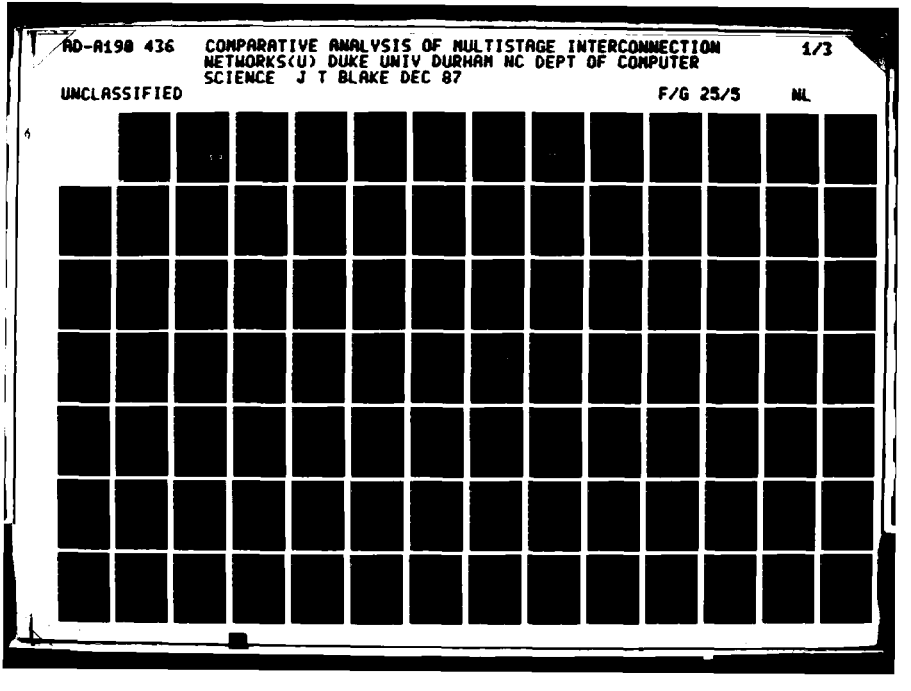
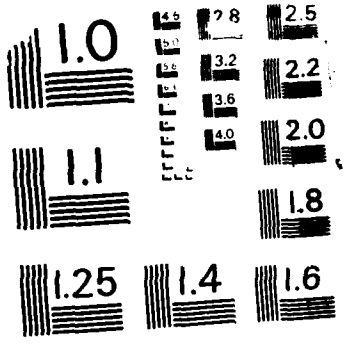
AD-A190 436 COMPARATIVE ANALYSIS OF MULTISTAGE INTERCONNECTION NETWORKS(U) DUKE UNIV DURHAM NC DEPT OF COMPUTER SCIENCE J T BLAKE DEC 87 1/3

UNCLASSIFIED F/G 25/5 NL





COMPARATIVE ANALYSIS OF MULTISTAGE INTERCONNECTION NETWORKS

by

James T. Blàke

Department of Computer Science
Duke University

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS None
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Duke University 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State, and ZIP Code) Department of Computer Science Duke University Durham, NC 27706	7b ADDRESS (City, State, and ZIP Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11 TITLE (Include Security Classification)
COMPARATIVE ANALYSIS OF MULTISTAGE INTERCONNECTION NETWORKS (UNCLASSIFIED)

12 PERSONAL AUTHOR(S)
James T. Blake

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final	FROM _____ TO _____	December 1987	

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Multistage Network. Performance Analysis. Performability Analysis. Reliability Analysis. Shuffle-Exchange Network.

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

See back

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL James T. Blake	22b TELEPHONE (Include Area Code) (919) 493-3996
	22c OFFICE SYMBOL

DD Form 1473, JUN 86 *Previous editions are obsolete*

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

ABSTRACT

This thesis provides a comparative analysis of various interconnection networks and multiprocessor systems. The principle interest is in the analysis of the reliability and composite measures of performance and reliability of interconnection networks that connect processors to memories in large multiprocessor systems. Specifically, the Shuffle-Exchange multistage interconnection Network (SEN) and its variants are evaluated and compared. Comparison is based on reliability, composite measures of performance and reliability, and cost.

Closed-form expressions for the computation of the available bandwidth for multiprocessor systems with a capability for graceful degradation are developed. Then, the time-dependent reliability of the SEN and three fault-tolerant schemes aimed at improving system reliability are examined. These schemes are the redundant network, the extra-stage network, and the network augmented with intrastage links. Exact closed-form expressions for the time-dependent reliability of the $N \times N$ Shuffle-Exchange Network (SEN), the 8×8 and 16×16 SEN with an additional stage (SEN+), and the 4×4 and 8×8 Augmented SEN (ASEN) are derived.

Upper and lower bounds useful for the analysis of larger SEN+ and ASEN networks are derived. Numerical results for networks as large as 1024×1024 are provided. A comparison of these networks shows that, on the basis of reliability, the ASEN is superior to the SEN, SEN+, and the redundant SEN (2-SEN). The results for the SEN+ are extended to the case of an (uniform) Omega network. Further, through the novel use of hierarchical decomposition, results on the reliability of ASENs are extended to include imperfect coverage and on-line repair.

In the last chapter, performability analysis of a complete multiprocessor system is conducted. The crossbar and the Omega networks are used to represent the interconnection network and two levels of detail are presented for analyzing the crossbar. Bottleneck and sensitivity analysis of the multiprocessor system are also performed. Markov chains and Markov reward models are used in the analysis. In addition, the criteria for the lumping of states in a Markov chain is extended to Markov reward models.

Comparative Analysis of Multistage Interconnection Networks

LTC James T. Blake
HQDA, MILPERCEN (DAPC-OPA-E)
200 Stovall Street
Alexandria, VA 22332

Final report **December 1987**

A thesis submitted to Duke University, Durham, NC
in partial fulfillment of the requirements for the degree of Ph.D. in Computer Science.

COMPARATIVE ANALYSIS OF MULTISTAGE INTERCONNECTION NETWORKS

by

James T. Blake

Department of Computer Science
Duke University

Date: _____

Approved:

_____
Kishor S. Trivedi, Supervisor

_____
Joanne Bechta Dugan

_____
Ambuj Goyal

_____
Merrell L. Patrick

_____
Donald J. Rose

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science in the Graduate School of Duke University

1987

COMPARATIVE ANALYSIS OF MULTISTAGE INTERCONNECTION NETWORKS

by

James T. Blake

Department of Computer Science
Duke University

Date: _____

Approved:

_____
Kishor S. Trivedi, Supervisor

_____
Joanne Bechta Dugan

_____
Ambuj Goyal

_____
Merrell L. Patrick

_____
Donald J. Rose

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science in the Graduate School of Duke University

1987

Abstract

This thesis provides a comparative analysis of various interconnection networks and multiprocessor systems. The principal interest is in the analysis of the reliability and composite measures of performance and reliability of interconnection networks that connect processors to memories in large multiprocessor systems. Specifically, the Shuffle-Exchange multistage interconnection Network (SEN) and its variants are evaluated and compared. Comparison is based on reliability, composite measures of performance and reliability, and cost.

Closed-form expressions for the computation of the available bandwidth for multiprocessor systems with a capability for graceful degradation are developed. Then, the time-dependent reliability of the SEN and three fault-tolerant schemes aimed at improving system reliability are examined. These schemes are the redundant network, the extra-stage network, and the network augmented with intrastage links. Exact closed-form expressions for the time-dependent reliability of the $N \times N$ Shuffle-Exchange Network (SEN), the 8×8 and 16×16 SEN with an additional stage (SEN+), and the 4×4 and 8×8 Augmented SEN (ASEN) are derived.

Upper and lower bounds useful for the analysis of larger SEN+ and ASEN networks are derived. Numerical results for networks as large as 1024×1024 are provided. A comparison of these networks shows that, on the basis of reliability, the ASEN is superior to the SEN, SEN+, and the redundant SEN (2-SEN). The results for the SEN+ are extended to the case of an (uniform) Omega network. Further, through the novel use of hierarchical decomposition, results on the reliability of ASENs are extended to include imperfect coverage and on-line repair.

In the last chapter, performability analysis of a complete multiprocessor system is conducted. The crossbar and the Omega networks are used to represent the interconnection network and two levels of detail are presented for analyzing the crossbar. Bottleneck and sensitivity analysis of the multiprocessor system are also performed. Markov chains and Markov reward models are used in the analysis. In addition, the criteria for the lumping of states in a Markov chain is extended to Markov reward models.

Acknowledgements

I wish to express my appreciation to Dr. Kishor Trivedi for his patience, guidance, and criticism during the past few years.

To Dr. Vijay Kumar, who is responsible for stimulating my interest in the general area of interconnection networks for large parallel-processing applications, I extend a special thanks.

Also, I want to thank my colleagues at Duke. In particular, I want to thank Gianfranco Ciardo, Phil Chimento, Dr. Joanne Bechta Dugan, Dr. Andrew Reibman, Earl Smith, Dr. Roger Smith, and Malathi Veeraraghavan for their interest in my research and for the many helpful comments they made during my stay at Duke.

Finally, I appreciate all the encouragement my wife, Sharon, gave me during this endeavor.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, combined performance and reliability measures are used to evaluate the interconnection networks in large multiprocessor systems. Then, this work is extended to the analysis of an entire multiprocessor system consisting of processors, memories, and an interconnection network. The specific networks examined are the crossbar and the Shuffle-Exchange multistage interconnection Network (SEN) and its variants.

Separately modeling the reliability and performance of such networks is not new; many researchers have examined either reliability or performance as separate measures of a network's "goodness." In general, however, the reliability analysis of these networks has been limited to finding the probability that a given source can communicate with a given destination, which is called *two-terminal reliability*; simulation to examine *multi-terminal reliability*; or analytic arguments for stating the fault-tolerance properties of a network. This type of analysis is too crude to permit a useful assessment of a large multiprocessor system (MPS) designed to permit graceful degradation. Previous work on performance has concentrated on the permutation capabilities of these networks under a no-fault assumption; or, when faults

are allowed, analytical work has been limited to special classes of permutations, since the optimal realization of arbitrary permutations is known to be intractable. Also, bandwidth analysis has been limited in a similar manner.

In this thesis, reliability analysis of different topologies will be conducted by "normalizing" the complexities of the different networks based on gate count. Thus, a standardized basis can be used to compare different fault-tolerant schemes. Combinatorial methods and Markov models are used in the analysis; and, whenever possible, exact reliability expressions are derived.

Several researchers have looked at combining performance and reliability. The term for this combined measure has been coined as *performability* by Meyer [60]. Previous work on the theoretical development of performability can be found in [31], [61], and [62]; some examples have been presented in [48], [60], and [90].

While it is recognized that many measures may be used for combining performance and reliability, the focus will be on three such measures. They are: the average instantaneous performance level at time t, the average accumulated work until time t, and the distribution of the cumulative work until system failure. These measures include, as special cases, several "pure" performance measures (the maximum and minimum performance levels and their product with the time-to-failure random variable); the distributions of these performance measures; and "pure" reliability measures (the distribution of a system's lifetime and the mean time to failure).

In the remainder of this chapter, the salient features of multiprocessor systems and interconnection networks will be presented. Then, in the next chapter, a more thorough examination of Multistage Interconnection Networks (MINs) will be conducted. (The emphasis in this chapter is on unique-path MINs and the methods used to add fault tolerance to these networks.) The

following chapter contains a description of the networks to be analyzed. The remaining chapters are devoted to performance, reliability, and performability analysis of the networks. A detailed analysis of a complete multiprocessor system using three different interconnection network models is also included as a final example of the application of performability analysis.

1.1 Multiprocessor Systems

In recent years, significant advances have been made in parallel processing. Real-time applications which require enormous computing power appear to be the driving force behind these endeavors. Execution rates exceeding one billion instructions per second are required for many applications such as image processing and weather forecasting. These execution rates appear to be unachievable on uniprocessors performing serial instruction execution. Multiprocessor systems using many processors executing in parallel, however, have the ability to perform at these rates. As mentioned in [103], there are several experimental multiprocessor systems employing a large number of processing elements (PEs) in various stages of development, and today multiprocessor systems with hundreds and even thousands of processors exist. These systems are composed of three major components: processors, common memory modules, and an interconnection network.

Figure 1.1 provides a simplified view of these large multiprocessor systems. These systems consist of sources (Ss), an interconnection network (IN), and destinations (Ds). The sources are processors or PEs, and the destinations may be either memory modules (MMs) or other PEs. The IN is used to provide a communication path between particular source-destination (S-D) pairs.

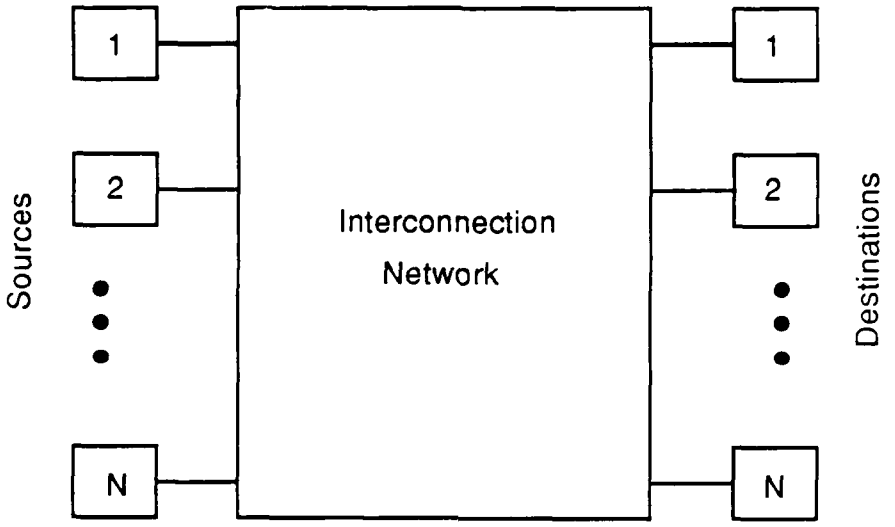


Figure 1.1: Simplified Multiprocessor System.

As the number of processors used in these multiprocessor systems increases, so does the need to insure that the communication network between the system components does not become a bottleneck to achieving the desired concurrent processing speeds.

In order to take advantage of the high computation speeds of today's powerful microprocessors in a multiprocessor architecture, the communication between these processors must be extremely efficient. Furthermore, the network that performs processor-to-processor or processor-to-memory connections must be robust. That is, the IN must be reliable and relatively insensitive to a small number of failures in the components which comprise the network. A brief survey of interconnection methods is found in [30].

1.1.1 Multiprocessor Organization

A large multiprocessor computer utilizing an IN can usually be classified as a Single-Instruction Multiple-Data (SIMD) organization or a Multiple-Instruction Multiple-Data (MIMD) organization. In fact, some architectures provide a combination of these two organizations.

In SIMD organizations, all PEs receive the same instruction broadcast from a central control unit, but they operate on different data sets from distinct data streams. One can think of these multiprocessor systems as a synchronous array of parallel processors. These types of machines are usually designed to perform vector computations over arrays of data. In MIMD organizations, subsets of the PEs operate in concert using a particular set of instructions. All PEs derive their data sets from the same shared-memory structure.

SIMD computer organizations usually use a given interconnection network (IN) based on four decision criteria [42]:

1. operation modes,
2. control strategies,
3. switching methodologies, and
4. network topologies.

Since SIMD machines operate in a lock-step fashion, a synchronous operation mode, rather than an asynchronous mode, is used. A centralized control strategy is usually preferred over distributed control. With this strategy, all switching elements are controlled by a single controller. While three switching methodologies (circuit, packet, and combined) can be identified, circuit

switching is generally used in SIMD machines. In a circuit-switched environment, a physical path is established between each *S-D* pair, whereas in a packet-switched environment, data is broken into small packets and each packet is routed through the IN without establishing a physical path. Circuit switching is preferred if long, bulky transmissions are required between *S-D* pairs. Finally, both static and dynamic topologies exist in INs. Static INs are usually chosen for SIMD machines. In a static IN, once a physical path is established between a given *S-D* pair, no reconfiguration of the switching elements (SEs) and links along this this path is made. In a dynamic IN, links can be reconfigured to satisfy other *S-D* requests.

In a MIMD computer organization, each processing element contains some local memory, so the frequency with which each PE requests access to the IN is expected to be less than in a SIMD. The MIMD computer organization may use both synchronous and asynchronous operation modes. Distributed control of the components of the IN is often used, so self-routing networks are common. The switching methodology may be any of the three mentioned for SIMD machines, and the network topology is heavily dependent on the size of the multiprocessor system and the perceived application.

1.1.2 Network-Oriented Architecture

In [25], a network-oriented view of multiprocessor organizations is presented. The two common network-oriented systems are: the processor-to-memory and the processing element-to-processing element (PE-to-PE) architectures. Each PE is composed of a processor and a local memory. In the processor-to-memory architecture, sources are the processors and the destinations are the memory modules (MMs). The interconnection network is bidirectional, and it is used to fetch instructions and data stored in the MMs. This is a *shared-*

memory interprocessor communication system, and the associated multiprocessor system is often referred to as a *tightly-coupled* system. In this system, the interconnection network can be expected to be heavily loaded. In the PE-to-PE architecture, each PE is connected to the network via both an input and output link of a unidirectional interconnection network. The instructions and data for each PE are considered to be contained in the local memory associated with that PE, so the network is used only for inter-PE communication. The loading on this network will be far less than on a comparable processor-to-memory network. The multiprocessor systems using this type of network are often called *loosely coupled*, and their inter-communication strategy is called *message passing.*

1.2 Interconnection Networks

Interconnection strategies for multiprocessor systems range from the time-shared bus (Figure 1.2) to the crossbar switch. The time-shared bus is inexpensive, but it does not permit simultaneous communication between distinct components attached to the bus. Even the fastest of these buses causes the multiprocessor system using it to become inefficient when a moderate number of components attempt to communicate in a time-shared manner. Bus-oriented multiprocessor systems may provide acceptable performance for systems with up to 30 processors, but, given the current state of technology, it is unlikely that a shared-bus architecture would be viable for systems with 1000 or more processors [94]. The key distinction between the bus and the MINs that are examined in this thesis is that the bus allows transmission between just two units at any time; whereas a MIN allows a number of parallel transmissions to take place. Usually a bus is a slower, although less expensive, network than the MIN.

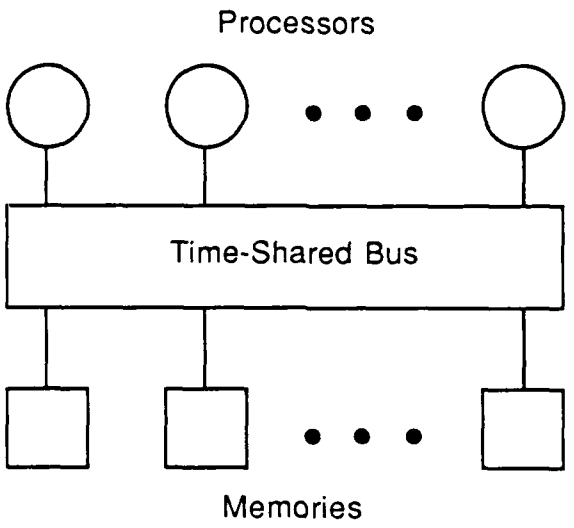


Figure 1.2: Multiprocessor System Using a Bus Architecture.

Point-to-point communications are also used in today's multiprocessor systems. In a graphical representation of point-to-point interconnections, the PEs are the vertices and the dedicated links are the arcs. In Figure 1.3, the mesh and ring are illustrated. In these networks, there is often a bound placed on the number of processors/memories that a given processor can be connected to. As the size of the network grows, the bandwidth of these networks becomes too small for real-time applications.

The fastest of the interconnection strategies is the crossbar switch (Figure 1.4). It allows simultaneous connections between all source-destination pairs as long as no two sources request the same destination. However, for N sources and N destinations, the crossbar switch requires $O(N^2)$ connections. Thus, for large N, the use of a crossbar is prohibitively expensive. In fact, its cost may dominate the cost of the entire multiprocessor system. Furthermore,

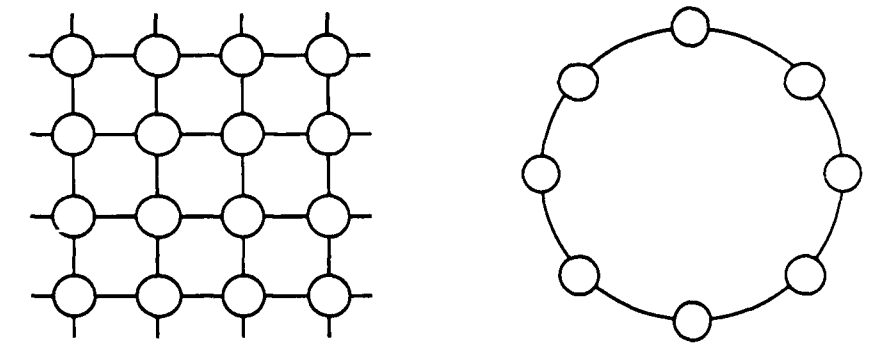


Figure 1.3: Point-to-Point Communications.

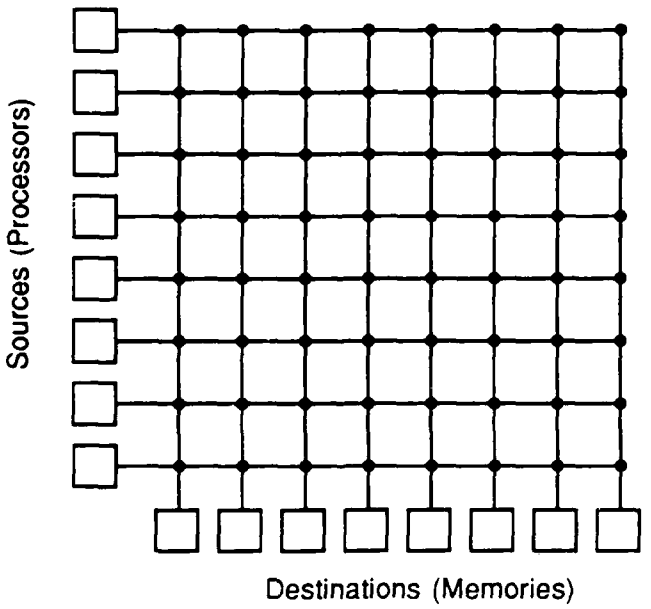


Figure 1.4: Crossbar Switch.

effective use of the available bandwidth may not be achieved; thus providing very little benefit in terms of the crossbar's actual throughput [94].

The Multistage Interconnection Network (MIN) is a compromise between the IN extremes. It offers simultaneous communications at a lower cost than the crossbar, has a smaller number of connections leading out of a source or into a destination, and for large systems, it has a higher bandwidth than the time-shared bus. A MIN has several stages of switching elements (small crossbar switches) arranged so that many source-destination connections can be made as long as no two connections require a common link. Figure 1.5 is an illustration of a 16×16 Shuffle-Exchange Network which is an unique-path MIN. The hardware complexity of this network, expressed in terms of the number of required switching elements, is $O(N \log N)$.

In multiprocessor systems, the amount of parallelism that can be achieved is often a function of the parallel accessibility of data by the PEs. Depending on the degree of fault-tolerance that the system enjoys, the presence of switching element and/or link failures may seriously degrade the permutation capability and bandwidth of these systems [77].

A number of unique-path MINs have been proposed, and a multitude of evaluation metrics have been used to analyze these MINs; however, no one network appears as the clear choice for a given application. This thesis will examine a unique-path MIN called the Shuffle-Exchange multistage interconnection Network (SEN), which is representative of several proposed MINs. Some variants of this MIN are also examined. Because the most critical properties of a MIN in a large multiprocessor system are reliability and performance, the emphasis will be on a combined evaluation measure for these INs. In gracefully degrading multiprocessor systems, faults can be tolerated in the processors, memories, and/or the IN. These systems require

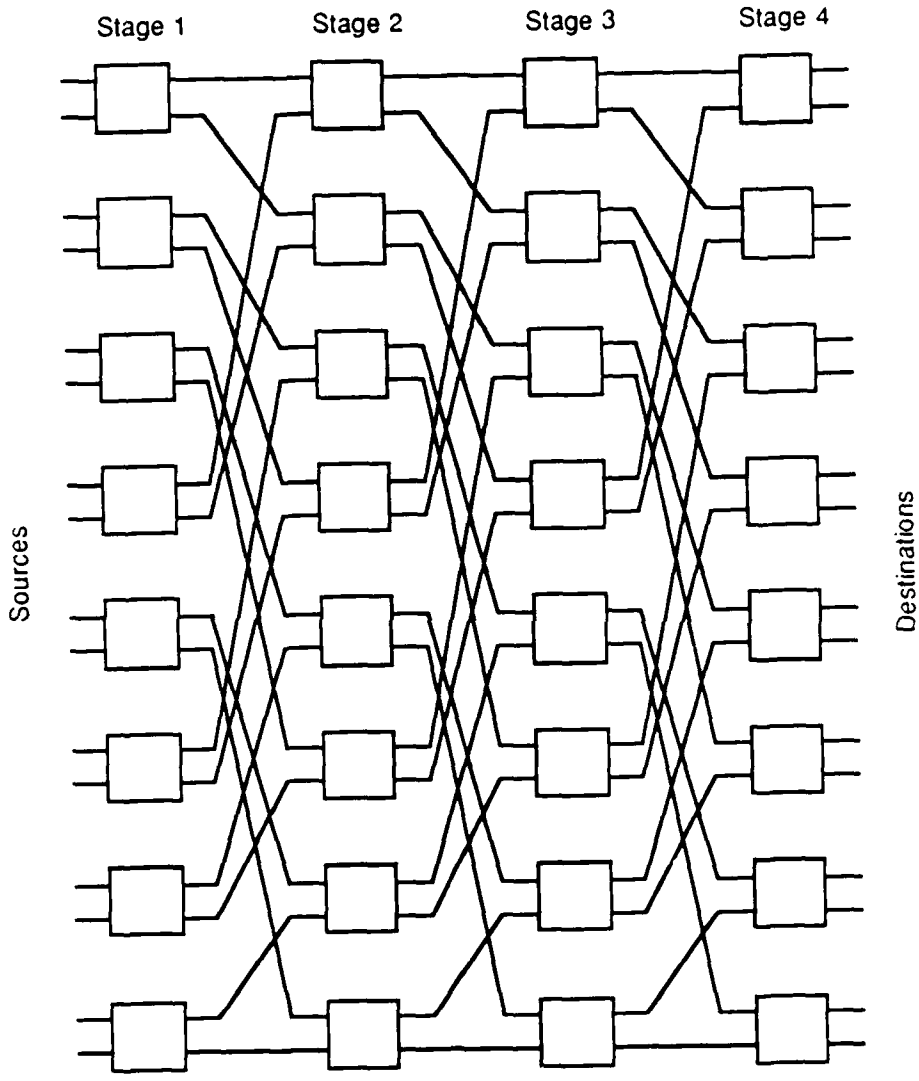


Figure 1.5: 16×16 SEN.

new performance-related measures which are more informative than traditional measures. So new measures such as computational availability and performability will be used to deal with these systems.

Chapter 2

Multistage Interconnection Networks

2.1 Introduction

Multistage interconnection networks represent a large subset of the interconnection networks proposed for large-scale multiprocessor systems [86]. In this chapter, the basic building block of the MIN, the switching element, is described. Then, the three major classes of MINs are discussed, followed by a description of the characteristics of unique-path and multiple-path MINs. The last section reviews the basic fault models used to analyze multistage interconnection networks.

2.2 Switching Element Description

The basic building block of a MIN is the switching element (SE). The switching element is essentially a $c \times d$ crosspoint switch. There are c input links and d output links attached to the SE. These SEs are then interconnected in a particular pattern to form a specific multistage interconnection network. For clarity of explanation, let $c = d = 2$. Switching elements of this size are frequently encountered in MINs because of the simplicity of their design.

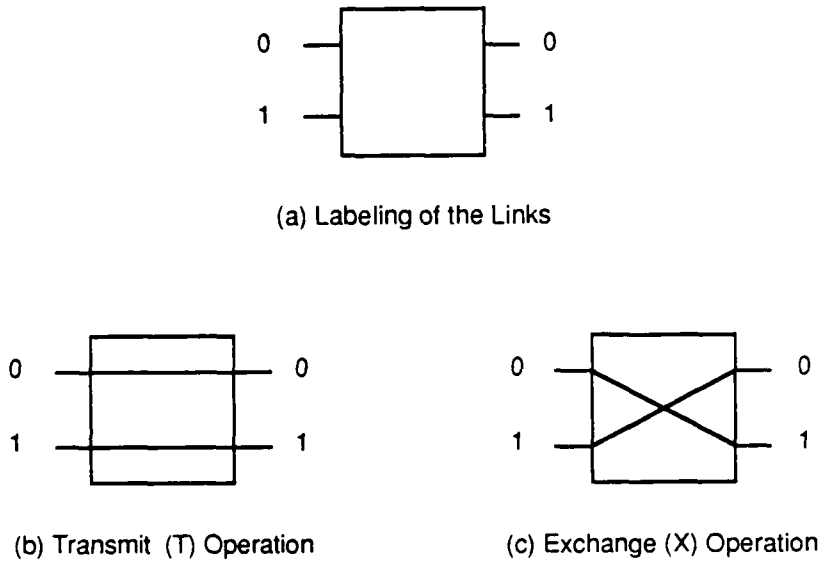


Figure 2.1: 2×2 Switching Element.

Figure 2.1 shows a 2×2 switching element and the two operations it can perform. Figure 2.1(a) shows the labeling of the input and the output links. The SE can either transmit (T) the inputs directly through itself as in Figure 2.1(b) or exchange (X) the inputs as in Figure 2.1(c). In general, the MINs examined in this thesis will be constructed from 2×2 SEs.

2.3 MIN Classification

MINs are often classified based on their connection capability and their ability to realize permutations. The three major classes are *strictly non-blocking*, *rearrangeably non-blocking*, and *blocking networks* [50].

A *strictly non-blocking network* can realize any permutation of its inputs. It can connect any source to any non-busy destination without regard for the current state of the network. Such networks have received considerable

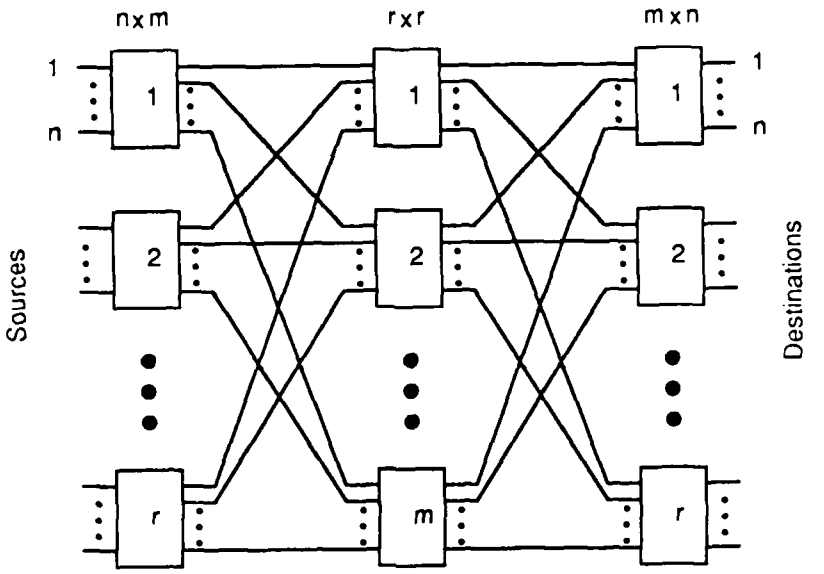


Figure 2.2: Clos Network.

attention in connection with telephone switching systems. The Clos network [21] (Figure 2.2) is an example of such a network. The hardware complexity of the strictly non-blocking networks, however, is $O(N(\log N)^2)$, so they are not suitable for multiprocessing systems.

A *rearrangeably non-blocking network* can also realize any permutation on its inputs. It can connect any source to any non-busy destination, but it may require the rearrangement of existing connections by changing switching element settings. The Benes network [11] (Figure 2.3) is a member of this class, and it has been studied extensively for use in synchronous data permutations and asynchronous interprocessor communications [30]. These networks have a hardware complexity of $O(N \log N)$. From a cost perspective, these networks may be acceptable for multiprocessor systems; however, for networks of moderate size, the routing algorithms used for rearranging the existing connections make them too slow.

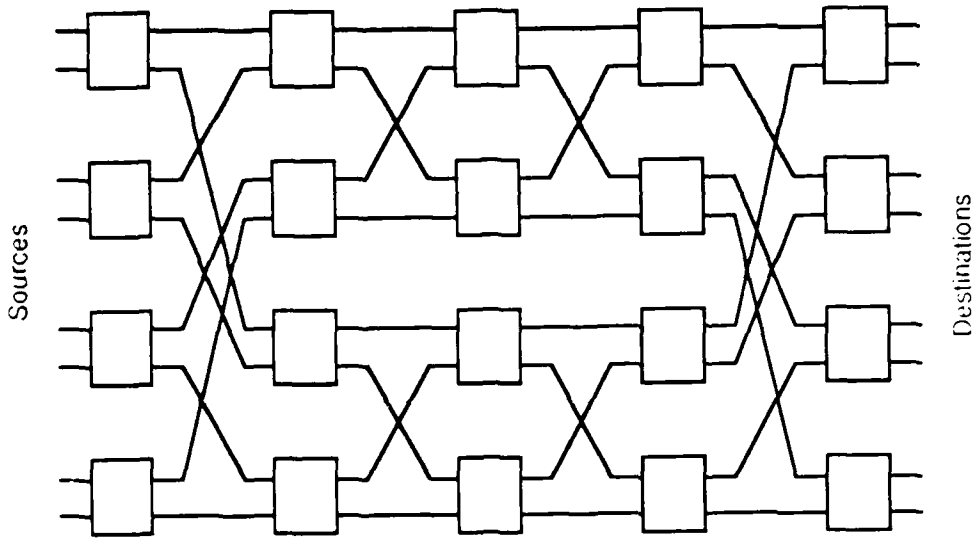


Figure 2.3: Benes Network.

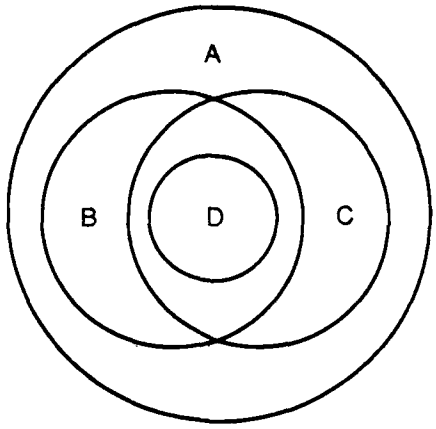
In *blocking networks*, simultaneous connections of more than one source-destination pair may require the use of common links. Thus, one or more connections may be blocked. Many networks in this class have been studied extensively. Examples are the Baseline [104], SW Banyan [33], Omega [54], Indirect binary n-cube [70], and Delta [69]. These networks have a hardware complexity of $O(N \log N)$, but in most implementations of these networks, they are only half as complex as the rearrangeably non-blocking networks. Several of the networks in this class were shown to be topologically equivalent to the Baseline network in [104]. The basic networks in this class are often called unique-path MINs meaning that there exists only one path be-

tween any source-destination pair. This structure prevents such MINs from realizing every arbitrary permutation. However, unique-path MINs can realize many permutations useful for synchronous parallel computations [54,70]. Furthermore, the simplicity of their distributed routing algorithms have made them very useful for multiprocessor applications.

MINs are attractive networks for tightly-coupled multiprocessor systems, and offer a good balance between cost and performance [1]. Popular among the MINs considered for large multiprocessor systems are networks with distributed routing algorithms which obviate the need for a central controller to operate the MIN. Further, those networks which also possess the *self-routing* property are often used because of the ease of setting the switching elements with a destination tag generated by the source. Examples are the Omega [54] and the Delta [68] networks.

2.4 Unique-Path MINs

Figure 2.4 shows a Venn diagram for the classes of unique-path MINs. The Banyan networks introduced by Goke and Lipovski in [33] form the most general class of unique-path MINs. Within this class are two large subclasses, they are: (1) the Generalized Shuffle Networks (GSN) introduced by Bhuyan and Agrawal in [14], and (2) the Delta networks introduced by Patel in [69,68]. A GSN connects M sources to N destinations for arbitrary values of M and N. The Delta network connects a^n sources to b^n destinations through $a \times b$ crossbar switches at each stage. Included within the intersection of these two classes of networks are the MINs constructed from 2×2 SEs. In [104], Wu and Feng showed the topological equivalence of several of these networks to the Baseline network. The Baseline [104], Data manipulator (modified) [104],



A: Banyan

B: GSN (M x N for arbitrary M and N)

C: Delta (a^n x b^n)

D: Baseline
Data Manipulator (modified)
Flip
Indirect Binary n-cube
Omega
Regular SW Banyan (S = F = 2)
Reverse Baseline
SEN

Figure 2.4: Relationship of Selected MINs to the Class of Banyan Networks.

Flip [8], Indirect binary n-cube [70], Omega [55], Regular SW banyan ($S = F = 2$) [33], Reverse baseline [104], and SEN are topologically equivalent.

2.4.1 Characteristics

Information is passed through the MIN in one of two ways: (1) circuit switched, or (2) packet switched. In a circuit-switched mode, a source is granted a path through the network to a given destination, and it holds that path until it completes its data transfer. In this mode, a source communicates with a destination along a physical connection that is established through several switching elements. The links and SEs along this path are used exclusively by the S-D pair.

In a packet-switching mode, the information each source sends to a destination is broken into small packets. These packets then individually compete

for a path through the network. No dedicated, physical path from the source to the destination exists. Instead, each switching element must have the capability to store and forward the individual packets, and packets compete for links within the IN. Packet switching can improve the throughput of the MIN over that obtained by the use of circuit switching, but it will increase both the S-D transmission delay and the cost of the MIN since each SE must have a buffering capability.

Unique-path MINs have many properties that make them attractive for multiprocessor systems, including an $O(N \log N)$ hardware cost as opposed to the $O(N^2)$ hardware cost of crossbar switches, the ability to provide up to N simultaneous connections, $O(\log N)$ path lengths, and the existence of simple, distributed routing algorithms.

MINs with $\log N$ stages also have two other important properties:

1. there exists an unique path from each S to each D, and

2. distinct S-D paths may have common links.

These properties lead to two significant disadvantages. First, a S-D connection may be blocked by a previously established connection (even if the destinations involved are distinct) causing poor performance in a random-access environment. Second, the failure of even a single link or SE disconnects several source-destination paths, lowering reliability. The reduction in performance due to blocking and the decrease in reliability due to the lack of fault tolerance become increasingly serious with the increase in size of the network because the number of paths passing through a given link increases linearly with N [53].

While MINs can be built from any combination of switching elements [14], for the sake of brevity and clarity, the SEN presented in this thesis is defined

for $N = 2^n$ sources, N destinations, and n stages, each stage consisting of $N/2$ switching elements. The stages are numbered from 1 to n, and the switches in each stage are numbered from 0 to $N/2 - 1$.

2.4.2 Permutation Issues

The ability of a MIN to realize any arbitrary permutation is often used as a performance measure. The failure of a single SE in an unique-path MIN can have a significant impact on this measure. For example in [72], the number of distinct permutations that are admitted by a $2^n \times 2^n$ MIN which consists of n stages, using 2×2 SEs is $2^{n \cdot 2^{n-1}}$. Now, if one of the SEs in the network becomes stuck-at-T or X, the number of admissible permutations by the faulty network is reduced by one-half. Furthermore, several sources cannot be connected to certain destinations. For example, if the faulty switching element is in stage k, $1 \leq k \leq n$, there are some 2^k sources where each source cannot be connected to 2^{n-k} particular destinations.

It was proposed in [29] that these unique-path networks be augmented by adding one additional stage, so that in the event of a single faulty switch, one is still able to achieve all the permutations possible in the fault-free network using at most two passes through the network. This introduces the concept of multiple-path MINs. Their purpose is to improve the fault tolerance of the IN with a modest increase in network complexity.

2.5 Multiple-Path MINs

In setting up a connection (or routing a packet in a packet-switching environment), multiple-paths MINs allow an alternate path to be chosen whenever conflicts arise with other connections or when faults develop in the network. Thus, multiple-path MINs have higher reliability than unique-path MINs.

The multiple-path MIN may also enjoy increased performance in a random-access environment.

Some research has been done on the fault-tolerance properties of various multiple-path MINs. For example, in [67] Parker and Raghavendra introduce the Gamma network and examine its permutation capabilities. The Gamma network is a multiple-path MIN with $N = 2^n$ sources, N destinations, and $\log_2 N$ stages. Each stage has N 3×3 SEs. The various paths are represented in the redundant number system. In [76], the terminal reliability of the Gamma network and two of its variants (Bigamma and Monogamma) is examined. The analysis is restricted to terminal reliability since the multi-terminal reliability problem is intractable [6].

Ciminiera and Serra introduce another fault-tolerant MIN in [19]. This multiple-path MIN is called the F network. The $N \times N$ F network has N SEs in each of $\log_2 N$ stages and uses 4×4 SEs. No reliability analysis is attempted, instead it is shown that multiple paths exist between each S-D pair.

More recently, Raghavendra and Varma introduced the INDRA (Interconnection Networks Designed for Reliable Architectures) class of multiple-path networks in [78]. The Indra network with $N = 2^n$ inputs and N outputs achieves R redundancy ($R \geq 2$) when the network is constructed using $\log_R N + 1$ stages of $R \times R$ SEs; each stage has N SEs, and N must be a power of R. The Indra network also uses multiple connecting links to the sources and destinations that make it (R-1)-switch fault-tolerant in the first and last stages. R^2 paths exist between each S-D pair. The reliability analysis in [78] is limited to terminal reliability.

2.5.1 Fault-Tolerance Issues

Often, as the number of components in a conventional multiprocessor system increases, so does the rate of failure of the system. In [7], this type of behavior is referred to as "coherence." The criteria for judging the design of a fault-tolerant network can be found in [20].

In traditional fault-tolerant architectures, where failure-free operation is desired for long time intervals, emphasis is placed on the use of hardware replication and redundancy to obtain the desired reliability goals. In the case of large-scale parallel computing with homogeneous processors, the redundancy needed for fault tolerance is inherent in the design itself. The objective in these systems is to allow the system to gracefully degrade down to some specified level of performance [77]. However, when planning such a large multiprocessor system, the fault tolerance of the IN which connects redundant sources to redundant destinations is often overlooked. While unique-path MINs are no more susceptible to component failures than a redundant network, the effects of such failures are far more dramatic. This is especially true in large multiprocessor systems.

In large multiprocessor systems, hardware fault tolerance can be achieved in two ways: (1) at the *system level*, and (2) at the *processor/component level*. Hardware fault-tolerance at the system level is achieved by successfully identifying the fault, isolating it, and performing system reconfiguration and recovery. This fault-tolerant technique is preferred over redundancy and data replication at the processor level since it requires much less hardware overhead [77].

In [65], three techniques are mentioned for providing fault tolerance in a MIN. They are:

1. software,

2. hardware/software, and

3. redundant-path hardware.

The purely software approach includes methods such as error-detecting and error-correcting codes. These methods, however, are oriented toward insuring that correct data is received at a destination given that the data is ever received. In the hardware/software approach, one uses redundancy at the component level to achieve fault tolerance. If, for example, triple-modular redundancy is used, the hardware costs are roughly tripled. The third technique, the use of redundant paths, can be achieved either inherently in the network design as in [11] or [67], or by the addition of extra hardware to achieve redundant paths between each *S-D* pair. Three ways to add extra paths are: through additional links, additional stages, and duplication of an existing network.

2.5.2 Switch versus Link Complexity

There are two ways for a given MIN to possess the multiple-path property. Multiple paths may be inherently present in the definition of the MIN, or they may be created by augmenting the topology of an existing unique-path MIN. In any case, they have a higher hardware cost than unique-path MINs in terms of

1. the number of stages of switching elements,

2. the number of switching elements per stage, and/or

3. the size of the switching elements.

These three factors contribute to what is usually called the *switch complexity* of a MIN. Another measure of the cost of a MIN is its *link complexity*, which

depends on the number of interstage links, the number of intrastage links (if any), and the number of stages. Link complexity is an important measure because the implementation of MINs is often input/output or pin limited, at every level of integration. For instance, at the integrated circuit level, if each integrated circuit contains one SE, the size of the switching element is usually determined by the number of pins available and not by the complexity of the logic in the switch. Also, at the wafer scale integration level, if a MIN with a large number of sources and destinations were to be laid out on a single wafer, the links would be the limiting factor on the chip [102]. That is, the links would consume most of the area of the chip, rather than the SEs. Of the two types of links in MINs, interstage links tend to be more expensive than the intrastage links [53].

2.5.3 Routing Considerations

The routing strategy is a key issue in multiple-path MINs. The topology of a multiple-path MIN may allow rerouting to be done only at the source or some fixed points in the network. In that case, a busy link, a faulty link or a faulty switching element encountered while setting up a path may necessitate backtracking to a stage where a fork exists in an attempt to find an alternate path. Backtracking may be eliminated if the paths between every source-destination pair in a multiple-path MIN have a fork at every stage. As might be expected, multiple-path MINs which use backtracking tend to have lesser hardware complexity than nonbacktracking MINs. But backtracking MINs may be difficult to implement since they require bidirectional paths and reverse queues [51].

The proper sequencing of packets in a packet-switched environment is another problem that must be addressed by the routing strategy. Failure

to properly sequence packets can cause computational inconsistencies. If buffering is used to overcome this problem, this will lead to further increases in hardware and buffering delays. This problem can be resolved by using virtual circuit techniques or otherwise restricting the paths used when the proper sequence of packets has to be maintained.

As mentioned before, the performance of multiple-path MINs is usually better than that of unique-path MINs since alternate paths can be used to reduce the effect of blocking in a random-access environment.

2.6 Fault Models

A fault model captures the effects of physical failures on the operation of a system. For MINs, there are three fault models in use:

1. stuck-at fault model,
2. link fault model, and
3. switch fault model.

