RELIABILITY EVALUATION OF FAULT-TOLERANT COMPUTING SYSTEMS AND NETWORKS

by

Meera Balakrishnan

CENG Technical Report 92-06

A Dissertation Presented to the

FACULTY OF THE GRADUATE SCHOOL

UNIVERSITY OF SOUTHERN CALIFORNIA

In Partial Fulfillment of

Requirements for the Degree

DOCTOR OF PHILOSOPHY

(Electrical Engineering)

To America

Acknowledgements

I thank Prof. C. S. Raghavendra, Dr. Andrew Reibman of AT&T Bell Laboratories, Prof. Michel Dubois and Prof. William Harris for being on my thesis committee and for enabling me to finish this dissertation by coming to USC from the states of Washington, New Jersey and Minnesota. I take this opportunity to express my deepest gratitude to my advisors Prof. C. S. Raghavendra and Dr. Andrew Reibman; their realization and understanding that I constantly needed to digress from the mainstream to make any kind of progress made this thesis an enjoyable learning experience for me. For this, and for their constant support and encouragement, I shall always be grateful. I owe a special debt of thanks to Prof. James Yee for teaching me much through his excellent courses and for opening doors to areas I had never heard of before; I enjoyed exploring and felt like Alice in Wonderland! I consider it my privilege to acknowledge a great and kind teacher, Prof. Joshua Chover of the Mathematics Department, University of Wisconsin, Madison for his skillful unraveling of the mysteries of stochastic processes; thank you, Prof. Chover for an opportunity to learn from you. I thank Prof. Michel Dubois for helping me realize the value of analytical solutions in my first few years at USC and Prof. Alice Parker for her encouragement while I adjusted to a new culture.

I thank my friends in Salvatori, my home for many years: Bill Bates, Mary Zittercob, Christine Estrada and Carol Gordon with whom I have shared many a laugh – Diane Demetras, Lea Vasquez, Donnalyn Combest, Dawn and Alex, thank you. Lucille, Irene and Neela, thank you for making 27^{th} March 92 a pleasant and memorable day.

I thank my colleagues Suresh Chalasani, Hua-Chun Lin, Rajendra Boppana, Ge-ming Chiu, Robert Tien, Pei-ji Yang, Mitch Mlinar, Rajagopalan Srinvasan, Shiv Prakash, Dhabaleswar Panda, Amit Majumdar, Rajesh Gupta, Saumya Ramarao, Ted Cheng, Charlie Njinda and Jorge Seidel for sharing the ups and downs of graduate-student life at USC. I specially thank Rajagopalan Srinivasan and Shiv Prakash.

This thesis would have been very difficult to finish without the friendly environment of the ECE Department, the University of Wisconsin at Madison, my adopted department. I wish to thank Profs. Saluja, Ramanathan, Kime and Chalasani for their kindness and for letting me participate in many departmental activities.

I must not forget my husband Rajiv and parents who have constantly been with me through these years and who I took for granted while I spent year after year in graduate school– I have PHinisheD!

Finally, the financial assistance provided by the J. N. Tata Foundation, Bombay, India and the Zonta Amelia Earhart International Foundation is gratefully acknowledged.

Abstract

While fault-tolerance and dependability requirements are traditionally specified for computer system designs of life-critical and space applications, the large-scale use of computers in the services domain coupled with the size and complexity of these systems has made such specifications commonplace for these applications also. Thus, mathematical modeling has become an integral component in the design of fault-tolerant systems to demonstrate that a given design meets its specifications. Unfortunately, dependability models are, in general, cumbersome and computationally expensive to construct and solve.

In this thesis, we focus on the computational aspect of reliability evaluation of fault-tolerant systems for aerospace applications and private telecommunications networks. In the first case, we analyze two state-space reliability models corresponding to systems with and without repair. Through this analysis we show that for non-repairable systems, the solution is particularly simple but that the more difficult repairable case has several properties which result in computational savings. We conjecture that other fault-tolerant reliability models will exhibit similar properties which can be exploited to design efficient solution techniques. The solution proposed by previous researchers unifies the solution of non-repairable and repairable systems, but is cumbersome. For private network reliability evaluation

through state-space modeling, the first major problem is in model construction. The state space is usually too large even to enumerate and simulation is often used to solve these models. Our approach is to partition the state space and to replace each subset with a single state. We then derive state transition rates for this model from the original (Markov) model. This approach is referred to as *lumping* in the literature. The lumped process is generally non-Markovian but can be approximated by a Markov Chain of the same small size. This approximate Markov model can be solved by standard Markov solvers and network reliability computed. In this thesis, we present a heuristic to construct a lumped Markov Chain. The heuristic takes advantage of differences in time-scale in fault-tolerant system models. The underlying idea in this thesis is to use special properties and symmetries offered by fault-tolerant system models to design computationally efficient solution procedures.

