of distance reliability in star network,” in *Proc. of the 21st IEEE International Parallel & Distributed Processing Symposium IPDPS*, March 2007, pp.1-6.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER 6

CONCLUSION AND DISCUSSION

In this dissertation, three different reliability measures have been proposed to investigate the robustness of the star network under three different failure (node, link, and combined node/link) models, respectively. The first is the distance reliability. The second is the degradation of the container having at least one operational optimal path between two given nodes in the star graph. The most practical one is the third one: substar reliability.

The distance reliability, probability of having an operational path with the optimal distance between two given nodes u and I, poses stringent requirements on the connection of two arbitrary nodes in the star graph; i.e. not only do two nodes have to be connected, but the distance between them must be the shortest. The combinatorial method was used to determine the distance reliability when the number of faults is bounded especially for the node permutation label having a single cycle representation. For each of the failure models, two different cases depending on the relative positions of the source & destination, were analyzed to compute *DR*. Conservative *DR* comparisons with the hypercube was carried out when the shortest distance between two given nodes is the same for both the star graph and the hypercube. Even the *DR* in the hypercube is higher than that in the star graph (for a single cycle representation), this study gave us an

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

understanding of *DR* in the star graph. And the *DR* in the star graph showed an appreciable improvement with the increase the shortest distance. The *DR* in the star graph was expected to be closer to that of the hypercube when more than one cycles representation and the numerous ways of constructing disjoint paths were considered. The communication between a given node and its basic antipode was considered as a special case of *DR* to further demonstrate the fault tolerance of star networks. Lower bounds on the antipode reliability were derived and compared with the antipode reliability in the hypercube. Comparisons results showed that the antipode reliability in the star graph is higher than that in the hypercube due to the fact that there is more than one antipode for a given node when the size of the star graph is greater than 3 while a given node in hypercube has a single antipode.

The degradation of a container having at least one operational optimal path with the shortest distance between two given nodes in a star graph was examined to measure the system effectiveness in the presence of failures under three different failure models, respectively. For each failure model, two different cases depending on the symbol in the first position being "1" or not were considered to assess the star network. The states of the degradation of a container were modeled by a Markov chain. The values of MTTF for each transition state were computed and compared with the similar size containers in the hypercube. Comparisons showed that the star graph is more robust than the hypercube.

Two different methods, the probability fault model and the approximation approach, were used to derive the substar reliability. An upper bound on the $(n$-$1)$-star reliability in an n-dimensional star network under the probability fault model were derived by only considering the first three terms since the fourth one is negative and dominate the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

remaining ones, while an approximation on the (*n*-1)-star reliability by considering only disjoint (*n*-1)-stars under the partitioning along the fixed dimensions was computed. A search algorithm was developed to find the (*n*-1)-star reliability in the star graph for a given number of node faults, where faults were generated by a random permutation generator. For each scenario, the simulation was carried out for 10,000 iterations. The upper bound under the probability fault model and the approximation results by the partitioning along the fixed dimensions, were shown to be in good agreement especially for the low value of the node reliability, also consistent with and close to the simulation results. Conservative comparisons between similar size star graphs and hypercubes proved that the star graph is more tolerant than the hypercube in terms of the (*n*-1)-network reliability.

Future research work includes:

- The determination of the distance reliability between the source node with more than one cycle representation and the destination. Computer modeling to find out the distance reliability and comparisons with the hypercube are necessary to further investigate the robustness of the star graph.
- The (*n*-1)-star reliability when considering the link failure only or combined node/link failure is going to be determined in the future. Simulation results are necessary to further verify the numerical results.
- A more detailed analysis using the probability fault model or the fixed partitioning is appreciated to obtain a reasonably accurate approximation on the actual (*n*-*k*)-star reliability for $k>1$. Simulation to find the (*n*-*k*)-star reliability is also demanded to verify the numerical derivations.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