In the *stuck-at fault model*, failures are assumed to cause a switching element to remain in a particular state regardless of the control inputs given to it, thus restricting the ability of the SE to set up proper connections. The affected switching element can be used to set up paths if the stuck-at state is also the required state. The *link fault model* assumes that a failure affects an individual link of a switching element, leaving the remaining part of the switching element operational. The *switch fault model* is the most conservative of the three and assumes that a failure makes a switching element totally unusable [50]. Analysis of networks in this thesis will use the *switch fault model*. Note, however, that a *link fault model* can simulate the *switch fault model*, but not vice versa.

In the next section, a detailed description of the SEN and its variants will be presented. Also included is a description of the crossbar switch.

Chapter 3

Description of MINs to be Analyzed

In this chapter, descriptions of the networks selected for analysis will be presented. The networks are: (1) the crossbar network, (2) the Shuffle-Exchange MIN (SEN), (3) the Shuffle-Exchange MIN with an additional stage (SEN+), (4) the Redundant SEN, and (5) the Augmented SEN (ASEN).

While the SEN and its variants were selected for analysis, this work can be extended to many other MINs since the SEN is just one network in a large class of topologically equivalent MINs that include the Omega, Indirect binary n-cube, and Baseline [104].

3.1 Crossbar Network

An $N \times M$ crossbar network allows all possible connections between the N inputs, termed sources (Ss), and the M outputs, termed destinations(Ds). In general, N does not have to equal M, but to permit comparisons with the other networks in this thesis, only $N \times N$ crossbar networks will be considered. Figure 1.4 illustrates this network.

As long as no two sources request the same destination, any arbitrary permutation (one-to-one mapping) is possible. Hence, the crossbar network

is non-blocking. However, when two or more sources request the same destination, contention at the destination input port will decrease the available bandwidth of this network. As mentioned earlier, the network complexity is $O(N^2)$ which is not practical for large multiprocessor systems.

3.2 Shuffle-Exchange MIN (SEN)

The class of MINs to which the SEN belongs is termed Delta networks. The specific SENs to be examined will have $N = 2^n$ inputs and N outputs. There is an unique path between each source-destination pair. The SEN has n stages, and each stage has $N/2$ switching elements (SEs). The stages are labeled from 1 to n, and the switching elements at each stage are labeled from 0 to $N/2 - 1$. The interconnection pattern between the stages is the $2 \times 2^{n-1}$ shuffle permutation. The position of switching element i in stage j can be denoted as $\mathrm{SE}_{i,j}$.

Figure 3.1 illustrates a SEN for $N = 8$. An 8×8 SEN has 8 sources, 8 destinations, and 3 stages each with 4 SEs. The *network complexity*, defined as the total number of switching elements in the MIN, is $(N/2)(\log_2 N)$, which for this example is 12.

The SEN is a self-routing network. That is, a message from any source to a given destination is routed through the network according to the binary representation of the destination's address. For example in an 8×8 SEN, if $S = 000$ wants to send a message to $D = 101$, the routing can be described as follows: $S = 000$ presents the address of $D = 101$ plus the message for D to the SE in stage 1 to which $S = 000$ is connected ($\mathrm{SE}_{0,1}$). The first bit of the destination address ($\underline{1}01$) is used by $\mathrm{SE}_{0,1}$ for routing. So output link 1 of $\mathrm{SE}_{0,1}$ is selected. At $\mathrm{SE}_{1,2}$ the second bit of D ($1\underline{0}1$) is used and output link 0 of $\mathrm{SE}_{1,2}$ is used. Finally, at $\mathrm{SE}_{2,3}$ the third bit ($10\underline{1}$) of D is used and output

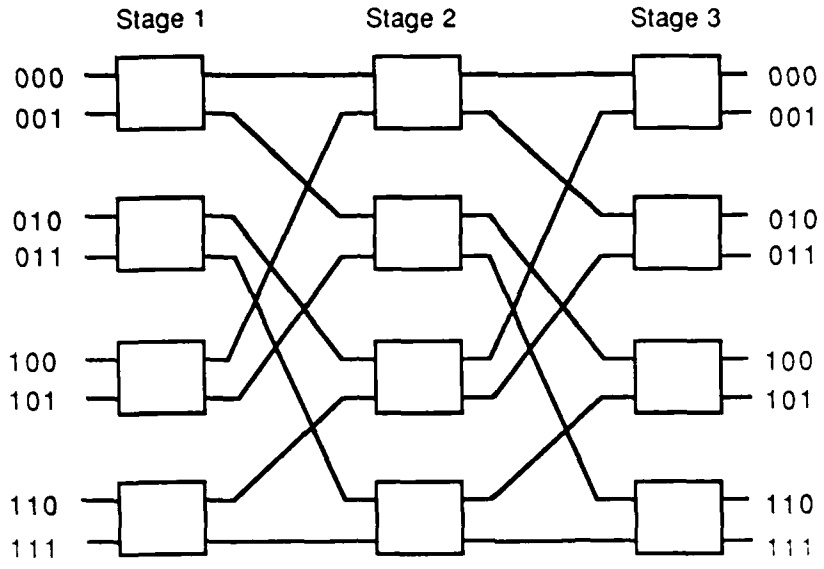


Figure 3.1: 8×8 Shuffle-Exchange Multistage Interconnection Network.

link 1 of $SE_{2,3}$ is selected. So $S = 000$ delivers the message to $D = 101$ using only the destination's address for routing control. Figure 3.2 shows this S-D connection.

3.3 Shuffle-Exchange MIN Plus (SEN+)

An $N \times N$ SEN+ network is an $N \times N$ SEN with an additional stage. Figure 3.3 shows an 8×8 SEN+. The first stage (labeled stage 0) is the additional stage. The addition of the extra stage requires implementation of a different control strategy. Several control strategies for the SEN+ network can be selected. However, the strategy chosen may affect both the bandwidth and the reliability of the network.

Adding a stage to the SEN allows two paths for communication between each source and every destination. (Recall that the SEN is an unique-path

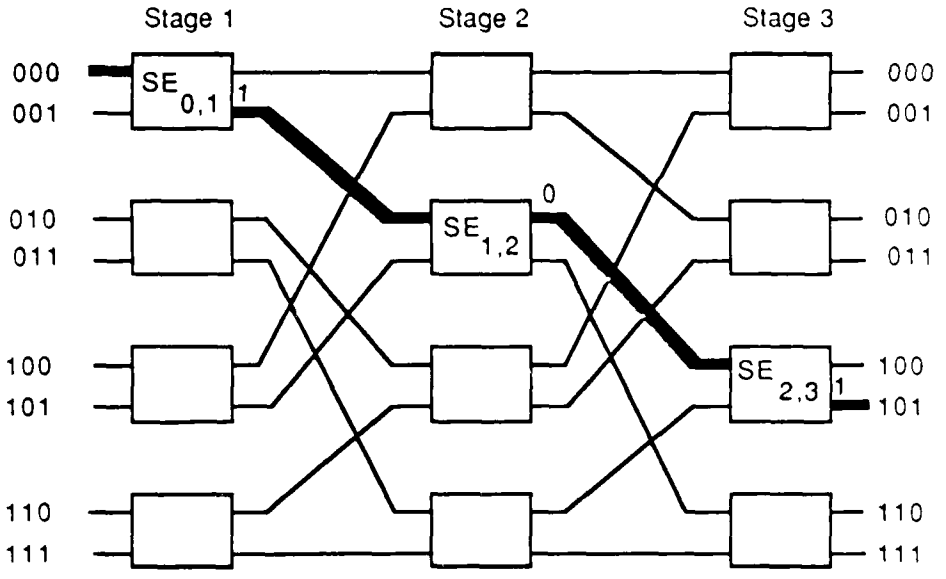


Figure 3.2: Routing for Communications Between $S = 000$ and $D = 101$ in the 8×8 SEN.

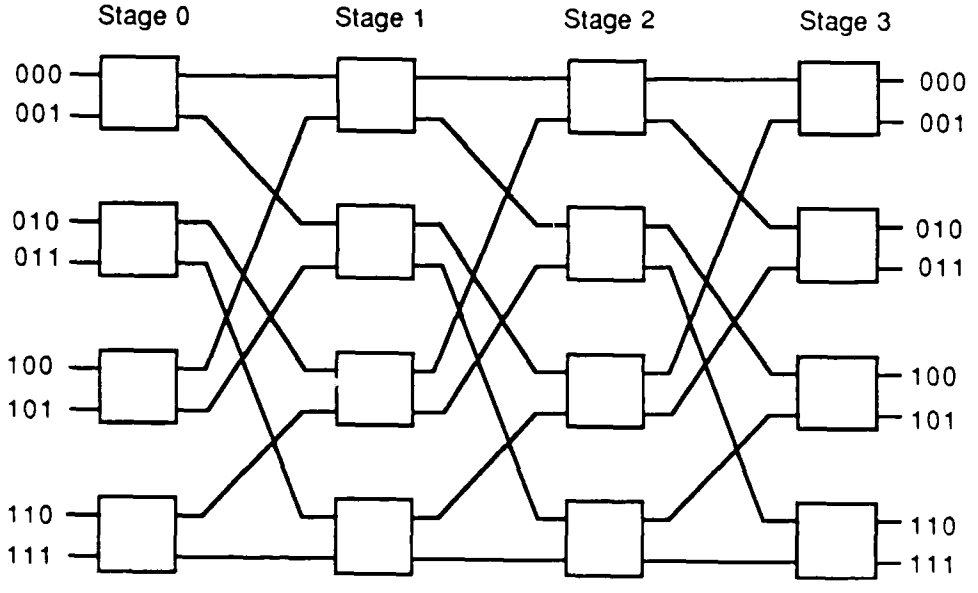


Figure 3.3: 8×8 Shuffle-Exchange Multistage Interconnection Network with an Extra Stage.

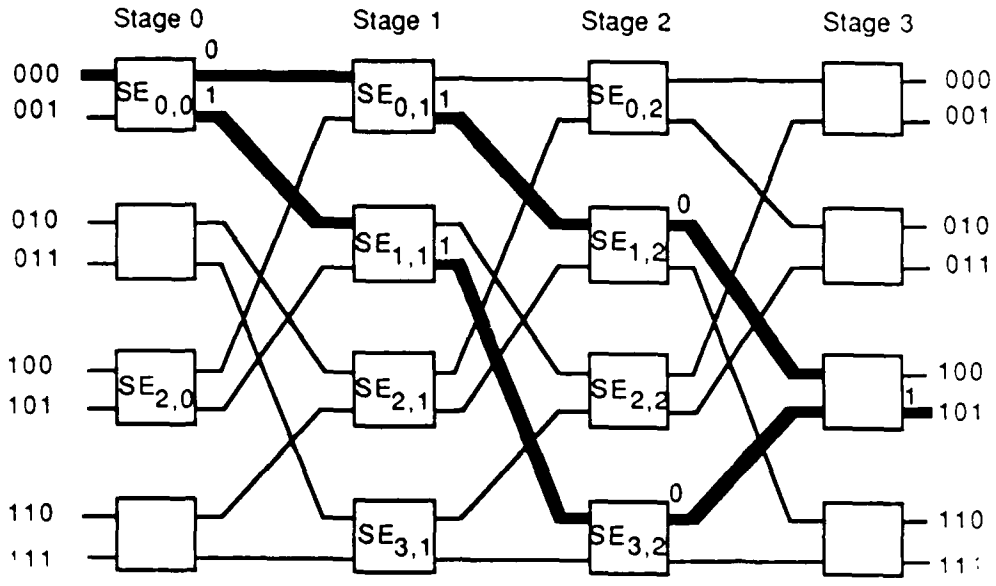


Figure 3.4: Two Paths for Routing Communications Between $S = 000$ and $D = 101$ in the 8×8 SEN+.

MIN.) While the paths in the first and last stages of the SEN+ are not disjoint, the paths in the intermediate stages do traverse disjoint links. As can be seen in Figure 3.4, $S = 000$ can reach $D = 101$ by two paths. So path redundancy is achieved in the SEN+ at the expense of one extra stage added to the SEN. The network complexity is $(N/2)(\log_2 N + 1)$. Thus, the cost of the SEN+ over that of the SEN is $N/2$ switches or a fractional increase of $1/\log_2 N$, small indeed for large N. One question to be addressed in Chapter 5 is how much increase in reliability is obtained by this amount of redundancy.

Since the purpose of the extra stage in the SEN+ is for reliability enhancement, several control strategies may be considered. First, a switching element in stage 0 remains in a straight-through (T) setting until it detects a failure of the switching element in stage 1. Then, the SE in the first stage selects the exchange (X) configuration for subsequent memory accesses. This

strategy allows two paths for each S-D pair given that failures only occur in the second stage; however, it ignores the status of the SEs in $\log_2 N$ of the stages.

In the second strategy, a switching element in stage 0 uses the T setting until a failure in a SE along the path from a given S to a given D is detected. At that time, the SE in stage 0 is placed in the X setting for all future accesses between that S-D pair. In this way, two paths between each S-D pair are realized given that the failures occur only in the intermediate stages of the SEN+.

Finally, one can modify the second strategy so that if a failure occurs in the last stage of the SEN+, then the network reconfigures itself so that no further accesses are made to the two Ds attached to the SE in the last stage. Since several paths are no longer considered, this will reduce congestion within the reconfigured network. In the remainder of this thesis, the unmodified second strategy will be considered.

Figure 3.3 shows that the network complexity for the 8×8 SEN+ is 16. There are 8 sources, 8 destinations, and 4 (i.e., $\log_2 N + 1$) stages each with 4 SEs.

3.4 Redundant SENs

Another scheme for providing fault-tolerance in unique-path MINs is the complete replication of the network. Let K be the number of copies of the network, then since these networks are arranged in parallel the K-redundant network is $(K-1)$ fault-tolerant. The cost of a K-redundant SEN is at least K times the cost of the SEN since K copies are necessary and additional links are required from the sources to the network and from the network to the destinations. The case of $K = 2$ will be considered in Chapter 5.

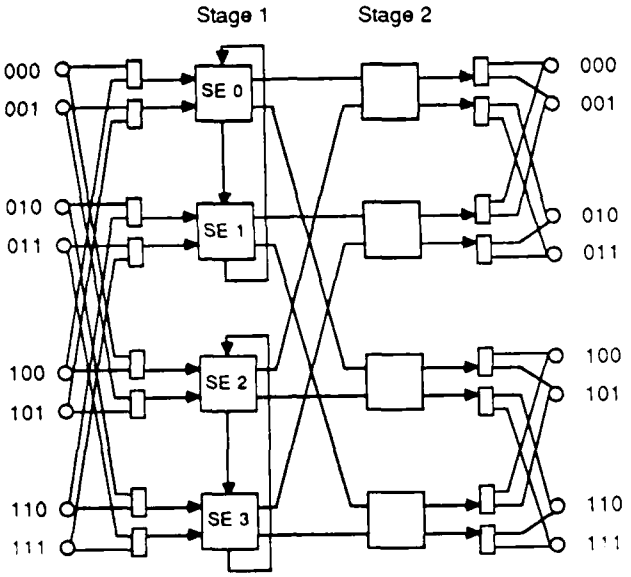


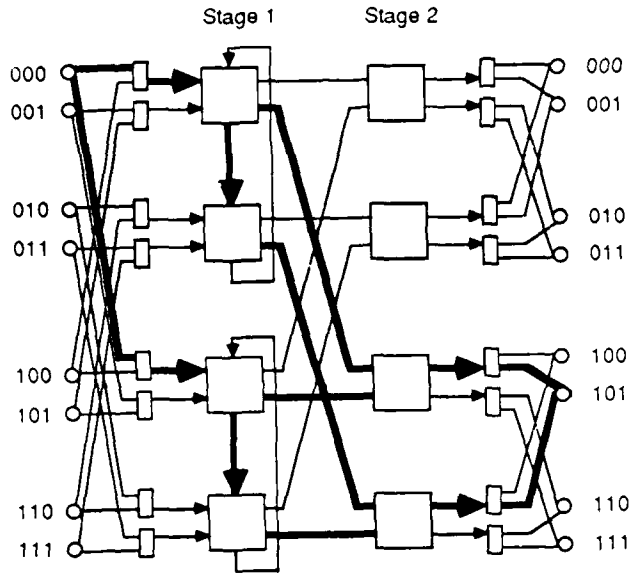
Figure 3.5: 8×8 Augmented Shuffle-Exchange Multistage Interconnection Network.

3.5 Augmented SEN (ASEN)

An Augmented Shuffle-Exchange Network (ASEN) is a SEN with one less stage, additional intrastage links called *auxiliary links*, multiplexers, demultiplexers, and a slightly more complex switching element. The ASEN obtained from modification of the corresponding SEN constructed from 2×2 SEs is considered in this thesis. (In [53], this MIN is called an ASEN-2.) The ASEN has N 2×1 multiplexers, N 1×2 demultiplexers, and $\log_2 N - 1$ stages of $N/2$ switches. Figure 3.5 shows an 8×8 ASEN. The SEs in the last stage are of size 2×2 or SE_2. (This is the basic SE used to construct the SEN and SEN+ networks.) The remaining switching elements are of size 3×3 denoted as SE_3. In each stage, the SEs can be grouped into *conjugate pairs*. That is, the SEs in such a pair are connected to the same pair of SEs in the next stage. These conjugate pairs can then be grouped into *conjugate subsets*,

where a conjugate subset is composed of all SEs in a particular stage that lead to the same subset of destinations. The ASEN achieves the multiple-path property by permitting two SEs in the same conjugate subset that are not a conjugate pair to communicate through auxiliary links. The SEs which communicate through the use of auxiliary links are called a *conjugate loop.* The conjugate loops are formed in such a way that the two switches forming a loop have their conjugate switches in a different loop. These pairs of loops are called *conjugate loops.* Observe that this construction of the network has two benefits. First, the network can tolerate the failure of both switches in a conjugate loop. Second, it also provides a topology which lends itself to on-line repair and maintainability. That is, a loop can be removed from the ASEN without disrupting the operation of the network. In stage 1 of the 8×8 ASEN shown in Figure 3.5, SEs 0, 1, 2, and 3 form a conjugate subset; within that subset, SEs 0 and 2 are a conjugate pair; and SEs 0 and 1 form a conjugate loop. Figure 3.6 shows the multiple paths between $S = 000$ and $D = 101$. The network complexity for the $N \times N$ ASEN is $(N/2)(\log_2 N - 1)$, but the SEs are not all of size SE_2.

A self-routing algorithm is also used for the ASEN. Each source has a primary multiplexer and SE and a secondary multiplexer and SE. Each source attempts entry into the ASEN via its primary multiplexer and SE. If either primary component is faulty, the request is sent to the secondary multiplexer. If the secondary multiplexer is faulty, the ASEN is failed. For stages 1 through $n - 2$, requests are first routed through the usual output link; if it is busy or if the successor SE (in the next stage) is faulty, routing is attempted via the auxiliary link. A faulty demultiplexer at the output of the ASEN is regarded as a failure of its associated SE in stage $n - 1$. So the algorithm essentially enables a SE to detect a failure of its successor SE and re-route the request



8x8 ASEN Showing Multiple Paths Between S = 000 and D = 101.

Figure 3.6: 8×8 ASEN Showing Multiple Paths Between $S = 000$ and $D = 101$.

whenever possible. The ASEN is failed if a request that is not blocked, does not find a path to its destination.

Chapter 4

Performance

4.1 Introduction

Depending on the application, a number of performance criteria are available for evaluating competing MIN designs. For example, the number and/or classes of permutations realizable, the fault-tolerance properties, control complexity, expected throughput, expected bandwidth, and expected delay may be considered when selecting a MIN for a specific application.

First, a review of previous work on performance measures for networks is presented. The principal efforts in this area are concerned with the permutation capability, probability of acceptance, and expected bandwidth. Next, the usefulness of the bandwidth as a reward rate for performability models of MINs viewed as a separate system and as a component of a complete multiprocessor system is discussed. This is followed by the development of analytic expressions for the bandwidth of the crossbar network and the unique-path MIN.

4.2 Previous Work

One performance measure that has been studied extensively is the *permutation capability* of a network. This measures the connectivity of the number of *S-D* pairs realizable in the network. Several researchers have examined this measure for various multistage interconnection networks. For example, in [2] the Extra Stage Cube (ESC) is introduced. It is tolerant of a single switch failure. The ESC is a Generalized Cube with an additional stage and 1×2 demultiplexers and 2×1 multiplexers on both sides of the first and last stages. Adams *et al.* address permutation issues and mention that fault-tolerant interconnection networks can help achieve reliability goals in a multiprocessor system. However, no reliability analysis is performed. In the case of a MIN, the *permutation capability* refers to the fraction of all possible permutation requests that can be realized with no blocking [95].

One shortcoming of shuffle-exchange interconnection networks is that only one path exists from every source, S_i, to every destination, D_j. Thus, two different settings of switching elements will result in two different permutations. Consequently, if a switch does become faulty, many permutations will not be admissible by the network. To overcome this deficiency, it was proposed in [29] that these networks be augmented by adding one additional stage, so that in the event of a single faulty switch, one is still able to realize all the permutations using at most two passes through the network. This introduced a class of interconnection networks, called *two-path interconnection networks.* In these networks, any source can be connected to any destination through two disjoint paths. Therefore, if a switch in the network becomes stuck-at-T or X, any source can still be connected to any destination, and all permutations can still be realized by the faulty network in two passes [72].

This work was extended by Padmanabhan and Lawrie to *R-path interconnection networks.* In [66], Multipath Omega networks are introduced. This paper explains how to construct an R-redundant path MIN where R is the number of disjoint paths between a S-D pair. In [65], the construction of the Modified Omega network (which is similar to the network in [66]) is discussed. The Modified Omega network is an Omega network with a sufficient number of additional switching elements and links to provide a desired level of $(R-1)$ fault-tolerance. The permutation capability of these networks is also discussed in [65].

The concept of multiple passes through a network is embodied in a fault-tolerance measure of MINs called *dynamic full access. Dynamic full access* refers to the ability of the network with PEs as both sources and destinations to transfer data from one PE to another PE in a finite number of passes either directly or by routing the data through other PEs. Because this technique requires the intermediate storage of data, it is more suited to packet-switched networks [77].

As mentioned earlier, in a fully-operational Delta network with an additional stage, the problem of performing arbitrary permutations in a multiple number of passes was shown to be equivalent to the vertex-coloring problem in graph theory [77]. The general problem of realizing a permutation in the minimum number of passes through the network is intractable, so a restricted class of permutations are analyzed. Graph-theoretic techniques are used and both the fault-free and faulty-SE cases are examined. This modified Delta network is equivalent to the SEN+.

Another measure used to quantify the circuit-switching performance of a MIN is the *probability of acceptance* [68]. This measure is the probability that, in a random access environment, a request submitted by a source

is accepted by a destination without getting blocked by other requests or connections in the network. This probability is usually evaluated by assuming that all the sources simultaneously generate their requests for connection with a probability p, aimed at uniformly chosen destinations, at the beginning of a cycle. If these requests arrive at a switch requiring the same output link, the requests that are serviced are chosen at random and the others are blocked and dropped. The *probability of acceptance* is defined as the ratio of the expected number of successful requests to the expected number of the requests submitted by the sources.

Expected bandwidth is another commonly used metric for analyzing MINs. The expected bandwidth is defined as the average number of destination requests accepted per cycle, conditioned on the rate of destination requests. This is the measure used in this thesis as the *reward rate* for a given configuration of a degradable network. Rewards will be discussed in a subsequent chapter.

The crossbar network has the highest possible bandwidth. In a crossbar, as long as no two sources request the same destination, all requests will be accepted. However, in an environment where requests are issued in a random fashion, the memory bandwidth of a crossbar is much less than its capacity [12]. As might be expected, in a MIN the bandwidth will be even less because of additional conflicts in the network. Interference analysis of MINs has been studied in [68], [26], and [96].

Kruskal and Snir [46] examine the performance of MINs assuming fault-free operation. Both the buffered and unbuffered Banyan networks are examined in a packet-switching environment. In the unbuffered case, they derive an asymptotic equation for the probability that a request issued at a source arrives at its intended destination. This probability is inversely proportional

to the number of stages in the network. It was shown in [54] that the bandwidth of a SEN is very high for operations that do not conflict; however, in [46] under the assumption of a random access pattern from the sources to the destinations, they found the effective bandwidth in a MIN to be $O(N/\log N)$. This means that contention within the network reduces bandwidth by a factor of $O(\log N)$. In [71], the problem of hot-spot contention in SENs was investigated. In this model, the destinations are shared memory modules, and it allows a small number of accesses to be made to a specific memory while all other accesses are uniformly distributed. The results show a rapid decrease in effective bandwidth as the correlation of accesses increases.

In [24], Das and Bhuyan use simulation to determine the reliability and performance of a multiprocessor system with three interconnection networks in a random access environment: a multiple-bus, a crossbar, and a MIN with a centralized controller. Since deriving analytical solutions for the bandwidth of a randomly truncated multiprocessor system using a MIN or a multiple-bus structure is extremely difficult, simulation is used to obtain results. The model assumes that the multiprocessor system is executing a task requiring I processors and J memories. To determine the reliability of the system they require that at least I processors and J memories are operational and that they can communicate. Then, the bandwidth is determined using exactly I processors and J memories. Previous performance analyses for these networks were done in [14], [13], and [12]. However, the analytical models used for the MIN and multiple-bus interconnection network do not hold when random faults are considered.

A *chained network* was introduced in [99] which is similar to the ASEN presented in [53]. The chained network provides redundant paths between every source-destination pair so that all single faults and many multiple faults

can be tolerated. The proposed network meets the criteria for the design of a fault-tolerant network listed in [20], and it also has a bandwidth comparable to that of a crossbar. In [100], the performance of such a network was studied. An analytical model was employed to evaluate the bandwidth of the network operating under both fault-free and fault-present conditions. Simulations were utilized to explore the average delay when buffers are incorporated into the network, and it was demonstrated that network delay can be reduced by controlling the threshold value. In addition, performance degradation caused by a single fault in a network was investigated. They use the Baseline network as an example to illustrate their scheme, and perform a probabilistic failure analysis of a circuit-switched MIN and a simulation for the analysis of a MIN with output buffers in the SEs (under a packet-switching assumption). Bandwidth analysis was performed on an unbuffered MIN operating both with and without faults.

4.3 Bandwidth as a Performance Measure

The average number of busy memories (memory bandwidth) will be used as the performance level (reward rate) for a particular system configuration. This is an appropriate choice of performance metric for the multiprocessor system since the efficiency of the system will be limited by the ability of the processors to randomly access the available memories.

In the case of a crossbar switch, contention for the memories occurs only at the memory ports since the crossbar switch is non-blocking. But, in the case of the SEN network, contention occurs inside the interconnection network, as well, since this network is a blocking network. That is, if two or more processors compete for the same output link of a SE, only one request will be successful and the remaining requests will be dropped.

Over time, components of the multiprocessor system can be expected to fail, and as a result, the performance of the system can be expected to decrease. To determine the performance of the crossbar, the model developed by Bhandarkar [12] to obtain the average number of busy memories or memory bandwidth will be used, and an extension of the performance model in [68] will be used for the SEN network.

In determining the bandwidth of a given configuration of the multiprocessor system, the assumptions stated in [68] for analysis of circuit-switched networks will be used. The assumptions are:

1. At the beginning of each memory access cycle, every operational processor issues a request with the same probability.

2. The requests are randomly and uniformly distributed among all memories.

3. Blocked requests in any cycle are ignored. A new set of requests is issued in each cycle.

Assumption 3 may appear to oversimplify the model since, in practice, blocked requests are normally resubmitted during the next network cycle. However, work performed by [12] and others on more complex problems, and studies done by Patel [68], indicate that assumption 3 has only a minor impact on the results obtained. Furthermore, this assumption makes the analysis more tractable.

In the following two sections, the bandwidths of the crossbar and the unique-path MIN are developed. Let p_{in} denote the probability that a processor issues a request during a particular memory request cycle, and p_{out} denote the probability that a particular memory receives a request at its

input link. Since it is assumed that requests are not buffered in the interconnection network, nor are multiple requests accepted at a memory on any cycle, computation of the memory bandwidth for the multiprocessor system is accomplished in a straightforward manner.

4.4 Crossbar Bandwidth

In the case of an $n \times n$ crossbar switch, the probability that a particular processor requests a particular memory is p_{in}/n for a given network cycle. So the probability that a particular processor does not issue a request for a particular memory is $(1 - p_{in}/n)$. By the independent event assumption, the probability that a particular memory is not requested by any processor is $(1 - p_{in}/n)^n$. Therefore, the probability that a particular memory is selected by at least one processor is just the complement of this value, or

$$p_{out} = 1 - (1 - \frac{p_{in}}{n})^n. \tag{4.1}$$

The bandwidth (BW) for the system, which is the average number of memories requests accepted in a particular memory access cycle, is just p_{out} times n, hence

$$BW_{xbar} = n(1 - (1 - \frac{p_{in}}{n})^n). \tag{4.2}$$

In the presence of memory and/or processor failures, this equation must be modified since the number of operational memories will not, in general, equal the number of operational processors. In [12], a detailed combinatorial and Markovian analysis was performed to determine the bandwidth in the asymmetric case. Let i denote the number of operational processors and j denote the number of operational memories. Further, let $\ell = min\{i, j\}$ and $m = max\{i, j\}$. Then for $p_{in} = 1.0$, Bhandarkar found the average

bandwidth of the system to be accurately predicted by the formula,

$$BW_{xbar} = m(1 - (1 - 1/m)^{\ell}). \tag{4.3}$$

4.5 MIN Bandwidth

Now consider the $N \times N$ MIN with switching elements of size $n \times n$. Number the stage to which the processors are attached as stage 1, and the last stage to which the memories are attached as stage ν. The switching elements are $n \times n$ crossbars, and the output of a particular link of a switching element can be denoted as p_i. This value is also the probability that there will be an input request for a SE in the next stage. A recurrence relation exists for computing these request probabilities. That is,

$$p_{i+1} = 1 - (1 - \frac{p_i}{n})^n. \tag{4.4}$$

Consider the SEN as a specific example. The probability of a request at the input of a SE in stage i, $i = 1, 2, \ldots, \nu$, can be denoted as p_{i-1}, then the probability of a request for an output of a SE at stage i will be p_i and can be computed as

$$p_i = (1 - \frac{p_{i-1}}{2})^2, \ i = 1, 2, \ldots, \nu. \tag{4.5}$$

Note that $p_0 = p_{in}$ (the probability that there is a request for the first stage) and $p_\nu = p_{out}$ (the probability that there is a request for a particular memory at its associated network output link). In the case of the 16×16 SEN, the probability of a request at the output link of a SE in stage 1 will be

$$p_1 = (1 - \frac{p_0}{2})^2 \ , \tag{4.6}$$

and the probability of a request for a given destination (the output link of stage 4) will be

$$p_4 = (1 - \frac{p_3}{2})^2 \ . \tag{4.7}$$

The bandwidth is then computed as the product of the request probabilities for a particular memory and the number of memories, hence from [68]

$$BW_{\text{MIN}} = N(1 - (1 - \frac{p_{\nu-1}}{n})^n). \tag{4.8}$$

Of course, assuming that each destination is equally likely to be requested by a given source, the bandwidth is simply the probability of a request for any destination times the number of destinations. The computation of bandwidth, however, is not so easy when the probability of requests for the destinations are *not* uniformly distributed or one or more SEs have failed. It is assumed that after a SE has failed, its output links will not be active. Thus, p_i from a failed SE in stage i is zero. Further, the request probabilities that feed a particular SE may not be equal. In the presence of failures, equation (4.8) must be modified to account for graceful degradation. Consider a particular input link to an $n \times n$ SE, say link 0 in Figure 4.1, and denote it by $p_{in,0}$. It may request a particular output link with equal probability, so it will *not* request a specific link with probability $(1 - p_{in,0}/n)$. Similarly, input link 1 will *not* request the same link with probability $(1 - p_{in,1}/n)$. The request probability for a specific output link, say i, as a result of the (perhaps unequal) request probabilities by the input links is then computed as

$$p_{out,i} = \begin{cases} 1 - \prod_{j=0}^{n-1}(1 - p_{in,j}/n) & \text{if the SE has not failed, and} \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

The bandwidth of the SE is then

$$BW_{SE} = \begin{cases} n(p_{out,i}) & \text{if the SE has not failed, and} \\ 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

The outputs of this SE will serve as inputs to n of the SEs in the next stage. At the final stage of the MIN, some memories may be inoperable so the network bandwidth is computed as the sum of the request rates for the

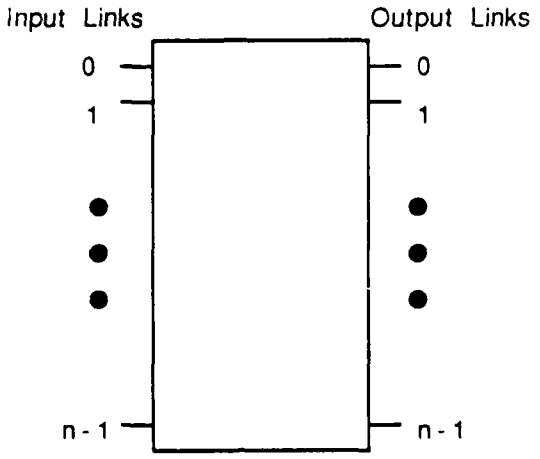


Figure 4.1: $n \times n$ Switching Element.

operational memories. Let N_0 denote the set of operational memories. Then,

$$BW_{MIN} = \sum_{j \in N_0} (p_{out})_j \ . \tag{4.11}$$

Equations (4.3) and (4.11) will be used to compute the bandwidth for the crossbar and the SEN networks, respectively.

It was mentioned that the SEN is a blocking network, whereas the crossbar was not. Assuming fault-free operation and $p_{in} = 1.0$, Figure 4.2 shows the degradation factor (BW/N) for these two networks as a function of the size of the network. For networks of size 256×256 and larger, the bandwidth of the crossbar is at least twice that of the SEN. However, recall that the cost of the crossbar is $O(N^2)$. If the crossbar is modeled as a system composed of demultiplexers/multiplexers as in [12], then the implication of equations (4.3) and (4.11) and Figure 4.2 is that the MIN is more susceptible to the failure-induced loss of bandwidth than the crossbar network.

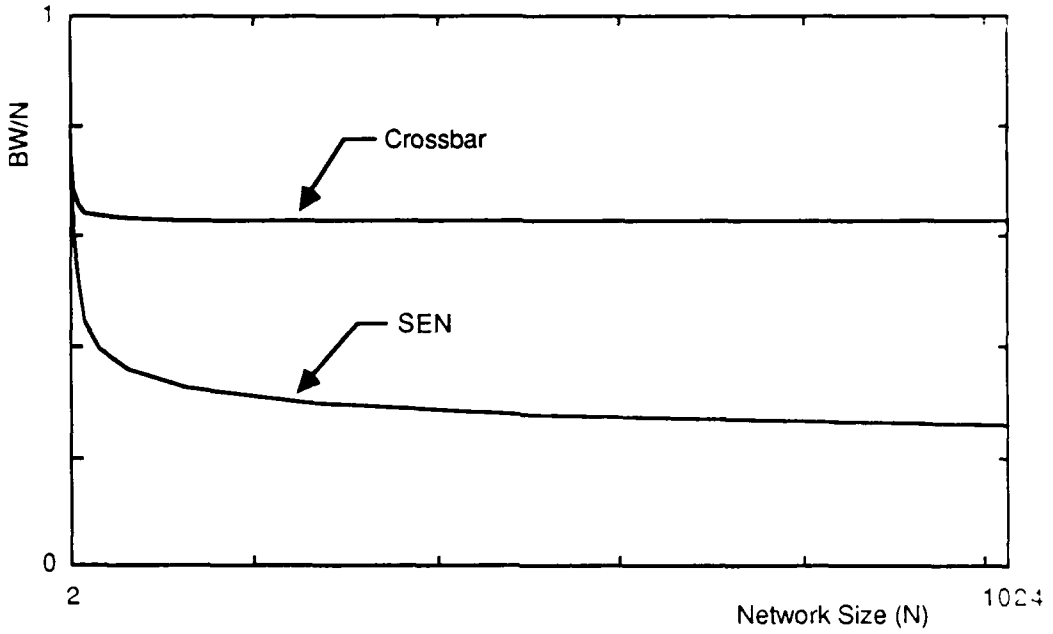


Figure 4.2: Bandwidth Degradation as a Function of Network Size.

4.6 Summary

Bandwidth will be used as the performance metric for analyzing the networks in this thesis. Analytic expressions for the bandwidth of a crossbar network and a MIN in a degradable environment have been presented and will be used to establish the reward structure associated with the Markov reward models discussed in Chapter 7 and in the analysis of a multiprocessor system in Chapter 8.

Chapter 5

Reliability

5.1 Introduction

A number of schemes have been proposed to increase the reliability and fault tolerance of Multistage Interconnection Networks (MINs). The modest cost of unique-path MINs make them attractive for large multiprocessor systems, but their lack of fault-tolerance is a major drawback. To mitigate this problem three hardware options are available: (1) replicate the entire network; (2) add extra stages; (3) and/or add additional links. Adding an additional network doubles the cost while adding an extra stage requires only $N/2$ additional SEs in an $N \times N$ network. Adding links not only increases the number of links, but it also requires a more complex switching element. Also, adding interstage links is not practical for large-scale VLSI applications [102]; however, adding intrastage links is still viable.

In this chapter, the reliability issues relating to MINs are examined. First, previous work in this area will be covered. Next, definitions of an operational network and a description of the measures used to compare the networks are introduced. Then transient reliability analysis of the crossbar, SEN, SEN+, and ASEN will be presented. Since the reliability of crossbar switches has

been studied under several connectivity assumptions, the emphasis in this chapter is on reliability analysis of MINs.

The analysis of the SEN and SEN+ networks is divided into four parts. Exact transient reliability analysis of small SEN and SEN+ networks is presented first. Then, lower and upper bounds for approximating the reliability of larger networks are derived. The lower bound obtained is compared to the exact solutions derived for the 8×8 and 16×16 SEN+ to verify that it is a close approximation of SEN+ reliability, and then this lower bound is used for analyzing SEN+ networks up to size 1024×1024. Next, a comparison of the mean time to failure ($MTTF$) of these networks is presented. Finally, a discussion on how network reliability is affected by the underlying component-lifetime-distributions is presented.

In Section 5.7, the reliability of the ASEN is analyzed. The exact reliability expressions for the 4×4 and 8×8 ASEN are derived. This is followed by the development of bounds. Then, these bounds are used to compare the $MTTF$, normalized mean-time-to-failure, cost, and mission time improvement factor of the networks.

It is shown that the lower-bound reliability of the ASEN dominates the upper-bound reliability of the SEN+. Furthermore, ASEN reliability analysis is extended to include imperfect coverage and on-line repair using a novel hierarchical approach. Block diagrams have been used to model the steady-state and instantaneous availability of systems with independent repair [83]. In this chapter, a *two-level hierarchical approach* is used to model the *reliability* of a repairable system. The top level is a reliability block diagram while the bottom level is a Markov chain. In this analysis, the increased complexity of the SEs in the network is considered instead of assuming that the various components have identical failure rates.

5.2 Previous Work

There are several papers which address reliability issues pertaining to MINs. A reliability analysis of the C.mmp and Cm* was performed in [44], but only processor and memory failures were considered. In [43], reliability of the crossbar, shared bus, and multiport memory structures was analyzed using graph models. And in [3], the fault tolerance of MINs, considering control line and link failures in the SEs, was examined. The emphasis was on finding the critical faults that destroy the *dynamic full access* (DFA) property, but DFA between specified source and destination subsets was not considered. (Note that DFA may require several passes through the network.)

The reliability issues pertaining to tightly-coupled multiprocessor systems using circuit-switched communications were discussed in [24]. This model considered processing elements (PEs), memory modules (MMs), and switch failures. A reachability matrix, constructed from a graph model, was modified depending on various faults. Given that a task requires a specified number of MMs and PEs, the system is considered operational as long as these resources and the DFA property between these resources exists. The system state was obtained by searching for a fully-connected system in the reachability matrix that satisfied the minimum resource requirements. Simulation results indicated that MINs are worse than crossbars if failures are taken into account, and the multi-bus performed the best because of the large number of alternate paths between PEs and MMs.

In addition, several researchers [2,19,59,66,65,67,76,75] have reported on the use of multiple-path MINs as a means of improving the fault-tolerance and reliability of interconnection networks. For example, in [67] the Gamma network is examined for the terminal reliability of the network, but neither PE and MM failures nor performance degradation are considered.

Redundancy graphs offer a convenient way to study multiple-path MINs to determine such properties as the number of faults tolerated or the type of rerouting possible. A redundancy graph depicts all the available paths between a given source-destination pair in a MIN. It consists of two distinguished nodes — the source S and the destination D — and the rest of the nodes correspond to the switching elements that lie along the paths between S and D. Its principal use is for terminal reliability calculations.

A general criterion for the evaluation of the robustness of the MIN is that every member of a subset of sources must have paths to every member of a subset of destinations given that each switch has a certain reliability. (The reliability of a switch is the probability that it is fault-free.) The probability that the above criterion is satisfied is called *multi-terminal reliability*. Two special cases of this criterion are of interest. The first case is when the subsets of sources and destinations contain exactly one element each. This leads to a measure called *two-terminal reliability*, or simply *terminal reliability*, which is the probability that a given source-destination pair has at least one fault-free path between them. The other special case of the multi-terminal reliability criterion is full connectivity between all the sources and all the destinations. This special case leads to the assumption that the MIN has failed whenever all the paths are disconnected between some source-destination pair, and it establishes the reliability of the MIN.

The criterion of *full connectivity* for a multiprocessor system is too narrow a view of reliability. It does not consider the ability of a system to operate in a degraded mode. It may be acceptable for a system to be considered operational as long as some subset of sources and destinations can communicate. This view of graceful degradation recognizes that the failure of a basic component should not cause system failure. Rather the system should be

able to detect any faulty module and also have the ability to reconfigure and continue to perform in a degraded mode. Analysis of the degradation behavior of such a system is done using a transient reliability analysis. Of course, even with transient analysis, one can still obtain the mean time to failure of the MIN, which is the expected time elapsed before network failure.

The focus of the reliability analysis that has been performed on MINs, however, has been either: (1) in terms of the average number of switch failures tolerated and mean time to failure; or (2) on terminal reliability, a measure often used for packet-switching applications. Analysis using the former measure can be found for the F-Network [20]; the Augmented C-Network (ACN) and Merged Delta Network (MDN) [79]; the Augmented Bidelta Network (ABN) [52,51]; and the Modified Omega network [64]. In addition, terminal reliability analysis has been performed on the Gamma network in [76], INDRA network in [75], and the ACN, ABN and MDN networks in [51].

In [18], the SW-banyan network with added stage(s) composed of $f \times f$ switches is analyzed. Cherkassky *et al.* derive a reliability expression for this network. The expression considers both link and switch failures, but it assumes that the network can only tolerate $f - 1$ failures. Therefore it provides a rough lower bound since there are many operational configurations of the network which permit more than $f - 1$ failures. This underestimates network reliability.

In [51], Kumar compares the mean time to failure of the Augmented Shuffle-Exchange Network (ASEN) with that of several other MINs. $MTTF$ data on the INDRA [78], F [20], modified Omega [65], and SEN networks for $N = 8$ through $N = 1024$ are provided for comparison. In all cases the ASEN is superior. In this analysis, however, the lower bound is based on only one switching element type and the multiplexers and demultiplexers associated

with the network are ignored. A more detailed model for the reliability and $MTTF$ of the ASEN which incorporates added network complexity due to different types of switching elements and multiplexers and demultiplexers is considered in this chapter.

Network reliability analysis is known to be NP-hard [74]. It is for this reason that other authors (e.g., Das and Bhuyan in [24]) have resorted to Monte-Carlo simulation to examine "small" networks. In this thesis, exact reliability expressions for up to 16×16 networks are derived, and a closed-form tight lower bound for larger networks is presented. Using this lower bound, numerical answers for up to 1024×1024 networks are computed.

5.3 Definitions of an Operational Network

Before any reliability analysis can be performed, a clear understanding of what constitutes an operational network must be established. That is, what is meant by system failure? There are at least three definitions of an operational network:

1. The network is operational as long as every source can communicate with every destination.

2. The network is functioning properly as long as some source can communicate with some destination.

3. The network is operational as long as U sources can communicate with V destinations.

It should be clear that a network operating under definition 1 will have the shortest time to failure, while the same network operating under definition 2 has the longest time to failure. Since definition 1 is the view most often

used for modeling MINs, this definition will be adopted for the following analysis. However, for some network applications, the other two definitions are appropriate.

Also, it is assumed that the components of the network have independent lifetime distributions, and that they are either fully-operational or failed. That is, stuck-at-T or stuck-at-X faults are not considered.

5.4 Comparative Measures

In this section, the measures used to compare the networks are introduced. The measures are: the *reliability* as a function of mission time ($R(t)$), *mean time to failure* ($MTTF$), *normalized* $MTTF$ ($NMTTF$), *mission time improvement factor* ($MTIF$), and cost.

Let T be a random variable representing the lifetime of a particular system, then its *reliability* can be defined as

$$R(t) = \text{Prob}[T > t]. \tag{5.1}$$

The *mean time to failure* is simply the integral of the reliability over the interval from zero to infinity,

$$MTTF = \int_0^\infty R(t)dt \ . \tag{5.2}$$

The *normalized mean-time-to-failure*, $NMTTF$, is a comparative measure of reliability. It is defined as the ratio of the $MTTF$ of a network with redundancy and the $MTTF$ of the unique-path MIN.

Let Υ denote the time for the system to decrease from a fully-operational system (at time $t = 0$) to some specified reliability. Υ is an useful absolute measure of reliability in its own right because it provides information regarding the suitability of a given system for a particular mission. However,

a comparative measure is desirable for the analysis of the networks. The *mission time improvement factor* $MTIF$ [57] reflects the improvement in the maximum mission time for some desired minimum mission reliability as a result of adding redundancy to the SEN. For example, let $\Upsilon_{\text{SEN+}}$ be the time for the SEN+ to reach some desired mission reliability, R_{desired}, and Υ_{SEN} be the time for the basic SEN to reach the same mission reliability, then

$$MTIF(R_{\text{desired}}) = \frac{\Upsilon_{\text{SEN+}}}{\Upsilon_{\text{SEN}}} \tag{5.3}$$

represents the factor by which mission time is increased by using the SEN+ instead of the SEN.