Contents

List Of Figures

List Of Tables

Chapter 1

Introduction

A fault-tolerant computer system is one that can continue to correctly perform its specified tasks in the presence of hardware failures and software errors [Joh89]. The relevance of fault-tolerant computers in aerospace applications and life-critical applications is clear. Indeed, it was the space program in the early years that motivated research in the area of fault-tolerant computers. Since then computers have been playing an increasingly important role in peoples' everyday-lives, for example, in banking, airline reservations, libraries, stock exchanges, telephone systems and office automation. A computer outage in any of these applications affects a large number of people - in some cases with significant economic losses [Ser84] [Sie84] [TM84]. These applications usually employ a large number of processors and the increased failure rate is a serious concern. Thus, it is becoming increasingly common to specify fault-tolerant requirements in the design specification. Traditional fault-tolerant architectures are characterized by their usage of large amounts of resources during normal operation to provide very quick recoveries in case of failure. These methods are clearly justified for life-critical and aerospace applications

where a system failure has catastrophic results. In contrast, in the services domain, it is more appropriate to use as much of the resources as possible towards performance during normal operation. In the event of a failure, however, large portions of resources may be devoted to fault-tolerance while the system operates in a performance-degraded mode. This basic difference in philosophy along with the use of parallel and distributed architectures for high performance has given rise to new issues in fault-tolerant computing for parallel and distributed systems. Traditional fault-tolerant mechanisms may be used in individual nodes of a parallel or distributed system but additional mechanisms to handle more complicated failure modes are employed at a system level. A further discussion on the evolution of concepts in parallel and distributed fault-tolerant computing is available in [KR86]. In this thesis, we analyze some reliability models for aerospace applications and develop new reliability models for an application in the services domain, a private telecommunications network service. In the following paragraphs, we discuss the role of mathematical modeling in the design process, motivate the need for reliability modeling for fault-tolerant systems through an example and present a rationale for reliability evaluation of private telecommunication networks.

Mathematical modeling is an integral component of any design process. For any given design, mathematical models are used to demonstrate/verify that the design meets its specifications. Should the design fail to meet these specifications, it must be modified and evaluated again. Thus, the evaluation tools need to be as efficient as possible. Unfortunately, construction of mathematical models and

their solutions can both be cumbersome and intractable. Thus, for most real-world problems, the design loop remains open and many good designs can go un-noticed.

A fault-tolerant design specification typically includes qualitative design goals such as the *tolerance of single faults* and quantitative *dependability* [Lap85] requirements such as reliability, availability, safety, performability, maintainability and testability. These are probabilistic measures. For example, the *reliability* of a system is the probability that a system operates continuously in a time interval $(0, t]$ given that it operates at time zero and is thus a measure of the system's ability to deliver continuous service [Lap85]; it is a time-varying measure denoted by $R(t)$. Note that reliability can be defined even for non-fault-tolerant systems. Conversely, a fault-tolerant system is not necessarily reliable for a given application [Nel90]. This is not surprising because systems designed to tolerate hardware faults typically do so through the use of redundancy which increases the failure rate of the system; *massive redundancy, stand-by sparing* and *hybrid redundancy* [Bea78] are traditional architectures which employ hardware redundancy for tolerating faults in modules/uniprocessors. We now discuss the characteristics of a Triple Modular Redundancy architecture [Joh89] and illustrate some issues in reliability modeling through this example.