APPENDIX

MATLAB SIMULATION CODE

```
function [R_sub_sim] = simulation_R (n, lamda, t)

    N = factorial(n); % generate n! nodes

    No_n_1 = n*(n-1); sum = 0; R =0;

    max_fault = round( N* (1-exp(-lamda*t)) );

    % Number of fault nodes related with the node reliability

     iterations = 10000;

    % run 10,000 times to find the no. of (n-1)-stars remaining operational

     f = zeros(max_fault,n);% initialization of the fault matrix

    if (max_fault < n )

        R_sub_sim = 1; % if no. of faults is less than n, reliability is '1'

    else

        for k = 1:1:iterations

            for i=1:max_fault

                fault = randperm(n); % randomly generate a node fault

                f(i,:) = fault(1,:);

                % add each node fault to the faults matrix [dimension

                %(max_fault, n) ]
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
end

 % node faults matrix

f = sort(f,1);

% sort the elements in each columns (2 to n) in ascending order

count = 0;

% used to record the no. of different symbols in every column

%from 2 to n .

% This number is equal to the no. of (n-1)-star being destroyed.

for j = 2:1:n

          for i = 2:1:max_fault

                  if ( f(i,j) > f( i-1,j) )

                           count = count + 1;

                   end

          end

          count = count +1;

end % end of counting of the no. of distinct Sn-1 being destroyed

if (count >=  No_n_1)

          R = 0;

          % if count is greater than total no. of Sn-1s, then R=0

else

          R = 1;

 end

 sum = sum + R; % record how many times that R=1
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
        end % end of the iterations

        sum;

        R_sub_sim = sum/iterations;

    end % end of if statement

end
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

VITA

Graduate College
University of Nevada, Las Vegas

Xiaolong Wu

Local Address:
1600 E.University Ave, #208, Las Vegas, NV 89119

Home Address:
Rm.2008, No.22 Building, 1st District, Anhuili, Yayunchun, Chaoyang,
Beijing, P.R.China, 100101

Degrees:
Master of Science, Electrical and Computer Engineering, 2004
University of Nevada, Las Vegas, NV 89119

Master of Science, Mechanical Engineering, 2002
University of Nevada, Las Vegas, NV 89119

Bachelor of Science, Mechanical Engineering, 1998
Nanjing University of Aeronautics & Astronautics

Selected Publications:
Journal:
1. Xiaolong Wu and S. Latifi, "Substar Reliability Analysis in Star Networks," submitted to *Information Science.*
2. Xiaolong Wu, S. Latifi, and Y. Jiang, "Robustness assessment of star graph," under 2nd review, *Journal of Information Science and Engineering.*
3. Xiaolong Wu, J. Zheng, E. Regentova, Y. Jiang, and M. Yang, "Performance analysis of channel sub-rating schemes application to handoff calls in hierarchical cellular networks," under 1st review, *Computer & Electrical Engineering.*
4. J. Zheng, S. Latifi, E. Regentova, K. Luo, and Xiaolong Wu, "Diagnosability of star graphs under the comparison diagnosis model," *Information Processing Letters*, vol.93, no.1, Jan. 2005, pp.29-36.
5. B. M. Fu and Xiaolong Wu, "Acute response of microvessel small solute permeability to vascular endothelial growth factor (VEGF)," *Federation of American Societies for Experimental Biology (FASEB) J.,* 15(4):A55, 2001.

Conference:
1. Xiaolong Wu, S. Latifi, and Y. Jiang, "Markov reliability modeling of star networks," in *Proc. of 2007 International Conference on Parallel and Distributed Processing Techniques and Applications.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2. Xiaolong Wu, S. Latifi, and Y. Jiang, "A combinatorial analysis of distance reliability in star networks," in *Proc. of 21st IEEE International Parallel & Distributed Processing Symposium (PDPN), 2007, pp.1~6.*
3. Xiaolong Wu, J. Zheng, E. Regentova, and Y. Jiang, "Analysis of the effect of channel sub-rating in unidirectional call overflow scheme for call admission in hierarchical cellular networks," in *Proc.* of *2007 Vehicular Technology Conference (VTC).*
4. G. M. Guo, M. Yang, Xiaolong Wu, Y. Jiang, S. L. Ma, and G. S. Hao, "Automous and dynamic web service composition in wireless grids," in *Proc. of International Conference on Information Technology (ITCC),* vol.2, 2005, pp.574-579.
5. G. S. Hao, S. L. Ma, H. M. Guo, Xiaolong Wu, M. Yang, and J. Jiang, "Analyze grid from the perspective of a computing system," in *Proc. of International Conference on Information Technology (ITCC),* vol.2, 2005, pp.789-790.
6. Xiaolong Wu, J. Ma, Y. Jiang, B. M. Fu, W. Hang, J. S Zhang, and N. Li, "Instrumentation of YSZ Oxygen Sensor Calibration in Liquid Lead-Bismuth Eutectic," in *Proc. of 2005 IEEE International Symposium on Circuits and Systems (ISCAS),* vol.2, pp.1746-1749.
7. Y. Jiang, M. Yang, Xiaolong Wu, and M. Zhang, "Substream selection in MIMO system with reduced complexity," in *Proc. of International Conference on Information Technology (ITCC),* vol.2, 2005, pp.729-733.
8. J. Zheng, S, Latifi, E. Regentova, Xiaolong Wu, and K. Luo, "Diagnosabilty of star graphs under the comparison diagnosis model," in *Proc. of 2003 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA),* pp.731-735.
9. Xiaolong Wu, R. Clarksean, Y. T. Chen, and D. W. Pepper, "An Analysis of the Melt Casting of Metallic Fuel Pins," in *Proc. of ASME International Mechanical Engineering Congress and Exposition,* New Orleans, Nov. 17-22, 2002.

Honors and Awards

- GPSA Travel Grant, Summer 2007
- GPSA Travel Grant, Spring 2007
- Bally's Technologies Graduate Scholarship, 2006~2007
- Department Representative of Graduate & Professional Student Association (GPSA) of Dept. of Electrical & Computer Eng. Of UNLV, 06~current
- Great Research Training Assistantship (GREAT) from UNLV, June ~ Aug. 2006
- Graduate teaching and research assistantship, UNLV, Fall 2000 ~ present
- NUAA scholarship, P. R. China, 94~97

Dissertation Title: Investigation of the robustness of star graph networks

Dissertation Examination Committee:

Chairperson, Dr. Shahram Latifi, Ph.D.
Committee Member, Dr. Emma Regentova, Ph.D.
Committee Member, Dr. Venkatesan Muthukumar, Ph.D.
Graduate Faculty Representative, Dr. Laxmi Gewali, Ph.D.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.