Finally, cost is a significant measure. Many times modifying a given system to provide fault-tolerance requires more than merely adding components. To properly compare different modification schemes, the cost of the schemes must be normalized on some basis. In the case of the SEN, the number of "equivalent" 2×2 SEs (SE_2) in the SEN+ and ASEN is used to normalize the cost. The ASEN is constructed from demultiplexers, multiplexers, 3×3 SEs (SE_3), and 2×2 SEs; whereas the SEN+ is composed entirely of 2×2 SEs. The SEs are considered crossbar switches so an $n \times n$ SE has $4n(n-1)$ gates [47], and the multiplexers/demultiplexers have $2(n-1)$ gates where n is the number of input/output links. The SEN+ is simply a SEN with $N/2$ additional 2×2 SEs. But in the ASEN, some of the 2×2 SEs have been replaced by 3×3 SEs and multiplexers and demultiplexers have been added. In order to make a fair comparison, gate counts in the network components are used to compensate for the differences in the network's construction. For example, a SE_2 has 8 gates whereas a SE_3 has 24, so a SE_3 is three times as complex as a SE_2. The "normalized" network complexity of an $N \times N$ ASEN is then $(3N/4)(1 + 2(\log_2 N - 2))$.

5.5 Crossbar Networks

Reliability analysis of the crossbar network has been studied in several papers. In [87], the C.mmp system from Carnegie Mellon University was studied. In that paper, the crossbar was considered as a single large switch. In [9] and [88], a more detailed model of the crossbar was considered by introducing the aspect of coverage. However, in any model that considers the crossbar as a single switch, the reliability analysis of such a model using a Markov chain has only two states. Also note that all three definitions of what constitutes an operational network will be identical from the perspective of the network.

In a later chapter, the crossbar network will be analyzed by decomposing the crossbar into demultiplexer/multiplexer components. The crossbar will then be considered as a component of an entire multiprocessor system. Definition 3 will be used to analyze this system. It will be shown that modeling the crossbar in more detail shows that the network has a much higher reliability than indicated by the simple model.

5.6 SEN and SEN+ Networks

In this section, the reliability of the unique-path Shuffle-Exchange multistage interconnection Network (SEN) and a variant of the SEN called the SEN+ are analyzed. The SEN+ network has an additional stage which is used in an attempt to increase the reliability of the basic SEN. However, this effort is not successful in all cases. A comparison of the SEN and SEN+ networks as a result of transient reliability analysis is presented, as well as a discussion of the distributional sensitivity of the reliability of these networks when their components have increasing-failure-rate (IFR) lifetime-distributions.

5.6.1 Exact Reliability Analysis

Let $r_{SE}(t)$ be the time-dependent reliability of the basic switching element. Reliability analysis for this SEN, and for all $N \times N$ SENs under definition 1, is straightforward. Since the SEN is an unique-path MIN, the failure of any switch will cause system failure, so from the reliability point of view, the network is composed of $(N/2)(\log_2 N)$ switching elements in series. Hence, the reliability of an $N \times N$ SEN is given by

$$R_{SEN}(t) = [r_{SE}(t)]^{\frac{N}{2}\log_2 N} . \tag{5.4}$$

For the 4×4 SEN, it is clear that the reliability is

$$R_{SEN}(t) = [r_{SE}(t)]^4 \tag{5.5}$$

since there are four identical SEs. The 4×4 SEN+ has six SEs; two in each of three stages. The four SEs which comprise the first and last stages are all necessary for full connectivity. The intermediate stage can tolerate one fault, so this stage has two SEs arranged in parallel. Therefore, computing the reliability of the 4×4 SEN+, arranged in this series-parallel fashion, the closed-form reliability expression is

$$R_{SEN+}(t) = [r_{SE}(t)]^4[1 - (1 - r_{SE}(t))^2] . \tag{5.6}$$

The purpose of the extra stage in the SEN+ is to increase the system's reliability, but by examining equations (5.5) and (5.6), it is evident that the 4×4 SEN+ is strictly *less* reliable than the corresponding SEN. This is because the number of components in the intermediate stages where the two paths between a *S-D* pair are disjoint is small when compared to the number of SEs in the first and last stages combined. (That is, there are only 2 SEs in the intermediate stage, but there are 4 SEs in the first and last stages combined.)

SEN+ networks are *not strictly* more reliable than SEN networks. The SEN+ networks are not more reliable until the aggregated number of components in the intermediate stages is sufficiently larger than the number of components in the first and last stages combined. For $N \geq 8$ the SEN+ *is strictly more* reliable than the SEN.

Modeling the reliability of 8×8 and 16×16 SEN+ networks is not as straightforward. Determining their reliability is more easily illustrated by using discrete-state, continuous-time Markov chains (CTMC) [98].

For the SEN+ networks, as the number of stages increases, the number of possible configurations for which the full connectivity specified in definition 1 is satisfied increases dramatically. To represent the configurations of a SEN+ as a CTMC, the states of the chain can be specified as $[(N/2)(\log_2 N + 1)]$-tuples where each position of the tuple is either a 1 or 0 corresponding to the "up" or "down" state of the respective SE. One would like to take advantage of the symmetry of the SEN+, and use a $(\log_2 N + 1)$-tuple where the switches are grouped by stages into the corresponding tuple positions. But the failure configurations of the network quickly destroy the network's fault-free symmetry.

The major problem with the CTMC approach to modeling the system's time-to-failure behavior is the exponential growth of the state space as the network's size increases. Essentially, the operational status of each SE in each state must be considered. For example, the 8×8 SEN+ has 16 SEs, so 2^{16} possible states must be considered. The state space can be reduced significantly by noting that all the switches in the first and last stages must function for the network to function. Now for the 8×8, at most, only 2^8 possible configurations must be considered. The initial state of a CTMC which models the lifetime behavior of an 8×8 SEN+ is (11111111) indicating that

all eight SEs in the intermediate stages are operational. The system's configurations can be represented by a non-homogeneous CTMC, thus allowing a time-dependent failure rate $\lambda(t)$ for each switching element. The reliability of a SE is thus given by

$$r_{\mathrm{SE}}(t) = e^{-\int_0^t \lambda(\tau)d\tau} . \tag{5.7}$$

Figure 5.1 is the CTMC representation for the 8×8 SEN+. Arcs that are not labeled are assigned the transition rate $\lambda(t)$; this was done to avoid cluttering the figure. Note that this chain has 36 states. Once the CTMC has been constructed, it is possible to reduce the size of the chain by using state lumping [32]. In this example, it was possible to reduce the chain to an equivalent one with only seven states. In Figure 5.2, a seven-state CTMC representation for this SEN+ is shown. For such an acyclic CTMC, the convolution integration method [98] can be used to solve for the state probabilities $P_i(t)$, and hence the system reliability $R_{\mathrm{SEN+}}(t)$ is the sum of the $P_i(t)$ over all the "up" states. Appendix A shows how the method can be applied to the solution of this Markov chain. The reliability of the 8×8 SEN+ is thus determined to be

$$R_{\mathrm{SEN+}}(t) = 2e^{-12\int_0^t \lambda(\tau)d\tau} + 4e^{-14\int_0^t \lambda(\tau)d\tau} - 8e^{-15\int_0^t \lambda(\tau)d\tau} + 3e^{-16\int_0^t \lambda(\tau)d\tau} \tag{5.8}$$

which can be written as

$$R_{\mathrm{SEN+}}(t) = 2[r_{\mathrm{SE}}(t)]^{12} + 4[r_{\mathrm{SE}}(t)]^{14} - 8[r_{\mathrm{SE}}(t)]^{15} + 3[r_{\mathrm{SE}}(t)]^{16}. \tag{5.9}$$

Assuming a constant failure rate $\lambda(t) = \lambda$, Figure 5.3 compares the reliabilities of the 8×8 SEN and SEN+ networks as functions of dimensionless parameter λt. These curves show that the reliability of the 8×8 SEN+ is greater than that of the corresponding SEN. In fact, it can be shown (see

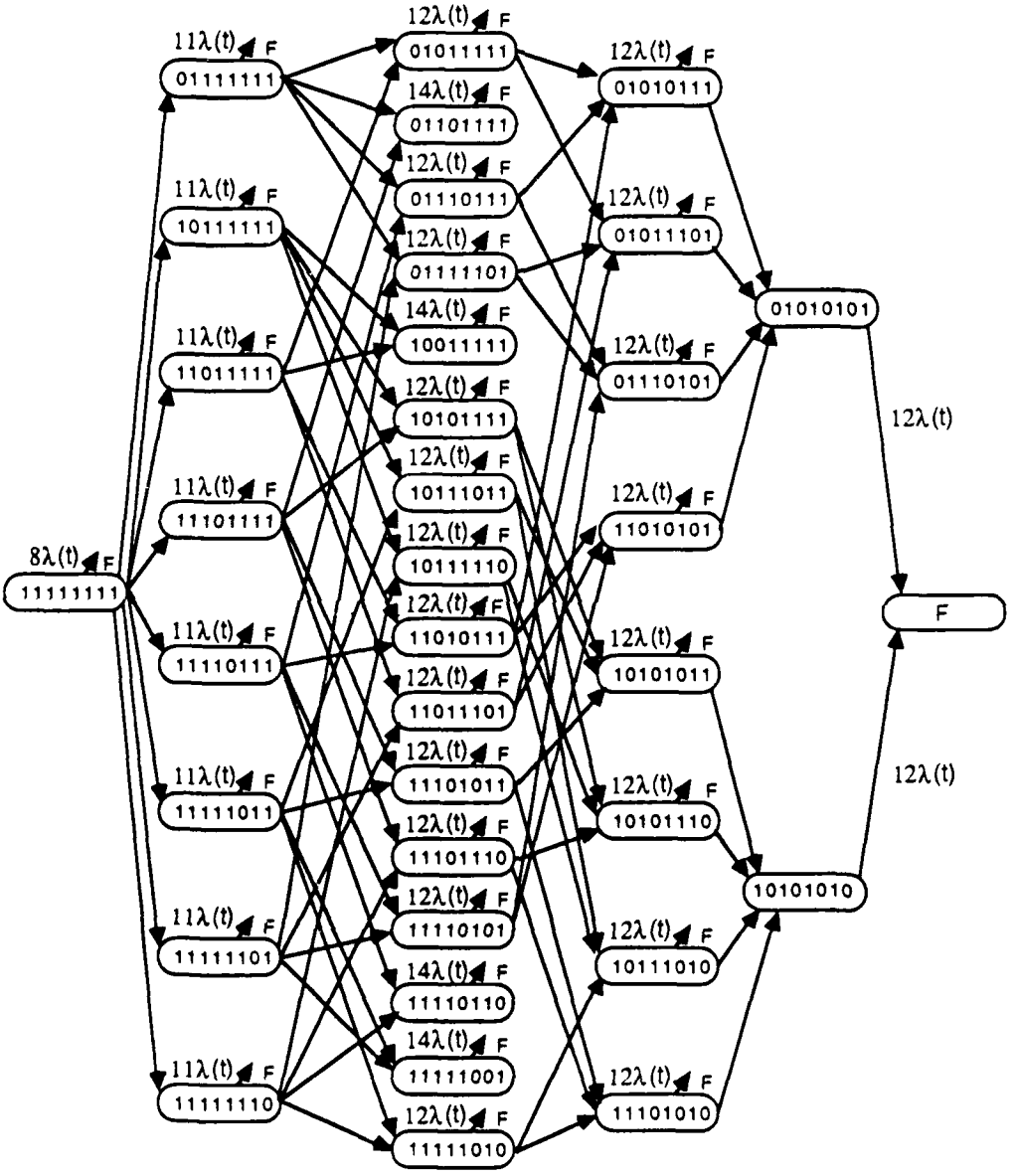


Figure 5.1: Continuous-Time Markov Chain Representing the Time to Failure of the 8×8 SEN+.

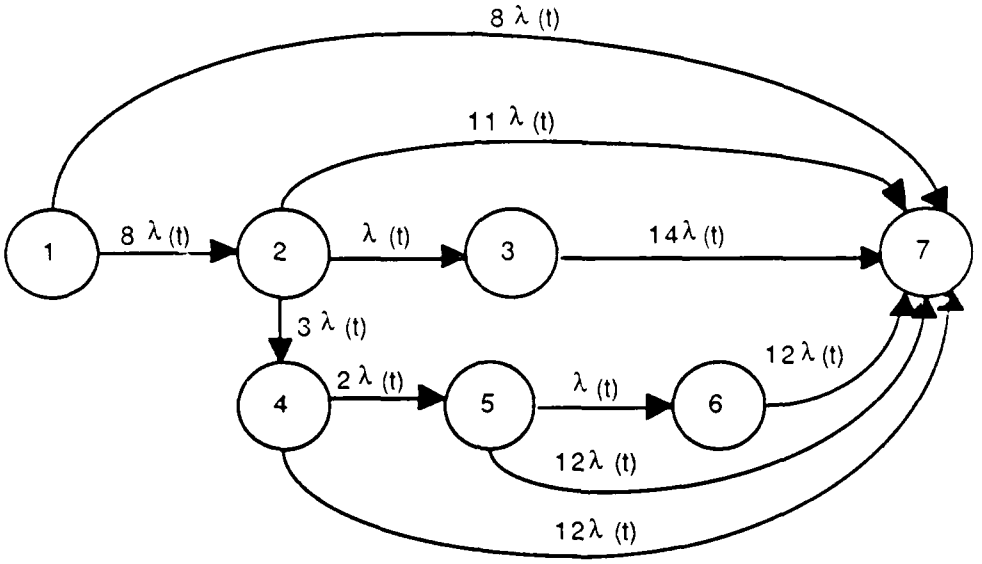


Figure 5.2: Reduced-State-Space CTMC of the Time to Failure of the 8×8 SEN+.

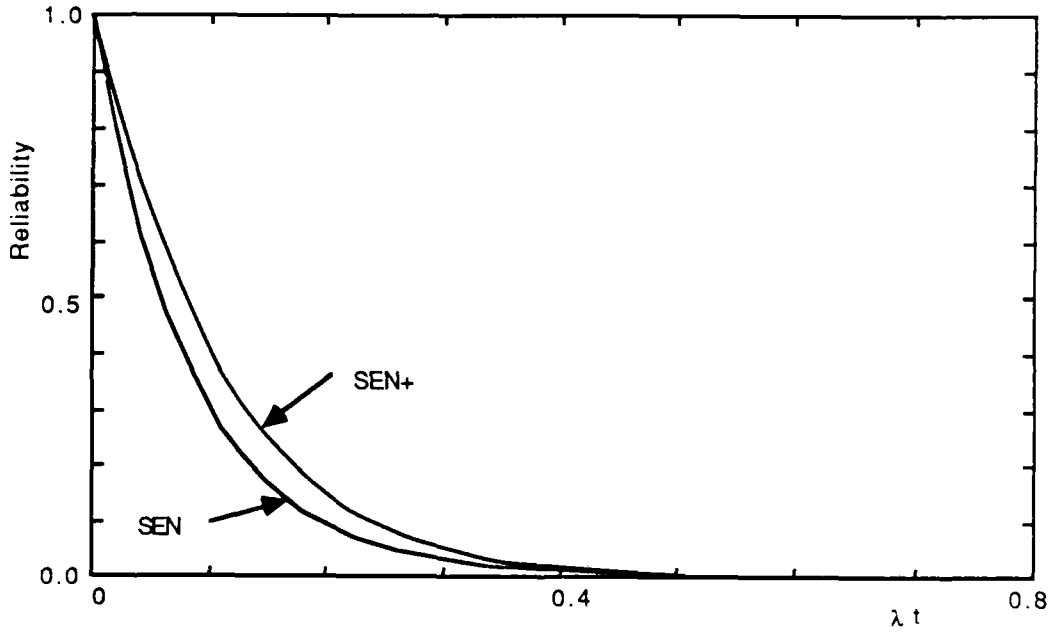


Figure 5.3: Comparison of the Reliabilities of the SEN and SEN+ Networks for the 8×8 Case.

Appendix B) that this result holds for any underlying component-lifetime-distribution. One needs only to solve

$$R_{\text{SEN+}} - R_{\text{SEN}} \geq 0 \ . \tag{5.10}$$

For the 8×8 case, let $r = r_{\text{SE}}(t)$, then using equations (5.4) and (5.9), the inequality

$$r^{12}(1 + 4r^2 - 8r^3 + 3r^4) \geq 0 \tag{5.11}$$

needs to be shown to hold for all $0 \leq r \leq 1$. For the equality condition there are three real roots (0, 1, and 1.929) and two complex roots. Further, over the open interval $(0, 1)$ for r, the strict inequality holds, hence the reliability of the SEN+ is strictly greater than that of the corresponding SEN.

All these reliability expressions can be interpreted either as time functions or as static functions of the reliability of the switching elements since the networks are assumed to possess only static redundancy. Thus for example,

$$R_{\text{SEN+}} = 2r^{12} + 4r^{14} - 8r^{15} + 3r^{16} \tag{5.12}$$

where r is the reliability (as a simple probability) of a switching element. In fact, $R_{\text{SEN+}}$ and R_{SEN} can be plotted as functions of r as in Figure 5.4 to obtain a graphical proof that $R_{\text{SEN+}} \geq R_{\text{SEN}}$ for all $0 \leq r \leq 1$.

While a Markov chain representation of the evolution of the system lifetime for the 8×8 SEN+ network has been presented, analysis of the next larger SEN+ using this approach is too expensive in terms of time and space. Considering only the intermediate stages, the 16×16 SEN+ has 2^{24} possible states. One might consider constructing the Markov chain by depth-first or breadth-first search looking for transitions to operational states starting from the "fully" operational state (no SEs failed). These search procedures will be very expensive because many paths may reach a given state and an exorbitant amount of checking for duplicates is involved. Note that if all "tuples"

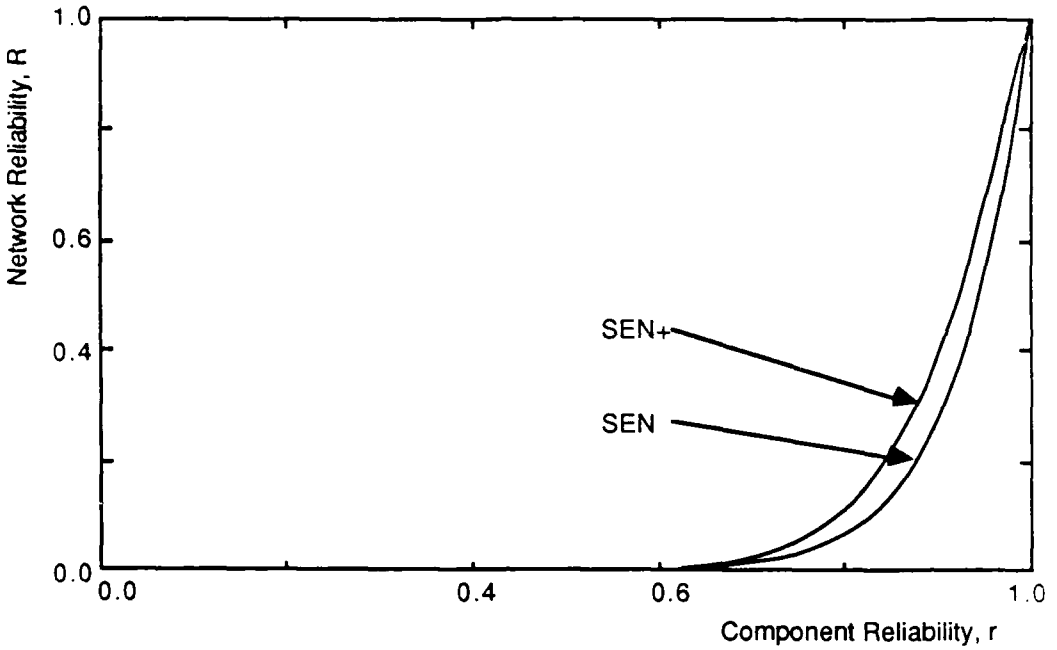


Figure 5.4: Comparison of the Reliabilities of the SEN and SEN+ Networks as a Function of the Reliability of a Switching Element for the 8×8 Case.

or switch configurations in which the network is operational are known, then one can easily find the reliability of the network as the disjoint sum of the tuple probabilities. In other words, there is no need to generate the transitions of the Markov chain. The earlier use of the CTMC was principally for pedagogical purposes as it will be used later in the performability analysis. It provides a clearer illustration of the evolution of the network under discussion. These networks, however, have no dynamic redundancy. That is, they do not have spares to replace failed components, so the analysis of these networks can also be performed using a graph-theoretic approach for multi-terminal graphs.

While the exponential complexity of algorithms used to find the "up" states of a system appears to be unavoidable, one can take advantage of the structure of the SEN+ to reduce the memory requirements and check-

ing for duplicates during the computation. To find the set of "up" tuples, number each of the SEs in the intermediate stages from 1 to M where $M = (N/2)(\log_2 N - 1)$. In the intermediate stages of the network, there are two disjoint paths so the SEs that comprise this portion of the network can be partitioned into two disjoint sets. Hence, there exists pairs (u, v) of SEs (one from each set) that disconnect the network. Each possible pairing is checked to see if it causes network failure, and those pairs that do are placed on a list. Next, start with the binary representation of $2^M - 1$ (all SEs operational) and check the binary representation of each number from $2^M - 1$ to $2^{M/2} - 1$ against the list to see if it is an operational tuple. This is accomplished by checking positions u and v in the binary representation. If they are not both 0s, then record an occurrence of i, the number of 1s in the binary representation, and keep track of the number of occurrences of i. If both positions are 0, discard the tuple. The expression for the reliability of the intermediate stages (IS) is then expressed as

$$R_{\mathrm{IS}}(t) = \sum_{i=M/2}^{M} a_i r_{\mathrm{SE}}(t)^i (1 - r_{\mathrm{SE}}(t))^{M-i}, \tag{5.13}$$

where the coefficient a_i is the number of "up" tuples with i operational SEs.

The reliability expression for the 16×16 SEN+ was determined to be

$$\begin{aligned} R_{\mathrm{SEN+}}(t) = \quad & r_{\mathrm{SE}}(t)^{28}[2 + 2r_{\mathrm{SE}}(t)^4 + 8r_{\mathrm{SE}}(t)^6 - 16r_{\mathrm{SE}}(t)^7 + 8r_{\mathrm{SE}}(t)^8 - \\ & 16r_{\mathrm{SE}}(t)^9 + 20r_{\mathrm{SE}}(t)^{10} - 8r_{\mathrm{SE}}(t)^{11} + r_{\mathrm{SE}}(t)^{12}] \, . \end{aligned} \tag{5.14}$$

A comparison of the reliabilities of the two networks, assuming a constant switch failure rate, is presented in Figure 5.5. Once again, the SEN+ is more reliable than the corresponding SEN.

At this point, the exact reliability expressions for the 8×8 and 16×16 SEN+ networks have been derived, and a comparison of the curves that represent

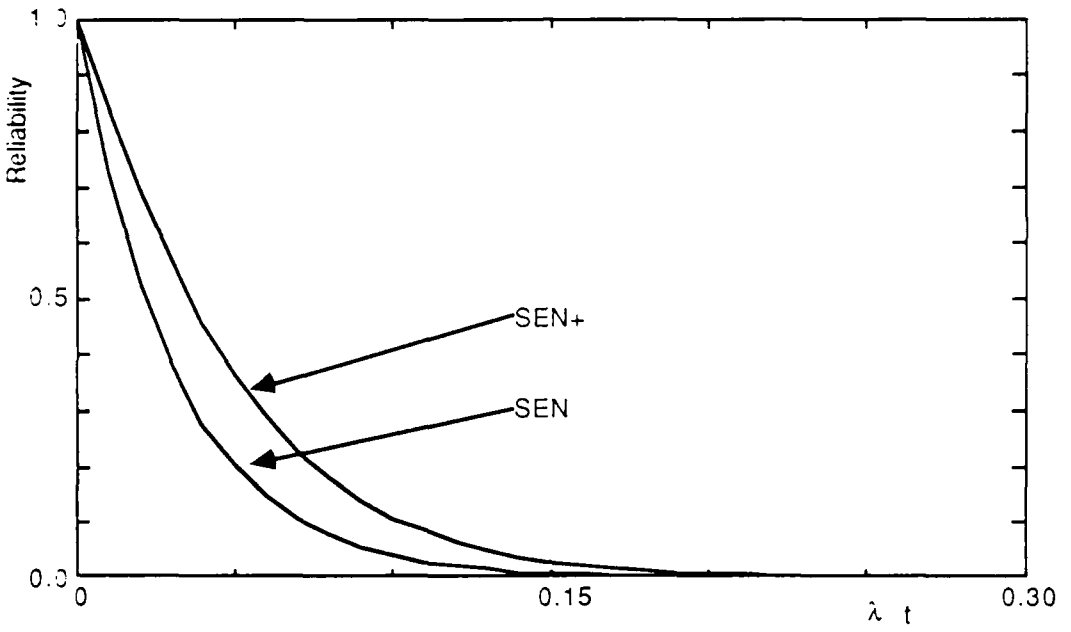


Figure 5.5: Comparison of the Reliabilities of the SEN and SEN+ Networks for the 16 × 16 Case.

their absolute measures of reliability with the corresponding SENs has been presented (Figures 5.3 and 5.5).

Now a comparative reliability measure ($MTIF$) for these networks will be used. Let $r_{SE}(t) = e^{-\lambda t}$, and set $\lambda = 1$, then Υ_{SEN} can be obtained from the closed-form expression

$$\Upsilon_{SEN} = -\frac{\ln R_{desired}}{\hat{M}} \tag{5.15}$$

where $\hat{M} = (N/2)(\log_2 N)$. To obtain Υ_{SEN+}, a nonlinear equation must be numerically solved. Let $R_{desired} = R_{SEN+}$ and $\Upsilon_{SEN+} = t$ in equations (5.6), (5.9), and (5.14). Then, Υ_{SEN+} is computed for specified values of $R_{desired}$ in these equations. The plot of $MTIF = \Upsilon_{SEN+}/\Upsilon_{SEN}$, as a function of required mission reliability for the 4×4, 8×8, and 16×16 networks is presented in Figure 5.6.

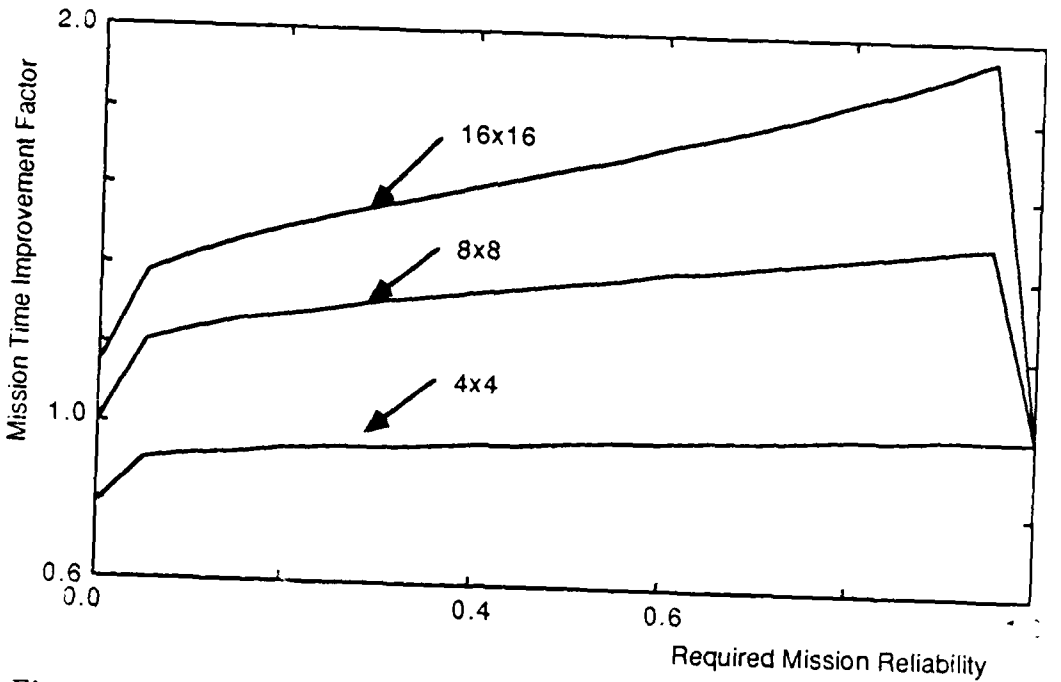


Figure 5.6: Comparison of the Mission Time Improvement Factor of the Networks for the 4×4, 8×8, and 16×16 Cases.

The figure shows that from a reliability perspective, as network size increases, it becomes more advantageous to choose the SEN+ network over the SEN. For example, consider a reliability requirement of 0.95 for a particular mission. In the 8×8 case, the improvement achieved by the SEN+ over the basic SEN is only a factor of 1.25; while for the 16×16 case, the gain is nearly two-fold. Also note that after some relatively high reliability requirement, $MTIF$ decreases rapidly with further increases in the reliability requirement. In the extreme case (component reliability equal to one), then redundancy provides no improvement in system reliability.

5.6.2 Reliability Bounds for Large Networks

As network size increases, explicitly modeling the reliability of the SEN+ networks using Markov chains or tuples becomes rather complex. Since for

each S-D pair there are two disjoint paths within the intermediate stages of the SEN+ network, one has to determine if the failure of the $(k+1)^{st}$ SE in this group of stages causes system failure conditioned on the fact that the first k SE-failures did not cause system failure. Now since each S-D pair has two disjoint paths, each such pair must be examined. So, for a 1024×1024 SEN+, there are 2^{21} paths and each path has $\log_2 1024 + 1 - 2 = 9$ SEs through the intermediate stages. Therefore, approximation techniques for determining the reliability of the larger SEN+ networks are a practical and necessary alternative.

Lower Bounds

To obtain a lower bound, observe that as many as one-half of the switching elements in the intermediate stages of an SEN+ can be failed, and yet the network is still operational. Figure 5.7 illustrates this condition for the 8×8 SEN+. If one models the intermediate stages as a system consisting of a parallel arrangement of two series subsystems each with $(N/4)(\log_2 N - 1)$ switches, then the lower bound of reliability can be obtained using reliability block diagrams. This provides a series system of three subsystems — the first and last are series subsystems and the middle subsystem is a parallel-series subsystem. The reliability expression resulting from the "lower-bound" block diagram as shown in Figure 5.8 is

$$\begin{aligned} R_{lb}(t) &= [r_{SE}(t)]^N \cdot \left[1 - \left[1 - r_{SE}(t)^{\frac{N}{4}[\log_2 N - 1]}\right]^2\right] \\ &= 2[r_{SE}(t)]^{\frac{N}{4}(\log_2 N + 3)} - [r_{SE}(t)]^{\frac{N}{2}(\log_2 N + 1)} . \end{aligned} \tag{5.16}$$

A similar technique is used by Padmanabhan in [64] to obtain a lower bound for the reliability of redundant path networks using an independent link-fault model. (The switch-fault model is used for the analysis in this paper.)

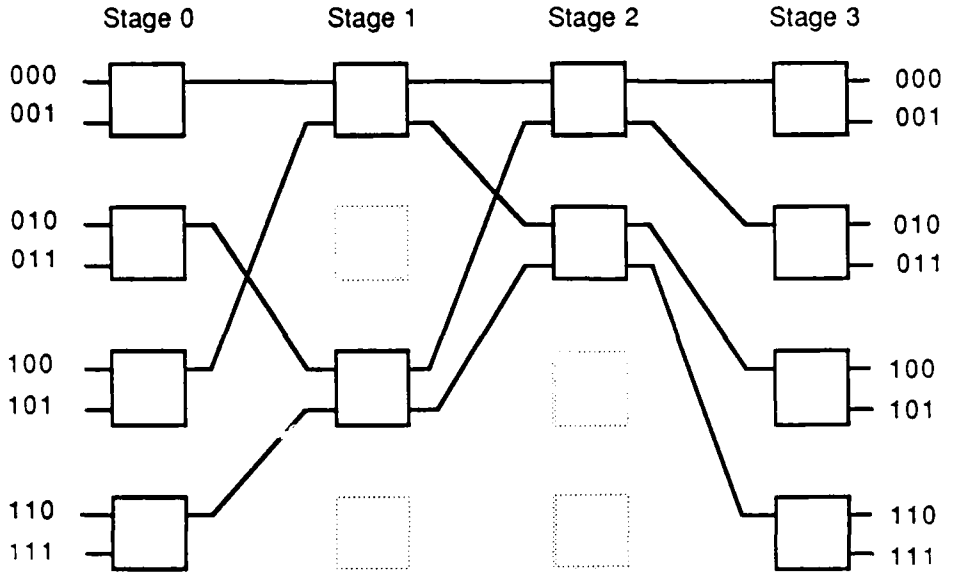


Figure 5.7: Illustration of the 8×8 SEN+ with One-half of the Switching Elements in the Intermediate Stages Failed.

Upper Bounds

To obtain an upper bound on the reliability of the SEN+, observe that each SE in a particular stage of the SEN+ shown in Figure 3.4 has a *conjugate* [51]. That is, for stages 1, ..., n there exists a pair of SEs in stage $i-1$ that are connected to a pair of SEs in stage i. For example, $SE_{0,0}$ and $SE_{2,0}$ are connected to $SE_{0,1}$ and $SE_{1,1}$. If a conjugate pair of SEs fail, then the network has failed. Assuming the network is operational as long as no conjugate pair in the intermediate stages fail and no SE in the first or last stages fail, an upper bound on the reliability of the SEN+ is obtained. This will overestimate system reliability since there are many combinations of failed SEs other than conjugates pairs that will cause the network to be failed. Figure 5.9 shows a representation of this configuration. (The upper bound can be improved further by taking advantage of the linkage interdependencies between stages,

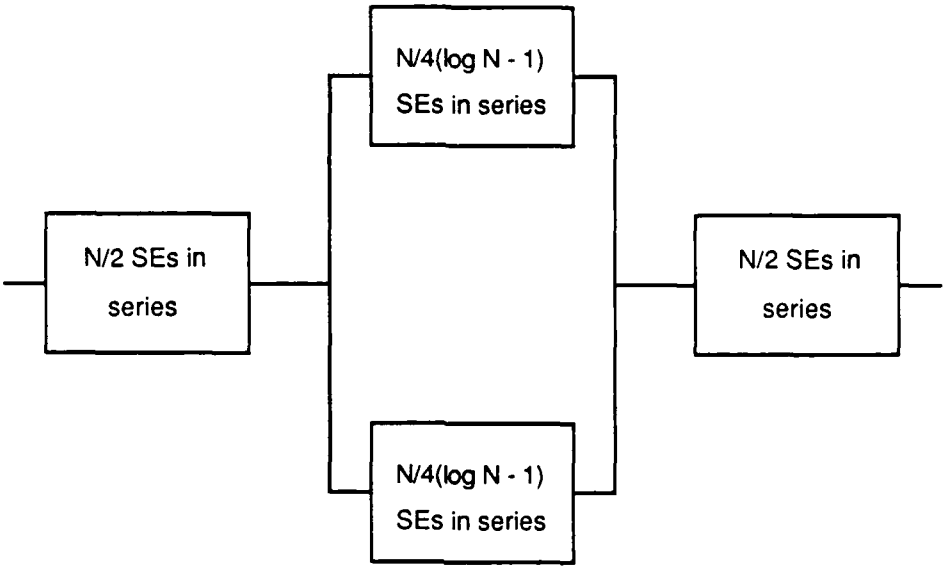


Figure 5.8: Reliability Block Diagram Representation of the Tight Lower-Bound Model for the SEN+ Networks.

and in larger networks, the improvement obtained may be significant.) The reliability expression using this upper bound is given by

$$R_{ub}(t) = [r_{\mathrm{SE}}(t)]^N \cdot [1 - (1 - r_{\mathrm{SE}}(t))^2]^{\frac{N}{4}(\log_2 N - 1)} . \qquad (5.17)$$

Figure 5.10 compares the upper (optimistic) and lower (conservative) bounds for an 8×8 SEN+ network with the exact reliability expression (5.9).

Finding an upper bound for system reliability is usually not the center of attention in real world applications. One usually wants a conservative indication of how long the system will be operational, and upper bounds present an optimistic view of the world. The lower bound provides the probability that the system will be operational at some specified time. The expectation is that the real system is at least this good. If the gross lower bound provides sufficient assurance that the system will be operational over the time interval

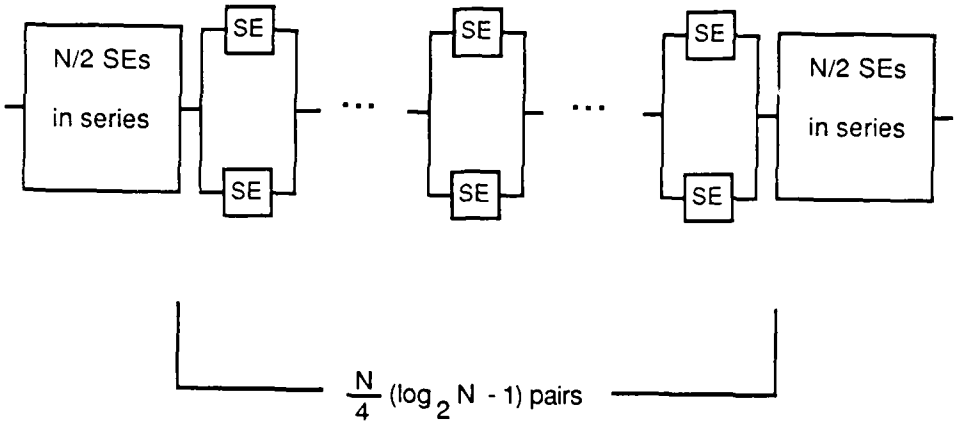


Figure 5.9: Reliability Block Diagram Representation of the Upper-Bound Model for the SEN+ Networks.

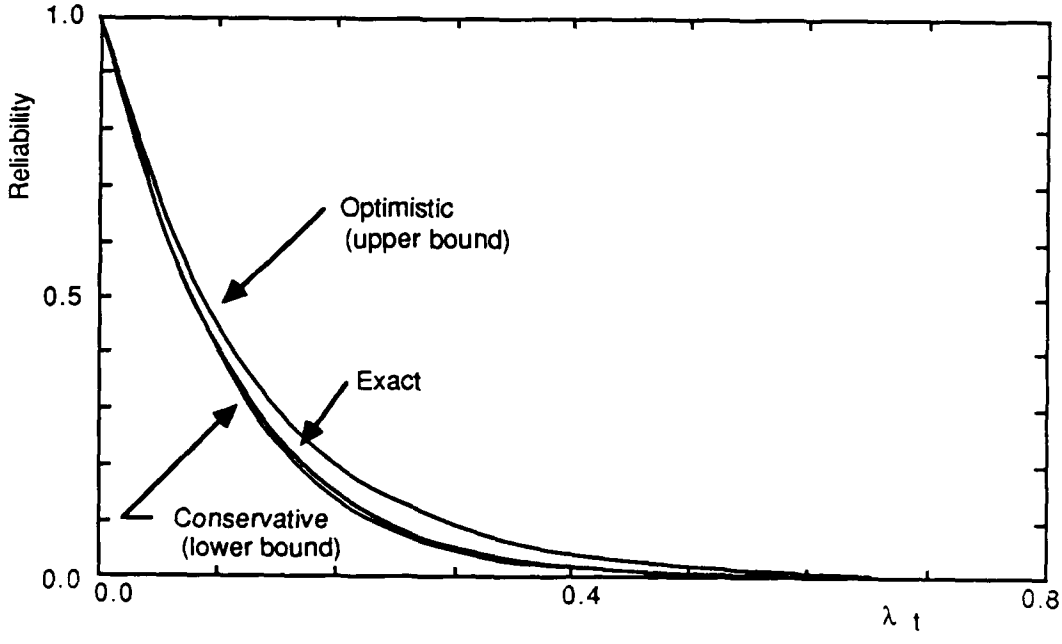


Figure 5.10: Comparison of the Upper and Lower Bounds with the Exact Reliability of the 8×8 SEN+.

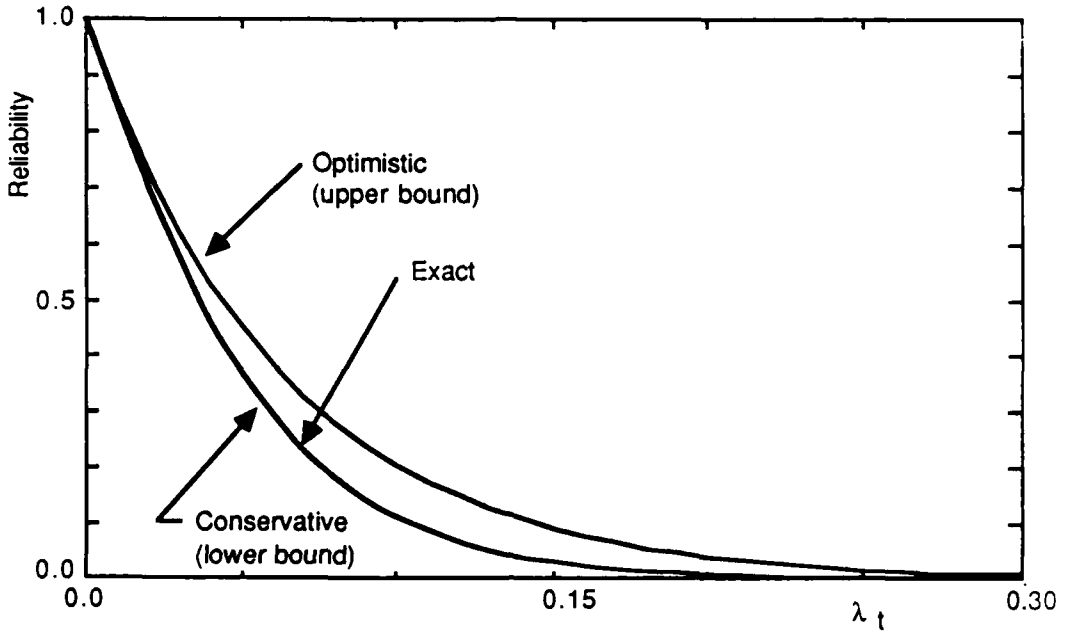


Figure 5.11: Comparison of the Upper and Lower Bounds with the Exact Reliability of the 16×16 SEN+.

of interest, then no further effort at obtaining a better approximation or the exact reliability expression is necessary.

The above analysis is repeated for the 16×16 networks. In Figure 5.11, the upper and lower bounds are compared with the exact solution, equation (5.14), for the $R_{\text{SEN+}}(t)$ for $N = 16$. The "lower bound" model closely approximates the exact solution for the SEN+ network. From the above comparisons, it is clear that the bound of equation (5.16) is a reasonable approximation to the actual reliability of SEN+ networks.

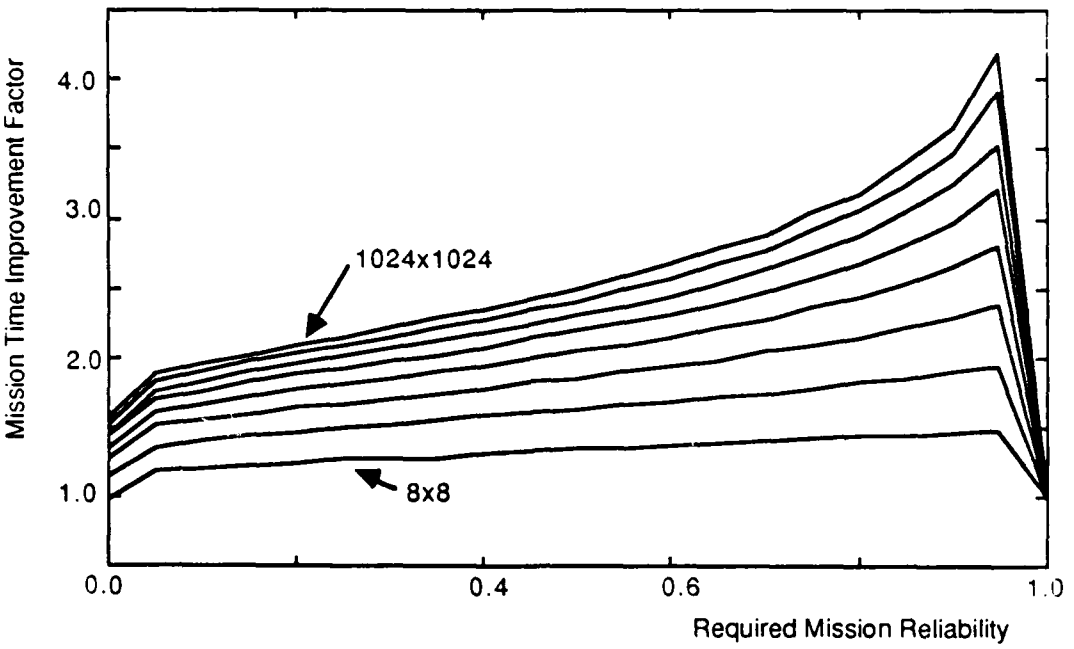


Figure 5.12: Comparison of the Mission Time Improvement Factor of the Networks from Size 8 × 8 to 1024 × 1024 Using the Lower-Bound Model.

5.6.3 Network Comparisons

Mission Time Improvement Factor

Using the lower bound model, the $MTIF$ for 8×8 through 1024×1024 networks were computed. As shown in Figure 5.12, a dramatic reliability improvement is obtained by simply adding an extra stage to the SEN networks.