The Triple Modular Redundancy (TMR) architecture is designed to protect the system against a single processor/module failure. In this architecture, the module or processor is triplicated. The three modules are tightly synchronized and outputs sent to a voter which propagates the majority output. This fault-tolerance mechanism uses the notion of *fault-masking* and the system operates until the masking

capability is exceeded. For the TMR system, a second module failure or voter failure causes system failure. In NMR systems, $\lfloor (N-1)/2 \rfloor$ module failures can occur before the masking capability is exceeded. This architecture, and other traditional architectures are suitable for applications requiring high reliability where the cost and additional resources allocated for fault-tolerance during normal operation are justified. The reliability of this architecture has been extensively studied [MA70] and illustrates several characteristics of reliability functions of fault-tolerant systems. Some results are as follows. The reliability of a TMR architecture, $R_{TMR}(t)$, is higher than that of a single module for $t < ln(2) * MTTF$ (Mean-Time-to-Failure of a single module) but lower than that of a single module for times greater than this value. Therefore, a TMR architecture for a given application (which decides t) places a lower bound on the quality of individual modules for the architecture [Joh89] [Sie82]. $MTTF_{TMR}$ is only five-sixth that of a single module. Yet, the TMR architecture and its generalization the NMR are useful fault-tolerant architectures for life-critical intervals of time. But, their usefulness can only be identified by observing their reliability functions; the MTTFs are misleading. The NMR architecture has successfully been used in short-term high-reliability systems such as medical applications, space-shuttle ascent/re-entry [Skl76] and industrial process control applications. A typical requirement for such applications is a 0.9999999 reliability at the end of a three hour period [Joh89]. Similarly, through reliability modeling, the NMR architecture with stand-by sparing was identified to be a good architecture for space applications needing 0.95 reliability at the end of ten years [MA70].

The architectures discussed so far enhance reliability of single modules and are characterized by heavy resource usage during normal operation. Although for parallel and distributed systems, these methods are normally avoided, resource consumption might be acceptable for real-time distributed applications and the idea of fault-masking can be used. Hardware replication is both expensive and reliability demands on individual processors can be unreasonably high for TMR type of architectures to be practical [KR86]. In parallel and distributed computing, the entities voted on are results of replicated tasks running on processors assigned to them and there is no separate voter. Thus, voting must be done in software through the exchange of information between tasks (including the faulty ones) and a consensus reached despite incorrect information floated by faulty processors. Algorithms based on the *Byzantine General's* problem have been proposed and researched [LSP82] [PSL80]. But, these methods result in a large number of task replications and significant communication overheads.

For high-performance parallel and distributed architectures, system-level fault-tolerance is achieved by detecting, containing, locating faults, isolating faulty components, reconfiguring the system, rolling back to a point of fault-free operation and recovering in a full or degraded mode. Each of these system-level steps must be done efficiently and each constitutes an area of research in itself [Pra86]. Each node of the parallel or distributed system can in addition have further protection. The interconnection network in parallel systems and the dedicated communications network in a distributed system become crucial components because system-level fault-tolerance mechanisms depend on information exchange [KR86]. Thus, the

reliability of the underlying communications network or interconnection network is an important issue and application specifications often require high network reliability for the duration of the application. Note that we are concerned about the reliability of a specific application rather than reliability of the entire system.

In a networked environment, re-configuration of tasks in any application is logical, that is, a new processor is assigned to the incomplete task and to other tasks, if necessary. Thus, it is conceivable to decide upon a set of alternate processors for incompletely run tasks and require that a given set of processors be connected reliably for the duration of the application. In this thesis, we develop and analyze reliability models for private networks with alternate routing capabilities. Private networks are logical networks built from dedicated lines leased from a public telecommunications network and meant for the use of a single customer such as a large corporation and can be used as the underlying network for distributed system applications. Further details are presented in Chapter 4. In the next section, we present a review on dependability modeling and previous work in the area. In Section 1.2 we discuss the rationale for analyzing individual reliability models in this thesis and the main contributions.

1.1 Dependability Modeling: Background

In order to evaluate the hardware dependability of any fault-tolerant architecture, the first step is to construct an appropriate behavioral model. A common approach is to assign a *status* to each hardware module. The definition of the status depends on the specific system being modeled and the dependability measure of interest.

For example, the status of a module in a non-repairable system could be "working" and "failed"; in a repairable system a module could have the status "in repair" in addition. A vector of the module-statuses specifies the *state* of the fault-tolerant system. Often, this information can be represented more concisely in terms of the *number* of homogeneous modules in any particular status and/or by removing dependent information.

If all failure and repair rates are assumed to be constant, then, $\{\underline{N}(t); t \geq 0\}$ is a homogeneous continuous time Markov Chain where $\underline{N}(t)$ is the state-vector, the elements of which are independent variables describing the status of the hardware modules. For fault-tolerant models, the state-space is finite, but can get large. The permissible state transitions depend on the dependability measure of interest. The measure is obtained by solving the Markov Chain; sometimes several measures can be obtained by solving a single Markov Chain. Since Markov Chain dynamics are equivalent to the dynamics of linear time-invariant systems, the solution is essentially a solution to a system of ordinary differential equations with constant coefficients and a given initial probability distribution.

The Markov model described above is a simple hardware dependability model of a fault-tolerant architecture. This has been used as the underlying mathematical model for ARIES [Ng76], a tool for reliability and availability analysis [1]. More advanced models have been built since ARIES. ARIES was responsible for unifying most classical reliability models (combinatorial models and reliability integrals [Fle71] [Mat72] [MA70] [RA73]) used for computer system evaluation till the 1970s.

[1] ARIES, developed by Ng and Avizienis was targeted to model very high reliability computer systems for space missions.

Markov models can model complex system behavior that classical models like reliability block diagrams and fault-trees cannot. For example, classical reliability models cannot model repairable systems [2]. Markov models can also be used to derive measures other than reliability. Several new measures have been introduced [Bea78] [Mey80] [Lap85] [GCdSSL86] and conveniently derived from the underlying Markov process and associated Markov Reward processes [ST89]. In this thesis, we such state-space models for network reliability evaluation.

An important concept in reliability modeling is the concept of *coverage*. Coverage is defined as the probability that the system successfully re-configures given that a failure has occurred. Since successful re-configuration is dependent, among other things, on error detection, fault location and switching, the reliability of the mechanisms responsible for these actions has a major impact on the reliability of the system. Coverage is very easily accounted for in a Markov model; in classical models it is messy. Bouricius, Carter and Schneider [BCS69] introduced the coverage parameter and showed that very small changes in this parameter cause significant changes in the reliability function [BCS69]. Further, they showed that higher reliability values are more sensitive to coverage. Arnold [Arn73] confirmed the sensitiveness of reliability of systems with repair. In Chapter 4 of this thesis, we see yet another example of the sensitivity of the reliability function to coverages in network reliability models. Here, coverage parameters are used to model the pattern-sensitive nature of a network's recovery after failure. Modeling of imperfect detection/switching for networks through coverage is available in [RZ90].

[2] Buzacott and Arnold modeled repairable computer systems as Markov Processes to study reliability.

The advances in dependability modeling since ARIES have been in the following directions. In model specification and generation, using for example, Petri-nets [Mur89] [DTGN84] [DT89] [Mol85] [MWT91], in modeling the coverage parameter for accurate estimations [DT89] [McG83], in reduction of the state-space of the model [BT86] [MdG89] [ST87] [SBG79], in relaxing exponentially distributed sojourn time assumptions for states [SBG79], in deriving closed form solution methods symbolic in system parameters and time [Mey82] or symbolic in time only [MRT87] [ST87] and in solution methods for Markov Processes [CS84] [Gra77] [GM84] [MRT87]. A concise comparison of various tools developed with one or more of the the above features is available in [MT83] and [MT86]. For a comprehensive account of numerical methods in transient analysis, see [RST89] [Rei87]. Following is an example which simultaneously estimates the coverage factor and reduces the size of the state space in a reliability model; in numerical solution methods, it eliminates stiffness.

Since the reliability measure is sensitive to coverage, researchers have sought to model this parameter accurately. An immediate consequence of this detailed modeling is that the state space becomes very large. This is due to the fact that the time for recovery is not exponentially distributed and requires several states with exponentially distributed sojourn times to approximate the recovery process in a Markov model. One method of avoiding the problem of state-space explosion has been suggested by Bobbio and Trivedi [BT86] and takes advantage of the differences in the magnitude of the sojourn times in the states representing the fault-handling and recovery process and those representing the failure process. This allows the