Mean Time to Failure

In this section, the mean time to failure of the networks is discussed, where

$$MTTF = \int_0^\infty R(t)dt \ . \tag{5.18}$$

Noting that $R(t)$ has the form $\sum_i [a_i r_{\mathrm{SE}}^i(t)]$, one can perform this integration symbolically and get a closed-form result. In the case that $r_{\mathrm{SE}}(t)$ is assumed

to be the Weibull reliability function, then

$$r_{SE}(t) = e^{-\lambda_W t^\alpha} \ . \tag{5.19}$$

In this case, using [98]

$$\int_0^\infty e^{-\lambda_W t^\alpha} dt = (\frac{1}{\lambda_W})^{\frac{1}{\alpha}} \Gamma(1 + \frac{1}{\alpha}) \ , \tag{5.20}$$

one obtains

$$MTTF_W = \sum_i a_i [(\frac{1}{i\lambda_W})^{\frac{1}{\alpha}} \Gamma(1 + \frac{1}{\alpha})] \ , \tag{5.21}$$

where $\Gamma()$ denotes the *gamma function.* Thus in the case of the 8×8 SEN+, from equation 5.9 one obtains

$$MTTF_W = (\frac{2}{12^{\frac{1}{\alpha}}} + \frac{4}{14^{\frac{1}{\alpha}}} - \frac{8}{15^{\frac{1}{\alpha}}} + \frac{3}{16^{\frac{1}{\alpha}}})(\frac{1}{\lambda_W})^{\frac{1}{\alpha}} \Gamma(1 + \frac{1}{\alpha}) \ . \tag{5.22}$$

In the special case of the exponential distribution, a further simplification provides

$$MTTF_E = \sum_i \frac{a_i}{i\lambda_E} \ . \tag{5.23}$$

So in the above case,

$$MTTF_E = (\frac{2}{12} + \frac{4}{14} - \frac{8}{15} + \frac{3}{16}) \cdot \frac{1}{\lambda_E} = \frac{179}{1680\lambda_E} \ . \tag{5.24}$$

Figure 5.13 plots the $MTTF$ of the SEN and SEN+ networks as a function of the network size N ($\log_2$ scale is used on the x-axis). Both the lower-bound model and the exact solution for the (size 2, 4, 8 and 16) SEN+ networks are shown. The "•" marks overlying the $MTTF$ for the SEN+ lower-bound curve show the exact solutions. Observe that the $MTTF$ of the SEN+ networks for sizes 2 and 4 are less than their corresponding SEN, and as previously stated, for networks of size 8 and larger, the $MTTF$ for the SEN+ networks is dominant. In fact, for the lower-bound model, direct integration of equation (5.16) yields the closed-form answer for the $MTTF$:

$$(MTTF_{lb})_E(N) = \frac{2}{N\lambda_E} \left[\frac{3\log_2 N + 1}{(\log_2 N + 1)(\log_2 N + 3)} \right] . \tag{5.25}$$

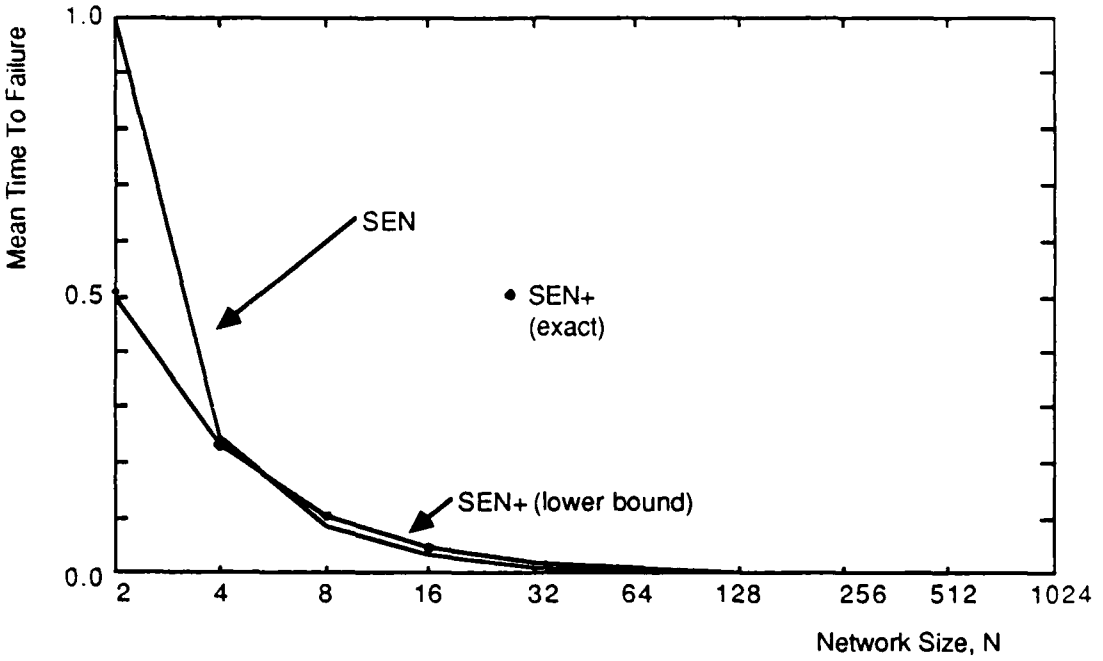


Figure 5.13: Comparison of the Mean Time to Failure of SEN and SEN+ Networks from Size 2×2 to 1024×1024.

These curves are helpful, but a single curve that compares the $MTTF$ for a given network size is more revealing. For this purpose, the normalized mean-time-to-failure is used for specified network sizes.

The normalized mean-time-to-failure is an appropriate comparative measure of reliability for networks because it is the ratio of the $MTTF$ of a network with redundancy divided by the $MTTF$ of the basic network. In Table 5.1, data is provided for both the lower and upper bounds for the SEN+ network. Noting that the $MTTF$ for the SEN is $2/(N\lambda \log_2 N)$, and using equation (5.25), the asymptotic value of the $NMTTF$ for the lower-bound model for the SEN+ is determined to be 3. By examining the $NMTTF$ for the SEN+, one observes that the exact values are close to the lower-bound model. It is expected that the exact values will remain close to the lower-bound model as the network size increases since the series arrangement of SEs

Size	$MTTF * \lambda$				$NMTTF$		
	SEN	SEN+			SEN+		
N	EXACT	LB	EXACT	UB	LB	EXACT	UB
2	1	0.50000	$\frac{1}{2}$	0.50000	0.5000	0.5000	0.5000
4	$\frac{1}{4}$	0.23333	$\frac{7}{30}$	0.23333	0.9333	0.9333	0.9333
8	$\frac{1}{12}$	0.10417	$\frac{179}{1680}$	0.11525	1.2500	1.2785	1.3830
16	$\frac{1}{32}$	0.04643	$\frac{37630211}{783029520}$	0.05830	1.4857	1.5378	1.8656
32	$\frac{1}{80}$	0.02083		0.02969	1.6667		2.3752
64	$\frac{1}{192}$	0.00942		0.01509	1.8095		2.8973
128	$\frac{1}{448}$	0.00430		0.00764	1.9250		3.4227
256	$\frac{1}{1024}$	0.00197		0.00386	2.0202		3.9480
512	$\frac{1}{2304}$	0.00091		0.00194	2.1000		4.4698
1024	$\frac{1}{5120}$	0.00042		0.00097	2.1678		4.9664

Table 5.1: $MTTF$ and $NMTTF$ Ratios for the $N \times N$ SEN and SEN+ Networks.

in the first and last stages of the network will tend to be a limiting factor of reliability. Note also that as the network size increases, the upper bound diverges from the lower bound. It is evident that for larger networks, it is desirable to find a tighter upper-bound model. However, emphasis should be placed on the lower bound since assurance of some minimum level of reliability is desired.

In terms of cost, the ratio of the number of switching elements used in a network with redundancy divided by the number of SEs in the basic network is also an useful measure. In Table 5.2, a comparison of the complexities of these networks is presented.

Another method for improving the reliability of a MIN is through the use of multiple copies. This method of adding fault tolerance uses K replications of the basic network (K-SEN) to achieve $(K-1)$-fault-tolerance. The same assumption stated by Ciminiera and Serra [20] and Padmanabhan [64] is

Size	Network Complexity		Ratio
	SEN	SEN+	
N	$\frac{N}{2}(\log_2 N)$	$\frac{N}{2}(\log_2 N + 1)$	$\frac{SEN+}{SEN}$
2	1	2	2.0000
4	4	6	1.5000
8	12	16	1.3333
16	32	40	1.2500
32	80	96	1.2000
64	192	224	1.1667
128	448	512	1.1429
256	1024	1152	1.1250
512	2304	2560	1.1111
1024	5120	5632	1.1000

Table 5.2: Network Complexity for the $N \times N$ SEN and SEN+ Networks.

used in this analysis. That is, each basic network is considered as a single component of the replicated network, so a component is failed whenever one of its SEs has failed. Then the reliability of a K-SEN is

$$R_{K-\text{SEN}}(t) = 1 - [1 - R_{\text{SEN}}(t)]^K. \tag{5.26}$$

Note, however, that this method of adding fault tolerance is not very effective since the improvement factor is proportional to $\log K$ [20]. For the purpose of comparison with the SEN+, the case where $K = 2$ is considered. The $MTTF$ of the 2-SEN is

$$MTTF_{2-\text{SEN}} = 2MTTF_{\text{SEN}} - \frac{MTTF_{\text{SEN}}}{2}.$$

Figure 5.14 plots the $NMTTF$ of these two redundant networks (the SEN+ and the 2-SEN) as a function of N (using $\log_2$ scale on the x-axis). For the SEN+, the $NMTTF$ is an increasing function of network size, whereas for the 2-SEN, the $NMTTF$ is independent of network size. It provides a

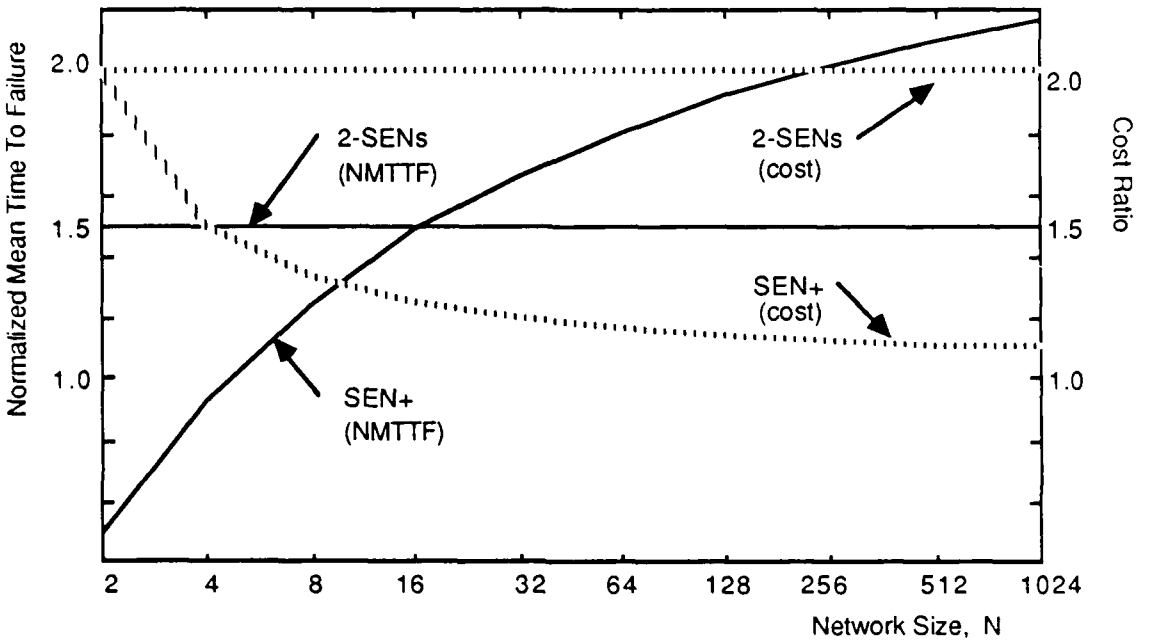


Figure 5.14: Comparison of the Normalized Mean-Time-To-Failure and the Ratio of the Number of Switching Elements for the SEN+ and 2-SEN Networks from Size 2×2 to 1024×1024.

$NMTTF = 1.5$. For networks of size 16 and larger, the reliability improvement achieved by using an extra stage is superior to that obtained by using a pair of SENs.

It is interesting to compare the cost of these networks, too. The ratios of the network complexities for the SEN+ and the 2-SEN divided by the basic SEN are also plotted in Figure 5.14. By using the 2-SEN, the number of SEs is twice that of the basic SEN. Observe that, in the figure, the SEN+ is superior to this network for size 32 and larger.

For the SEN+, as network size increases, the ratio of the network complexities levels off very quickly while the corresponding $NMTTF$ continues to increase at a significantly higher rate. This points out that the cost of adding an extra stage to larger networks is small compared to the gain in reliability which is possible. Hence, for large networks, the SEN+ is less expensive than using a pair of SENs in terms of additional hardware, and it is more reliable as well.

5.6.4 Distributional Sensitivity

A common assumption in the transient analysis of multistage interconnection networks is that individual components have exponentially distributed lifetimes. This means that each component has a constant failure rate. In other words, the conditional probability that the component will fail in the interval Δt given that it has survived until time t is the same as the conditional probability that it will fail in the same interval Δt given that it has survived until time $t + \tau$. Often this assumption is challenged. It seems more appealing to believe that the component is more likely to fail as time increases. A Weibull distribution with shape parameter $\alpha > 1$ models such an increasing-failure-rate (IFR) behavior.

What is the impact on the system's reliability of using an IFR distribution for component lifetime? Consider the 8×8 SEN. Recall that the failure of any component will cause system failure, so the 8×8 SEN can be modeled as a series system with 12 components. Now consider two distributions for an individual component's lifetime. An exponential distribution with CDF $F_E(t) = 1 - e^{-\lambda_E t}$ and a Weibull IFR distribution with CDF $F_W(t) = 1 - e^{-\lambda_W t^\alpha}$. In order to assess the sensitivity of the reliability comparison of SEN and SEN+ networks, one needs to "equalize" the two distributions in some manner. First do this "equalization" by letting the $MTTF$ of individual *components* be the same for the two distributional assumptions. Specifically,

$$\frac{1}{\lambda_E} = (\frac{1}{\lambda_W})^{\frac{1}{\alpha}} \Gamma(1 + \frac{1}{\alpha}) \ . \tag{5.27}$$

Solving for the scale parameter of the Weibull distribution,

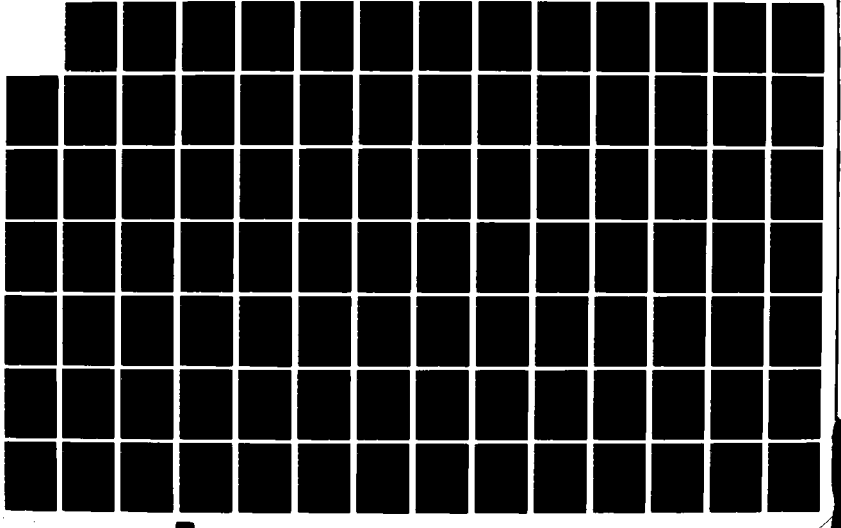
$$\lambda_W = [\lambda_E \Gamma(1 + \frac{1}{\alpha})]^\alpha \ . \tag{5.28}$$

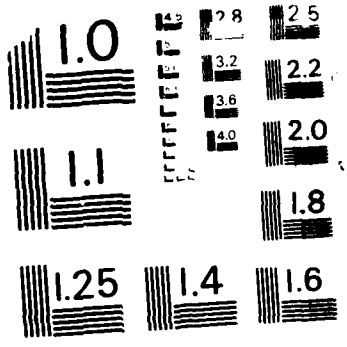
Figure 5.15 shows the system reliability curves for the 8×8 SEN and SEN+ networks assuming $\lambda_E = 0.1$, $\alpha = 1.5$, and solving for the scale parameter $\lambda_W = 0.02712$ so that the $MTTF$ of the individual components is equal. As expected, the SEN+ is more reliable than the SEN. In the figure, one can see that the constant-failure-rate assumption for individual component lifetimes underestimates the system's reliability if the underlying component distributions have an IFR behavior. This means that the standard assumption of exponentially distributed component lifetime-distributions provides a conservative estimate of the system's reliability. The same behavior has been observed for larger networks as well.

Another way to "equalize" the two distributions is to equate the system $MTTF$s under the two distributional assumptions for the individual components. For a series system with n components, the system $MTTF$ is

AD-A190 436 COMPARATIVE ANALYSIS OF MULTISTAGE INTERCONNECTION NETWORKS(U) DUKE UNIV DURHAM NC DEPT OF COMPUTER SCIENCE J T BLAKE DEC 87 2/3

UNCLASSIFIED F/G 25/5 NL





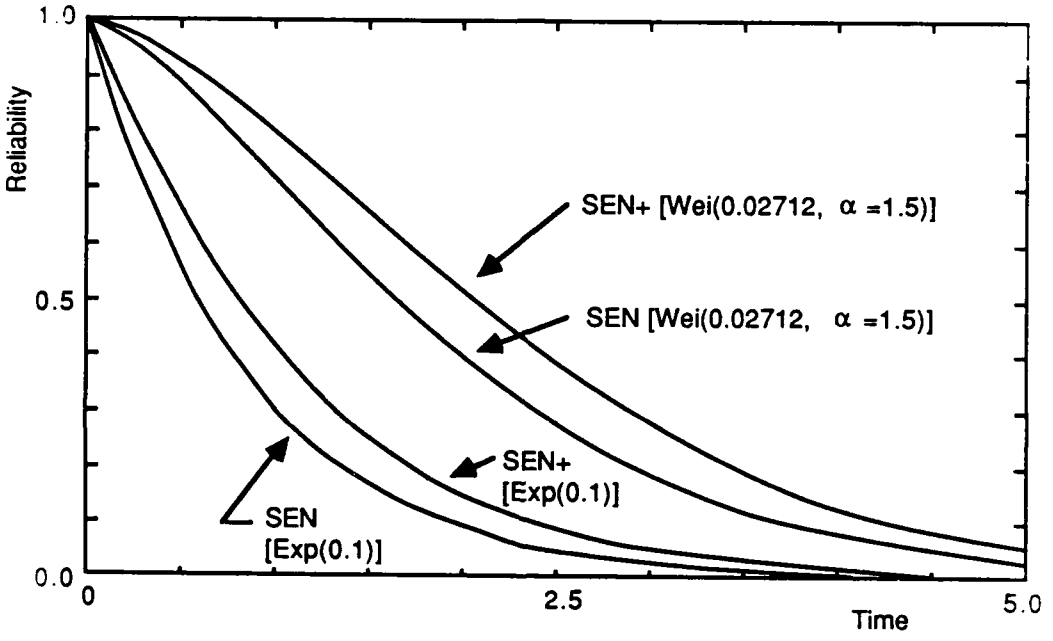


Figure 5.15: Comparison of the Reliabilities of the 8×8 SEN and SEN+ Networks When the Components Have Either an Exponential or Weibull Lifetime Distribution and the *Component* Means are Equalized.

equated as

$$\frac{1}{n\lambda_E} = (\frac{1}{n\lambda_W})^{\frac{1}{\alpha}}\Gamma(1+\frac{1}{\alpha}) \ . \tag{5.29}$$

Solving for the scale parameter λ_W, one gets

$$\lambda_W = n^{(\alpha-1)}[\lambda_E\Gamma(1+\frac{1}{\alpha})]^{\alpha} \ . \tag{5.30}$$

For the 8×8 SEN with $\lambda_E = 0.1$, $\alpha = 1.5$, and $n = 12$; $\lambda_W = 0.093959$. Using equations (5.22) and (5.24) one can determine λ_W for the corresponding SEN+. The expression is

$$\lambda_W = \left[\frac{1}{MTTF_E}\Gamma(1+\frac{1}{\alpha})[\frac{2}{12^{\frac{1}{\alpha}}} + \frac{4}{14^{\frac{1}{\alpha}}} - \frac{8}{15^{\frac{1}{\alpha}}} + \frac{3}{16^{\frac{1}{\alpha}}}]\right]^{\alpha} \ . \tag{5.31}$$

With $\lambda_E = 0.1$, and $\alpha = 1.5$, the scale parameter for the Weibull distribution is $\lambda_W = 0.0845373$. Figure 5.16 shows the system reliability curves under both distributional assumptions. Examining the system's reliability curves after equating the *system* $MTTF$s shows crossover points. The IFR assumption provides a higher system reliability for short missions as expected, and the constant-failure-rate assumption yields superior reliability for longer missions.

One might think when the *system* $MTTF$s are equal under the two distributional assumptions that one should expect to see a crossover point as in Figure 5.15 when the *component* means were equal. This is not the case because the exponential and Weibull distributions do not allow the $MTTF$s to scale in the same fashion. For example, for a series system of n components each having exponentially distributed lifetimes, the system $MTTF$ is simply $1/n$ times the component $MTTF$. But, for the Weibull case, the system $MTTF$ is $(1/n)^{1/\alpha}$ times the component $MTTF$.

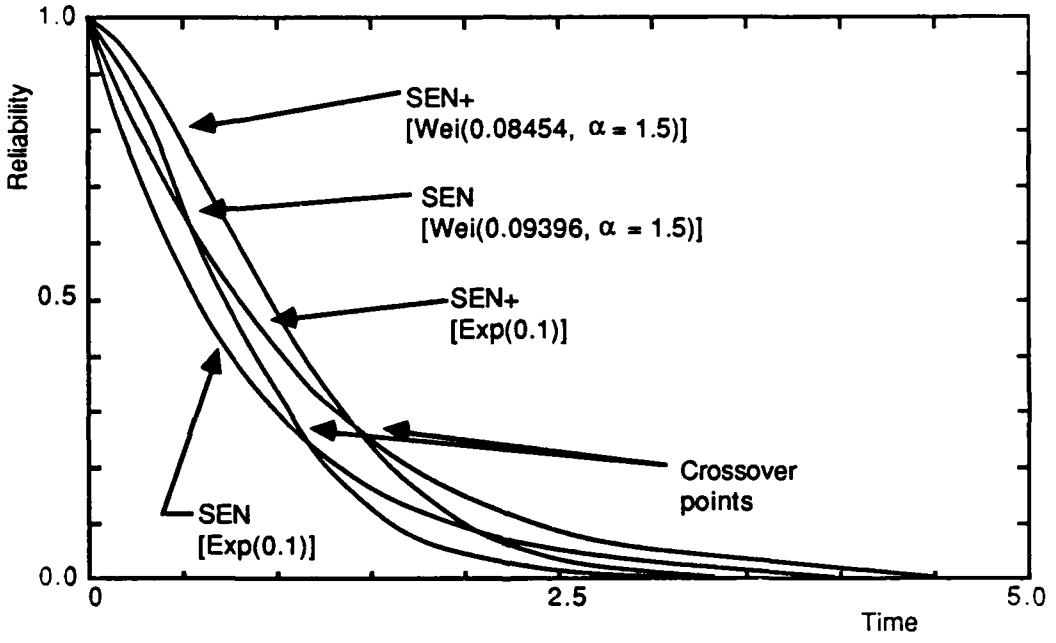


Figure 5.16: Comparison of the Reliabilities of the 8×8 SEN and SEN+ Networks When the Components Have Either an Exponential or Weibull Lifetime Distribution and the *System* Means are Equalized.

5.7 ASEN Network

5.7.1 Exact Reliability Analysis

An exact reliability analysis of the 4×4 and 8×8 ASEN is performed by determining the cut sets of each network and then computing the number of operational configurations. Since the ASEN is a multiple-path MIN, the routing algorithm as well as the topology must be considered in deriving the reliability expressions for the network. The adaptive routing algorithm as described in [53] considers a 2×2 SE in the last stage and its associated demultiplexers as a series system, so these three elements can be considered as a single component, and based on gate count, a failure rate of $\lambda_{2m} = 1.5\lambda_2$ can be assigned to this grouping of elements. Also, let λ_3 be the failure rate of the 3×3 SE and λ_m be the multiplexer/demultiplexer failure rate. Then based on gate count, $\lambda_3 = 3\lambda_2$ and $\lambda_m = \lambda_2/4$. The time-dependent reliability expression for the 4×4 ASEN is

$$R(t) = e^{-4\lambda_m t}\left[2e^{(\lambda_{2m}+2\lambda_m)t} + (2e^{2\lambda_m t} - 4e^{\lambda_m t} + 1)e^{2\lambda_{2m}t}\right]. \tag{5.32}$$

For the 8×8 ASEN, the reliability expression is

$$\begin{aligned} R(t) = \Big[&(4e^{(4\lambda_{2m}+4\lambda_m)t} - 16e^{(3\lambda_{2m}+4\lambda_m)t} + 16e^{(2\lambda_{2m}+4\lambda_m)t})e^{2\lambda_3 t} \\ &+ \Big((8e^{4\lambda_m t} - 16e^{3\lambda_m t} + 4e^{2\lambda_m t})e^{4\lambda_{2m}t} \\ &+(-32e^{4\lambda_m t} + 64e^{3\lambda_m t} - 16e^{2\lambda_m t})e^{3\lambda_{2m}t} \\ &+ (32e^{4\lambda_m t} - 64e^{3\lambda_m t} + 16e^{2\lambda_m t})e^{2\lambda_{2m}t}\Big)\, e^{\lambda_3 t} \\ &+(4e^{4\lambda_m t} - 16e^{3\lambda_m t} + 20^{2\lambda_m t} - 8e^{\lambda_m t} + 1)e^{4\lambda_{2m}t} \\ &+(-16e^{4\lambda_m t} + 64e^{3\lambda_m t} - 80e^{2\lambda_m t} + 32e^{\lambda_m t} - 4)e^{3\lambda_{2m}t} \\ &+ (16e^{4\lambda_m t} - 64e^{3\lambda_m t} + 80e^{2\lambda_m t} - 32e^{\lambda_m t} + 4)e^{2\lambda_{2m}t}\Big] \\ &e^{-(4\lambda_3+8\lambda_m)t}. \end{aligned} \tag{5.33}$$

5.7.2 Reliability Bounds for Large Networks

Deriving the exact reliability expressions for SEN+ and ASEN networks of size 16 and larger is computationally difficult. For example, the CTMC used to represent the various degraded configurations of the 16×16 ASEN could have $2^{40} - 2^{38} = 15 \cdot 2^{38}$ possible states, and the exponential growth of the state space for larger networks makes the construction and solution of the CTMC intractable. For each S-D pair there are two or more disjoint paths within the intermediate stages of the ASEN network. One has to determine if the failure of the $(k+1)^{st}$ SE in this group of stages causes system failure conditioned on the fact that the first k SE-failures did not cause system failure. Each S-D pair has disjoint paths, and each path must be examined. Therefore, approximation techniques are considered for determining the reliability of the larger networks.

Lower Bounds

At the input side of the ASEN, the multiplexers are not considered an integral part of a given 3×3 SE. That is, a multiplexer can be failed, and as long as at least one of its two associated SEs (e.g., SEs 0 and 1 in Figure 3.5) is operational, the network may be operational. But, if two multiplexers grouped with each SE on the input side are considered as a series system, then a conservative estimate of the reliability of these three components is obtained. Their failure rate will be $\lambda_{3m} = 3.5\lambda_2$. Finally, these aggregated components and the SEs in the intermediate stages can be arranged in pairs of conjugate loops. To obtain the pessimistic (lower) bound on the reliability of the ASEN, it is assumed that the network is failed whenever more than one loop has a faulty element or more than one SE in a conjugate pair in the last stage fails. After this simplification of the ASEN, the lower-bound

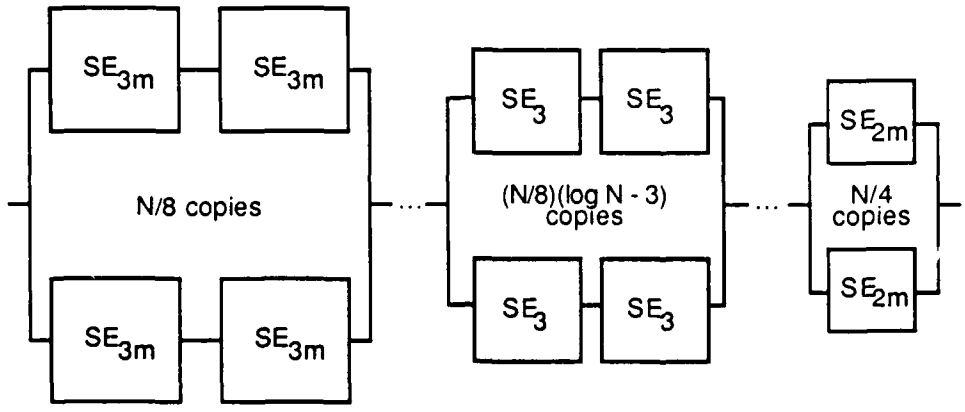


Figure 5.17: Lower-Bound Reliability Block Diagram for the ASEN.

construction from [53] can be modified to reflect the reliability block diagram which is shown in Figure 5.17. For $N \geq 8$, the reliability expression for the lower bound of the ASEN is

$$R_{\mathrm{ASEN}_{lb}}(t) = (1-(1-e^{-2\lambda_{3m}})^2)^{\frac{N}{8}}(1-(1-e^{-2\lambda_3})^2)^{\frac{N}{8}(\log_2 N-3)} (1-(1-e^{-\lambda_{2m}})^2)^{\frac{N}{4}}. \quad (5.34)$$

The ASEN can tolerate any single loop failure or the failure of any single switch in the last stage.

Upper Bounds

To obtain an upper bound for the ASEN, observe that each source is connected to two multiplexers and each SE has a conjugate. If it is assumed that the ASEN is operational as long as one of the two multiplexers attached to a source is operational and as long as a conjugate pair is not faulty, as many as

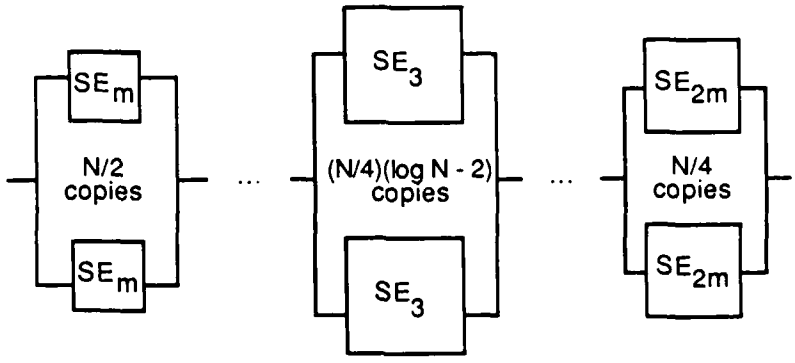


Figure 5.18: Upper-Bound Reliability Block Diagram for the ASEN.

one-half of the components can fail and the ASEN may still be operational. This permits the use of a simple reliability block diagram for the optimistic (upper) bound as shown in Figure 5.18. The expression for the upper bound of the ASEN reliability is

$$R_{\text{ASEN}_{ub}}(t) = (1-(1-e^{-\lambda_m})^2)^{\frac{N}{2}}(1-(1-e^{-\lambda_3})^2)^{\frac{N}{4}(\log_2 N-2)} (1-(1-e^{-\lambda_{2m}})^2)^{\frac{N}{4}}. \quad (5.35)$$

In Figure 5.19, the exact reliability, upper, and lower bounds derived for the 8×8 ASEN are plotted. Also shown in Figure 5.19 is the upper bound for the SEN+. The ASEN lower bound is strictly greater than the upper bound of the SEN+ for $t > 0$. So the worst case reliability of the ASEN is still better than the best case reliability of the SEN+. The ASEN is clearly superior to the SEN+ even for small networks in spite of the fact that it has increased complexity.

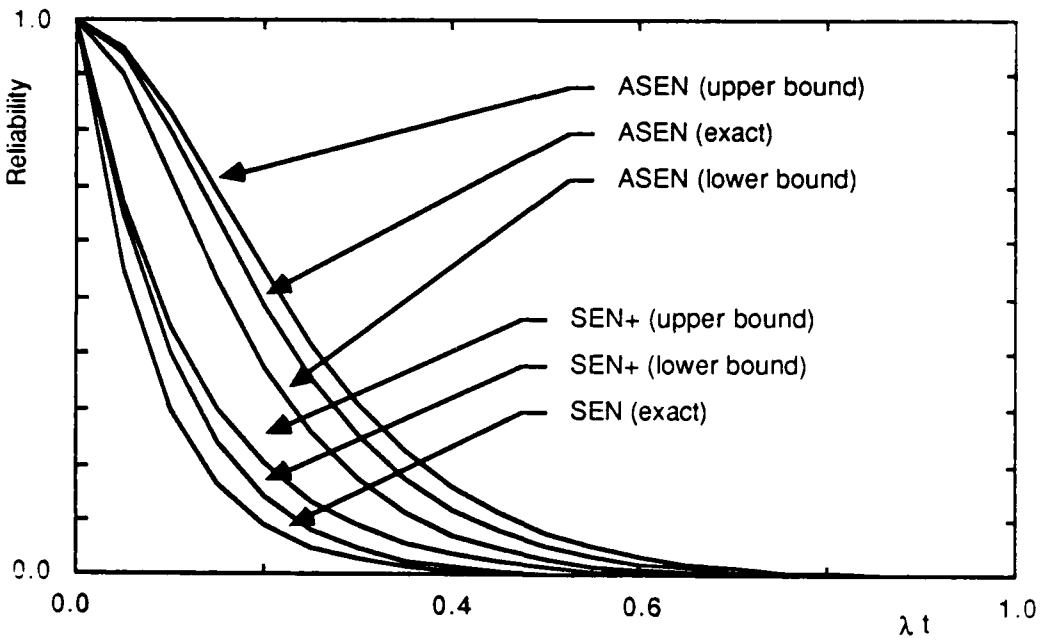


Figure 5.19: Comparison of the Network Reliabilities for the 8×8 Network.

5.7.3 Network Comparisons

In this section, the reliability of the ASEN is compared to both the SEN and SEN+ networks.

Reliability and Cost

In Table 5.3, absolute and relative measures are used to compare the networks. For $N = 8$ and larger, the $MTTF$ of the SEN+ is greater than that of the SEN; for $N = 4$ and larger, the ASEN's $MTTF$ is superior to both. The $NMTTF$ data for the SEN+ and ASEN show that as the size of the network increases, the reliability advantage of the ASEN is significantly greater than that of the SEN+. In particular, note that the $NMTTF$ upper bound of the SEN+ is much smaller than the $NMTTF$ lower bound of the ASEN.

Size	$MTTF * \lambda$				$NMTTF$		
	SEN	SEN+		ASEN	SEN+		ASEN
N	EXACT	LB	UB	LB	LB	UB	LB
4	0.25000	0.23333	0.23333	0.75000	0.9333	0.9333	3.0000
8	0.08333	0.10417	0.12450	0.18912	1.2500	1.4940	2.2690
16	0.03125	0.04643	0.06250	0.08527	1.4857	2.0000	2.7280
32	0.01250	0.02083	0.03125	0.04607	1.6667	2.5000	3.6860
64	0.00521	0.00942	0.01563	0.02712	1.8095	3.0010	5.2080
128	0.00223	0.00430	0.00781	0.01676	1.9250	3.4989	7.5078
256	0.00098	0.00197	0.00391	0.01067	2.0202	4.0038	10.9240
512	0.00043	0.00091	0.00195	0.00693	2.1000	4.4928	15.9591
1024	0.00020	0.00042	0.00098	0.00456	2.1678	5.0176	23.3473

Table 5.3: $MTTF$ and $NMTTF$ Ratios for the $N \times N$ Networks.

Based on the number of equivalent SE_2s, Table 5.4, shows the complexities of the networks. For larger networks, the ASEN is more than twice as complex as the SEN+. If differences in the component complexities are ignored, then the ASEN will appear to be even less costly than the basic SEN since it will have $N/2$ fewer SEs. In comparison with the SEN+, the ASEN would have N fewer SEs.

Figure 5.20 plots both the ratio of the $NMTTF$ and of the cost of the ASEN to the SEN+ as a function of network size (using a $\log_2$ scale on the x-axis). For the case of the ASEN, the growth in $NMTTF$ is much faster than the corresponding increase in cost as network size increases. For example, for $N = 1024$ the ASEN is more than twice as expensive as the SEN+, but it is also more than ten times more reliable. (The asymptotic cost ratio ASEN/SEN is 3.)

Size	Network Complexity			Ratio	
N	SEN	SEN+	ASEN	$\frac{SEN+}{SEN}$	$\frac{ASEN}{SEN}$
4	4	6	4	1.5000	1.0000
8	12	16	20	1.3333	1.6670
16	32	40	64	1.2500	2.0000
32	80	96	176	1.2000	2.2000
64	192	224	448	1.1667	2.3333
128	448	512	1088	1.1429	2.4286
256	1024	1152	2560	1.1250	2.5000
512	2304	2560	5888	1.1111	2.5556
1024	5120	5632	13312	1.1000	2.6000

Table 5.4: Network Complexity for the $N \times N$ Networks.

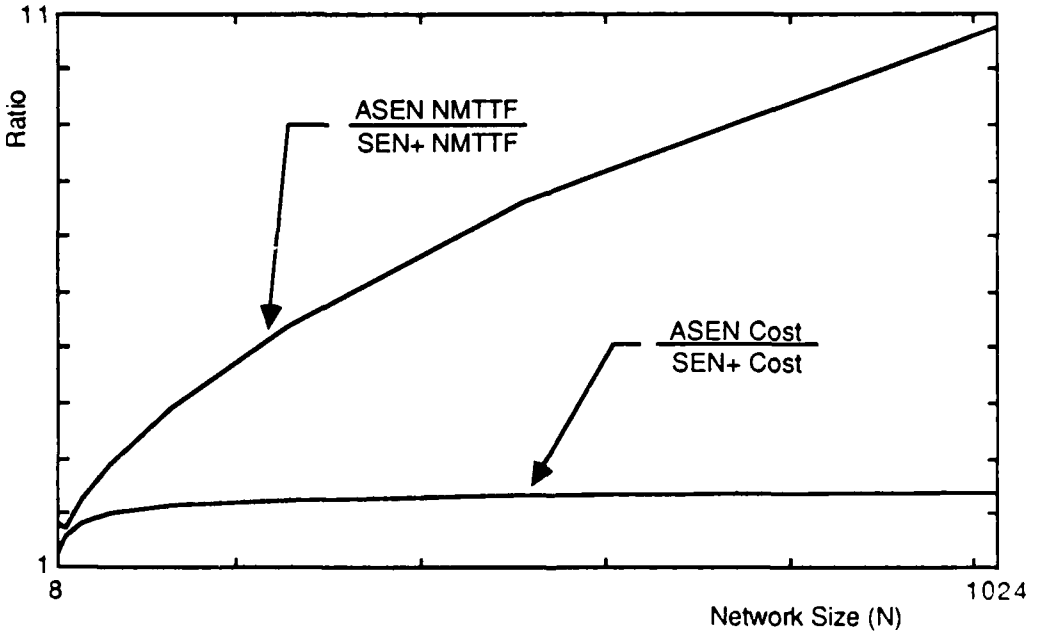


Figure 5.20: Ratios of the $NMTTF$ and the Cost of the ASEN to the SEN+.

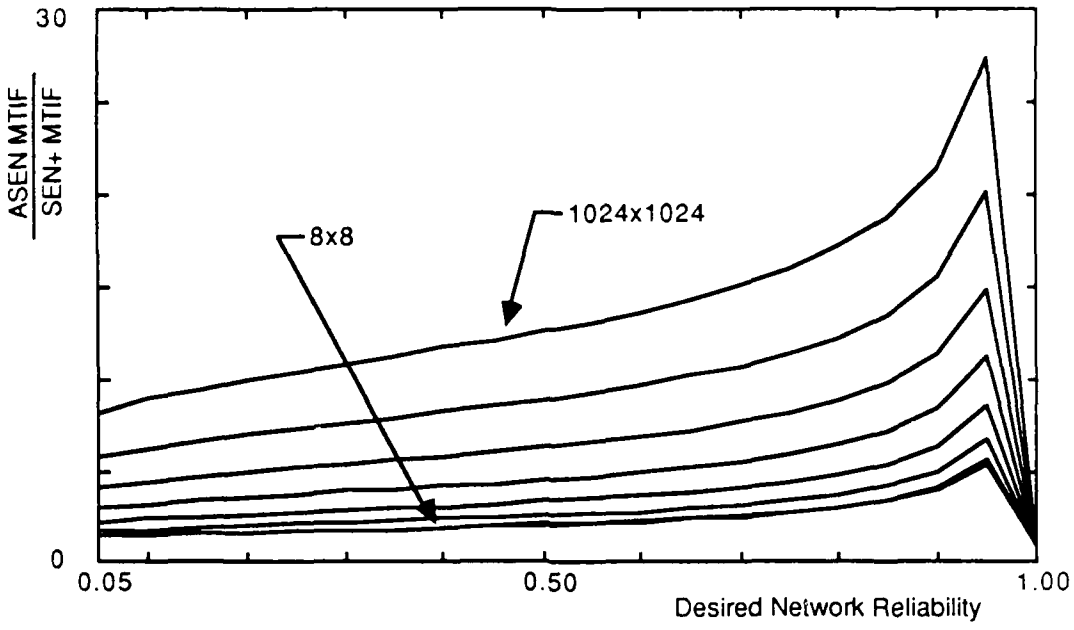


Figure 5.21: Ratio of the Mission Time Improvement Factor of the ASEN to the SEN+ for Networks from Size 8×8 to 1024×1024 Using the Lower-Bound Model.

Mission Time Improvement Factor

The $MTIF$ for 8×8 through 1024×1024 networks, was computed using the lower-bound model. In Figure 5.21, the ratio of the $MTIF$ of the ASEN to that of the SEN+ is plotted. Observe the dramatic increase in reliability achieved by the ASEN in Figure 5.21. This shows that the ASEN is superior to the SEN+.

5.7.4 Extensions to Reliability Analysis of ASEN

Previous reliability analysis of the ASEN has examined terminal reliability and the $MTTF$ (a single-valued measure) using bounds. In Sections 5.7.1 and 5.7.2, this work was extended to transient reliability analysis of these networks and derivation of the closed-form reliability expressions for small

networks. In this section, a further extension is made by considering *imperfect coverage* and *on-line repair* in the reliability analysis.

If the usual approach of an overall Markov model to incorporate imperfect coverage and/or on-line repair were taken, then analysis would be restricted to an 8×8 ASEN network. Instead, a hierarchical approach is used to model rather large ASEN networks. In the lower-bound block diagram model shown in Figure 5.17, each parallel combination can be considered to be a single "pseudo" component which is modeled as a Markov chain. This lower-level Markov model can be designed to incorporate imperfect coverage and/or on-line repair from which pseudo-component reliability can be determined. The overall system reliability is then obtained by taking a top-level block diagram model and multiplying individual pseudo-component reliabilities. For other uses of the hierarchical approach to reliability modeling, the reader is referred to [84].

Imperfect Coverage

It is often the case that, when a component in a system fails, the detection, isolation, and reconfiguration procedures of the system are less than perfect. This notion of imperfection is called *imperfect coverage*, and it is defined as the probability that the system successfully accomplishes system reconfiguration given that a component failure occurs [17,4]. Denote this probability as c. Imperfect coverage is an important factor in considering the reliability of interconnection networks since as their size increases, the number of components increases, and the potential for an uncovered fault to occur increases as well.

Consider the lower-bound model of the ASEN shown in Figure 5.17. Each parallel arrangement of two SE_{2m} can be considered as a pseudo component

denoted as PC_{2m} whose reliability, given imperfect coverage, can be computed from a simple 3-state Markov model:

$$R_{PC_{2m}}(t) = e^{-2\lambda_{2m}t} + 2ce^{-\lambda_{2m}t}(1 - e^{-\lambda_{2m}t}). \tag{5.36}$$

The first term in equation (5.36) represents the probability that both SE_{2m} are operating concurrently, and the second term represents the probability of operation with one of the two SEs after successful reconfiguration of the system given that one of the two SEs fail.

Each series-parallel arrangement of SE_{3m} and each such arrangement of SE_3 can be considered as a pseudo component in a similar fashion. The reliability expressions are:

$$R_{PC_{3m}}(t) = e^{-4\lambda_{3m}t} + 2ce^{-2\lambda_{3m}t}(1 - e^{-2\lambda_{3m}t}), \text{ and} \tag{5.37}$$

$$R_{PC_3}(t) = e^{-4\lambda_3 t} + 2ce^{-2\lambda_3 t}(1 - e^{-2\lambda_3 t}), \tag{5.38}$$

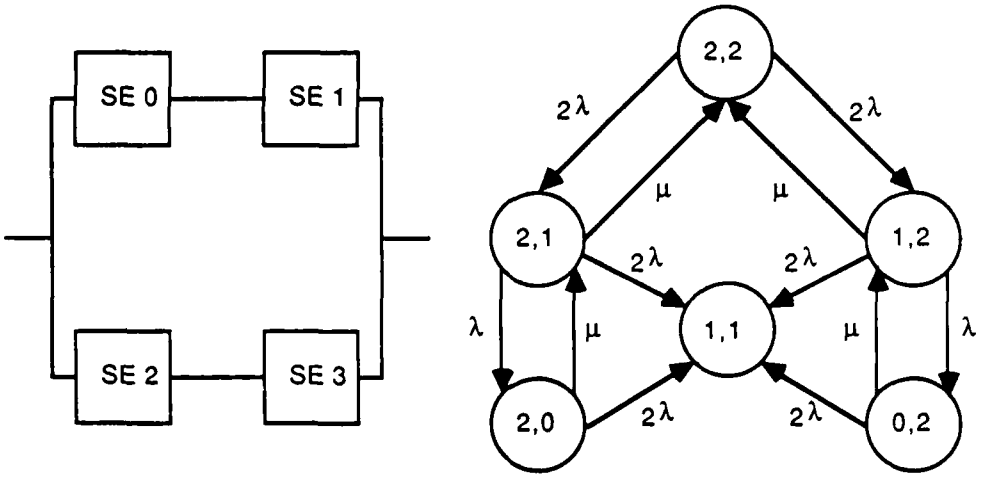
respectively. Hence, the reliability expression for the lower-bound model of the ASEN which allows for imperfect coverage is given by

$$R_{\text{ASEN}}(t) = [R_{PC_{3m}}(t)]^{\frac{N}{8}}[R_{PC_3}(t)]^{\frac{N}{8}(\log_2 N - 3)}[R_{PC_{2m}}(t)]^{\frac{N}{4}}. \tag{5.39}$$

As will be shown later, even a coverage factor of 0.95 has a significant effect on the ASEN's reliability.