detailed model to be decomposed into a failure occurrence model and a fault-and-error-handling sub-model [DT89]. The sub-model is first solved for the coverage parameter and then included in the fault-occurrence model with the fast states accounted for by the coverage parameter. The fault-occurrence model is solved first and is followed by a disaggregation step which assigns appropriate (approximate) probabilities to the fast states. A reliability evaluator called HARP [DTSG86], based on the behavioral decomposition described above offers different sub-models for coverage calculation and uses this value in a fault-occurrence Markov model. McGough and Trivedi [MST85] studied the mechanism of approximating the reliability by the behavioral decomposition method. Their results show that for any semi-Markov reliability model the approximation of fast transitions by instantaneous coverage yields conservative reliability estimates. A similar issue is addressed by Geist et.al [GSB89]; they show that, applied to globally time dependent failure processes (such as the Weibull failure-time distribution), this approximation is not necessarily conservative. In summary, the decomposition method described above allows for accurate estimates of the coverage parameter while simultaneously avoiding the explosion of the state-space for time-homogeneous Markov models. In this thesis, we use coverage to approximately account for behavior we do not wish to model explicitly. With this brief survey, we summarize the motivation and contributions of this thesis, next.

1.2 Motivation and Contributions

In this thesis, we analyze state-space reliability models for high reliability space applications and private network applications. We believe this to be a useful measure to analyze because it is a special case of the performability measure [Mey80], a measure suggested for integrating the performance attribute with the fault-tolerant attribute of any computing system. Further, as noted earlier, it is useful to evaluate the reliability of the underlying communications facility in a distributed system and to study the effect of different alternate routing and repair strategies; the behavior of these reliability functions is not well understood. One reason for this is the size of the state space and the huge computation involved in obtaining reliability. But, reliability models vary vastly in model construction and solution methods, and are easy or difficult depending on the model assumptions and on the level of abstraction sought. Besides, there is no single method for evaluating all reliability models efficiently. Methods capable of solving arbitrary reliability models are cumbersome and often inefficient for solving a specific instance of a reliability model. On the other hand, specialized methods often make too many restricting assumptions and it might be impossible to represent the problem at hand without imposing unnecessary restrictions. For example, the Petri net model is a very general model (representation) of system behavior and is independent of the solution method. (It has the added advantage of serving as a viable interface between the designer and the modeler.) That is, one may solve a problem represented as a Petri net through discrete event simulation and use the same representation to generate a Markov model by restricting the behavior to be Markovian. If the Markovian model is

acceptable, an expensive solution through simulation can perhaps be avoided and the problem solved more efficiently. Similarly, some Markov models can be solved more efficiently than others. Thus, for a given model, one must carefully analyze the underlying assumptions and choose an appropriate solution method. A useful modeling methodology must combine the ease of a Petri Net or a high level representation of system behavior and identify an efficient solution method for the problem at hand. This calls for an automatic and intelligent interface for analyzing the high-level representation and to match it to the most efficient solution method applicable. This matching currently requires the skill of experts in the area. But, efficient *matching* methods suited to automation can only be identified through a good understanding of the issues through a study of individual systems and is the motivation for this thesis. The philosophy of this thesis is as follows.

The behavior of fault-tolerant systems as represented in a mathematical model is not completely arbitrary. A given model is usually rich in *behavioral symmetry* and this can often be exploited to reduce the complexity of the solution considerably. This is analogous to spatial symmetries (regular topologies) which are routinely exploited to simplify the operation of parallel processing systems, interconnection networks and local area networks; sometimes they are instrumental in simplifying behavioral models also [AM91] [ST89]. In Chapters 2 and 3 of this thesis, we carefully analyze the ARIES models for non-repairable and repairable fault-tolerant systems to seek out special properties and symmetries that can simplify the solution and yield computational savings. The **contributions of this thesis** are summarized below.

The contributions of this thesis are in the evaluation of reliability of fault-tolerant computer systems for aerospace applications and private telecommunication networks. The focus is on the reduction of computation through identifying special properties and symmetries offered by fault-tolerant systems. The ARIES model for aerospace applications was proposed by Ng and Avizienis [Ng76] and is one of the earliest models to introduce state dependent coverages and to show that a Markov model could represent all other (combinatorial) models commonly used in the early 70s for reliability modeling. The model construction is particularly simple and so is not an issue in this case. The main contribution of Chapter 2 of this thesis is in proving the diagonalizability of the transition rate matrix of the Markov Chain for ARIES non-repairable systems; a particularly simple version of the ACE algorithm [MRT87] can replace the method proposed in ARIES. While the solution method in [NA80] is applicable to both non-repairable and repairable systems, it is unnecessarily cumbersome for either. In Chapter 3 we demonstrate that several properties of the rate matrix can be exploited to yield a considerable savings in computation.