On-Line Repair

One characteristic of the ASEN is that it lends itself to on-line repair and maintainability. But modeling this behavior has not been previously addressed. Previous reliability analysis of ASENs is extended by employing hierarchical decomposition in modeling such behavior. Each pair of conjugate loops is a series-parallel arrangement of four switching elements. This



(a) Pair of Conjugate Loops (b) CTMC Representation

Figure 5.22: Markov Chain Representation of a "Pseudo" Component.

grouping can be considered as a pseudo component and the failure and repair behavior of this PC can be modeled using a discrete-state, continuous-time Markov chain. The reliability expression of the pseudo component is obtained, and then this reliability function is used as input to the lower-bound model of the ASEN.

Figure 5.22 shows: (a) a pair of conjugate loops from Figure 3.5, and (b) the CTMC representation of the failure and repair behavior of the pseudo component. Tuple (i, j) represents the number of operational components in each loop. For example, $i = 2$ means both SE 0 and SE 1 are operational. Furthermore, switches are replaced in pairs even though only one SE in the loop may be failed. Repair then takes the same time to replace one or both SEs in a loop. Let the failure rate of each component be λ and the repair rate be μ.

For reliability, the concern is with continuous operation given that on-line repair is conducted. Note that state $(1,1)$ is an absorbing state. Let $P_{1,1}(t)$ be the transient probability of that state, then $1 - P_{1,1}(t)$ is the reliability of the pseudo component. The 6-state CTMC in Figure 5.22 can be reduced to a 4-state CTMC; then the transient solution of the state probabilities is accomplished using Laplace transforms, solution of a system of linear equations, partial fraction expansion, and inversion back to the time domain. The highest-order denominator of the Laplace transform solution of the 4-state CTMC is a quartic equation with four real roots. One root is zero, the other three roots are determined by using the usual explicit closed-form expressions found, for example, in [73]. Let $\bar{P}(s)$ denote the Laplace transform of the transient probability of being in the absorbing state($P_{1,1}(t)$), then

$$\bar{P}(s) = \sum_{i=1}^{4} \frac{A_i}{(s + x_i)} \tag{5.40}$$

where the $-x_i$ are the real roots of the denominator, and the A_i are the constant coefficients. Then

$$P_{1,1}(t) = \sum_{i=1}^{4} A_i e^{-x_i t}, \text{ and} \tag{5.41}$$

$$R_{PC}(t) = 1 - P_{1,1}(t). \tag{5.42}$$

Once $R_{PC}(t)$ has been determined, the reliability of the ASEN with on-line repair is found by replacing each pair of conjugate loops with its PC in the lower-bound model of the ASEN. For small networks, SHARPE (see Appendix C) can be used directly to compute system reliability, but for larger networks, numerical instabilities were avoided by using a program written specifically for the present problem.

In Figure 5.23, the reliability of the 256×256 ASEN is plotted using the upper and lower-bound models under three assumptions:

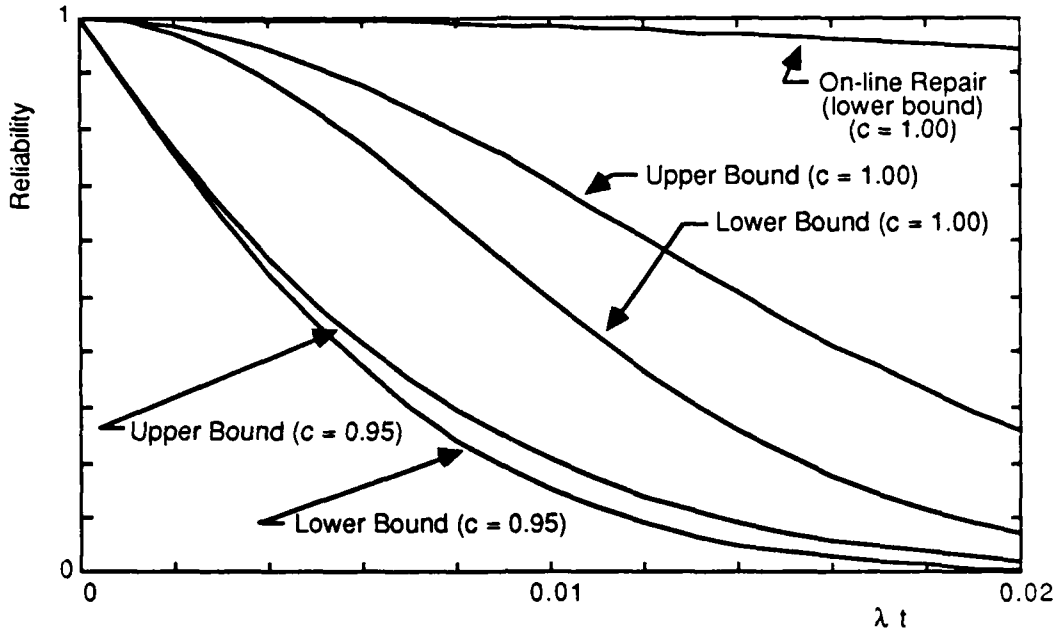


Figure 5.23: Reliability of the 256 × 256 ASEN.

1. An imperfect coverage factor ($c = 0.95$).

2. Perfect coverage ($c = 1.00$).

3. On-line repair (lower-bound model only, $c = 1.00$).

Assume $\lambda_{3m} = 3.5$, $\lambda_3 = 3$, $\lambda_{2m} = 1.5$, and $\mu = 500,000$. This is equivalent to assuming a failure rate of 1×10^{-6} SEs per hour using a "normalized" SE and a repair rate of one loop per one-half hour. The figure presents three views of the ASEN. Even the slightest probability (0.05) of unsuccessful reconfiguration has a significant impact on ASEN reliability. On the other hand, on-line repair enhances the reliability of the ASEN in a profound way. For example, Table 5.5 shows the impact of imperfect coverage and on-line repair on the reliability of the 256 × 256 ASEN. At time $t = 0.01$, the reliability ranges from 0.15 to 0.99. Table 5.6 compares the $MTTF$ of the ASEN under three assumptions using the lower-bound model. As network size in-

Reliability of the 256×256 ASEN at $t = 0.01$				
Lower Bound		Upper Bound		On-Line Repair
$c = 1.00$	$c = 0.95$	$c = 1.00$	$c = 0.95$	$c = 1.00$
0.43	0.15	0.70	0.21	0.99

Table 5.5: Impact of Imperfect Coverage and On-Line Repair on the 256×256 ASEN.

Lower Bound with	Network Size (N)							
	8	16	32	64	128	256	512	1024
Repair	0.6111	0.3880	0.2538	0.1696	0.1152	0.0791	0.0547	0.0381
$c = 1.00$	0.1891	0.0853	0.0461	0.0271	0.0168	0.0107	0.0070	0.0046
$c = 0.95$	0.1781	0.0763	0.0381	0.0200	0.0106	0.0055	0.0028	0.0013

Table 5.6: $MTTF$ of ASEN Under Three Model Assumptions.

creases, the improvement in $MTTF$ with on-line repair over the models with no repair increases. For example, as network size increases from 8×8 to 1024×1024, the ratio of the $MTTF$ with on-line repair increases from 3.23 to 8.28 for $c = 1.00$.

5.8 Summary

In this chapter, the transient reliability of the Shuffle-Exchange Network (SEN) and three fault-tolerant schemes for improving the reliability of this network were examined. These schemes are the SEN+, 2-SEN, and ASEN. Exact closed-form expressions for the time-dependent reliability of the SEN and the 8×8 and 16×16 SEN with an additional stage (SEN+) were derived independent of the assumptions regarding the underlying component-lifetime-distributions. Also, for the networks examined, the exponential distribution provides a conservative estimate of the reliability of these MINs if the components have an increasing-failure-rate lifetime-distribution.

Further, a tight reliability lower bound for larger SEN+ networks was derived and used to provide numerical results for networks as large as 1024× 1024. A comparison of these networks shows that, on the basis of reliability, the SEN+ is superior to the SEN and the redundant SEN.

Next, exact closed-form expressions for the reliability of 4×4 and 8×8 ASEN networks were derived. Also derived were the upper and lower bounds for approximating the reliability of larger ASEN networks by "normalizing" the networks based on the gate complexities of their components. The bounds obtained were compared to the exact solutions derived for the 8×8 ASEN to show that they are a reasonable approximation of ASEN reliability, and then these bounds were used for analyzing ASEN networks up to size 1024×1024. A comparison of the mean time to failure, cost, and mission time improvement factor of the SEN+ and ASEN networks was presented, and it was shown that, on the basis of reliability, the ASEN is superior to the SEN, 2-SEN, and SEN+. Finally, through the novel use of hierarchical decomposition, results on the reliability of ASENs were extended to include imperfect coverage and on-line repair.

Chapter 6

Selecting the Optimal Switching Element Size for SEN and SEN+

A significant amount of the reliability analysis presented in Chapter 5 was concerned with the SEN and the SEN+. In this chapter, the analysis is extended to the (uniform) Omega network [54] for the purpose of finding the optimum switch size for maximizing interconnection network reliability.

Consider an $N \times N$ Omega network, where $N = m^n$, constructed using $m \times m$ crossbar switches and $m * m^{n-1}$ shuffles connecting the stages, where $m = 2^l$, for l a positive integer. There are $\log_m N$ stages of N/m switches per stage. The Omega network shall be referred to as SEN_m and the Omega network with an additional stage as SEN+_m. The additional stage will make the network $(m-1)$-fault-tolerant in the intermediate stages since, in this portion of the network, there are m disjoint paths between each S-D pair.

Let $r_{\text{SE}_m}(t)$ be the reliability of the $m \times m$ switching element. The exact reliability expression for the Omega network is given by

$$R_{\text{SEN}_m}(t) = [r_{\text{SE}_m}(t)]^{\frac{N}{m} \cdot \lceil \log_m N \rceil} . \tag{6.1}$$

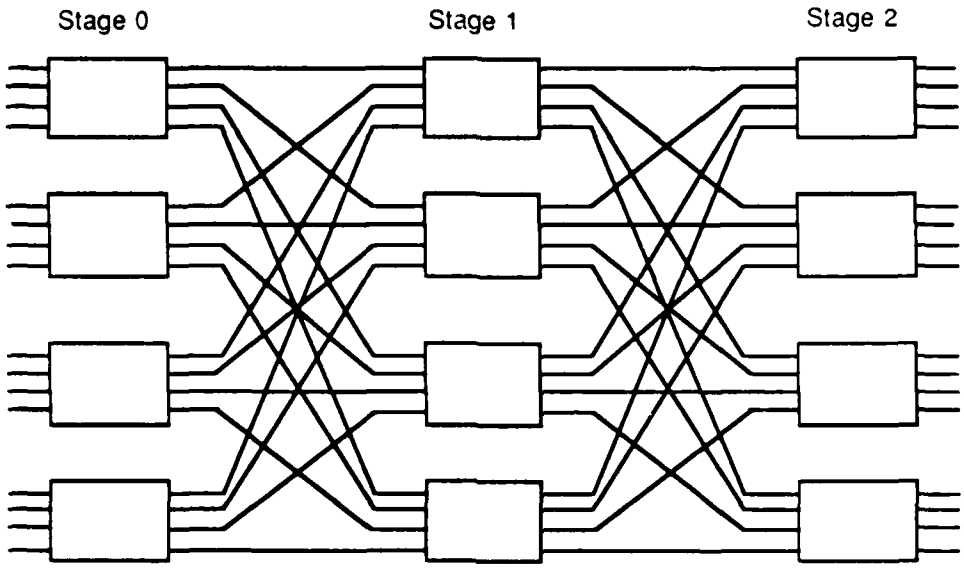


Figure 6.1: 16 × 16 Omega Network with 4 × 4 Switches (SEN+$_4$).

The reliability expressions for the lower and upper bounds for the Omega network with the additional stage are:

$$R_{lb_m}(t) = [r_{\mathrm{SE}_m}(t)]^{\frac{2N}{m}} \cdot [1 - [1 - r_{\mathrm{SE}_m}(t)^{\frac{N}{m^2}(\log_m N - 1)}]^m], \text{ and} \quad (6.2)$$

$$R_{ub_m}(t) = [r_{\mathrm{SE}_m}(t)]^{\frac{2N}{m}} \cdot [1 - (1 - r_{\mathrm{SE}_m}(t))^{\frac{N}{m}}]^{(\log_m N - 1)}. \quad (6.3)$$

Figure 6.1 shows the arrangement of a 16×16 SEN+$_4$ network. The expression for the reliability of the last two stages is equivalent to that of the basic Omega network which is

$$R_{\mathrm{SEN}_4}(t) = [r_{\mathrm{SE}_4}(t)]^8 \, . \quad (6.4)$$

The exact reliability expression for the corresponding SEN+$_4$ network as shown in Figure 6.1 is

$$R_{\mathrm{SEN}+_4}(t) = [r_{\mathrm{SE}_4}(t)]^8 \cdot [1 - (1 - r_{\mathrm{SE}_4}(t))^4] \, . \quad (6.5)$$

The hardware for the SEs in this network, however, will have a higher gate complexity than the 2×2 SEs used earlier. Since m denotes the size of the $m \times m$ SE, let $f(m)$ be the cost or complexity of the SE, where f is a function of m. It is generally accepted that, in terms of gate complexity, $f(m) = 4m(m-1)$ [47]. Now one can express the complexity of the SE_m in terms of the basic SE_2 used earlier. The equation is

$$r_{SE_m}(t) = [r_{SE}(t)]^{\frac{f(m)}{f(2)}} .$$

Then

$$r_{SE_m}(t) = [r_{SE}(t)]^{\frac{4m(m-1)}{4 \cdot 2}} = [r_{SE}(t)]^{\frac{m(m-1)}{2}} .$$

This provides an expression for the reliabilities of these two networks in terms of SE_2. Now equation 6.1 can be rewritten as

$$R_{SEN_m}(t) = [r_{SE}(t)]^{\frac{N(m-1)}{2}(\frac{\log N}{\log m})} . \tag{6.6}$$

Since $0 \leq r_{SE}(t) \leq 1$, the network reliability will be maximized for $m = 2$.

The reliability expression for the lower bound expressed in equation (6.2) becomes

$$R_{lb_m}(t) = [r_{SE}(t)]^{N(m-1)} \cdot [1 - [1 - r_{SE}(t)^{\frac{N(m-1)}{2m}(\log_m N - 1)}]^m]. \tag{6.7}$$

Using an exhaustive search, it can be shown that for $N \leq 1024$, expression (6.7) is maximized for $m = 2$. The cost (C) functions for each of these networks can be expressed as well. For the basic network the cost is given by

$$C(N,m) = 4N(m-1)(\frac{\log N}{\log m}). \tag{6.8}$$

It is clear that cost is minimized for $m = 2$. For the network with the additional stage the cost is expressed as

$$C(N,m,+) = 4N(m-1)(\frac{\log N}{\log m} + 1). \tag{6.9}$$

So, for the redundant path network, the optimum switch size for minimizing the cost is also $m = 2$.

In summary, based on reliability and hardware cost, a designer should choose a 2×2 SE for constructing an SEN network. Similarly, for $N \leq 1024$, the optimum switch size for the SEN+ network is $m = 2$.

Chapter 7

Performability

7.1 Introduction

In this chapter, combined performance and reliability measures for unique-path Multistage Interconnection Networks (MINs) are examined. While the Shuffle-Exchange MIN (SEN) will be the specific network considered, a number of other MINs are topologically equivalent. Many measures may be used for combining performance and reliability, but the focus here will be on three such measures. Of interest is the "average instantaneous reward rate at time t", the "average accumulated reward until time t", and the "distribution of the cumulative reward until system failure". These measures include, as special cases, several "pure" performance measures (the maximum and minimum reward rates and their product with the time-to-failure random variable); the distributions of these performance measures; and "pure" reliability measures (the distribution of a system's lifetime and the mean time to failure).

Separately modeling the reliability and performance of networks is not new. Recently, however, some research has been done on combining performance and reliability/availability analysis for a few interconnection networks. In [23], performance and reliability for the crossbar and the multiple-bus ar-

chitectures are combined as a single measure — computation availability. Markov chains are used for the analysis of the computation availability for these systems. A closed-form expression is derived for the reliability of the multiple-bus architecture considering graceful degradation. The results show that the reliability of the multiple-bus is better than that of the crossbar. Also, after some time t and depending on the number of buses, the computation availability of the multiple-bus exceeds that of the crossbar.

More recently, in [63] performability measures associated with the processing elements of Hypercube-based networks are examined. The disconnection probability of a network is used to compute the coverage factor for the system.

The purpose of this chapter is to show the applicability of Markov reward models for the analysis of interconnection networks. Determining the performance of an interconnection network under *all* possible failure configurations is a very difficult problem, but a methodology is shown in this chapter through analysis of the SEN. Then, a detailed analysis of a complete multiprocessor system is performed in Chapter 8.

7.2 Previous Work

The evolution of a degradable system through various configurations with different sets of operational components can be represented by a discrete-state, continuous-time Markov chain (CTMC). In performability terminology, this CTMC is referred to as a structure-state process. Associated with each state of the CTMC is a reward rate that represents the performance level of the system in that state. Each state represents a different system configuration. Transitions to states with smaller reward rates (lower performance levels) are generally characterized as failure transitions, and, in the case of

repairable systems, transitions to states with higher performance levels are characterized as repair transitions. The set of reward rates associated with the states of a structure-state process is referred to as the reward structure. The structure-state process combined with the reward structure constitutes a *Markov reward model* (MRM).

The choice of performance measure to be used for determining reward rates is a function of the system to be evaluated. Often a raw measure of system capacity such as the instruction execution rate may be the appropriate reward rate. For interconnection networks, the appropriate measure is bandwidth (BW). At other times, a queueing-theoretic performance model may be used to compute the reward rates. Since the time-scale of the performance-related events (bandwidth) is at least two orders of magnitude less than the the time-scale of the reliability-related events (component failures), steady-state values of performance models are used to specify the performance levels or reward rates for each structure state.

For degradable systems, a significant measure is the amount of accumulated work that can be produced by a given system over some specified time interval. Beaudry [10] proposed an algorithm to compute the distribution of accumulated reward until system failure for nonrepairable systems. In [61], the distribution function of the cumulative work during a specified period of time is considered as the performability measure. Goyal and Tantawi [36] and Donatiello and Iyer [27], provide efficient numerical algorithms to compute the distribution of accumulated reward in general acyclic structure-state processes.

In [48], another numerical algorithm was proposed that used numerical inversion of the double Laplace transform equations to obtain the performability measure. The algorithm presented has time complexity $O(k^4)$ where k

is the number of states of the Markov reward model. This algorithm applies to the computation of the distribution of accumulated reward for a general CTMC and arbitrary reward structure. The algorithm has been recently improved to an $O(k^3)$ execution time by Smith *et al.* in [88]. This algorithm makes the solution of larger Markov reward models practical.

In the next section, the notation usually associated with performability analysis will be introduced.

7.3 Notation

To facilitate the development of the notation for Markov reward models, let T be the time until system failure. Then, the system reliability is given by

$$R(t) = \text{Prob}[T > t] \; . \tag{7.1}$$

The evolution of the system in time is represented by the discrete-state stochastic process $\{Z(t),\ t \geq 0\}$. At time t, $Z(t)$ is the structure state of the system, and $Z(t) \in \Psi = \{1, 2, \ldots, k\}$, where Ψ represents the state space of the CTMC and k denotes the number of states in the structure-state process. If the holding times in the structure states are exponentially distributed, then $Z(t)$ is a homogeneous CTMC. Let q_{ij}, $i, j \in \{1, \ldots, k\}$, be the transition rate from state i to state j. Then $Q = [q_{ij}]$ is the k by k transition rate matrix where

$$q_{ii} \;\; = \;\; - \sum_{j=1, j \neq i}^{k} q_{ij}.$$

Also, let $P_i(t)$ denote the probability that the system is in state i at time t. That is, $P_i(t) = \text{Prob}[Z(t) = i]$. The transient-state probability vector $\underline{P}(t)$

may be computed by solving a matrix differential equation [98],

$$\dot{\underline{P}}(t) \;=\; Q^{\mathrm{T}}\,\underline{P}(t) \quad , \tag{7.2}$$

where the transpose of a vector or matrix is indicated by a superscript T.

To represent the reward structure, let r_i denote the reward rate associated with structure-state i. Then the vector $\underline{r}$ defines the reward structure. To represent the reward rate of the system at time t, let $X(t) = r_{Z(t)}$.

From the state probabilities we can obtain the instantaneous availability

$$A(t) = \sum_{i \in \mathrm{UP}} P_i(t)$$

where UP is the set of operational states. The expected reward rate at time t is

$$E[X(t)] = \sum_i r_i P_i(t),$$

also known as the *computation availability* [10].

The steady-state probability vector $\underline{\pi}$ of the Markov chain is the solution for the linear system (assuming that the CTMC is irreducible):

$$\begin{aligned} Q^{\mathrm{T}}\,\underline{\pi} \;&=\; 0, \text{ and} \\ \sum_i \pi_i \;&=\; 1 \,. \end{aligned}$$

Methods of solving this system are discussed by Stewart and Goyal in [93]. From the steady-state probabilities, we can obtain the steady-state availability

$$A = \sum_{i \in \mathrm{UP}} \pi_i,$$

and the steady-state computation availability

$$\lim_{t \to \infty} E[X(t)] = \sum_i r_i \pi_i.$$

For nonrepairable systems, these measures are not of interest since the steady-state availability and expected reward rate as time approaches infinity are zero.

Further, let $Y(t)$ be the accumulated reward until time t. It is the amount of reward accumulated (the amount of work done) by a system during the interval $(0,t)$, and it is equal to the area under the $X(t)$ curve. That is,

$$Y(t) = \int_0^t X(\tau)d\tau \quad . \tag{7.3}$$

If we use bandwidth to construct the reward structure, then from equation (7.3), $Y(t)$ represents the number of requests that the IN is capable of satisfying by time t.

The expected value of the accumulated reward can be determined by

$$\begin{aligned} E[Y(t)] &= E[\int_0^t X(\tau)d\tau] \\ &= \int_0^t E[X(\tau)]d\tau \\ &= \sum_i r_i \int_0^t P_i(\tau)d\tau. \end{aligned} \tag{7.4}$$

$E[X(t)]$ and $E[Y(t)]$ provide the first moments of their underlying distributions. However, if one is interested in the behavior of $Y(t)$ far from the mean (e.g., when a system is required to have a high probability of completing a specified amount of work in a particular time interval), the central moments may not provide accurate information. Instead, the distributions themselves are required.

The distribution of reward accumulated in the interval $(0,t)$ evaluated at x is:

$$y(x,t) \equiv \text{Prob}[Y(t) \leq x],$$

and its complement is :

$$y^c(x,t) \equiv \text{Prob}[Y(t) > x],$$

where x is a specified amount of performance (work) to be achieved. Methods of computing $Y^C(x,t)$ are discussed in [48] and [88]. In case the CTMC has one or more absorbing states, it is useful to analyze the accumulated reward until absorption (failure), $Y(\infty)$. Let H_i be a random variable denoting the time spent in state i until system failure, and let r_i be the bandwidth in state i; then the total number of requests that the can be handled prior to system failure, $Y(\infty)$, can be computed as

$$Y(\infty) = \sum r_i H_i. \tag{7.5}$$

The distribution function of $Y(\infty)$ can be computed by constructing another CTMC with the transition rate matrix Q' so that $q'_{ij} = q_{ij}/r_i$ for $r_i > 0$ and solving for the time to absorption for the new CTMC [10].

Table 7.1 summarizes the information currently available on performability measures. The table shows that measuring combined performance and reliability/availability for various systems has experienced increasing levels of sophistication over the past few years. Early models considered only transient measures and models without repair. As interest in finding ways to analyze more complex systems increased, distributional measures and repair behavior were considered. In the table, the Laplace-Stieltjes Transform (LST) is denoted by ~ (e.g., $G^{\sim}(u) = \int_0^\infty e^{-ux} dG(x)$) and the Laplace Transform (LT) by * (e.g., $f^*(s) = \int_0^\infty e^{-sx} f(x)dx$). Each measure's properties are indicated. The properties are whether the quantity measured is instantaneous (I) or cumulative (C); steady state (S) or transient (T); and whether the measure is a distribution function (DF) such as the probability mass function (pmf) or the cumulative distribution function (CDF) or a central moment (M). The references cited are related to the work on the corresponding measures. While the list is not necessarily exhaustive, it does provide sufficient reference for obtaining additional information on the corresponding measure. As shown in

the table, the algorithms used in [88] provide the most advanced analytical methods for evaluating all Markov reward model measures of interest.

Measures used to characterize the behavior of Markov reward models of MINs without repair are the reliability, $R(t)$; the expected reward rate at time t, $E[X(t)]$; the expected accumulated reward at time t, $E[Y(t)]$; and the distribution of accumulated reward until absorption $\mathcal{Y}(x,\infty) \equiv \lim_{t\to\infty} \mathcal{Y}(x,t)$.

After offering an intuitive explanation of the influence of reward rates on system performance, the 4×4 SEN will receive an exact analysis, and an 8×8 SEN will be analyzed using an approximation technique. Current difficulties encountered in modeling larger SENs will be discussed, as well.

7.4 Markov Reward Model for the SEN

An unique-path Multistage Interconnection Network (MIN) can be viewed as a gracefully degradable system. The MIN is a nonrepairable system; and as such, its evolution can be represented by an acyclic Markov chain. The states that the continuous-time Markov chain progresses through enroute to system failure are the configurations of a structure-state process [61]. Each state in the CTMC has a reward rate associated with it that represents the rate at which the MIN can perform useful work while in that state.

Before beginning the analysis, an intuitive argument about the merits of a single measure which combines performance and reliability will be presented. Unique-path MINs provide a single path between a given source-destination (S-D) pair; so with the failure of any one switching element (SE), some source is disconnected from some destination. In fact, several S-D pairs may be disconnected.

If one defines a MIN as being operational as long as no SE has failed, reliability analysis is straightforward. For example, by analyzing the MIN

Measure : Computation Method (Technique or Equation)	C or I: S or T: M or DF	References
$P_i(t)$: $\dot{\underline{P}}(t) = Q^{\mathrm{T}}\underline{P}(t)$ π_i : $\underline{0} = Q^{\mathrm{T}}\underline{\pi}$	I : T : pmf I : S : pmf	Reibman '87 [80] Trivedi '82 [98] Stewart '85 [93]
$A(t)$: $\dot{\underline{P}}(t) = Q^{\mathrm{T}}\underline{P}(t)$ $A(\infty)$: $\underline{0} = Q^{\mathrm{T}}\underline{\pi}$	I : T : M I : S : M	Reibman '87 [80] Trivedi '82 [98] Stewart '85 [93]
$R(t)$: $\dot{\underline{P}}(t) = Q^{\mathrm{T}}\underline{P}(t)$ $E[X(t)]$: $\sum_i r_i P_i(t)$	I : T : CDF I : T : M	Shooman '68 [85] Beaudry '78 [10]
$E[Y(t)]$: $\sum_i r_i \int_0^t P_i(\tau)d\tau$	C : T : M	Goyal '87 [35] Reibman '87 [82]
$\mathcal{Y}(x,t)$: $[sI + uR - Q]\cdot \mathcal{Y}^{\sim *}(u,s) = \underline{e}$: $R\frac{\partial y}{\partial x} + \frac{\partial y}{\partial t} = Q\underline{y}$	C : T : CDF	Smith '87 [88] Smith '87 [89]
$\mathcal{Y}(x,\infty)$: $\dot{\underline{P}}(t) = Q^{\mathrm{T}}\underline{P}(t)$	C : S : CDF	Beaudry '78 [10]

Table 7.1: Performability Measures Summary

as a system consisting of SEs connected in series. Analytically, let $r_{SE}(t)$ be the reliability of an individual SE at time t and $R(t)$ be the reliability of the MIN at time t, then $R(t)$ is simply the product of the individual reliabilities assuming that the SEs behave independently. Further, assume that the SEs are identical and each has an exponentially distributed lifetime, with parameter λ, then the time-to-failure of the MIN will also be exponentially distributed with parameter $M\lambda$, where M is the number of switching elements in the MIN. This condition, however, is not very comforting since it implies that the $MTTF$ is $1/M\lambda$, and thus the $MTTF$ decreases as the network complexity increases. Observing that the network complexity of a SEN is a function of the number of sources (N) and equals $(N/2)(\log_2 N)$, it is clear that obtaining large (say 1024×1024) SENs with a long lifetime will require SEs with a very long lifetime. For example, a 1024×1024 SEN composed of SEs with an exponentially distributed lifetime with parameter $\lambda = 10^{-6}$ failures/hour will have 5120 SEs and a $MTTF$ of only 8 days. It is doubtful such a system would find many applications.

From definition 2 in Chapter 5, a MIN is operational so long as some source can communicate with some destination. This view permits a number of ways to analyze the MIN. The traditional way is to model the MIN as a continuous-time Markov chain. But even in this simple model one is implicitly associating a performance level with each state. Consider the performance level associated with each state to be either a 1 or a 0. A reward rate of 1 associated with a state means that work is performed at the rate of 1 unit per unit time while in that state. Then denote the reward rate (r) associated with each structure-state i as r_i.

The reliability analysis can be done in terms of a performability model by letting T be the time until system failure. Let $r_i = 1$ for all operational

states and $r_i = 0$ for all failure states, the system reliability is

$$R(t) = \text{Prob}[T > t] = \lim_{\tau \to \infty} \text{Prob}[Y(\tau) > t]. \tag{7.6}$$

The structure-state process, $Z(t)$, for the SEN will be represented by an homogeneous continuous-time Markov chain assuming that the time spent in any particular operational state (holding time) is an exponentially distributed random variable. It is possible, however, to release this restriction of exponentially distributed holding times using at least three different approaches. The approaches that can be used are:

- a non-homogeneous CTMC [97];
- semi-Markov, structure-state process [49]; or
- the method of stages [22,41].

Analysis of the evolution of $Z(t)$ begins by selecting the appropriate reward structure. For each structure-state $i \in \Psi$, let the bandwidth in that particular configuration be the fixed reward rate r_i. So, from equation 7.3, $Y(t)$ represents the number of requests that the MIN is capable of satisfying by time t.

7.5 Reward Rate's Influence on Performance

How does the reward rate affect the performance that the model predicts the physical system will attain? The three curves in Figure 7.1 represent different levels of performance as reflected by the assumption made about failures and the reward rates chosen for each operational structure state. If one ignores the possibility that components within a particular system may fail, and if a constant reward rate is associated with each structure state,

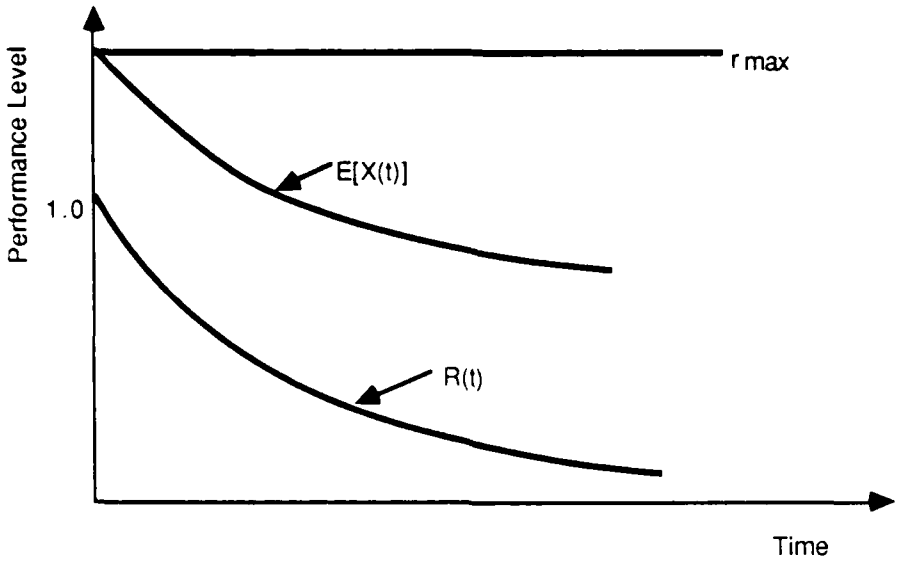


Figure 7.1: Impact of the Underlying Reward Structure on Performance Level as a Function of Time.

then as the system evolves in time, its performance level (the rate at which it does work) will be constant. Associating the maximum reward rate r_{max} $(= max_i\{r_i\})$ of any structure state with every structure state, an upper bound is obtained on the rate at which work is accomplished. This can be called a "pure" performance model. (Similarly, the minimum reward rate r_{min} $(= min_i\{r_i\})$ could be used for a performance model, but for nonrepairable systems, $r_{min} = 0$.)

Figure 7.1 shows r_{max} for a hypothetical system. If failure of the components is permitted, two additional possibilities exist. A performability model that associates a reward rate of 1 with each operational configuration and a reward rate of 0 with the failed structure states is simply a traditional model of the system's time-to-failure. The complementary distribution of the time-to-failure distribution is the system's reliability as function of time — $R(t)$.

This approach to reward rates may underestimate the system's ability to perform useful work. On the other hand, if the reward rate that is assigned to an operational structure state actually represents the productive capacity of that particular configuration, one gets a more accurate picture of the performance degradation that occurs as the system evolves. The third curve shows $E[X(t)]$,the average instantaneous reward rate at time t. The value of $E[X(t)]$ can be bounded by the following two inequalities:

$$\begin{aligned} 0 &\leq E[X(t)] \leq r_{max}, \text{ and} \\ r_{min} &\leq E[X(t)] \leq r_{max}, \ t \leq T \end{aligned}$$

where r_{min} is the smallest non-zero reward rate for the system. For discussion of nonrepairable systems, r_{min} is defined to be the smallest reward rate in an operational state.

Figure 7.2 shows three interpretations of a system's performability. These curves are specially weighted versions of the complementary distribution of the system's time-to-failure CDF. These curves, as functions of time, answer the question, "What is the probability that the system will deliver at least x amount of work before the system fails?" The curves in Figure 7.2 depict the effects of three different (perhaps) time-varying weighting assumptions. The upper curve plots Prob$[r_{max}T > x]$. This provides an upper bound. The interpretation here is that whatever state the system is in (as long as it is still operational) one gains as much benefit there as in the fully-operational state. In the case of a MIN, suppose that one arbitrarily decides that the system is considered operational as long as K sources can communicate with K destinations. Then, even though one or more components within the network may have failed, leading to a reduced bandwidth, this configuration is considered to be performing as if it were operating at full bandwidth. A rather optimistic view.

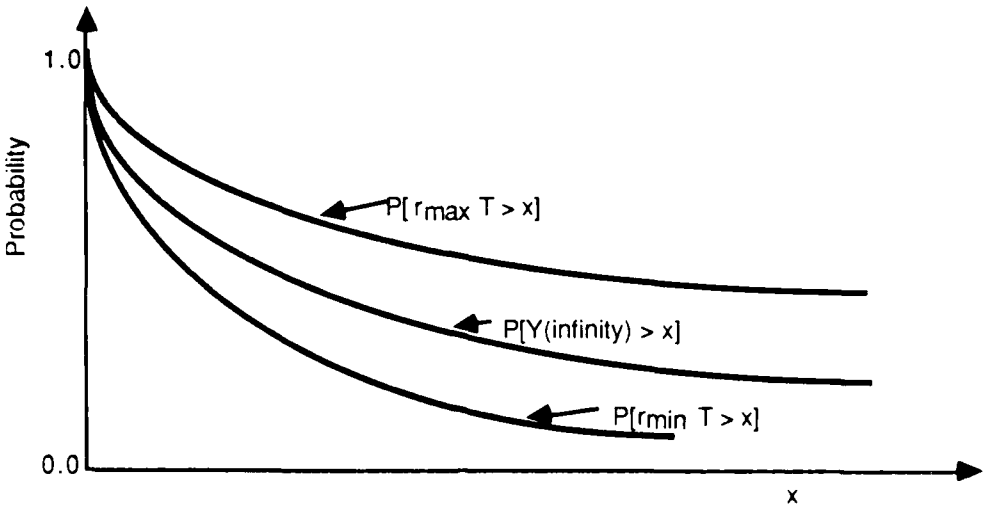


Figure 7.2: Establishing Bounds for the Complementary Distribution of Accumulated Work.

On the other hand, the lower curve plots Prob$[r_{min}T > x]$, and it provides a lower bound on the system's performability. This implies that whatever operational structure state the system is in, only minimal benefit is obtained from the system. That is, since the system's excess capacity cannot be used, this value will be discounted in determining the probability that the system will ever produce a specified amount of work. Again for the MIN, consider the K processors and K memories requirement. Then assign the smallest bandwidth of *any* of the operational configurations to *all* the operational states even though the network will be capable of performing well above that level for most of its lifetime. This will portray a rather pessimistic view.

The third interpretation is to view the MIN as a gracefully degradable system. Now, define the reward rate associated with each state as the bandwidth of that particular configuration. The center curve of Figure 7.2 shows

Prob$[Y(\infty) > x]$. Now, as the CTMC model of the system evolves, varying levels of performance will be produced in each state (with $r_i = 0$ for failure states). Here it is explicitly recognized that as components fail, the system's ability to produce useful work may be degraded and also that the system will accumulate work at decreasing rates as time progresses. This view corresponds to a more realistic view of a system's performance. Hence, the basis for the reward rates associated for various configurations can have a significant impact on decisions made about a particular system's ability to perform useful work.

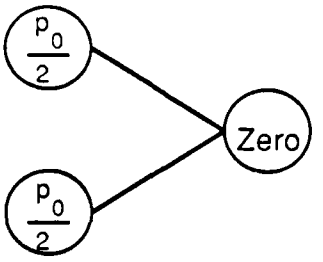
7.6 Bandwidth Computation with SHARPE

Aside from the familiar pen and paper drills for computing measures of interest, SHARPE [84] was used as a modeling tool since it allows system analysis using several different model types and permits computation of $E[Y(t)]$, $E[X(t)]$, $R(t)$, and the distribution function of $Y(\infty)$. Appendix C contains a brief description of SHARPE which was developed at Duke University.

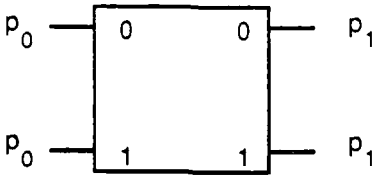
SHARPE can be used to compute the bandwidth of the SEN as it degrades over time in the presence of failures. The SEN can be modeled as a system with geometrically distributed input requests; where, on each memory request cycle, each source makes a request for some destination with a probability p. When a SE has failed, the assumption is that its output links will not be active. Thus p_i for a failed SE in stage i is zero. Further, the computation of p_i given that the two inputs $(p_{i-1,j})$ that feed a particular SE are not equal, is computed as

$$p_i = (1 - \frac{p_{i-1,0}}{2})(1 - \frac{p_{i-1,1}}{2}), \tag{7.7}$$

where j denotes the input link to a SE.



(a) SHARPE Model of 2x2 Switching Element.



(b) 2x2 Switching Element.

Figure 7.3: SHARPE Graphical Model of a Switching Element.

To use SHARPE to compute the bandwidth (BW) of a SEN when various SEs are permitted to fail, start with a single switching element. The basic idea is to model a SE as a graph with two input nodes and to compute the CDF of the time to transit the graph as the first-order statistic (or *minimum*). In Figure 7.3, observe that the distribution for each of the two input nodes is $p/2$, and the distribution for the output node is zero. Recall that p represents the probability of a request for either of the two destinations, so $p/2$ represents the probability of a request for a specific destination. Half of the time the request at an input will be for the upper output link, and half of the time it will be for the lower output link. Since queueing of requests is not allowed, if both input links simultaneously request the same output link, then only one request will be successful. The other request is dropped. The decision as to which request succeeds is random and each is equally likely. Of course,

Figure 7.3(a) models a specific output and each SE has two such outputs, so the BW for the SE is twice the BW of the specific output link.

One can justify the use of $p/2$ and the first-order statistic for obtaining p_1 by examining Figure 7.3(b). First, the first-order statistic is the probability that at least one request is made for a particular output link. This is equivalent to one minus the probability that there is no request for that output link. Now, consider input link 0 in the figure. With probability p_0, it has a request for either output link 0 or 1. Since a request for either output link is equally likely, with probability $p_0/2$ the request is for output link 1; so with probability $(1 - p_0/2)$ output link 0 is *not* requested. Similarly, if one considers input link 1, one obtains the same probability of no request for output link 0. Therefore, the combined probability of no request for output link 0 is $(1 - p_0/2)(1 - p_0/2)$ or $(1 - p_0/2)^2$. One minus this quantity is the first-order statistic as claimed. Furthermore, in the SHARPE model by using $p_0/2$ as the probability of a request made by an input link for a given output link (Figure 7.3(a)), the same value for bandwidth is obtained as in the method for computing bandwidth discussed in Chapter 4 where p_0 is the probability that a given source requests a particular destination. Since there are two outputs, the BW of the SE is twice p_1.

To model SENs of arbitrary size, simply use the inputs for the 2×2 SE in Figure 7.3(a) to represent a pair of inputs for the SEN. The output of the SE serves as the one input to the next stage and so on. So the sources of the SEN are the leaves of a full binary tree, and a single destination is the root. Figure 7.4 shows the SHARPE representation of a single destination for a 4×4 SEN.

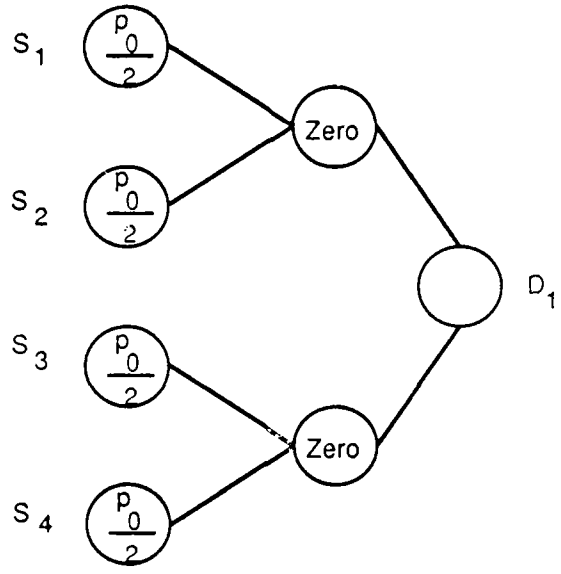


Figure 7.4: SHARPE Model of a Single Destination in a 4 × 4 SEN.

7.7 Analysis of 4 × 4 SEN

In this section and the one that follows, the SEN will be analyzed under the assumption that the interconnection network is operational as long as some source can communicate with some destination. This was definition 2 introduced in Chapter 5. This is a very loose interpretation of network reliability, but the purpose in using this definition is to show the importance that performability analysis has in establishing comparative criteria for INs. In the subsequent section, it will be shown how a variation of definition 3 can be used to solve larger problems.

A 4 × 4 SEN has 4 sources, 4 destinations, and 2 stages. Each stage has 2 SEs. Since this MIN has a total of 4 SEs, each of which can be in one of 2 states (operational or failed), one can easily model all possible states (2^4). Each configuration (combination of failed and operational SEs) in the MIN

has an associated bandwidth. Let $p_{in} = 1.0$, this means for each cycle there will be a request for some destination on each input link of the SEN.

Figure 7.5 shows the Markov chain representation of this system. It is assumed that the time-to-failure of each SE is exponentially distributed with parameter *lambda* (λ). Each state is represented by a 4-tuple where position 1 corresponds to the first SE in stage 1 and positions 2 through 4 represent the states of the SEs as shown in Figure 7.6. A 1 in position i, $1 \leq i \leq 4$, means SE_i is operational. A zero means the SE has failed.

Solving the Markov chain of Figure 7.5, produces the CDF of the time-to-failure of the 4×4 SEN, and its $MTTF$. The complementary distribution of the time-to-failure is also of interest since this is the reliability of the 4×4 SEN. However, this complementary distribution may represent more than reliability. If $r_{min} > 1$, then it also provides a gross lower bound on the performability of this SEN. This implies that the MIN works equally well (providing a performance level of one per unit time) in all states prior to failing. This value can be significantly different than the performance that *should* be expected from a MIN. The failure of one or more SEs does not necessarily imply that no source can talk to any destination. Rather, it says that the MIN is operating at a degraded level of performance. While the MIN is in some particular configuration, it can perform connections between some source-destination pairs at a certain rate; as SEs become inoperable, that rate will be diminished. So what is wanted is a measure of the cumulative work that the MIN produces prior to its failure. (In a failed state, the performance level is zero.)