For problems, such as the problem of network reliability evaluation which are known to be in class #P [Col87] even for simplified versions of the problem, the first major difficulty is the size of the state space. Possible solution methods are: (i) simulation, (ii) lumping. We discuss the two solution options and outline our contribution. For models with large state spaces, it is customary to write a Monte-Carlo simulation program to track the evolution of system behavior. (The mechanics of system behavior can be described compactly using a Petri Net model

or some high-level language.) For models of fault-tolerant systems, a solution by simulation is specially inefficient if done naively because the failure, repair and reconfiguration processes operate on time scales which are at least three orders of magnitude apart. This very disadvantage is being used by researchers in simulation to design efficient simulation methods to solve such models [GSH+92]. Our approach in this thesis is to partition the large state space into subsets and to replace each subset by a single state. We then derive transition rates for this *lumped* process and solve the resulting approximate lumped model by Markovian methods. We have applied this method successfully to develop a heuristic for reliability evaluation of fault-tolerant private networks in Chapter 4. In Chapter 5 we present an analysis of this heuristic. In this analysis, behavioral symmetries take the form of strong and weak Markov lumpability; when symmetries do not exist or are difficult to verify (weak lumpability is difficult to verify), a reasonable approach is to develop a meaningful heuristic through an analysis of the behavioral model. Such a procedure is necessarily approximate since, in the absence of symmetries, it is usually impractical to solve the problem exactly. Lumping is different from *decomposition/aggregation* (as in [Cou77]) in the following respect. In [Cou77], the large Markov Chain is first decomposed into several smaller Markov Chains; each is individually solved and results combined to yield an approximate solution to the original chain. In lumping, the large Markov Chain is reduced to a single Markov Chain of a smaller size and solved in terms of the lumped states. Depending on the application, the state probabilities may or may not have to be distributed among

the original states. Since we are only concerned with reliability models, we do not require the second step in this thesis.

In summary, the philosophy is to detect and use symmetries to advantage wherever possible, but to use meaningful, perhaps approximate methods when properties are hard to detect or known to be non-existent. The thesis is organized as follows. Chapters 2 and 3 are analyses of the ARIES model for non-repairable and repairable systems, respectively. In Chapter 4 we detail our heuristic for approximate Markov model construction for private fault-tolerant networks and follow up with an analysis of the heuristic in Chapter 5. Concluding remarks are presented in Chapter 6.

Chapter 2

On Reliability Modeling of Non-Repairable Fault-Tolerant Computer Systems

2.1 Introduction

A major application of non-repairable fault-tolerant computer systems is in the aerospace industry. Indeed, it was the space program in the early years that motivated research in the area of fault-tolerant computers. In [MT83] the term *Ultrahigh-Reliability* has been used to indicate the stringent reliability requirements demanded by these non-repairable computer systems. With such demands on the reliability measure, mathematical modeling is often the only means of assessing and comparing the reliability of two or more ultra-reliable systems.

Several mathematical models for reliability prediction have been developed and are available as software packages ([BDT+87] [MAG82] [SBG79]). These packages accept a high-level description of a fault-tolerant system and internally generate an equivalent mathematical model of the system. Various dependability measures are

then derived by solving the mathematical model. For a review and critical evaluation of several of these models, we refer the reader to [MT83] and [MT86]. The ARIES model proposed by Ng and Avizienis [NA80] is a unified reliability model developed for evaluating fault-tolerant computer systems. Based on the Markov modeling technique, it unifies most reliability models proposed in the 1970s. Further, the solution for the reliability function $R(t)$ is an analytical expression which is in the form of a weighted sum of pure negative exponentials. Hence, the solution is completely specified by a set of parameter pairs (parameters of the exponentials and their corresponding multipliers) which depend on the fault-tolerant system under evaluation. However, the solution is of this form only if the state transition rate matrix of the Markov Chain is diagonalizable.