Now consider the CTMC as the underlying structure-state process for the Markov reward model, and associate a reward rate (the bandwidth) with each operational state in the CTMC. Using the method described in [10], this

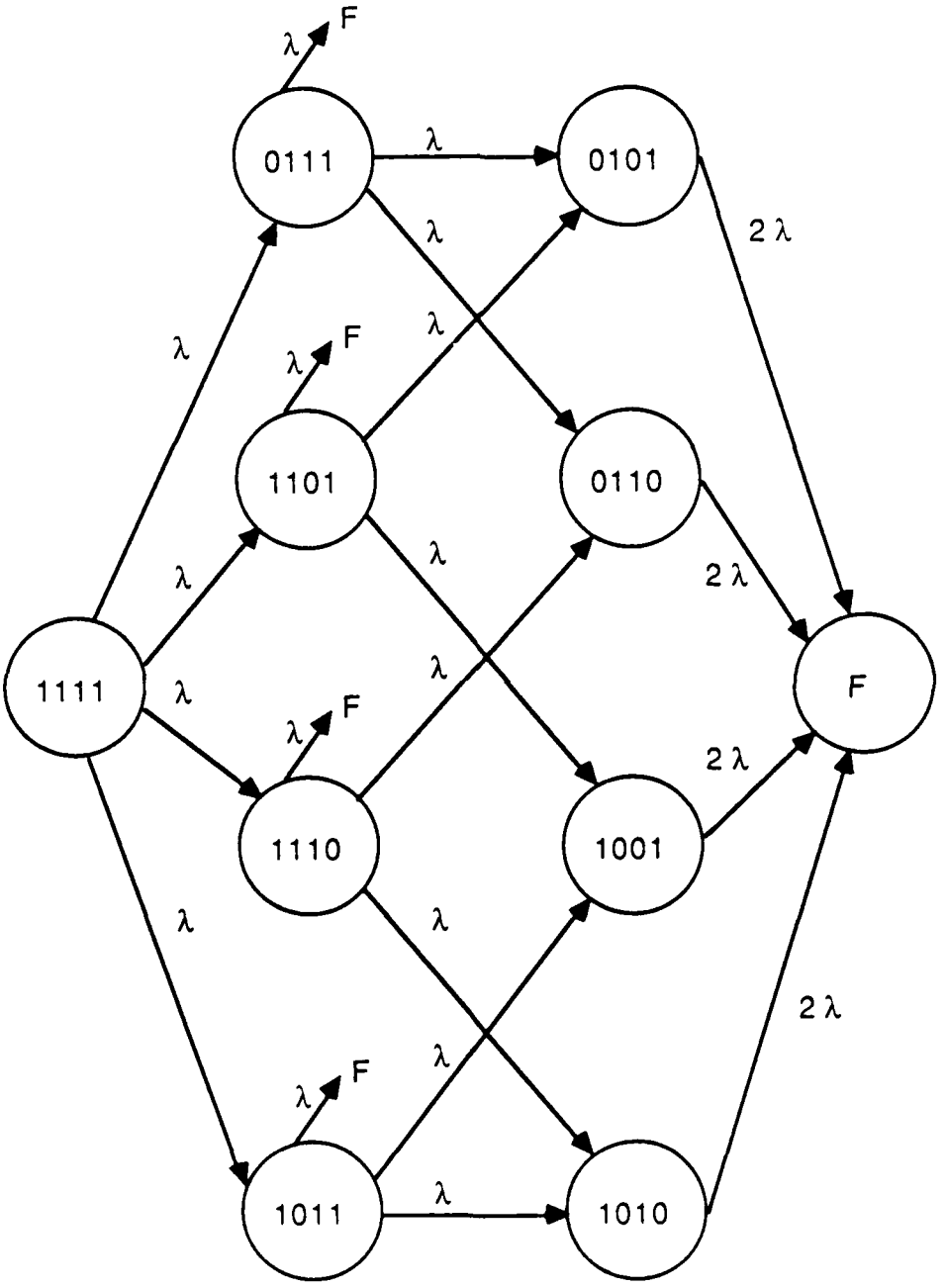


Figure 7.5: Markov Chain Representation of 4 × 4 SEN with Failure Rate λ for each SE.

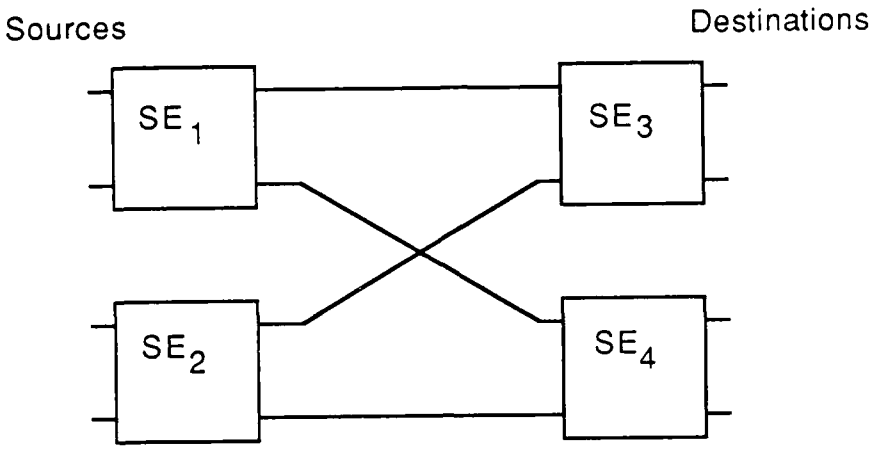


Figure 7.6: Correspondence of the SEs in the 4×4 SEN to the Markov Chain State Description.

Markov reward model can then be solved for the CDF of the accumulated reward until absorption for the 4×4 SEN.

Figure 7.7 plots the reward rate as a function of time. For this and the next two figures, $\lambda = 0.1$ and $p_{in} = 1.0$. If it is assumed that the SEs do not fail ($\lambda = 0.0$), the $r_{max} = 2.4375$ curve shows the constant upper bound for the reward rate for the 4×4 SEN. If failures ($\lambda = 0.1$) are considered, the $E[X(t)]$ curve shows the average instantaneous reward rate at time t over the interval from $t = 0$ until system failure. The reliability curve, $R(t)$, is plotted over the same interval and assumes $r_i = 1$ for the operational states and $r_i = 0$ for the failed states. Of these curves, $E[X(t)]$ properly reflects the performance level of the gracefully degradable 4×4 SEN.

Using the reward rates r_{max} and $E[X(t)]$ from Figure 7.7, one can show how the expected performability is affected. In Figure 7.8, $r_{max}t$ and $E[Y(t)]$

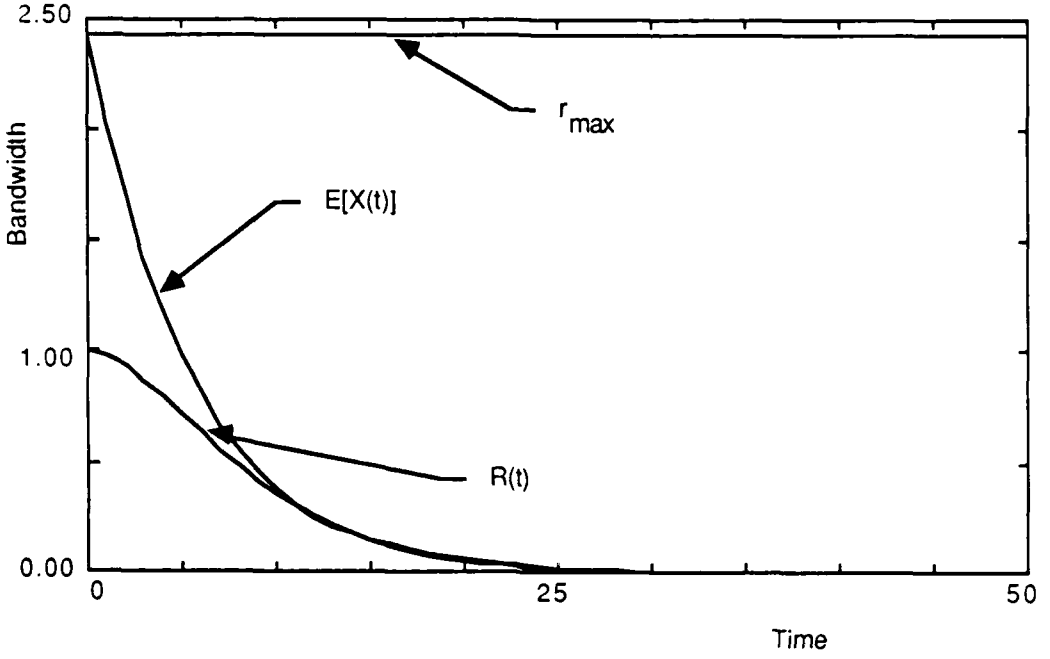


Figure 7.7: Reward Rate of the 4×4 SEN as a Function of Time.

are plotted over the time-interval for which the system is operational. The lower curve is the average performability as a function of time. As one can see, expectations about how much a given system can produce over a particular time-interval of interest is dependent on what assumptions were made about reliability and performance. The value of $E[Y(t)]$ can be bounded by the following two inequalities:

$$\begin{aligned} 0 &\leq E[Y(t)] \leq r_{max}t, \text{ and} \\ E[Y(t)] &\leq E[Y(\infty)] \leq r_{max}min\{t, MTTF\}. \end{aligned}$$

Finally, in Figure 7.9, three views of the performability of the 4×4 SEN are presented. The figure shows the complementary distribution of the system's time-to-failure using three different weighting functions. Assigning each operational state a reward rate equal to r_{max} produces an optimistic view of the SEN's performability. When each operational state is assigned the minimum

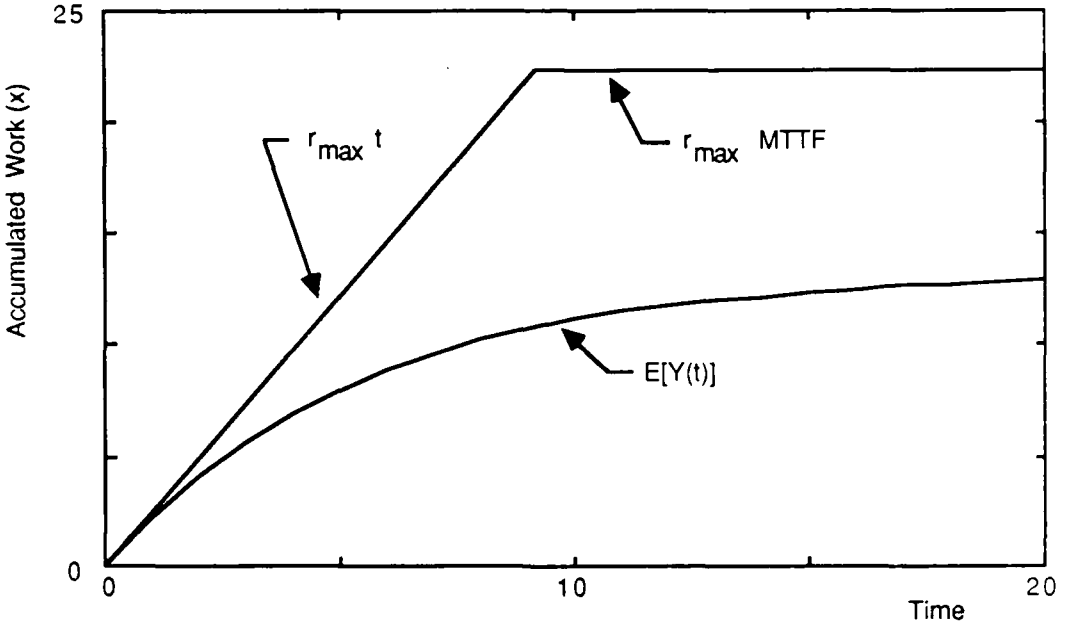


Figure 7.8: Expected Accumulated Work for the 4 × 4 SEN.

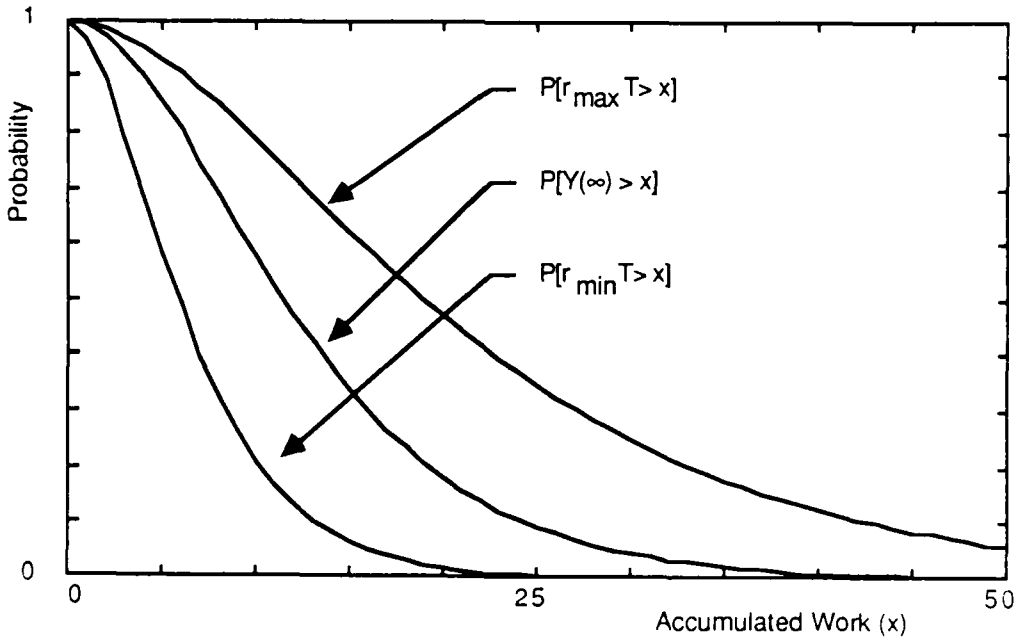


Figure 7.9: Complementary Distribution of the Accumulated Work Until System Failure for the 4 × 4 SEN.

reward rate, a pessimistic view of performability is obtained. The center curve represents the performability (performance and reliability) of the 4×4 MIN. This shows $Y(\infty) = \sum r_i H_i$ for the MIN, where the reward rate associated with each operational state is the bandwidth that the 4×4 SEN is capable of producing when in that configuration. This presents a realistic view of the SEN's performability.

To summarize, scaling the complementary distribution of the CDF, produces two views of the SEN's performability. Plotting $r_{min}T$, where a minimum reward is assumed to be accrued for each operational state, produces a lower bound on MIN performability, and plotting $r_{max}T$ provides an upper bound on MIN performability. The complementary distribution of the CDF of accumulated reward Prob$[Y(\infty) > x]$, which considers the BW as the appropriate reward rate for this degradable system, represents the probability that a specified amount of work will be completed before system failure. One can easily see the large difference that each interpretation has on performance. The particular application for which the MIN is intended will have an influence on which curve is most appropriate. For instance, in Figure 7.9, if one is only interested in whether some source can talk to some destination, then the lower curve is appropriate. If one feels that performance in a degraded condition is important, then the middle curve is appropriate. And finally, if one feels that performance in a degraded state is just as good as performance in a fully-operational state, then the upper curve is appropriate.

7.8 Analysis of 8×8 SEN

For the 4×4 SEN an explicit solution for its performability was obtained. This can be attributed to the fact that there were only 4 SEs, and hence 16 possible states. Specification of the structure-state process and the computation of the

rewards for each structure state could be accomplished with only moderate effort. The 8×8 SEN (see Figure 3.1) has 12 switching elements, so it has 4,096 distinct states. Some collapsing of states is possible, but the resulting state space is still large. For example, 333 states can be collapsed into one final state, but this still leaves more than 3700 states to deal with. This Markov reward model can still be generated and solved, but computation of the reward rate (bandwidth) associated with each state becomes tedious, and the computation of the reward rates for larger SENs would be impractical. Consider a 1024×1024 SEN for example. There are 2^{5120} possible states which is 2^{5041} times larger than Avogadro's number (6.02×10^{23}). Most people will agree that computing the bandwidth associated with each structure state is not worth the effort.

Since computation of the reward for each state is not possible, a suitable approximation for modeling the system must be found. One solution is reduction of the state space by means of truncation. Two feasible approximations are available. First, one may decide where to truncate as a function of the bandwidth. That is, truncate the state space by allowing all states with reward rates less than say 75% of the maximum bandwidth to be coalesced into an absorbing state. Or second, the truncation criterion may be a function of the number of failed switching elements. The second method has been suggested in [34], [35], and [56] as a way of reducing the state space in the analysis of other computer system models. This method has an intuitive appeal. The rationale is that when some number of switches (say k) have failed the difference between the $MTTF$ of the system with k and $k+1$ failures will be insignificant. In this thesis, the usefulness of the first approximation technique in the analysis of MINs will be demonstrated.

Table 7.2 provides a partial listing of the bandwidth computations for the 8×8 SEN in the presence of failed switching elements. One can see that at least 3 structure states (configurations) where 3 SEs have failed have a higher bandwidth than at least 2 states with only 2 failed SEs. The k versus $k+1$ total-failures approach for approximating the behavior of such SENs will truncate after all states with two failed SEs are considered, whereas the bandwidth approach will truncate in an asymmetric fashion in order to include those configurations that have more than two SEs failed yet still deliver the desired level of performance.

Consider the performability of the 8×8 SEN when its performance level in a given operational structure state is required to be equal to or greater than a specified percentage of the fully-operational SEN's bandwidth. Table 7.3 shows the number of operational structure states in the CTMC which models the 8×8 SEN where acceptable *performance* is predicated on maintaining a minimum bandwidth capability. Observe that even for 60% of maximum bandwidth, the truncated state space has only 57 operational states, whereas a CTMC based on a zero-bandwidth criterion could have up to 4,095 operational states. Hence truncation in this manner does decrease the state space. It is a practical approach, as well, because multiprocessor systems with N processors connected to N memories (or other processors) should be designed to permit some level of fault-tolerance; otherwise the complexity of the interconnection networks for such systems would make their usefulness to a broad market cost prohibitive. One way to achieve desired levels of performance is to design the system to operate in a way that permits some of the processors, memories, and components of the interconnection network to be inoperable and yet still allow an acceptable (but degraded) level of performance to be maintained. For many real-time systems, graceful degra-

Configuration	Bandwidth
All Switching Elements Operational	4.132
1 SE failed	
in stage 1	3.480
in stage 2	3.285
in stage 3	3.099
2 SE failed	
1 in stage 1 and 1 in stage 2 (case 1)	2.959
both in stage 2 (case 2)	2.719
1 in stage 2 and 1 in stage 3 (case 1)	2.676
1 in stage 1 and 1 in stage 3	2.610
1 in stage 1 and 1 in stage 2 (case 2)	2.490
both in stage 1 (case 1)	2.438
both in stage 2 (case 2)	2.438
1 in stage 2 and 1 in stage 3 (case 2)	2.252
both in stage 2 (case 1)	2.066*
both in stage 3	2.066*
3 SE failed	
one in each of the 3 stages (case 1)	2.350*
(case 2)	2.115*
(case 3)	2.089*
(case 4)	1.620

Note: Bandwidth computation assumes that the probability of a request from each source is 1.0 ($p = 1.0$).

*Indicates non-monotonicity of bandwidth as a function of the number of failed switching elements.

Table 7.2: Partial Listing of Bandwidth Capacity in the Presence of Failed Switching Elements (8×8 SEN).

Performance (% of BW)	BW	Number of Operational States	$MTTF$ ($\lambda = 0.1$)
100	4.132	1	0.8333
75	3.099	13	1.7424
70	2.892	20	1.8636
65	2.686	25	1.9242
60	2.479	57	2.3864

Note: BW computation based on average request rate $p = 1.0$ for each source.

Table 7.3: Number of States in a CTMC Where Performance is a Function of Specified Percentages of the Maximum Bandwidth (8×8 SEN).

dation is essential. By combining performance and reliability such gracefully degradable systems can be modeled to obtain a more meaningful measure of a system's effectiveness.

Assume that one wants to model the 8×8 SEN whose full CTMC has 4,096 states. Here the bandwidth computations become cost prohibitive and tedious, so the first truncation method will be used. What will such an approach reveal about the full-scale model? First, one can compute the $MTTF$ based on the specified bandwidth percentages. The mean of the system's lifetime provides the $MTTF$ for the system and is a lower bound on the its reliability. The mean of the accumulated reward provides a lower bound on performability. One way to make use of this truncation method is to iteratively compute the accumulated reward CDF for specified thresholds with progressively lower bandwidth percentages as the minimum reward rate criterion for operability. This is a variation of the tree pruning idea presented in [56]. The idea is to construct a small CTMC, using a high bandwidth cutoff, and solve for its performability. Then, if the results do not meet or exceed a specified decision criterion for the amount of work expected from a given MIN, expand the size of the CTMC by allowing transitions from the current operational structure states to new states. The bandwidths for

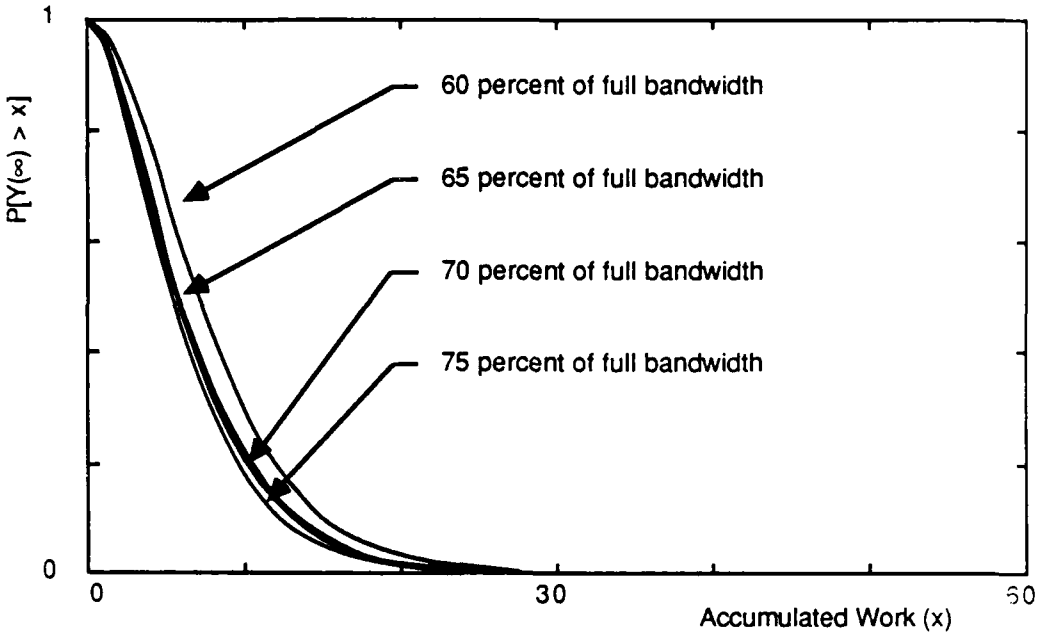


Figure 7.10: Complementary Distribution of the Accumulated Work for Specified Percentages of Full Bandwidth.

the new states are computed, and if they fall below the new reduced bandwidth requirements, they are not added to the CTMC. For those states whose bandwidth is still above the threshold, add them to the CTMC and consider transitions from these new states until all transitions from an added state fall below the threshold. The performability model is then solved, and its results are checked. This procedure is continued until it is determined if the system under consideration will meet the work standard. In the extreme, one must build a complete CTMC for the system. The same idea can be used for MINs with a specified minimum bandwidth. Starting from the full bandwidth and moving toward the specified minimum in an iterative fashion. Figure 7.10 shows the computation of the complementary distribution for the accumulated reward for 75, 70, 65, and 60 percent of full bandwidth for the 8×8 SEN.

7.9 Summary

In this chapter, it was shown that performability, a combined measure of performance and reliability, is a more useful measure than either of its components for determining the "goodness" of a multistage interconnection network. It was also demonstrated that for MINs of size 8×8 and larger, truncation of the state space as a function of bandwidth is a useful approximation technique. Of current interest is finding an algorithmic way of computing all possible bandwidths and/or finding a method of getting tight bounds on the performability of the MINs when approximation techniques are used for the analysis.

Chapter 8

Analysis of a Multiprocessor System

8.1 Introduction

Traditional evaluation techniques for multiprocessor systems use Markov chains and Markov reward models to compute measures such as mean time to failure, reliability, performance, and performability. In this chapter, parametric sensitivity analysis is performed on Markov models to determine their sensitivity to changes in the component failure rates. Using such analysis, one can guide system optimization, identify parts of a system model sensitive to error, and find system reliability and performability bottlenecks.

First performance, reliability, and performability measures for models of three architectural alternatives of a multiprocessor system are considered. Then, for these models, the sensitivity of the mean time to failure, unreliability, and performability to changes in component failure rates are examined. The sensitivities are used to identify bottlenecks in the three system models.

The MultiProcessor System (MPS) considered consists of 16 processors (Ps), 16 shared-memory modules (Ms), and an interconnection network (IN) for communication between the processors and the memories. The crossbar or the Omega network are the assumed interconnection network, and

two implementations of the crossbar are considered. The Omega network is equivalent to a SEN with 4×4 switching elements.

Closed-form combinatorial expressions, Markov chains, and Markov reward models are used in the analysis. The use of state lumping permits the computation of reliability and performability measures for a system with 16 processors, 16 memories, and an Omega network.

It is shown that both the requirement for processor-memory connectivity and the metric for comparison influence the preference for one architectural alternative over the others.

In the performance domain, this chapter builds upon and extends the work by Bhandarkar [12]; in the reliability domain, it builds upon the work of Siewiorek [86] and Siewiorek *et al.* [87]; and in the performability domain, it builds upon the earlier work by Beaudry [10], Meyer [60], and Smith *et al.* [90].

8.2 MPS Model Descriptions

Consider a MPS which consists of 16 processors (Ps), 16 shared memories (Ms), and an interconnection network (IN) that connects the processors to the memories. Three approaches to modeling the interconnection network will be considered.

First, the interconnection network may be modeled as one large switch. In this case, the IN is simply a crossbar switch, and the multiprocessor system is the well-known C.mmp system (see Figure 8.1).

Second, a more detailed model of the crossbar switch can be developed as shown in Figure 8.2 where the crossbar is considered to be composed of sixteen 1×16 demultiplexers and sixteen 16×1 multiplexers. In this arrangement,

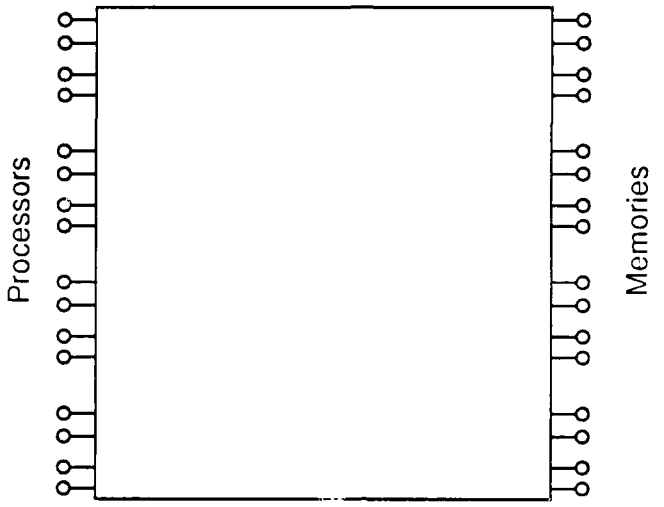


Figure 8.1: Multiprocessor System Using a Crossbar Switch as a Single Component Interconnection Network.

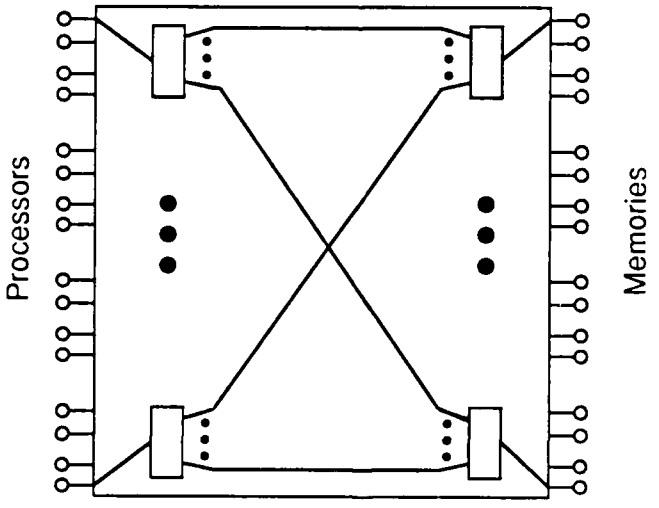


Figure 8.2: Multiprocessor System Using a Crossbar Switch Composed of Multiplexers/Demultiplexers as the Interconnection Network.

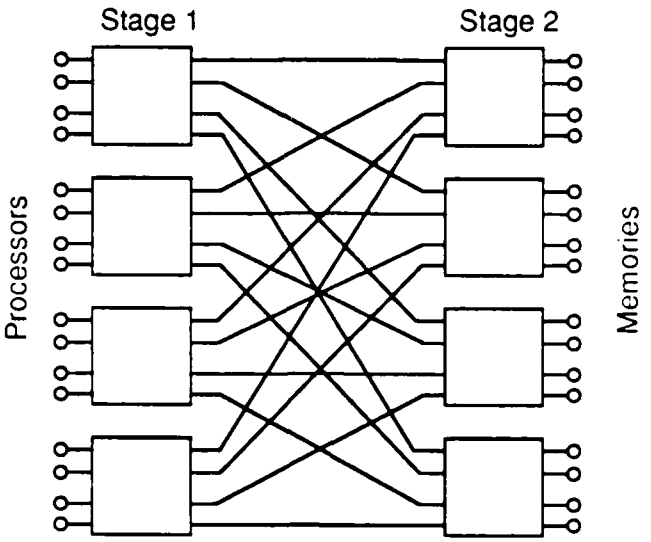


Figure 8.3: Multiprocessor System Using an Omega Network with 4×4 Switching Elements as the Interconnection Network.

each processor is connected to a demultiplexer and each memory is connected to a multiplexer.

The third model to be considered implements the IN with an Omega network constructed from eight 4×4 switching elements (SEs). This network has two stages and is a reasonable alternative to a crossbar implementation of the interconnection network since the complexity of the crossbar is $O(N^2)$ whereas that of the Omega network is $O(N \log N)$ where N is both the number of inputs and the number of outputs to the network. The MPS using the Omega network as its interconnection network is shown in Figure 8.3.

Each of the three MPS architectures will be referred to in a way that characterizes its IN. The three architectures are:

SYS_s which assumes that the interconnection network is a single component.

SYS_d considers a detailed model of the crossbar switch; it assumes the interconnection network is composed of individual demultiplexers and multiplexers.

SYS_Ω MPS using an Omega network with 4×4 switching elements.

The *switch-fault* model will be used for the subsequent analysis. As mentioned before, the primary assumption in this model is that a component being represented in a particular model is an atomic structure, and therefore, the failure of any device which is a part of this structure will cause a total failure of the component. Partial or degraded operation of the component is not considered. For example, if a gate in a multiplexer malfunctions, then the multiplexer is considered inoperative and its output is ignored.

Markov models will be used as the principal modeling tool for analyzing the three MPS architectures. Events that decrease the number of operational components are associated with failure. When a component of the system fails, a recovery action must be taken (e.g., shutting down a failed processor so that it does not fill memories with spurious data), or the whole system will fail and enter a failure state F. The probability that the recovery action is successfully completed is known as the coverage [17]. In general, the analysis in this chapter will assume perfect coverage so system failure occurs as a result of the accumulation of component failures. It has been shown, however, that coverage is very important in non-repairable systems [16,4]. This is because for degradable systems operating in an environment with imperfect coverage, the notion of failure may be the result of the cumulative effects of component failures *or* as the disastrous result of a coverage failure. The extension of the analysis to incorporate imperfect coverage is straight-forward, and its effect on reliability and the complementary distribution of accumulated reward until

system failure will be considered in the latter part of the section on numerical results.

8.3 Measures of Interest

In this section, a brief review of the performance, reliability, and performability measures used for analyzing the three MPS architectures will be discussed. Then, methods to compute parametric sensitivities will be presented.

8.3.1 Performance

The average number of busy memories (memory bandwidth) will be used as the performance level (also called the reward rate) for a particular system configuration. This is an appropriate choice of performance metric for the MPS since the efficiency of the system will be limited by the ability of the processors to randomly access the available memories.

In the case of a crossbar switch, contention for the memories occurs at the memory ports since the crossbar switch is non-blocking. But, in the case of the Omega network, contention occurs inside the interconnection network as well since this is a blocking network. That is, if two or more processors compete for the same output link of a SE, only one request will be successful and the remaining requests will be dropped.

Over time, components of the MPS can be expected to fail, and as a result, the performance of the system can be expected to decrease. To determine the performance of the crossbar, the model developed by Bhandarkar [12] to obtain the average number of busy memories will be used, and an extension of the performance model in [68] will be used for the Omega network. Also, the assumptions stated in [68] for the analysis of circuit-switched networks will be used.

generally, however, Markov chains and Markov Reward M used.

The evolution of a degradable system through various different sets of operational components can be represente continuous-time Markov chain (CTMC), $\{Z(t),\ t \geq 0\}$ $\Psi = \{1, 2, \ldots, k\}$. For each $i, j \in \Psi$, let q_{ij} be the transiti i to state j, and define

$$q_{ii} = -\sum_{\substack{j=1 \\ j \neq i}}^{k} q_{ij}.$$

Then, $Q = [q_{ij}]$ is the k by k ***transition rate matrix.*** Let P_i $i]$ be the probability that the system is in state i at time t. T probability row-vector $\underline{P}(t)$ can be computed by solving a r equation [98],

$$\dot{\underline{P}}(t) = \underline{P}(t)Q.$$

Methods for computing $\underline{P}(t)$ are compared in [80].

The state space can be partitioned into two sets: UP, the s states, and DOWN, the set of failure or down states. If a are absorbing failure states, then ***system reliability*** can be ol state probabilities,

$$R(t) = \sum_{i \in \mathrm{UP}} P_i(t).$$

Associated with each state of the CTMC is a ***reward rate*** the performance level of the system in that state. The CTMC rates are combined to form a ***Markov reward model*** [40]. Each a different system configuration. Transitions to states with rates (lower performance levels) are component failure tra repairable systems, transitions to states with higher perform

repair transitions. The choice of a performance measure for determining reward rates is a function of the system to be evaluated. For an interconnection network (IN), the appropriate measure is bandwidth.

Let r_i denote the reward rate associated with state i, and call $\underline{r}$ the *reward vector*. The reward rate of the system at time t is given by the process $X(t) = r_{Z(t)}$. The expected reward rate at time t is

$$E[X(t)] = \sum_i r_i P_i(t).$$

This quantity is also called the *computation availability* [10].

If $Y(t)$ denotes the amount of *accumulated reward* (the amount of work done) by a system during the interval $(0, t)$, then

$$Y(t) = \int_0^t X(u)du. \tag{8.4}$$

Furthermore, using bandwidth to construct the reward vector, $Y(t)$ represents the number of requests that the IN is capable of satisfying by time t. The *expected accumulated reward* is

$$E[Y(t)] = E[\int_0^t X(u)du] = \sum_i r_i \int_0^t P_i(u)du. \tag{8.5}$$

In order to compute $E[Y(t)]$, let $L_i(t) = \int_0^t P_i(u)du$. Then, the row vector $\underline{L}(t)$ can be computed by solving the system of differential equations:

$$\dot{\underline{L}}(t) = \underline{L}(t)Q + \underline{P}(0). \tag{8.6}$$

Methods of solving this system of equations are discussed in [82].

A special case of the expected accumulated reward is the *mean time to failure* ($MTTF$). The $MTTF$ of a MPS is defined as

$$MTTF = \int_0^\infty R(t)dt. \tag{8.7}$$

The $MTTF$ is a special case of $E[Y(\infty)]$, with reward rate zero assigned to all DOWN states (which are assumed to be absorbing) and reward rate one assigned to all UP states. To compute $MTTF$, solve for $\underline{\tau}$ in

$$\underline{\tau}\hat{Q} = -\hat{\underline{P}}(0), \tag{8.8}$$

where $\hat{\underline{P}}(0)$ is the partition of $\underline{P}(0)$ corresponding to the UP states only. The matrix $\hat{Q}$ is obtained by deleting the rows and columns in Q corresponding to DOWN states. Any linear algebraic system solver can be used to solve this system of equations. Although one might like to use direct methods like Gaussian elimination; for large, sparse models, iterative methods are more practical [93]. The matrix $-\hat{Q}$ is a non-singular, diagonally-dominant M-matrix. Thus, the use of an iterative method such as Gauss-Seidel, SOR, or optimal SOR to solve equation (8.8) is guaranteed to converge to the solution [101]. Then,

$$MTTF = \sum_{i \in UP} \tau_i. \tag{8.9}$$

In case the CTMC has one or more absorbing states, it is useful to compute the *accumulated reward until absorption*, $Y(\infty)$. The distribution function of $Y(\infty)$ can be computed by constructing another CTMC with the transition rate matrix Q' so that $q'_{ij} = q_{ij}/r_i$ for $r_i > 0$ and solving for the distribution of the time to absorption for the new CTMC [10]. $E[X(t)]$, $E[Y(t)]$, and the distribution of $Y(\infty)$ are the performability measures that will be used to compare the three alternative MPS architectures.

8.3.3 Parametric Sensitivity Analysis

The results obtained from a model are sensitive to many factors. For example, the effect of a change in distribution on a stochastic model is often considered. Here, attention is concentrated on parametric sensitivity analysis, a technique to compute the effect of changes in the rate constants of a

Markov model on the measures of interest [82]. Parametric sensitivity analysis helps: (1) guide system optimization, (2) find reliability, performance, and performability bottlenecks in the system, and (3) identify the model parameters that could produce significant modeling errors.

One approach to parametric sensitivity analysis is to use upper and lower bounds on each parameter in the model to compute optimistic and conservative bounds on system reliability [92]. The approach in this chapter is to compute the derivative of the measures of interest with respect to the model parameters [35,91]. A bound on the perturbed solution can then be computed with a simple Taylor series approximation.

It is assumed that the transition rates q_{ij} are functions of some parameter λ. Then given the value of λ, one wants to compute the derivative of various measures with respect to λ (e.g., $\partial P_i(t)/\partial\lambda$). If $\underline{S}(t)$ is the row vector of the sensitivities $\partial P_i(t)/\partial\lambda$, then from (8.3) one obtains

$$\dot{\underline{S}}(t) \;=\; \underline{S}(t)Q + \underline{P}(t)V \tag{8.10}$$

where V is the derivative of Q with respect to λ. Assuming the initial conditions do not depend on λ,

$$\underline{S}(0) \;=\; \frac{\partial \underline{P}(0)}{\partial\lambda} \;=\; \lim_{t\to 0}\frac{\partial \underline{P}(t)}{\partial\lambda} \;=\; \underline{0}.$$

Then (8.3) and (8.10) can be solved simultaneously using,

$$[\dot{\underline{P}}(t), \dot{\underline{S}}(t)] \;=\; [\underline{P}(t), \underline{S}(t)]\begin{bmatrix} Q & V \\ 0 & Q \end{bmatrix} \quad , \quad [\underline{P}(0), \underline{S}(0)] \;=\; [\underline{P}_0, \underline{0}]. \tag{8.11}$$

Let η be the number of non-zero entries in Q, and let η_s be the number of non-zero entries in V.

For acyclic models, an efficient algorithm that requires $O(2\eta+\eta_s)$ floating-point operations (FLOPS) is discussed in [58]. For more general models

with cycles, one can use an explicit integration technique like Runge-Kutta. The execution time of explicit methods like Runge-Kutta is $O((2\eta + \eta_s)(q + v)t)$ FLOPS, where $q = \max_i |q_{ii}|$ and $v = \max_i |v_{ii}|$. To solve (8.11) with Uniformization [37], choose $q > \max_i |q_{ii}|$ and let $Q^* = Q/q + I$. Then

$$\underline{S}(t) = \frac{\partial}{\partial \lambda} \sum_{i=0}^{\infty} \underline{P}(0)(Q^*)^i e^{-qt} \frac{(qt)^i}{i!} = \sum_{i=0}^{\infty} \underline{\Pi}(i)' e^{-qt} \frac{(qt)^i}{i!}, \qquad (8.12)$$

where

$$\underline{\Pi}(i)' = \frac{\partial}{\partial \lambda} \underline{\Pi}(i) = \frac{\partial}{\partial \lambda} (\underline{\Pi}(i-1)Q^*) = \underline{\Pi}(i-1)'Q^* + \underline{\Pi}(i-1)\frac{\partial}{\partial \lambda} Q^*, \quad (8.13)$$

and

$$\underline{\Pi}(i) = \underline{\Pi}(i-1)Q^* \quad , \quad \underline{\Pi}(0) = \underline{P}(0). \qquad (8.14)$$

If the CTMC's initial conditions do not depend on λ, then $\underline{\Pi}'(0) = \underline{0}$. Also note that $\partial Q^*/\partial \lambda = V/q$. With a sparse matrix implementation, Uniformization requires $O((2\eta + \eta_s)qt)$ FLOPS. Both Runge-Kutta's and Uniformization's performance degrades linearly as q (or v) grows. Problems with values of q that are large relative to the length of the solution interval are called *stiff*. Large values of q (and v) are common in systems with repair or reconfiguration. An attractive alternative for such stiff problems is an implicit integration technique with execution time $O(2\eta + \eta_s)$ [80].

The sensitivity of $E[X(t)]$ can be derived from the sensitivities of the state probabilities

$$\frac{\partial E[X(t)]}{\partial \lambda} = \frac{\partial}{\partial \lambda} \sum_{i \in \Psi} r_i P_i(t) = \sum_{i \in \Psi} \frac{\partial r_i}{\partial \lambda} P_i(t) + \sum_{i \in \Psi} r_i S_i(t). \qquad (8.15)$$

Similarly, the sensitivity of $E[Y(t)]$ can be derived by differentiating equation (8.5),

$$\frac{dE[Y(t)]}{d\lambda} = \frac{\partial}{\partial \lambda} \sum_{i \in \Psi} r_i L_i(t) = \sum_{i \in \Psi} \frac{\partial r_i}{\partial \lambda} L_i(t) + \sum_{i \in \Psi} r_i \int_0^t S_i(u)du. \qquad (8.16)$$

As in the instantaneous measures case, methods for computing the cumulative state probability sensitivity vector, $\int_0^t \underline{S}(u)du$, include numerical integration, the ACE algorithm for acyclic models [58], and Uniformization.

For the special case of mean time to failure, differentiate equation (8.8) and then solve for $\underline{s}$,

$$\underline{s}\hat{Q} = -\underline{\tau}\frac{\partial \hat{Q}}{\partial \lambda} \tag{8.17}$$

where $\underline{\tau}$ is the solution obtained from equation (8.8). Then,

$$\frac{\partial MTTF}{\partial \lambda} = \sum_{i \in UP} \frac{\partial \tau_i}{\partial \lambda} = \sum_{i \in UP} s_i. \tag{8.18}$$

This linear system can be solved using the same algorithms used to solve equation (8.8).

8.3.4 Interpretation of Parametric Sensitivities

Having computed the derivative of some measure, say $MTTF$, with respect to various system parameters λ_i, there are at least three distinct ways to use the results. The first application is to provide error bounds on the solution when given bounds on the input parameters. Assume that each of the parameters λ_i is contained in an uncertainty interval of width $\Delta\lambda_i$. Then an uncertainty interval $\Delta MTTF$ can be approximately determined by

$$\Delta MTTF \simeq \sum_i \Delta\lambda_i \frac{\partial MTTF}{\partial \lambda_i}. \tag{8.19}$$

A second use of parametric sensitivities is in the identification of portions of a model that need refinement. There is some cost involved in reducing the size of the intervals $\Delta\lambda_i$ since it requires taking additional measurements or performing more detailed analysis. Assume the cost (or time) of reduction in $\Delta\lambda_i$ is proportional to $\Delta\lambda_i/\lambda_i$ and let

$$I = argmax_i \left| \lambda_i \frac{\partial MTTF}{\partial \lambda_i} \right|, \tag{8.20}$$

where $argmax_i|x_i|$ denotes the value of i that maximizes x_i. Then, refining parameter I is the most cost-effective way to improve the accuracy of the model.

A third application of parametric sensitivities is system optimization and bottleneck analysis. Assume that there are N_i copies of component i in the system and that the failure rate of component i is λ_i. Furthermore, assume the cost of the i^{th} subsystem is given by some function $c_i N_i \lambda_i^{-\alpha_i}$. Define the optimization problem:

$$\begin{aligned} &Maximize: && MTTF \\ &Subject\ To: && \sum_i c_i N_i \lambda_i^{-\alpha_i} \leq COST. \end{aligned} \tag{8.21}$$

Using the method of Lagrange multipliers [5], the optimal values of λ_i satisfy:

$$\frac{\lambda_i^{\alpha_i+1}}{c_i N_i \alpha_i} \frac{\partial MTTF}{\partial \lambda_i} = constant. \tag{8.22}$$

Let

$$I^* = argmax_i \left| \frac{\lambda_i^{\alpha_i+1}}{c_i N_i \alpha_i} \frac{\partial MTTF}{\partial \lambda_i} \right|. \tag{8.23}$$

Then, the most cost-effective point to make an incremental investment is in subsystem type I^*. In other words, the system bottleneck from the $MTTF$ point of view is subsystem I^*. In the numerical examples, this definition of bottleneck will be used. For convenience, also assume that $c_i = \alpha_i = 1$ for all i although other cost functions could be used. Later, in the numerical results section, these results are compared with those obtained using the second scaling approach.

8.4 Model Development

Before developing the Markov models for the three MPS architectures, closed-form combinatorial expressions are derived for obtaining measures of interest

for SYS_s and SYS_d. Such expressions are desirable from an analytic point of view, and in this section, closed-form combinatorial expressions for the reliability, $MTTF$, $E[X(t)]$, and $E[Y(t)]$ are derived for two of the three models. Then, Markov reward models are developed for all three architectures.

8.4.1 Combinatorial Approach

Combinatorial expressions are appealing for system analysis since computation of the measures of interest is often straightforward. In this section, closed-form expressions for the reliability and performability measures of interest are derived for SYS_s and SYS_d.