Let $\mathcal{B}$ denote the transition rate matrix. The solution approach in ARIES is to compute the matrix function $e^{\mathcal{B}t}$ using the Lagrange-Sylvester Interpolation formula. The state probability vector is then given by $\underline{P}(t) = e^{\mathcal{B}t}\underline{P}(0)$ and the reliability function by $R(t) = \sum_{i \in states} P_i(t)$. From standard matrix theory, if the eigenvalues of $\mathcal{B}$ are distinct, then, $\mathcal{B}$ is diagonalizable so that the minimal polynomial of $\mathcal{B}$ has distinct roots. For this case, the Lagrange-Sylvester formula[1] is a valid solution technique. If the eigenvalues are not distinct, however, $\mathcal{B}$ may or may not be diagonalizable. For this case of repeated eigenvalues, ARIES considers only the distinct eigenvalues (ignoring duplicates) and solves as in the previous case. Since this is a valid approach only when $\mathcal{B}$ is diagonalizable, it is not clear that the ARIES results are correct for all systems with repeated eigenvalues. In

[1]This formula, described by Equation 2.2 in this chapter, is applicable only if $\mathcal{B}$ is diagonalizable.

[MT83], Geist and Trivedi have given an example of an ARIES non-repairable fault-tolerant system in which the eigenvalues of the transition rate matrix are not distinct. Further, they have observed that the set of eigenvectors of the transition rate matrix are independent (i.e. $\mathcal{B}$ is diagonalizable) and that classical solution methods yield correct results. In this chapter we prove that for ARIES non-repairable systems, $\mathcal{B}$, the transition rate matrix is always diagonalizable so that the Lagrange-Sylvester formula is applicable despite repeated eigenvalues. Further, since our proof guarantees that the rate matrix is always diagonalizable, general methods for solving arbitrary Markov Chains can be tailored to solve the ARIES model efficiently.

2.2 ARIES Model and Solution Technique for Non-Repairable Fault-Tolerant Systems

We begin with a brief description of the ARIES for non-repairable fault-tolerant systems (for a detailed description, see [Ng76] and [NA80]). The input to ARIES consists of a set of parameters which completely specifies the fault-tolerant system. The definition of these parameters [Ng76] is reproduced here for easy reference.

N = Initial number of modules in the active configuration

S = Number of spare modules

D = Number of degradations allowed in the active configuration

C_a = Coverage factor for recovery from active-module failures

C_d = Coverage factor for recovery from spare-module failures

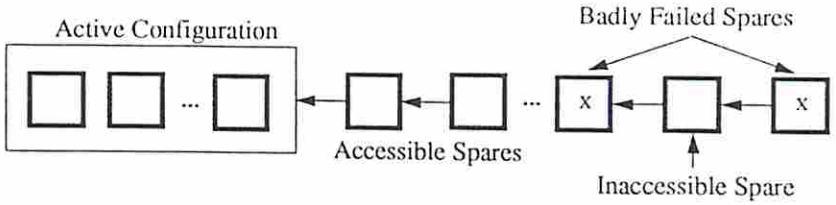


Figure 2.1: Arrangement of Spares in ARIES Non-repairable Systems.

λ = Failure rate of one active module

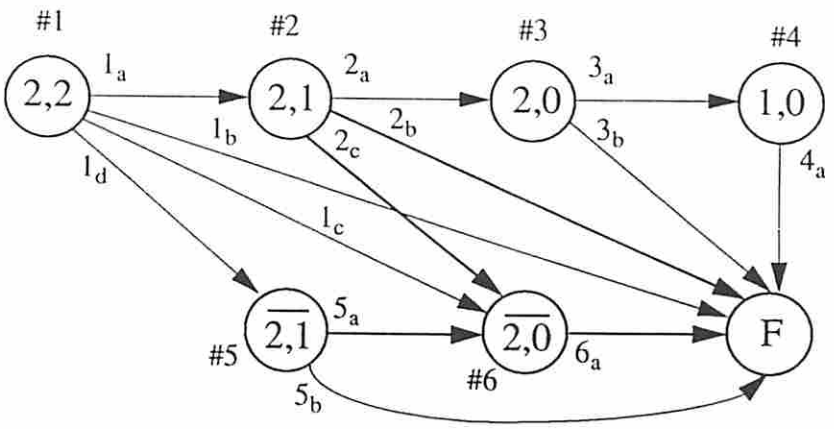
μ = Failure rate of one spare

$\underline{Y}$ = An integer vector which defines the number of modules in successive active configurations after successfully recovering from failure

$\underline{CY}$ = Coverage Vector for active and degraded configurations

Using the above input, the ARIES software package generates a Markov reliability model. ARIES assumes that the active configuration is a set of homogeneous modules and that all spares are identical. Both active modules and spares can fail but the failure rate of the spares may be different from that of the active modules. It is possible for a module (active or spare) to fail without being detected.