Let r_{ij} be the reward rate associated with the MPS having i processors and j memories operational (r_{ij} is obtained from equation(8.1)), let R_p be the reliability of a processor, and let R_m be the reliability of a memory. Also, let R_α be the reliability of the switch in SYS_s and let R_β be the reliability of a demultiplexer/multiplexer in SYS_d. Then the reliability of SYS_s can be expressed as

$$R_s(t) = \left(\sum_{i=K}^{N} \binom{N}{i} R_p^i (1-R_p)^{N-i}\right)\left(\sum_{j=K}^{N} \binom{N}{j} R_m^j (1-R_m)^{N-j}\right)(R_\alpha), \tag{8.24}$$

and the reliability of SYS_d is

$$\begin{aligned} R_d(t) \;=\; & \left(\sum_{i=K}^{N} \binom{N}{i} (R_p R_\beta)^i (1-(R_p R_\beta))^{N-i}\right) \\ & \left(\sum_{j=K}^{N} \binom{N}{j} (R_m R_\beta)^j (1-(R_m R_\beta))^{N-j}\right). \end{aligned} \tag{8.25}$$

Equations (8.24) and (8.25) can be rewritten by a power series expansion of factors like $(1-R_p)^{N-i}$. Then, by multiplying through and collecting terms,

one obtains:

$$R_s(t) = \sum_{i=K}^{N}\sum_{j=K}^{N} a_{ij;s} R_p^i(t) R_m^j(t) R_\alpha(t), \text{ and} \tag{8.26}$$

$$R_d(t) = \sum_{i=K}^{N}\sum_{j=K}^{N} a_{ij;d} R_p^i(t) R_m^j(t) R_\beta^{i+j}(t). \tag{8.27}$$

Assuming the component lifetimes are independent exponentially distributed random variables; $R(t)$, mean time to failure $(MTTF)$, $E[X(t)]$, and $E[Y(t)]$ are derived for these systems. Let λ be the processor failure rate, γ be the memory failure rate, δ_s be the failure rate of the IN in SYS_s, and δ_d be the failure rate of a demultiplexer/multiplexer. Then for SYS_s, the measures of interest are derived as:

$$R_s(t) = \sum_{i=K}^{N}\sum_{j=K}^{N} a_{ij;s} e^{-(i\lambda+j\gamma+\delta_s)t}, \tag{8.28}$$

$$MTTF_s = \sum_{i=K}^{N}\sum_{j=K}^{N} \frac{a_{ij;s}}{i\lambda+j\gamma+\delta_s}, \tag{8.29}$$

$$E[X(t)]_s = \sum_{i=K}^{N}\sum_{j=K}^{N} r_{ij} a_{ij;s} e^{-(i\lambda+j\gamma+\delta_s)t}, \text{ and} \tag{8.30}$$

$$E[Y(t)]_s = \sum_{i=K}^{N}\sum_{j=K}^{N} \frac{r_{ij} a_{ij;s}\left(1-e^{-(i\lambda+j\gamma+\delta_s)t}\right)}{i\lambda+j\gamma+\delta_s}. \tag{8.31}$$

And the expressions for SYS_d are:

$$R_d(t) = \sum_{i=K}^{N}\sum_{j=K}^{N} a_{ij;d} e^{-(i\lambda+j\gamma+(i+j)\delta_d)t}, \tag{8.32}$$

$$MTTF_d = \sum_{i=K}^{N}\sum_{j=K}^{N} \frac{a_{ij;d}}{i\lambda+j\gamma+(i+j)\delta_d}, \tag{8.33}$$

$$E[X(t)]_d = \sum_{i=K}^{N}\sum_{j=K}^{N} r_{ij} a_{ij;d} e^{-(i\lambda+j\gamma+(i+j)\delta_d)t}, \text{ and} \tag{8.34}$$

$$E[Y(t)]_d = \sum_{i=K}^{N}\sum_{j=K}^{N} \frac{r_{ij} a_{ij;d}\left(1-e^{-(i\lambda+j\gamma+(i+j)\delta_d)t}\right)}{i\lambda+j\gamma+(i+j)\delta_d}. \tag{8.35}$$

Often, when closed-form expressions for the reliability of a system are given, only static values for system reliability are presented. Closed-form expressions for other measures such as $MTTF$, $E[X(t)]$, and $E[Y(t)]$ can also be derived from a combinatorial model. In practice, however, expanding expressions like (8.24) and (8.25) to obtain coefficients like $a_{ij;s}$ and $a_{ij;d}$ can cause numerical difficulties.

8.4.2 Markov Models of the Architectures

In the case where the IN is viewed as a single component, construction and solution of the Markov chain to analyze the reliability and performability measures of the MPS is tractable, and it has been done in [88]. Each structure state of the Markov reward model is specified by a tuple pair (i, j) indicating the number of operational processors and memories, respectively.

If the interconnection network is modeled in more detail, the crossbar switch can be thought of as a combination of multiplexers and demultiplexers. In this case, a further refinement of the structure-state process can be made with respect to the IN. The failure rate of each processor and memory can be adjusted to account for the failure of the particular demultiplexer/multiplexer to which it is connected. Also, if a multiplexer is associated with each memory and a demultiplexer is associated with each processor, then the same Markov chain that was developed for SYS_s can be used by simply adjusting the failure rates of the processors and memories to account for their associated demultiplexers/multiplexers.

However, the size of Markov chain for the case of 4×4 SE components in the IN becomes a problem. The Markov chain must account for the failure behavior of the processors, memories, and SEs to which they are connected. If a state description explicitly accounts for the operational status of each

processor, memory, and SE, then a 40-tuple would be required, and there may be as many as 2^{40} states depending on the failure criteria used for the entire system.

If one examines Figure 8.3 more carefully, one will see that an Omega network without intermediate stages, as is the case for this MPS, has a great deal of symmetry. So the state description can be accomplished with an 8-tuple. The initial state is (44444444) where position i $(1 \leq i \leq 4)$ represents the number of functioning processors connected to an operational SE in position i. Similarly for the memories where $5 \leq i \leq 8$. One can see that this Markov chain embodies the concept of *bulk failures*. That is for a given i, either a processor (memory) may fail and the value at position i will decrease by one, or a SE may fail and the value at position i will become zero.

The number of states in a Markov chain using this representation may be as large as 5^8. If the MPS is determined to be operational as long as 12 processors can access 12 memories ($K = 12$), then this method of defining the states will produce a Markov chain with 4901 states, 26739 transitions, and a file requiring 1.5 megabytes of storage. While solving Markov chains of this size is tractable; for $K = 4$, the solution of a Markov chain with more than 64000 states is required. This is not practical.

What is needed is an efficient way to produce a reduced-state representation of the same system. There are three common approaches to the state reduction: *lumping*, *aggregation*, and *truncation*. *Lumping* will be discussed in the next section. For a discussion of state aggregation, see [15]. Truncation is discussed in [35].

State-Space Reduction

One approach to state-space reduction is to observe that there is an equivalence between a Markov chain representation of a system and a *Mealy machine*, which is a deterministic finite automata. That is, each state and arc has a label, and a transition function is easily derived using this information. Now, as an implication of the Myhill-Nerode Theorem, there exists an unique *minimum state* machine which can be constructed from the original machine (Markov chain). In [39], an algorithm for doing this construction is presented. The algorithm has $O(\kappa k^2)$ time complexity where k denotes the number of states and κ denotes the size of the input alphabet. While the time complexity of this algorithm appears to make this a viable technique for the current problem, there are several drawbacks with the actual implementation. For example, for $K = 12$ the Markov chain has 4901 states when the Omega network is used to represent the IN. Since there are 8 SEs, 16 Ps, and 16 Ms, the size of the input alphabet is $4 \times 6 \times 6 = 144$ for the current problem. This means that $O(144 \times 4901^2) = O(3.5 \times 10^9)$ steps are required to obtain the minimum-state Markov chain. Also note that the Markov chain must be completely constructed before one can do the reduction. Reducing the state space in this manner is referred to as "state lumping" and is explained in [45].

A more efficient approach is to "lump" the states as the Markov chain is constructed, thus avoiding the execution of a reduction algorithm after the Markov chain has been generated. In the case of the Omega network with two stages, this is possible by exploiting the symmetry and connectivity of the MPS. Consider Figure 8.3 again. Observe that a particular memory's view of the system is confined to the specific SE to which it is connected. Further observe that this SE's view of the system encompasses the status

of all four of the SEs in the first stage where the processors are connected. Each SE in the first stage can be up or down; and if it is up, it can have from zero to four functioning processors connected to it. The particular positions of the functioning processors is not important to the outputs of the SE to which they are connected (under the uniform access assumption [68]). Hence, tuples (34444444), (43444444), (44344444), and (44434444) are equivalent. Furthermore, the states to which these states transition can also be grouped into equivalence classes.

Generating the Markov chain in this fashion, one only has to consider one such tuple for each equivalence class in a breadth-first construction (BFC) of the Markov chain. Only one member of each class is added to the BFC queue and the transition rates from this representative state are adjusted to account for the lumping. (Note that the number of equivalence classes for a reliability model may be smaller than the number for a corresponding Markov reward model because the performance level of each state is ignored in the reliability model.) If the performance level for each state is considered before lumping, then the 4901 state Markov chain can be reduced to 145 states. This makes the development and solution of Markov chains with a lower connectivity requirement significantly easier.

Extension of Lumpability Requirements

In this section, the conditions for lumpability are extended to Markov reward models. The essential observation is that the underlying structure-state process of a Markov reward model can be suitably modified (transformed) to produce the same results as the Markov reward model. The reward rates associated with the structure states in the original process serve as the modifying variable.

Let $\mathbf{A} = \{\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_s\}$ be a partition of the k states of a Markov chain. Then a new process where each $\mathbf{A}_i \subseteq \mathbf{A}$ is a state, is termed a *lumped process*. Let $q_{i\mathbf{A}_j} = \sum_{m \epsilon \mathbf{A}_j} q_{im}$. Then $q_{i\mathbf{A}_j}$ represents the transition rate from state i into set $\mathbf{A}_j$.

The theorem in [45] is extended to the lumping of Markov reward models in the following corollary.

Corollary 1 *For a Markov reward model, a necessary and sufficient condition for lumpability with respect to a partition* $\mathbf{A} = \{\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_s\}$ *of the underlying structure-state process is that for every pair of sets* $\mathbf{A}_i$ *and* $\mathbf{A}_j$, $q_{m\mathbf{A}_j}/r_m$, $r_m > 0$, *have the same value for every structure state* m *in* $\mathbf{A}_i$.

Proof: From [10], every Markov reward model can be transformed into an equivalent Markov chain by an appropriate adjustment of the transition rates (q_{ij}) in the underlying structure-state process of the Markov reward model.

The resulting Markov chain is lumpable if it satisfies the theorem in [45]. In effect, the transition rates from a state in the original chain have been scaled by the reciprocal of the reward rate associated with that state (i.e., $q'_{ij} = q_{ij}/r_i$).

8.5 Numerical Results

The reliability of a system without repair can be determined from the solution of a general Markov reward model by simply assigning a reward rate of one to each operational state and a reward rate of zero to each failure state. This measure assumes that any operational configuration is as good as any other. However, the bandwidth that a multiprocessor system is able to achieve in a particular configuration is a more appropriate reward rate than the simple zero-one choice of the reliability model.

In this section, the three architectural alternatives are compared using 'pure' performance, assuming no failures; using a 'pure' reliability model that ignores performance differences; and then using combined measures — $E[X(t)]$, $E[Y(t)]$, and the complementary distribution of $Y(\infty)$. Also, for each model, the sensitivities of $MTTF$, $R(t)$, and $E[X(t)]$ to changes in the component failure rates are computed.

First, some single-valued measures of network performance and reliability are considered. Then, time-dependent system reliability and its sensitivity are presented. Next, the performability measures are examined, and the sensitivity of $E[X(t)]$ to changes in the component failure rates is analyzed. Finally, the effect of imperfect coverage on the reliability and the complementary distribution, $y^c(x)$, of accumulated reward, $Y(\infty)$, on the three MPS architectures will be analyzed. For notational convenience, $y^c(x) \equiv y^c(x, \infty)$.

In order to obtain the numerical results in this section, the Markov models were generated using the approach described in Section 8.4.2. To compute $y^c(x)$, the MRM was transformed into a CTMC using Beaudry's algorithm [10]. Then, the HARP package [28] was used to solve for the system reliability and $y^c(x)$. The Markov chain solvers developed by Reibman in [81] were used to solve for $E[X(t)]$, $E[Y(t)]$, and the sensitivities of the reliability and expected reward rate to changes in the component failure rates.

Failure data for the C.mmp system [86] will be used. By a parts count method, Siewiorek determined the failure rates per hour for the components to be:

	Processor	Memory	Switch
Failure Rates :	$\lambda = 0.0000689$	$\gamma = 0.0002241$	$\delta_s = 0.0002024$.

Like Siewiorek, throughout this section, component lifetime distributions are assumed to be exponentially distributed.

Gate count will be used as the basis for determining the failure rates of the components of the IN. From [47], an $n \times n$ crossbar switch requires $4n(n-1)$ gates where n is the number of inputs and outputs. An $n \times 1$ multiplexer requires $2(n-1)$ gates where n is the number of inputs to the multiplexer. A demultiplexer also requires $2(n-1)$ gates by similar reasoning. These numbers for gate count are based on a switching element construction which utilizes a tree-like arrangement of gates. For the 16×16 MPS, there are 960 gates in the simple 16×16 crossbar switch, 30 gates in a demultiplexer/multiplexer, and 48 gates in the 4×4 SE (assuming the SE uses a crossbar construction).

Using the switch-fault model assumption, let δ_s denote the failure rate of the 16×16 crossbar switch, then $\delta_s/960$ is the gate failure rate, $\delta_d = \delta_s/32$ is the demultiplexer/multiplexer failure rate, and $\delta_\Omega = \delta_s/20$ is the 4×4 SE failure rate.

8.5.1 Single-Valued Measures

In Table 8.1, three frequently used single-valued measures to compare the three candidate architectures are presented. Using equations (8.1) and (8.2), the bandwidth for each architecture can be computed. Assuming no failures, SYS_s and SYS_d have $BW = 10.3$, and SYS_Ω has $BW = 8.4$. On the basis of performance alone, SYS_s and SYS_d are indistinguishable, and SYS_Ω is the least preferred choice. Based on the mean time to failure ($MTTF$), SYS_Ω is no longer the last choice; SYS_d is the most reliable, and SYS_s is the least reliable. The cost of processors and memories is the same for all three architectures, so the cost of the IN is used to contrast the three MPS architectures where the cost is computed using a gate count. SYS_Ω is less than one-half as expensive as the other options, and this additional

Architecture	Bandwidth	$MTTF$		Cost
		$K = 12$	$K = 4$	
SYS_s	10.3	1322.3	3613.4	960
SYS_d	10.3	1537.9	6708.6	960
SYS_Ω	8.4	1497.2	6575.5	384

Table 8.1: Comparison of Architectures.

MPS	Failure Rate Parameter					
	Processors		Memories		Network	
	$K = 12$	$K = 4$	$K = 12$	$K = 4$	$K = 12$	$K = 4$
SYS_s	−21.3	−2.1	−1462.8	−2297.4	*− 4625.2*	*− 99839.4*
SYS_d	−35.0	−20.1	*− 1974.0*	*− 9069.5*	−0.9	−3.6
SYS_Ω	−35.5	−34.8	*− 1868.7*	*− 8655.7*	−10.6	−39.7

Table 8.2: Sensitivity of $MTTF$ with Respect to Parameters (Scaling factor $= \times((\lambda_i^2/N_i) \times 10^5)$).

consideration combined with the $MTTF$ data may make it the preferred choice.

Next, consider the sensitivity of the $MTTF$ estimates given in Table 8.1 to changes in component failure rates. For each model, using equation (8.18), the sensitivity of $MTTF$ with respect to processor failure rate, memory failure rate, and switching element failure rate is computed. Note that the different systems have different numbers of switching elements, with different failure rates. To find the system bottlenecks, the cost model described in Section 8.3.4 with $\alpha_i = c_i = 1$ is used. The parametric sensitivities are multiplied by a factor of λ_i^2/N_i. The results are shown in Table 8.2. The bottlenecks for each system configuration are italicized. Because SYS_s is most sensitive to switch failures, for this model, the switch is the reliability bottleneck. The memories are the bottleneck for the other two models.

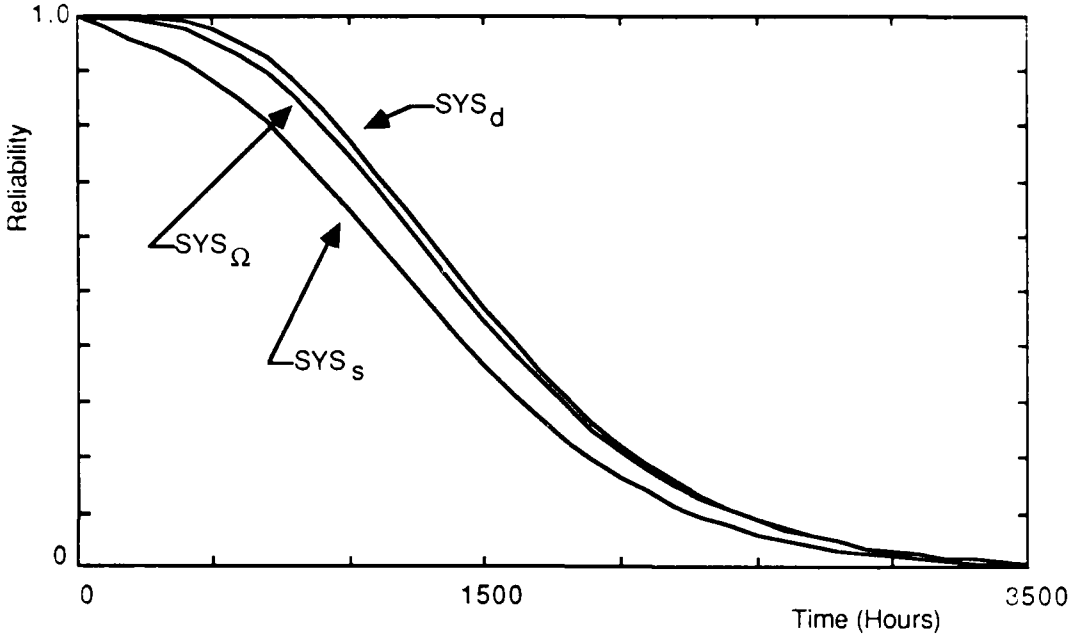


Figure 8.4: Comparison of the Reliabilities of the Three MPS Architectures for $K = 12$.

8.5.2 Reliability

In Figures 8.4 and 8.5, reliability as a function of mission time is plotted for the three MPS architectures. The reliability curves for $K = 12$ are plotted in Figure 8.4. Because SYS_s is vulnerable to a single-point switch failure, $R_s(t)$ is significantly less than $R_d(t)$ or $R_\Omega(t)$. Modeling the IN at the demultiplexer/multiplexer level increases the predicted reliability since the failure of individual components is not catastrophic. Also, observe that $R_\Omega(t) \leq R_d(t)$. A similar result is shown in Figure 8.5 ($K = 4$) except that now the degree of separation between the reliability of SYS_s and the other two architectures is even more pronounced and the difference between SYS_d and SYS_Ω is less discernible. This indicates that the reliability of the MPS design is insensitive to SYS_d or SYS_Ω as IN candidates when the connectivity requirement

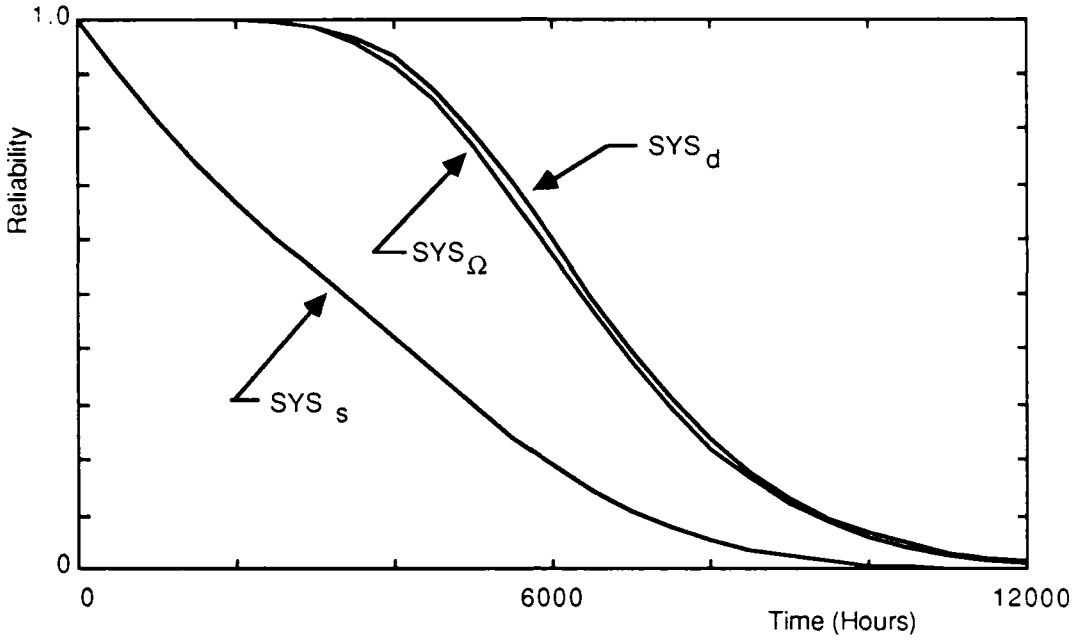


Figure 8.5: Comparison of the Reliabilities of the Three MPS Architectures for $K = 4$.

decreases. As with the $MTTF$ data, if cost is considered as well as reliability, SYS_{Ω} may be the preferred architecture.

Scaled parametric sensitivities for the SYS_s and SYS_{Ω} are plotted in Figures 8.6 and 8.7. The plot for SYS_d is omitted because it is almost identical to the plot for SYS_{Ω}. These parametric sensitivities are scaled by multiplying by the factor λ_i^2/N_i. Regardless of mission time, all three systems are insensitive to small changes in the processor failure rate. For SYS_s, the switch failure is the reliability bottleneck. For SYS_d and SYS_{Ω}, increased fault-tolerance in the switch makes the memories the reliability bottleneck, regardless of mission time.

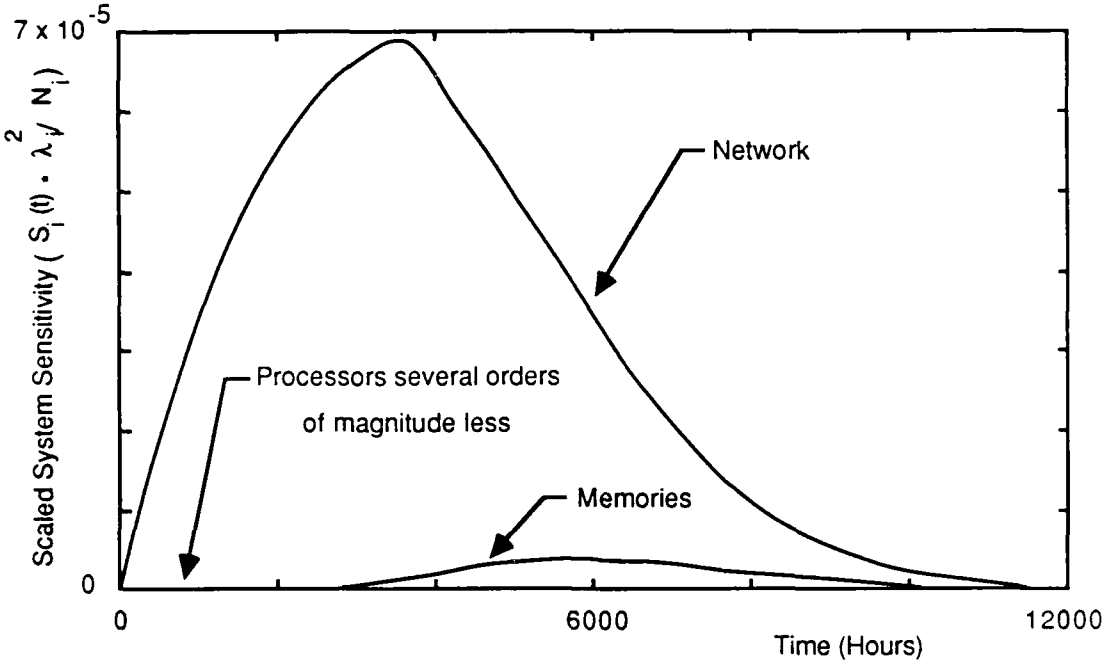


Figure 8.6: Scaled Parametric Sensitivity of Unreliability — Simple C.mmp Model.

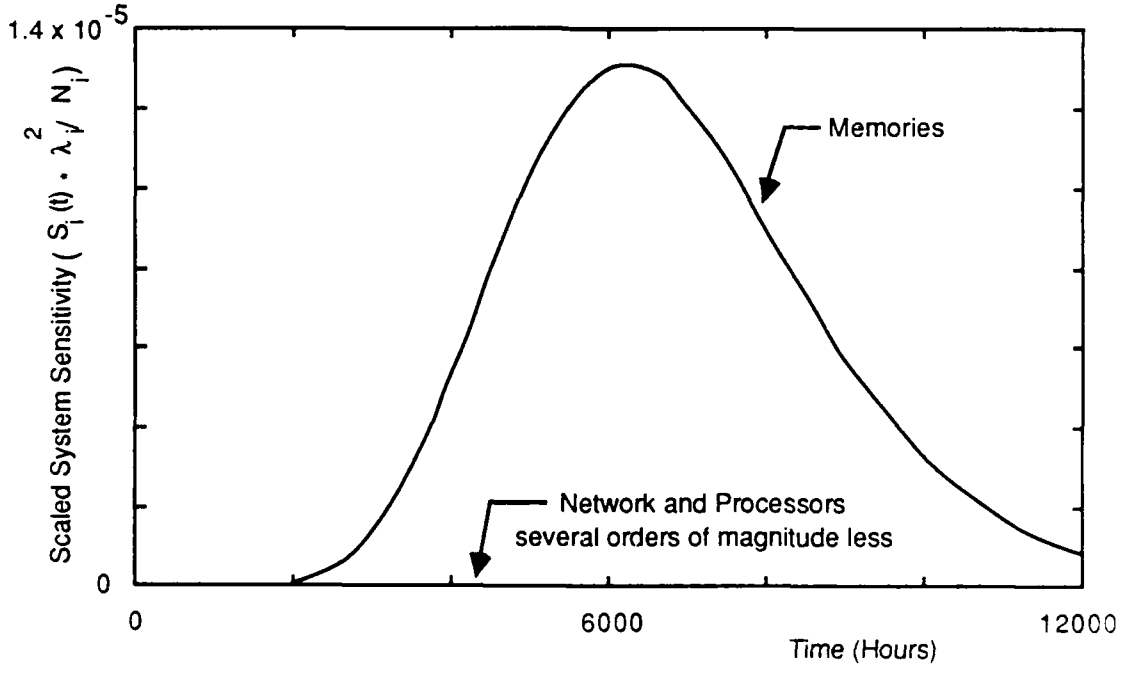


Figure 8.7: Scaled Parametric Sensitivity of Unreliability — Omega Network Model.

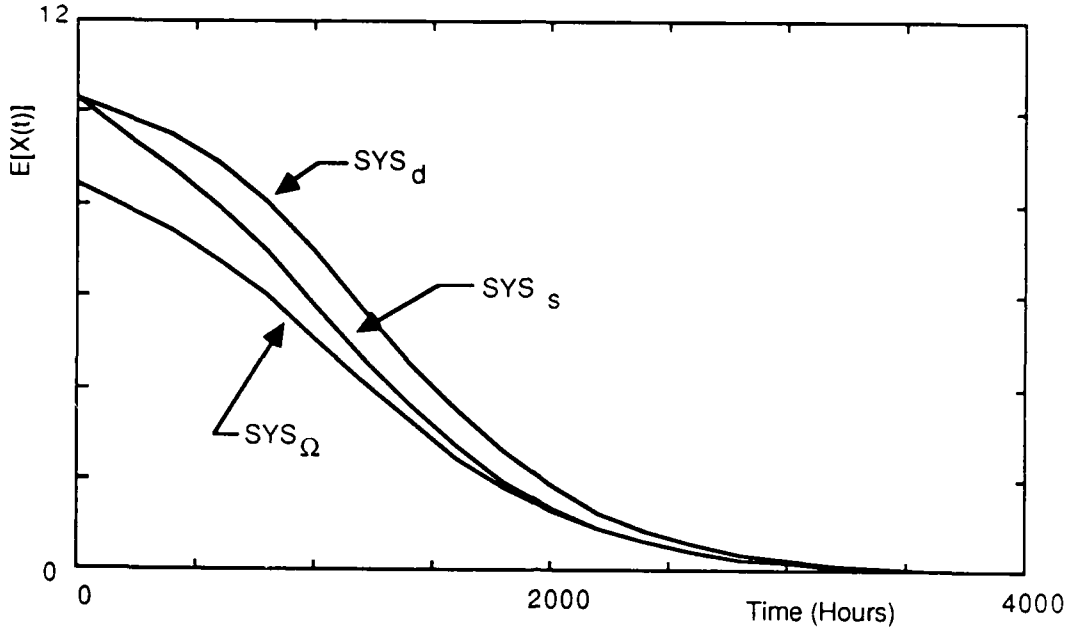


Figure 8.8: Comparison of the Expected Reward Rates at time t for the Three MPS Architectures for $K = 12$.

8.5.3 Performability

For $K = 12$, Figure 8.8 shows the expected system bandwidth at time t. SYS_d has the largest $E[X(t)]$, and SYS_s is significantly better than SYS_Ω for small values of time. For larger values of t, SYS_s and SYS_Ω are approximately equal. A different result is shown in Figure 8.9. SYS_d is still superior, but now for small values of t, SYS_s is superior to SYS_Ω and the converse is true for moderate values of t. This occurs because for small K, up to three SEs can fail in SYS_Ω and the system will still be operational, whereas for SYS_s when the IN fails, the system is down.

Parametric sensitivities for $E[X(t)]$ of the MPS models are plotted in Figures 8.10 and 8.11. Again, the plot for SYS_d is omitted because it is almost identical to the plot for SYS_Ω. These parametric sensitivities are scaled by multiplying by a factor of λ_i^2/N_i. Note that the sensitivities have an opposite

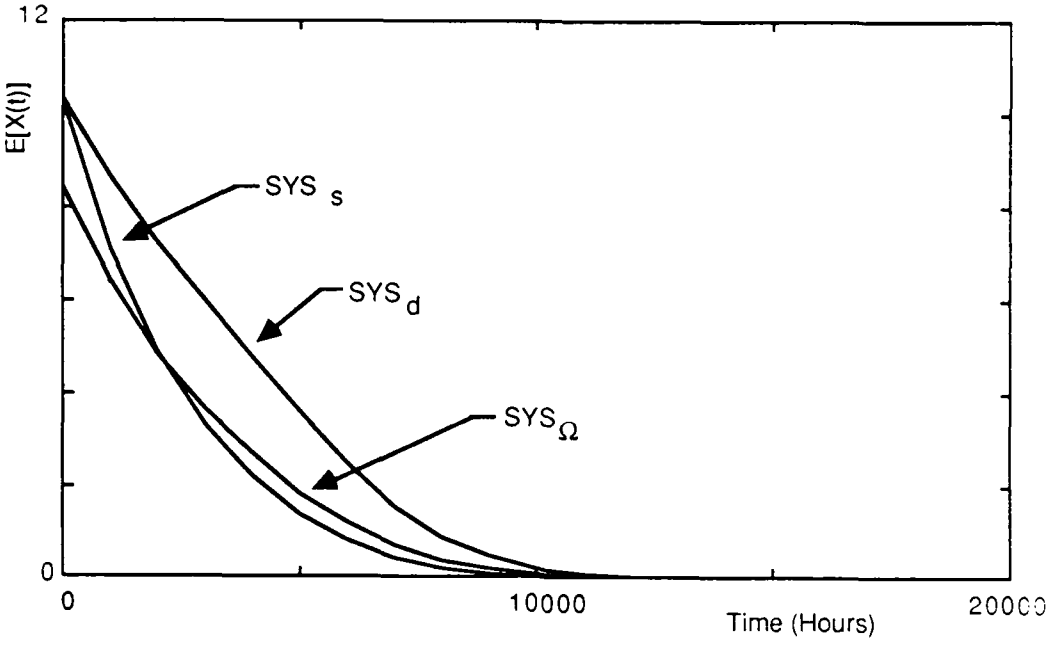


Figure 8.9: Comparison of the Expected Reward Rates at time t for the Three MPS Architectures for $K = 4$.

sign than the sensitivities of system unreliability; an increase in the failure rate increases unreliability but decreases the expected reward rate. Also, unlike the sensitivity of unreliability, the processor failure rate sensitivity curve is visible. Although it is unlikely that enough processors would ever fail to cause total system failure, a few processor failures might occur, reducing system performance. In SYS_s, the switch is the performability bottleneck. Because SYS_d and SYS_Ω have fault-tolerant switches, regardless of mission time, memories are their performability bottleneck.

The expected accumulated rewards for the three architectures are plotted in Figures 8.12 and 8.13 for $K = 12$ and $K = 4$, respectively. In Figure 8.12, the order of the architectures is SYS_d, SYS_s, and SYS_Ω. This is in contrast to the reliability curves of Figure 8.4 where the order of SYS_s and SYS_Ω were reversed. So even though SYS_s is less reliable than SYS_Ω, the larger

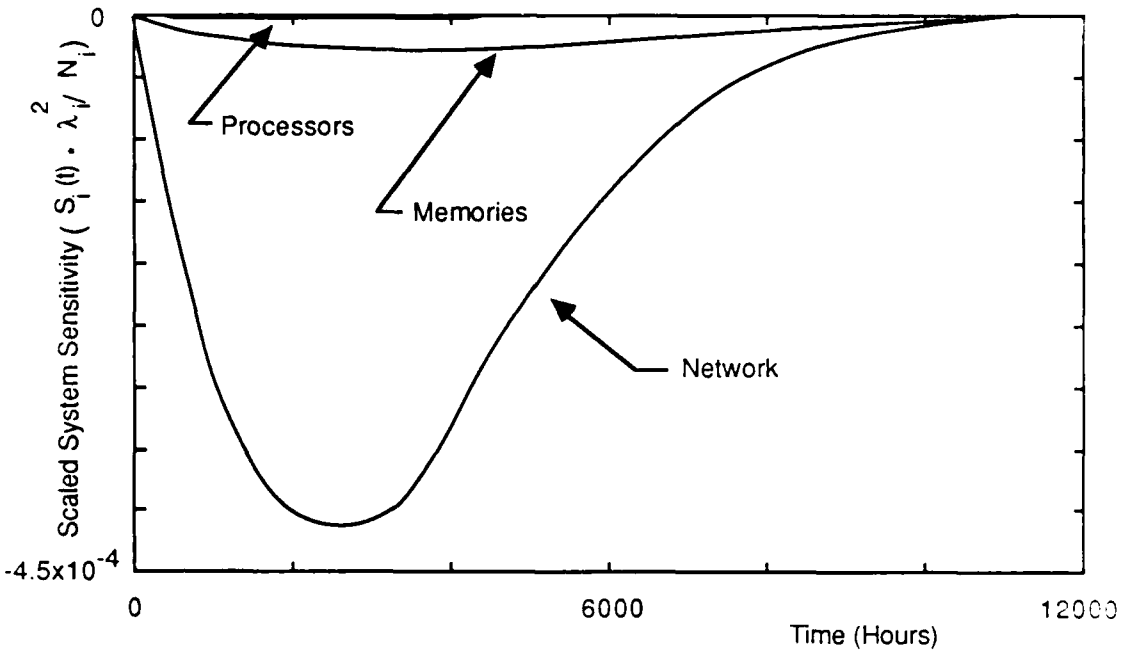


Figure 8.10: Scaled Parametric Sensitivity of Performance Level — Simple C.mmp Model.

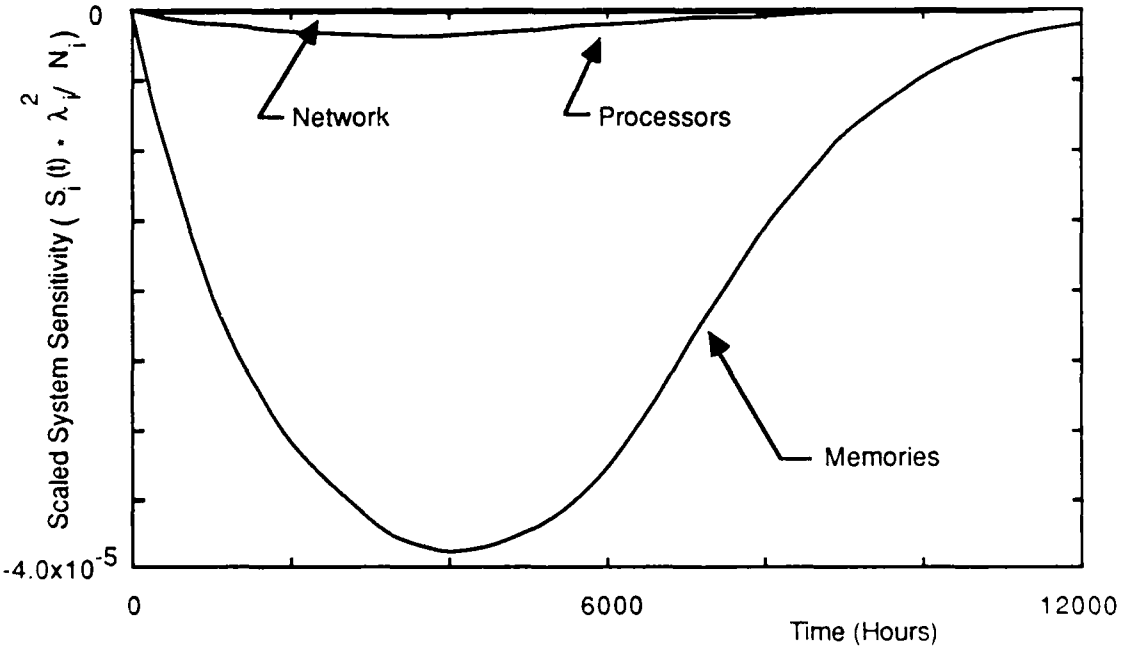


Figure 8.11: Scaled Parametric Sensitivity of Performance Level — Omega Network Model.

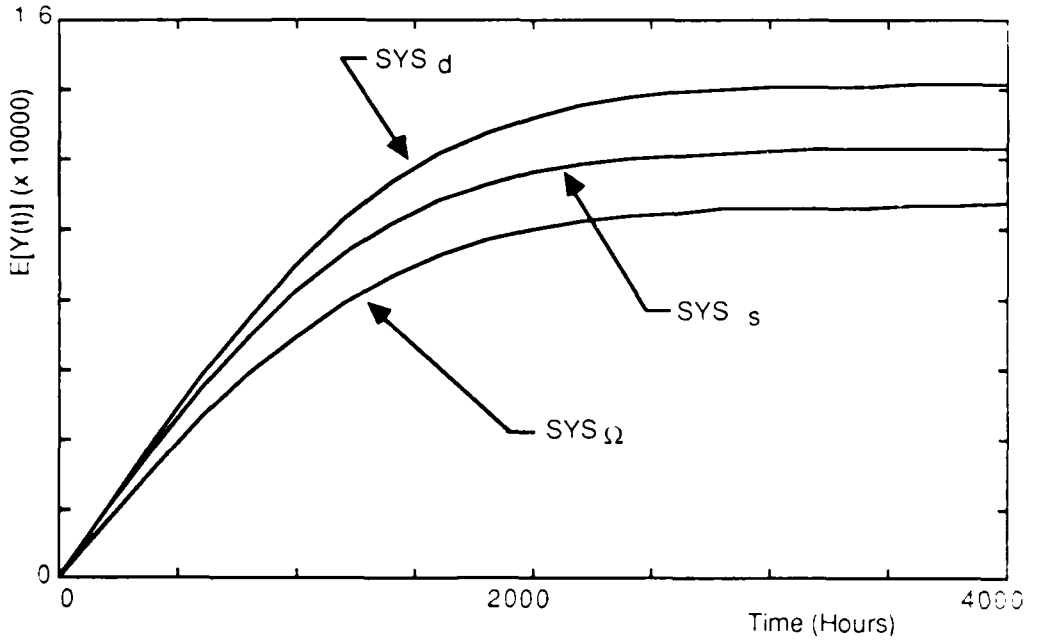


Figure 8.12: Comparison of the Expected Accumulated Reward by time t for the Three MPS Architectures for $K = 12$.

average bandwidth available in SYS_s while it is operational enables SYS_s to accomplish more work than SYS_Ω. For $K = 4$, SYS_s is preferred over SYS_Ω for small t, but the opposite is true for larger t. Also as expected, in Figure 8.13, SYS_d is clearly superior due to its larger possible bandwidth and the absence of bulk failures. For SYS_Ω, the failure of a single switching element may eliminate four processors or four memories; and in SYS_s, the failure of the IN immediately produces zero bandwidth.

The complementary distribution of accumulated reward until system failure is also analyzed. Prob$[Y(\infty) > x]$ will be larger for SYS_d since for a given K, it has a larger bandwidth than the corresponding SYS_Ω model, and unlike SYS_s and SYS_Ω, it does not permit bulk failures.

In Figure 8.14, the complementary distribution of accumulated reward is plotted for the three architectures. SYS_d is the dominating model as

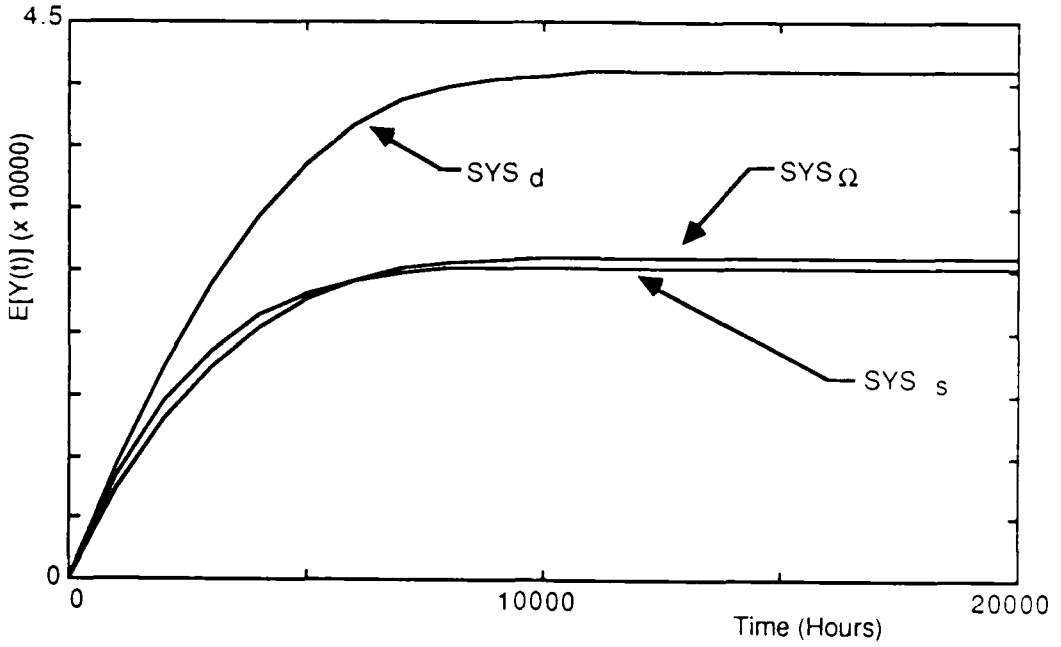


Figure 8.13: Comparison of the Expected Accumulated Reward by time t for the Three MPS Architectures for $K = 4$.

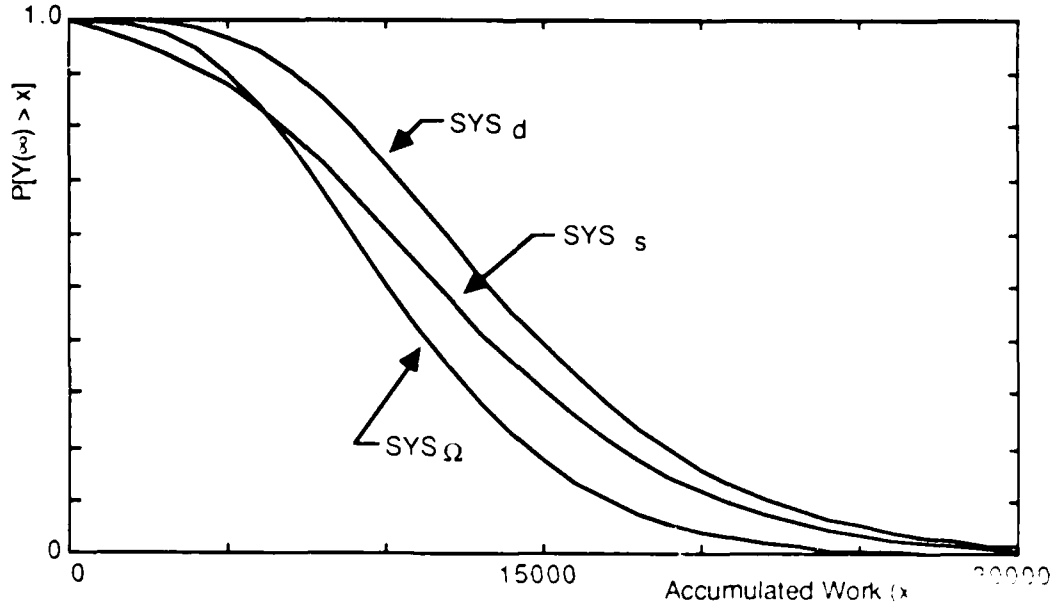


Figure 8.14: Comparison of the Complementary Distribution of Accumulated Reward Until System Failure for the Three MPS Architectures for $K = 12$.

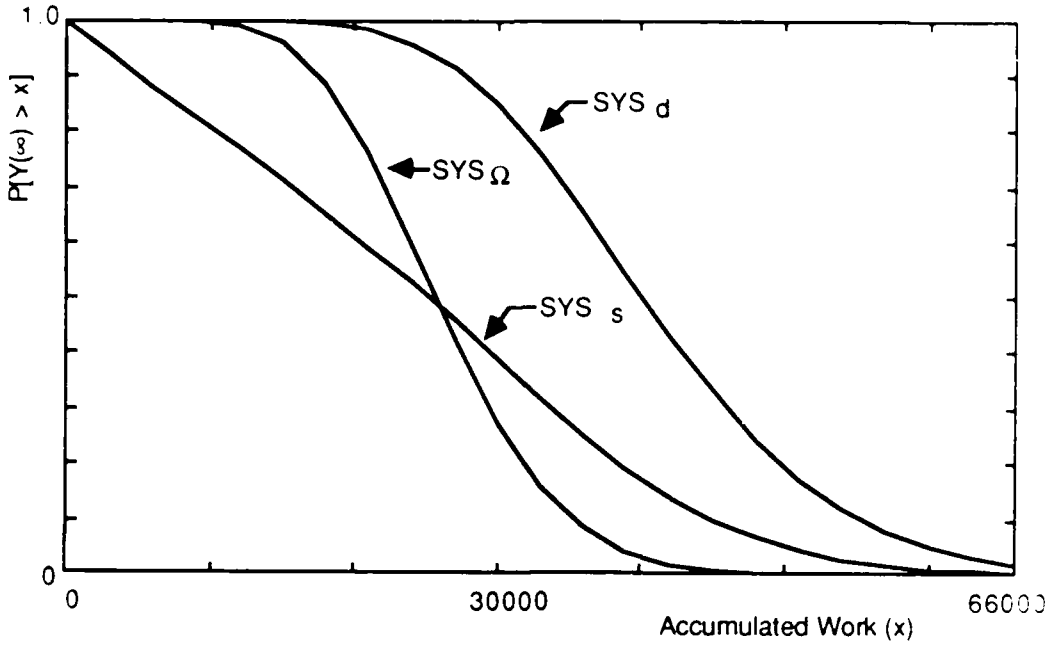


Figure 8.15: Comparison of the Complementary Distribution of Accumulated Reward Until System Failure for the Three MPS Architectures for $K = 4$.

expected. But, unlike the reliability curves of Figure 8.5, there is a crossover point for SYS_s and SYS_Ω. This shows that for small work requirements SYS_Ω would be preferred over SYS_s.

Prob$[Y(\infty) > x]$ is plotted for $K = 4$ in Figure 8.15. Since more "up" configurations are permitted for small K, the disparity between SYS_d and SYS_s is even more pronounced. Also note that now SYS_Ω reflects higher performability for nearly half of the possible work requirements. Also note that the spread between SYS_d and SYS_Ω is more pronounced from a performability perspective, as in Figure 8.15, than in terms of reliability, as shown in Figure 8.5.

8.5.4 Analysis with an Alternate Sensitivity Measure

As mentioned in Section 8.3.3, a second use of parametric sensitivities is in the identification of portions of a model that need refinement. Instead of using a cost function, as in the three previous subsections, relative changes, $\Delta\lambda_i/\lambda_i$ are considered in this subsection. This quantity is obtained by scaling the parametric sensitivities (multiplying each $\underline{S}(t)$ by λ_i). Using this approach changes the results obtained for SYS_s. With the "cost-based" measure used in Section 8.3.4, SYS_s $MTTF$ was most sensitive to switch failures for both $K = 4$ and $K = 12$. With the alternate scaling used here, the $MTTF$ of SYS_s is most sensitive to switch failures for $K = 4$, but for $K = 12$, it is most sensitive to memory failures. This indicates that if one wants to improve the $MTTF$ model for SYS_s, then K is also a factor in determining what component of the model should be refined.

Repeating the reliability sensitivity analysis with the alternate scaling, SYS_s is initially most sensitive to switch failures, but as mission time increases exhaustion of memory redundancy becomes a greater problem. For $t \geq 4000$, SYS_s reliability is most sensitive to changes in the memory failure rate. For $E[X(t)]$ of SYS_s, a similar crossover is observable at $t = 4000$. To improve the reliability or performability models for SYS_s for small t, the failure rate of the switch should be more accurately determined. For large values of t, the failure rate of the memory system should be more accurately determined.

8.5.5 Imperfect Coverage

To illustrate the effect of imperfect coverage on the three MPS architectures, the relative changes in $R(t)$ and $Y^c(x)$ as a result of imperfect coverage, c, will be considered for K 12. Specifically, the impact of a decrease in c

from $c = 1$ (perfect coverage) to $c = 0.95$ will be examined. Assume that each transition from an operational state i to another operational state j is successful with probability c. Then with probability $1 - c$, the system will fail as a result of unsuccessful reconfiguration.

In general, a coverage factor could be associated with each component type, but for the purpose of the current discussion, it is assumed that the factor is the same for each type. Now the effect on the curves in Figures 8.4 and 8.14 is to shift them down and to the left. Also the spread between the curves is reduced, but their relative position with respect to one another is unchanged. However, if the impact of imperfect coverage on the relative change in the independent variable is examined, some interesting observations can be made.

In the next two figures, the relative sensitivities of the three architectures to $c = 0.95$ as a function of the time (t) and work requirement (x) are shown. That is,

$$R_{SENS}(t) = \frac{R_{c=1}(t) - R_{c=0.95}(t)}{R_{c=1}(t)}, \text{ and} \tag{8.36}$$

$$y^c{}_{SENS}(x) = \frac{y^c{}_{c=1}(x) - y^c{}_{c=0.95}(x)}{y^c{}_{c=1}(x)} \tag{8.37}$$

From Figure 8.16, it can be seen that the reliability of SYS_d is more sensitive to imperfect coverage than the other two. Observe that at $t = 1000$ there is a 17% decrease in the reliability of SYS_d as a result of a 0.95 coverage factor. At $t = 2000$, the decrease is 23%. In Figure 8.17, SYS_Ω is most sensitive to a 0.95 coverage factor. At a work requirement of 10000, the relative decrease in $\text{Prob}[Y(\infty) > x]$ for SYS_Ω is 19%, and at $x = 20000$ the relative decrease is 25%.

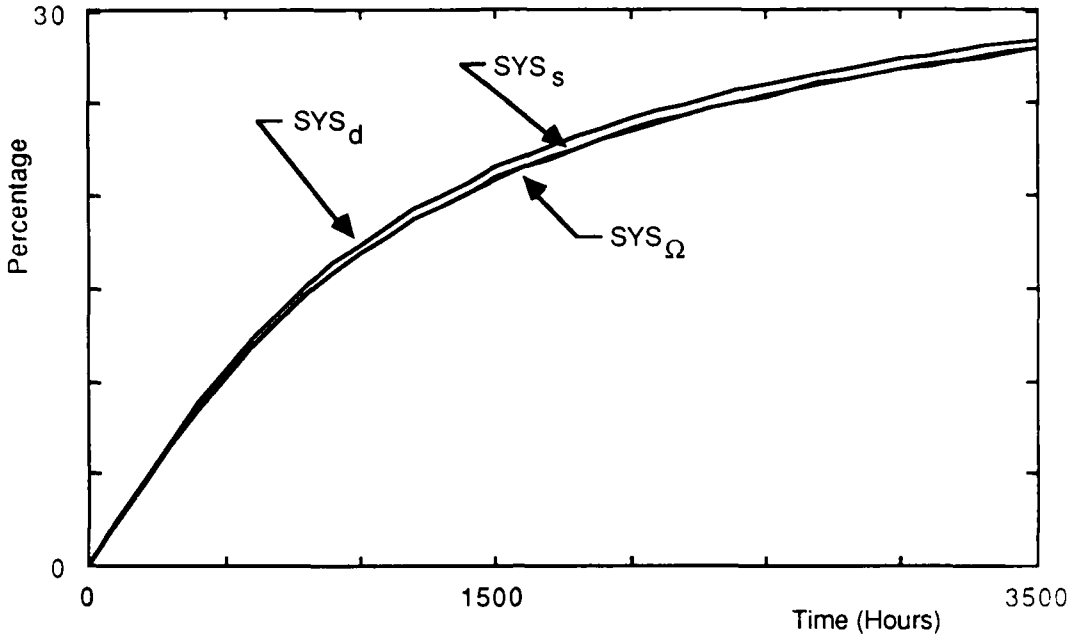


Figure 8.16: Relative Decrease in Reliability as a Result of a Decrease in the Coverage Factor from 1.00 to 0.95.

8.6 Summary

System modelers often rely on single-valued measures like $MTTF$. This oversimplification may hide important differences between candidate architectures. Time-dependent reliability analysis provides additional data, but unless whole series of models are run, it does not suggest where to spend additional design effort. In this chapter, the use of Markov reward models and parametric sensitivity analysis were discussed. Markov reward models allow modeling of the performance of degradable systems. Parametric sensitivity analysis helps identify critical system components or portions of the model that are particularly sensitive to error.

Three candidate architectures for implementing a multiprocessor system constructed from processors, shared memories, and an interconnection net-

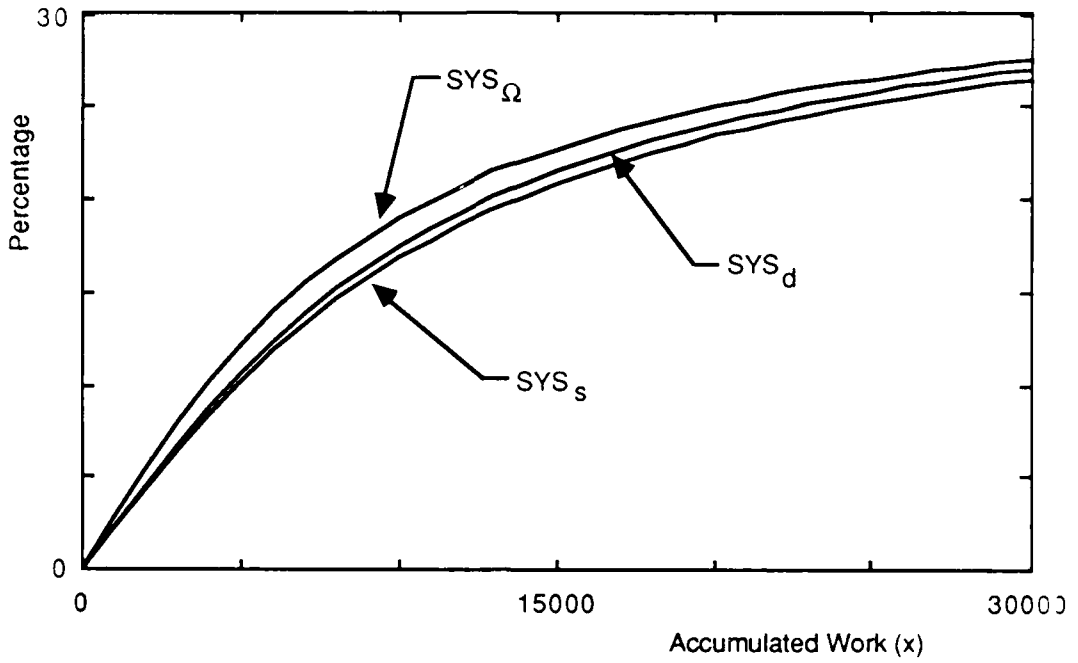


Figure 8.17: Relative Decrease in the Complementary Distribution of Accumulated Reward as a Result of a Decrease in the Coverage Factor from 1.00 to 0.95.

work were examined. The crossbar or the Omega network is used to represent the interconnection network and two implementations of the crossbar are presented. The use of state lumping allows computation of reliability and performability measures for realistic architectures.

Pure performance, reliability, and performability were used to evaluate the three multiprocessor system architectures. Based on performance alone, a MPS using a crossbar switch implemented as a single integrated component, SYS_s, or as a switch composed of independent demultiplexers/multiplexers, SYS_d, is the preferred architecture. On the basis of cost, SYS_Ω, utilizing an Omega network, is the least expensive. By all other measures considered, SYS_d is the best choice.

Using reliability to distinguish between architectures, SYS_d and SYS_Ω have similar lifetimes, and the differences between their lifetimes become less distinguishable as the minimum number of processor-memory pairs required for system operation decreases.

The use of pure performance or reliability measures in comparing architectures can be misleading. Using a combined measure provides a better metric for comparing competing systems with degradable characteristics. If one is concerned with the probability that the MPS will complete a specified amount of work before system failure, SYS_Ω is preferred over SYS_d for small work requirements and the converse is true for larger requirements.

To demonstrate the use of parametric sensitivity analysis in the evaluation of competing system designs, for each model, the parametric sensitivity of mean time to failure, system unreliability, and time-dependent expected reward were computed. By scaling with respect to a cost function, the identity of the reliability, performability, and $MTTF$ bottlenecks in each system were determined. The three models produced different results. The differences between the models highlight the need for detailed models and shows the role of analytic modeling in choosing design alternatives and guiding the design refinements.

Chapter 9

Conclusions

9.1 Summary

Performance, reliability, and performability issues for multiprocessor systems were examined in this thesis. The analysis centered on the interconnection network (IN) used in such systems since they are generally regarded as the bottleneck for achieving high speeds in large multiprocessor systems.

In the area of reliability analysis, a transient reliability analysis of the SEN, SEN+, and ASEN networks was performed. Exact closed-form expressions for the reliability of small networks were derived. These expressions are valid for any arbitrary component-lifetime distribution. Also derived were reasonably close lower bounds for approximating the reliability of larger SEN+ and ASEN networks. The lower bounds obtained were compared to the exact solutions derived for the smaller SEN+ and ASEN networks to verify that they are reasonable approximations of their respective network reliabilities. Then these lower bounds were used for analyzing SEN+ and ASEN networks up to size 1024×1024.

A comparison of the mean time to failure of these networks was presented, and it was shown that, on the basis of reliability, the ASEN is superior to the SEN, SEN+, and a parallel arrangement of two SENs.

The results for the SEN and SEN+ networks were extended to the case of an (uniform) Omega network, and it was shown that, based on both reliability and cost (in terms of gate complexity), the 2×2 switching element is the optimal switching element for the SEN, and for $N \leq 1024$, this switching element is optimal for the SEN+, as well.

Also, distributional sensitivity's influence on system reliability when modeling networks whose components have increasing failure rate (IFR) lifetime-distributions was discussed. It was shown that for the networks examined, the assumption that individual components have an exponential lifetime distribution is conservative if the actual distribution is increasing-failure-rate Weibull.

In the area of combined evaluation metrics, it was shown that performability, a combined measure of performance and reliability, is a more useful measure than either of its components — performance and reliability — for determining the "goodness" of a multistage interconnection network. Also it was shown that for MINs of size 8×8 and larger, truncation of the state space as a function of bandwidth is an useful approximation technique.

Finally, a detailed performability analysis of a multiprocessor system composed of 16 processors, 16 memories, and an interconnection network was performed. Three models of the interconnection were compared in the analysis. This analysis showed that detailed modeling of the IN is necessary in order to avoid erroneous conclusions about the efficiency of a multiprocessor organization.

9.2 Suggestions for Further Research

The bounds for the SEN+ and ASEN networks could be improved with the focus on obtaining converging upper and lower bounds as the size of the networks increase. Also, other recently suggested fault-tolerant networks should be analyzed and compared to those already in this thesis.

More work must also be done on determining how to properly model large multiprocessor systems. Simulations and crude approximations for evaluating measures of interest are currently all that is available. As a further extension to this effort, a technique for optimizing the combination of multiprocessor components to achieve a desired goal should be pursued.

Another goal of further research is to find an efficient algorithmic technique for computing all possible bandwidths and/or finding a method of getting tight bounds on the performability of the MINs when approximation techniques are used for analysis. In particular, the bandwidth computation of redundant path MINs in the presence of faults seems to present acute difficulty, and it ought to be pursued further. Also, a "normalized" or "standardized" basis for comparing competing interconnection network and multiprocessor designs should be established to provide a sound basis for evaluating the capabilities of these systems.

Appendix A

Convolution Integral Solution of CTMC

The integral (convolution) form of the Kolmogorov forward equation for the non-homogeneous Markov chain is given by [98]:

$$P_j(t) = P_j(0)e^{-\int_0^t q_i(\tau)d\tau} + \sum_k \int_0^t P_k(x)q_{kj}(x)e^{-\int_x^t q_j(\tau)d\tau}dx. \qquad \text{(A.1)}$$

where $P_i(t)$ is the probability of being in *state* i at time t, and the q's are the elements of the instantaneous transition rate matrix $\mathbf{Q}$.

The equations corresponding to the seven states of the CTMC for the 8×8 SEN+ network shown in Figure 5.2 are:

$$\begin{aligned}
P_1(t) &= e^{-16\int_0^t \lambda(\tau)d\tau} \\
P_2(t) &= \int_0^t P_1(x)(8\lambda(x))e^{-15\int_x^t \lambda(\tau)d\tau}dx \\
P_3(t) &= \int_0^t P_2(x)(\lambda(x))e^{-14\int_x^t \lambda(\tau)d\tau}dx \\
P_4(t) &= \int_0^t P_3(x)(3\lambda(x))e^{-14\int_x^t \lambda(\tau)d\tau}dx \\
P_5(t) &= \int_0^t P_4(x)(2\lambda(x))e^{-13\int_x^t \lambda(\tau)d\tau}dx \\
P_6(t) &= \int_0^t P_5(x)(\lambda(x))e^{-12\int_x^t \lambda(\tau)d\tau}dx
\end{aligned}$$

$$P_7(t) \;=\; 1 - \sum_{i=1}^{6} P_i(t) \; .$$

The system reliability being

$$R_{\mathrm{SEN+}}(t) = \sum_{i=1}^{6} P_i(t) \; . \tag{A.2}$$

The $P_i(t)$, $(i = 2,\ 3,\ 4,\ 5,\ 6)$ are determined in order of increasing subscripts. Leibnitz' rule for differentiating a definite integral [38] is restated here since it will be used several times in the solution of this system of equations.

Let

$$F(x) = \int_{a(x)}^{b(x)} f(x,y)dy \; .$$

If $f(x,y)$ has a continuous derivative with respect to x in the region $\alpha \le x \le \beta$, if $a(x) \le y \le b(x)$, and if $a(\cdot)$ and $b(\cdot)$ are differentiable, then

$$\frac{d}{dx}F(x) = \int_{a(x)}^{b(x)} \frac{\partial f(x,y)}{\partial x}dy + f[x,b(x)]\frac{db(x)}{dx} - f[x,a(x)]\frac{da(x)}{dx}$$

whenever $\alpha \le x \le \beta$. In particular,

$$\frac{d}{dx}\int_a^b f(x,y)dy = \int_a^b \frac{\partial f(x,y)}{\partial x}dy \; .$$

This equation is valid when $b = \infty$ or $a = -\infty$ provided the right-side is finite.

$P_1(t)$ is used to solve for $P_2(t)$.

$$P_2(t) = \int_0^t P_1(x)(8\lambda(x))e^{-15\int_x^t \lambda(\tau)d\tau}dx \; . \tag{A.3}$$

substituting for $P_1(x)$,

$$P_2(t) = 8\int_0^t [e^{-16\int_0^x \lambda(\tau)d\tau} \cdot e^{-15\int_x^t \lambda(\tau)d\tau}\lambda(x)]dx \; .$$

The second term inside the integral can be rewritten as

$$e^{-15\int_x^t \lambda(\tau)d\tau} = e^{(-15\int_0^t \lambda(\tau)d\tau + 15\int_0^x \lambda(\tau)d\tau)} \; .$$

Substituting and rearranging terms,

$$P_2(t) = 8e^{-15\int_0^t \lambda(\tau)d\tau} \int_0^t \left[e^{-\int_0^x \lambda(\tau)d\tau} \cdot \lambda(x) \right] dx .$$

Let $u = \int_0^x \lambda(\tau)d\tau$, then $\frac{du}{dx} = \lambda(x)$ by application of Leibnitz' rule. Hence,

$$\begin{aligned} \int_0^t e^{-\int_0^x \lambda(\tau)d\tau} \cdot \lambda(x)\, dx &= \int_0^t e^{-u} \frac{du}{dx} dx \\ &= e^{-u} \Big|_0^t \\ &= e^{-\int_0^x \lambda(\tau)d\tau} \Big|_0^t \\ &= 1 - e^{-\int_0^t \lambda(\tau)d\tau} \end{aligned}$$

Again by substitution, the expression for the transient probability of being in *state* 2 is determined to be

$$P_2(t) = 8\left[e^{-15\int_0^t \lambda(\tau)d\tau}\right]\left[1 - e^{-\int_0^t \lambda(\tau)d\tau}\right] . \qquad (A.4)$$

$P_2(t)$ is used to find $P_3(t)$ and so on.

Finally, the system's reliability is determined by summing over the "up" states.

$$\begin{aligned} R_{SEN+}(t) &= \sum_{i=1}^{6} P_i(t) \\ &= 2e^{-12\int_0^t \lambda(\tau)d\tau} + 4e^{-14\int_0^t \lambda(\tau)d\tau} - 8e^{-15\int_0^t \lambda(\tau)d\tau} + 3e^{-16\int_0^t \lambda(\tau)d\tau} . \end{aligned}$$

Appendix B

Reliability Dominance

To show that the system reliability of $2^n \times 2^n$ SEN+ networks (where $n \geq 3$) is strictly greater than the reliability of the corresponding SENs regardless of the underlying component lifetime-distribution, it must be shown that

$$R_{\text{SEN+}}(t) - R_{\text{SEN}}(t) \geq 0. \tag{B.1}$$

Let $r = r_{\text{SE}}(t)$ and observe that $0 \leq r \leq 1$. Then equation (B.1) can be expressed as a polynomial in r, and it must be shown that for any time t on the open interval $(0, 1)$ that equation (B.1) is greater than zero.

For the 8×8 SEN+ network,

$$R_{\text{SEN+}} = 3r^{16} - 8r^{15} + 4r^{14} + 2r^{12}. \tag{B.2}$$

And for the corresponding 8×8 SEN the reliability is

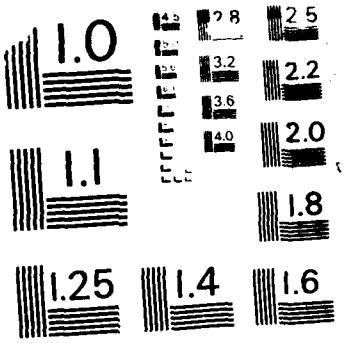
$$R_{\text{SEN}} = r^{12}.$$

As a first step, solve for the equality part of equation (B.1)

$$R_{\text{SEN+}} - R_{\text{SEN}} = 0$$

AD-A190 436 COMPARATIVE ANALYSIS OF MULTISTAGE INTERCONNECTION NETWORKS(U) DUKE UNIV DURHAM NC DEPT OF COMPUTER SCIENCE J T BLAKE DEC 87 3/3

UNCLASSIFIED F/G 25/5 NL



By substitution,

$$(r^{12})(3r^4 - 8r^3 + 4r^2 + 1) = 0. \tag{B.4}$$

Now it is given that at time $t = 0$ each system is operational, so the reliability of each network is 1. Thus, equation (B.4) can be further factored as

$$(r^{12})(r - 1)(3r^3 - 5r^2 - r - 1) = 0. \tag{B.5}$$

From equation (B.5), it is clear that there are roots at zero and 1, now only the remaining cubic expression

$$(3r^3 - 5r^2 - r - 1) = 0. \tag{B.6}$$

in equation (B.5) needs further examination.

Descartes' rule of signs can be used to determine the number of real roots of this polynomial. The rule states that the number n_+ of positive zeros of a polynomial $p(x)$ is less than or equal to the number of variations (v) in the sign of the coefficients of $p(x)$, where $p(x)$ is of the form

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0, \text{ and } a_n \neq 0. \tag{B.7}$$

Further, it states that the difference $v - n_+$ is an even integer. For equation (B.6), there is one sign change, so there is only one positive real root. There is a similar relationship between the number of sign changes in the coefficients of the polynomial $p(-x)$ and the number of negative real roots of $p(x)$. Again, for equation (B.6) we have two sign changes, so there must be either zero or two negative real roots.

Of course, application of *Descartes' rule of signs* is not essential for the cubic equation under consideration, but for higher-order equations, it can be very useful in determining how many real roots must be found. There are also a number of theorems for finding bounds on the locations of the roots

of polynomials, but discussion of these theorems is not necessary for this exposition.

For equation (B.6), there exists a closed-form expression for explicitly finding the real zeros. Using the method prescribed in [73] for example, equation (B.6) has exactly one real root equal to 1.929, and the two remaining roots are complex. From the roots of equation (B.4), it is clear that there are no zero crossings on the interval of interest. Simple substitution of $r = 0.5$ into equation (B.4) shows that this equation is positive over the entire interval. Thus for $0 < r < 1$, the inequality of equation (B.1) holds, and for $r = 0$ and $r = 1$, the equality holds.

Hence, the SEN+ network is strictly more reliable than the corresponding SEN regardless of the underlying component lifetime-distribution.

Appendix C

SHARPE Highlights

SHARPE (Symbolic Hierarchical Automated Reliability and Performance Evaluator) is a modeler's tool developed at Duke University. It allows the user to construct and analyze performance, reliability, availability, and Markov reward models. SHARPE provides seven model types: reliability block diagram, fault tree, acyclic Markov chain, cyclic irreducible Markov chain, cyclic Markov chain with absorbing states, acyclic semi-Markov chain, and general series-parallel graph. It allows a mixture of model types to be used in establishing a given application model. SHARPE also allows models to be combined hierarchically in the sense that the output of a submodel may used as a input to a (sub)model at a higher level. Therefore, SHARPE has a remarkable modeling capability in that it retains the efficiency of combinatorial solution methods where they are applicable, while providing the power and flexibility of Markov models.

SHARPE provides a symbolic solution in terms of time t for each of the model types. Within each model type, every individual component is characterized by a cumulative distribution function (CDF). SHARPE, however, places no interpretation on the CDF. This provides the modeler with the ca-

pability of adapting many system problems to the SHARPE framework. In a performance model for example, a CDF represents the time-to-completion of a task (component). In a reliability model, a CDF represents the time-to-failure of a component. In an availability model, a CDF represents the instantaneous probability that a component is not operational.

For each model type, components may have any distribution function that can be written as an exponential polynomial, including functions with a mass at zero and functions with a mass at infinity. The only exception is Markov chains; its components must have exponential distributions by definition.

The CDFs of individual components are specified as functions of the time parameter t, and SHARPE solves each model for a CDF in the same form. Because the solution CDF is symbolic in t and is in the same form as component CDFs, it is easy to combine the results obtained from different types of models.

SHARPE was written with portability in mind, and it is coded in C.

Bibliography

[1] G. B. Adams, D. P. Agrawal, and H. J. Siegel. A Survey and Comparison of Fault-Tolerant Multistage Interconnection Networks. *IEEE Computer*, 14–27, June 1987.

[2] G. B. Adams and H. J. Siegel. The Extra Stage Cube: A Fault-Tolerant Interconnection Network for Supersystems. *IEEE Transactions on Computers*, C-31(5):443–454, May 1982.

[3] D. P. Agrawal and J. Leu. Dynamic Accessibility Testing and Path Length Optimization of Multistage Interconnection Networks. In *Proceedings of the 4th International Conference on Distributed Computing Systems*, pages 266–277, May 1984.

[4] T. F. Arnold. The Concept of Coverage and Its Effect on the Reliability Model of a Repairable System. *IEEE Transactions on Computers*, C-22(3):251–254, March 1973.

[5] M. Avriel. *Nonlinear Programming: Analysis and Methods.* Prentice-Hall, Englewood Cliffs, NJ, 1976.

[6] M. O. Ball. Computational Complexity of Network Reliability Analysis: An Overview. *IEEE Transactions on Reliability*, R-35(3):230–239, August 1986.

[7] R. E. Barlow and F. Proschan. *Statistical Theory of Reliability and Lifetesting: Probability Models.* Holt, Rinehart and Winston, New York, 1975.

[8] K. E. Batcher. The Flip Network in STARAN. In *Proceedings of the International Conference on Parallel Processing*, pages 65–71, August 1976.

[9] S. J. Bavuso, J. B. Dugan, K. S. Trivedi, E. M. Rothmann, and W. E. Smith. Analysis of Typical Fault-Tolerant Architectures Using HARP. *IEEE Transactions on Reliability*, R-36(2):176–185, June 1987.

[10] M. D. Beaudry. Performance Related Reliability for Computer Systems. *IEEE Transactions on Computers*, C-27(6):540–547, June 1978.

[11] V. Benes. *Mathematical Theory of Connecting Networks.* Academic Press, NY, 1965.

[12] D. P. Bhandarkar. Analysis of Memory Interference in Multiprocessors. *IEEE Transactions on Computers*, C-24(9):897–908, September 1975.

[13] L. N. Bhuyan. A Combinatorial Analysis of Multibus Multiprocessors. In *Proceedings of the International Conference on Parallel Processing*, pages 225–227, August 1984.

[14] L. N. Bhuyan and D. P. Agrawal. Design and Performance of Generalized Interconnection Networks. *IEEE Transactions on Computers*, C-32(12):1081–1090, December 1983.

[15] A. Bobbio and K. Trivedi. An Aggregation Technique for the Transient Analysis of Stiff Markov Chains. *IEEE Transactions on Computers*, C-35(9):803–814, September 1986.

[16] W. G. Bouricius, W. C. Carter, J. P. Roth, and P. R. Schneider. On-Line Reliability Calculations to Achieve a Balanced Design of an Automatically Repaired Computer. In *Proceedings of the IEEE National Aerospace and Electronics Conference*, pages 243–246, Dayton, Ohio, May 1967.

[17] W. G. Bouricius, W. C. Carter, and P. R. Schneider. Reliability Modeling Techniques for Self-Repairing Computer Systems. In *Proceedings of the 24th Annual ACM National Conference*, pages 295–309, 1969.

[18] V. Cherkassky, E. Opper, and M. Malek. Reliability and Fault Diagnosis Analysis of Fault-Tolerant Multistage Interconnection Networks. In *Proceedings of the Fourteenth International Symposium on Fault-Tolerant Computing*, pages 246–251, June 1984.

[19] L. Ciminiera and A. Serra. A Connecting Network with Fault Tolerance Capabilities. *IEEE Transactions on Computers*, C-35(6):578–580, June 1986.

[20] L. Ciminiera and A. Serra. A Fault-Tolerant Connecting Network for Multiprocessor Systems. In *Proceedings of the International Conference on Parallel Processing*, pages 113–122, August 1982.

[21] C. Clos. A Study of Non-Blocking Switching Networks. *Bell System Technical Journal*, 32(2):406–424, March 1953.

[22] D. R. Cox. A Use of Complex Probabilities in the Theory of Stochastic Processes. *Proceedings of the Cambridge Philosophical Society*, 51:313–319, 1955.

[23] C. R. Das and L. N. Bhuyan. Computation Availability of Multiple-Bus Multiprocessors. In *Proceedings of the International Conference on Parallel Processing*, pages 807–813, August 1985.

[24] C. R. Das and L. N. Bhuyan. Reliability Simulation of Multiprocessor Systems. In *Proceedings of the International Conference on Parallel Processing*, pages 591–598, August 1985.

[25] N. J. Davis and H. J. Siegel. The Performance Analysis of Partitioned Circuit Switched Multistage Interconnection Networks. In *Proceedings of the International Symposium on Computer Architecture*, pages 387–394, June 1985.

[26] D. M. Dias and J. R. Jump. Analysis and Simulation of Buffered Delta Networks. *IEEE Transactions on Computers*, C-30(4):273–282, April 1981.

[27] L. Donatiello and B. R. Iyer. Analysis of a Composite Performance Reliability Measure for Fault-Tolerant Systems. *Journal for the Association of Computing Machinery*, 34(1):179–199, January 1987.

[28] J. B. Dugan, K. S. Trivedi, M. K. Smotherman, and R. M. Geist. The Hybrid Automated Reliability Predictor. *AIAA Journal of Guidance, Control and Dynamics*, 9(3):319–331, May-June 1986.

[29] K. M. Falavarjani and D. K. Pradhan. Fault-Diagnosis of Parallel Processor Interconnection Networks. In *Proceedings of the Eleventh International Symposium on Fault-Tolerant Computing*, pages 209–212, June 1981.

[30] T-Y. Feng. A Survey of Interconnection Networks. *IEEE Computer*, 12–27, December 1981.

[31] D. G. Furchtgott and J. F. Meyer. A Performability Solution Method for Degradable Nonrepairable Systems. *IEEE Transactions on Computers*, C-33(6):550–554, June 1984.

[32] E. Gelenbe and A. I. Mitrani. *Analysis and Synthesis of Computer Systems.* Academic Press, New York, 1980.

[33] L. R. Goke and G. J. Lipovski. Banyan Networks for Partitioning Multiprocessor Systems. In *Proceedings of the International Symposium on Computer Architecture*, pages 21–28, December 1973.

[34] A. Goyal and T. Agerwala. Performance Analysis of Future Shared Storage Systems. *IBM Journal of Research and Development*, 28(1):95–108, January 1984.

[35] A. Goyal, S. Lavenberg, and K. Trivedi. Probabilistic Modeling of Computer System Availability. *Annals of Operations Research*, 8:285–306, March 1987.

[36] A. Goyal and A. N. Tantawi. Evaluation of Performability for Degradable Computer Systems. *IEEE Transactions on Computers*, 36(6):738–744, June 1987.

[37] P. Heidelberger and A. Goyal. Sensitivity Analysis of Continuous Time Markov Chains Using Uniformization. In P. J. Courtois G. Iazeolla and O. J. Boxma, editors, *Proceedings of the 2nd International Workshop on Applied Mathematics and Performance/Reliability Models of Computer/Communication Systems*, pages 93–104, Rome, Italy, May 1987.

[38] D. P. Heyman and M. J. Sobel. *Stochastic Models in Operations Research: Volume II Stochastic Optimization.* McGraw-Hill Book Company, 1984.

[39] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Publishing Company, 1979.

[40] R. A. Howard. *Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes.* John Wiley and Sons, New York, 1971.

[41] M. C. Hsueh, R. K. Iyer, and K. S. Trivedi. A Measurement-Based Dependability Model for a Multiprocessor System. In P. J. Courtois G. Iazeolla and O. J. Boxma, editors, *Proceedings of the 2nd International Workshop on Applied Mathematics and Performance/Reliability Models of Computer/Communication Systems*, pages 337–352, Rome, Italy, May 1987.

[42] K. Hwang and F. A. Briggs. *Computer Architecture and Parallel Processing.* McGraw-Hill Book Company, 1984.

[43] K. Hwang and T-P. Chang. Combinatorial Reliability Analysis of Multiprocessor Computers. *IEEE Transactions on Reliability*, R-31(5):469–473, December 1982.

[44] A. D. Ingle and D. P. Siewiorek. Reliability Models for Multiprocessor Systems With and Without Periodic Maintenance. In *Proceedings of the Seventh International Symposium on Fault-Tolerant Computing*, pages 3–9, Los Angeles, CA, June 1977.

[45] J. G. Kemeny and J. L. Snell. *Finite Markov Chains.* Van Nostrand-Reinhold, Princeton, NJ, 1960.

[46] C. P. Kruskal and M. Snir. The Performance of Multistage Interconnection Networks for Multiprocessors. *IEEE Transactions on Computers*, C-32(12):1091–1098, December 1983.

[47] D. Kuck. *The Structure of Computers and Computations.* Volume 1, John Wiley and Sons, NY, 1978.

[48] V. G. Kulkarni, V. F. Nicola, R. M. Smith, and K. S. Trivedi. Numerical Evaluation of Performability and Job Completion Time in Repairable Fault-Tolerant Systems. In *Proceedings of the Sixteenth International Symposium on Fault-Tolerant Computing*, pages 252–257, July 1986.

[49] V. G. Kulkarni, V. F. Nicola, and K. S. Trivedi. The Completion Time of a Job on Multi-Mode Systems. *Advances in Applied Probability*, December 1987. To appear.

[50] M. Kumar. *Performance Improvement in Single-Stage and Multiple-Stage Shuffle-Exchange Networks.* PhD thesis, Rice University, Houston, Texas, July 1983.

[51] V. P. Kumar. *On Highly Reliable, High Performance Multistage Interconnection Networks.* PhD thesis, University of Iowa, December 1985.

[52] V. P. Kumar and S. M. Reddy. Design and Analysis of Fault-Tolerant Multistage Interconnection Networks With Low Link Complexity. In *Proceedings of the International Symposium on Computer Architecture*, pages 376–386, June 1985.

[53] V. P. Kumar and S. M. Reddy. A Fault-Tolerant Technique for Shuffle-Exchange Multistage Interconnection Networks. *IEEE Computer*, 30–40, June 1987.

[54] D. H. Lawrie. Access and Alignment of Data in an Array Processor. *IEEE Transactions on Computers*, C-24:1145–1155, December 1975.

[55] D. H. Lawrie. *Memory-Processor Connection Networks.* Report UIUCDCS-R-73-557, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, February 1973.

[56] V. O. Li and J. A. Silvester. Performance Analysis of Networks with Unreliable Components. *IEEE Transactions on Communications*, COM-32(10):1105–1110, October 1984.

[57] S. Makam, A. Avizienis, and G. Grusas. *UCLA ARIES 82 User's Guide.* Technical Report CSD-820830, UCLA, August 1982.

[58] R. A. Marie, A. L. Reibman, and K. S. Trivedi. Transient Solution of Acyclic Markov Chains. *Performance Evaluation*, 7(3):175–194, 1987.

[59] R. J. McMillen and H. J. Siegel. Performance and Fault-Tolerance Improvements in the Inverse Augmented Data Manipulator Network. In *Proceedings of the International Symposium on Computer Architecture*, pages 63–72, April 1982.

[60] J. F. Meyer. Closed-Form Solutions of Performability. *IEEE Transactions on Computers*, C-31(7):648–657, July 1982.

[61] J. F. Meyer. On Evaluating the Performability of Degradable Computing Systems. *IEEE Transactions on Computers*, C-29(8):720–731, August 1980.

[62] J. F. Meyer. Performability Modeling of Distributed Real-Time Systems. In G. Iazeolla, P. J. Courtois, and A. Hordijk, editors, *Mathematical Computer Performance and Reliability*, pages 361–372, Elsevier Science Publishers, B. V. (North Holland), 1984.

[63] W. Najjar and J. L. Gaudiot. Reliability and Performance Modelling of Hypercube-Based Multiprocessors. In P. J. Courtois G. Iazeolla and O. J. Boxma, editors, *Proceedings of the 2nd International Workshop on Applied Mathematics and Performance/Reliability Models of Computer/Communication Systems*, pages 305–319, Rome, Italy, May 1987.

[64] K. Padmanabhan. *Fault Tolerance and Performance Improvement in Multiprocessor Interconnection Networks.* PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, May 1984.

[65] K. Padmanabhan and D. H. Lawrie. A Class of Redundant Path Multistage Interconnection Network. *IEEE Transactions on Computers*, C-32(12):1099–1108, December 1983.

[66] K. Padmanabhan and D. H. Lawrie. Fault Tolerance Schemes in Shuffle-Exchange Type Interconnection Networks. In *Proceedings of the International Conference on Parallel Processing*, pages 71–75, August 1983.

[67] D. S. Parker and C. S. Raghavendra. The Gamma Network: A Multiprocessor Interconnection Network with Redundant Paths. In *Proceedings of the International Symposium on Computer Architecture*, pages 73–80, June 1982.

[68] J. H. Patel. Performance of Processor-Memory Interconnections for Multiprocessors. *IEEE Transactions on Computers*, C-30(10):771–780, October 1981.

[69] J. H. Patel. Processor-Memory Interconnections for Multiprocessors. In *Proceedings of the International Symposium on Computer Architecture*, pages 168–177, April 1979.

[70] M. C. Pease. The Indirect Binary n-Cube Microprocessor Array. *IEEE Transactions on Computers*, C-26:458–473, May 1977.

[71] G. F. Pfister, W. C. Brantley, D. A. George, S. L. Harvey, W. J. Kleinfelder, K. P. McAuliffe, E. A. Melton, V. A. Norton, and J. Weiss. The IBM Research Parallel Prototype (RP3): Introduction and Architecture. In *Proceedings of the International Conference on Parallel Processing*, pages 764–771, August 1985.

[72] D. K. Pradhan, editor. *Fault-Tolerant Computing: Theory and Techniques.* Volume I & II, Prentice-Hall, Englewood Cliffs, NJ. 1986.

[73] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge Press, 1986.

[74] J. S. Provan. Bounds on the Reliability of Networks. *IEEE Transactions on Reliability*, R-35(3):260–268, August 1986.

[75] C. S. Raghavendra, A. Avizienis, and M. D. Ercegovac. Fault Tolerance in Binary Tree Architectures. *IEEE Transactions on Computers*, C-33(6):568–572, June 1984.

[76] C. S. Raghavendra and D. S. Parker. Reliability Analysis of an Interconnection Network. In *Proceedings of the 4th International Conference on Distributed Computing Systems*, pages 461–471, May 1984.

[77] C. S. Raghavendra and A. Varma. Fault-Tolerant Multiprocessors with Redundant-Path Interconnection Networks. *IEEE Transactions on Computers*, C-35(4):307–316, April 1986.

[78] C. S. Raghavendra and A. Varma. INDRA: A Class of Interconnection Networks with Redundant Paths. *Real-Time Systems Symposium*, 153–165, December 1984.

[79] S. M. Reddy and V. P. Kumar. On Fault-Tolerant Multistage Interconnection Networks. In *Proceedings of the International Conference on Parallel Processing*, pages 155–164, August 1984.

[80] A. Reibman and K. Trivedi. Numerical Transient Analysis of Markov Models. *Computers and Operations Research*, 15(1):19–36, 1988.

[81] A. L. Reibman. *Transient Analysis of Large, Stiff Markov Models: Numerical and Approximate Solution Techniques.* PhD thesis, Department of Computer Science, Duke University, 1987.

[82] A. L. Reibman and K. S. Trivedi. Transient Analysis of Cumulative Measures of Markov Chain Behavior. 1987. Submitted for publication.

[83] S. M. Ross. *Introduction to Probability Models.* Academic Press, 3rd edition, 1985.

[84] R. Sahner and K. S. Trivedi. Reliability Modeling Using SHARPE. *IEEE Transactions on Reliability*, R-36(2):186–193, June 1987.

[85] M. L. Shooman. *Probabilistic Reliability: An Engineering Approach.* McGraw-Hill, New York, 1968.

[86] D. P. Siewiorek. Multiprocessors: Reliability Modeling and Graceful Degradation. In *Infotech State of the Art Conference on System Reliability*, pages 48–73, Infotech International, London, 1977.

[87] D. P. Siewiorek, V. Kini, R. Joobbani, and H. Bellis. A Case Study of C.mmp, Cm*, and C.vmp: Part II — Predicting and Calibrating Reliability of Multiprocessor. *Proceedings of the IEEE*, 66(10):1200–1220, October 1978.

[88] R. Smith, K. S. Trivedi, and A. V. Ramesh. Performability Analysis: Measures, an Algorithm, and a Case Study. *IEEE Transactions on Computers*, April 1988. Accepted for publication.

[89] R. M. Smith and K. S. Trivedi. Calculating the Distribution of Accumulated Reward Using Partial Differential Equations. In preparation, 1987.

[90] R. M. Smith and K. S. Trivedi. A Performability Analysis of Two Multiprocessor Systems. In *Proceedings of the Seventeenth International Symposium on Fault-Tolerant Computing*, pages 224–229, July 1987.

[91] M. Smotherman. *Parametric Error Analysis and Coverage Approximations in Reliability Modeling.* PhD thesis, Department of Computer Science, University of North Carolina, Chapel Hill, NC, 1984.

[92] M. Smotherman, R. Geist, and K. Trivedi. Provably Conservative Approximations to Complex Reliability Models. *IEEE Transactions on Computers*, C-35(4):333–338, April 1986.

[93] W. Stewart and A. Goyal. *Matrix Methods in Large Dependability Models.* Research Report RC-11485, IBM, November 1985.

[94] H. S. Stone. *High-Performance Computer Architecture.* Addison-Wesley, 1987.

[95] T. H. Szymanski and V. C. Hamacher. On the Permutation Capability of Multistage Interconnection Networks. *IEEE Transactions on Computers*, C-36(7):810–821, July 1987.

[96] S. Thanawastien and V. P. Nelson. Interference Analysis of Shuffle/Exchange Networks. *IEEE Transactions on Computers*, C-30(8):545–556, August 1981.

[97] K. Trivedi and R. Geist. Decomposition in Reliability Analysis of Fault-Tolerant Systems. *IEEE Transactions on Reliability*, R-32(5):463–468, December 1983.

[98] K. S. Trivedi. *Probability and Statistics with Reliability, Queueing and Computer Science Applications.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[99] N-F. Tzeng, P-C. Yew, and C-Q. Zhu. A Fault-Tolerant Scheme for Multistage Interconnection Networks. In *Proceedings of the International Symposium on Computer Architecture*, pages 368–375, June 1985.

[100] N-F. Tzeng, P-C. Yew, and C-Q. Zhu. The Performance of a Fault-Tolerant Multistage Interconnection Network. In *Proceedings of the International Conference on Parallel Processing*, pages 458–465, August 1985.

[101] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1962.

[102] D. S. Wise. Compact Layouts of Banyan/FFT Networks. In *Proceedings of the CMU Conference on VLSI Systems and Computations*, pages 186–195, Computer Science Press, 1981.

[103] C-L. Wu, editor. *IEEE Computer*, IEEE Computer Society, June 1985. Special Issue on Multiprocessing Technology.

[104] C-L. Wu and T-Y. Feng. On a Class of Multistage Interconnection Networks. *IEEE Transactions on Computers*, C-29(8):694–702, August 1980.

Biography

James T. Blake was born in Baltimore, Maryland, on July 14, 1948. He was awarded an Associate of Arts degree from Baltimore Junior College in 1968; a Bachelor of Science degree in Accounting with minors in Mathematics and Business from the University of Tampa, Florida, in 1973; and a Master of Science degree in Systems Engineering from the Naval Postgraduate School, Monterey, California, in 1984, where he graduated with distinction and received the Naval Electronic Systems Command Electronic Warfare Technology Award and the Armed Forces Communications and Electronics Association Honor Award.

He is a member of the Association of Computing Machinery (ACM), the Computer Architecture and the Computer System Performance and Measurement Special Interest Groups of the ACM, the Institute of Electrical and Electronics Engineers (IEEE), the IEEE Computer Society, and Sigma Xi.

Publications

J. Blake, A. Reibman, and K. Trivedi. *Sensitivity Analysis of Reliability and Performability Measures for Multiprocessor Systems.* Technical Report CS-1987-32, Department of Computer Science, Duke University, Durham, NC, 1987.

J. Blake and K. Trivedi. *Comparing Three Interconnection Networks Embedded in a Multiprocessor System.* Technical Report CS-1987-33, Department of Computer Science, Duke University, Durham, NC, 1987.

J. Blake and K. Trivedi,"Multistage Interconnection Network Reliability." *IEEE Transactions on Computers*, 1988. Accepted subject to revision.

J. Blake and K. Trivedi. *Reliabilities of Two Fault-Tolerant Interconnection Networks.* Technical Report CS-1987-36, Department of Computer Science, Duke University, Durham, NC, 1987.

J. Blake and K. Trivedi,"Reliability of the Shuffle-Exchange Network and Its Variants." To appear in *Proceedings of the Hawaii International Conference on System Sciences*, January 1988.

END

DATE

FILMED

DTIC

4/88