As shown in Figure 2.1, spares are switched into the active configuration in a linear fashion. Also, there could be failed but undetected spares in the linear array. Such spares are termed *badly* failed. The first badly failed spare in the linear array determines the number of working spares accessible to the fault-tolerant system. The state of the Markov Chain for ARIES non-repairable systems is an ordered pair where the first element denotes the number of active modules and the second



$1_a : 2\lambda C_a + 2\mu C_d$ $\quad 2_a : 2\lambda C_a + \mu C_d$ $\quad 3_b : 2\lambda(1 - C_a)$

$1_b : 2\lambda(1 - C_a)$ $\quad 2_b : 2\lambda(1 - C_a)$ $\quad 4_a : \lambda$

$1_c : \mu(1 - C_d)$ $\quad 2_c : \mu(1 - C_d)$ $\quad 5_a : 2\lambda C_a + \mu$

$1_d : \mu(1 - C_d)$ $\quad 3_a : 2\lambda C_a$ $\quad 5_b : 2\lambda(1 - C_a)$

$6_a : 2\lambda$

Figure 2.2: Example of an ARIES Markov Chain.

element denotes the number of spares available to the system. It is assumed that there is exactly one absorbing state in the Markov Chain, the failed state F. The transition rates between states are derived from the failure rates of the system in its various configurations. Figure 2.2 shows an example to illustrate the structure of the ARIES Markov Chain. The input which translates to this structure is $(2, 2, 1, C_a, C_d, \lambda, \mu, \underline{1}, \underline{C_a})$ where the elements of the vector correspond to N, S, D etc. respectively.

Note that in the Markov Chain of Figure 2.2 there are pairs of states with identical configuration (one state of each pair has an overbar in the figure). These pairs are not merged together because states with the overbar correspond to a

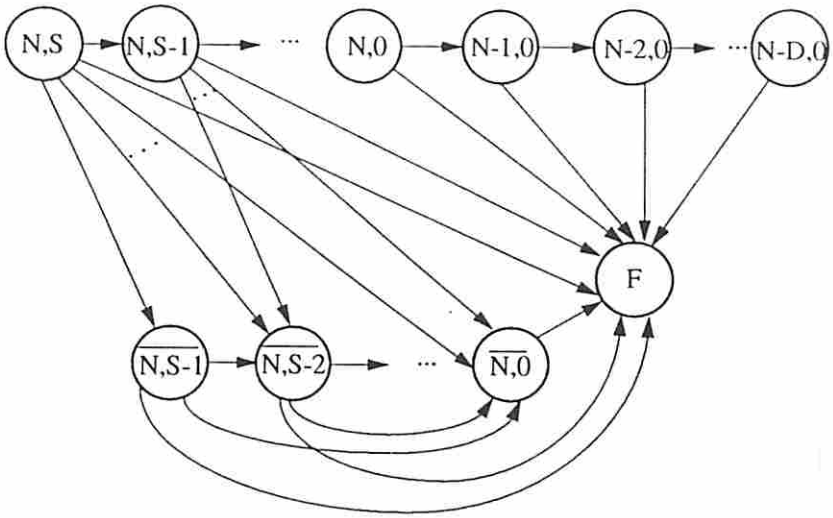


Figure 2.3: Markov Chain for all ARIES Non-Repairable Fault-Tolerant Systems.

situation where the system has badly failed spares in the linear array. In the presence of a badly failed spare, the system can never be *working* with an active configuration less than N (N is the number of working modules in the active configuration) because ARIES conservatively assumes that the entire system fails if a badly failed spare is switched into the active configuration. This essentially means that the degradation capability is lost once the system has an undetected bad spare. For systems which are not permitted to degrade, pairs with identical configurations are mergeable.

Figure 2.3 displays the structure of the Markov Chain for all non-repairable fault-tolerant systems represented by the ARIES model. From standard Markov modeling theory [Tri82], the probability that the system is in state $i, 1 \leq i \leq n-1$ at time t (state n is the failed state) is obtained by solving the following system of differential equations:

$$\frac{d}{dt}\underline{P}(t) = \mathcal{B}\underline{P}(t) \tag{2.1}$$

subject to the initial condition

$$\underline{P}(0) = [P_1(0), P_2(0), ...P_{n-1}(0)]^T = [1, 0, ...0]^T$$

The above initial condition is due to the fact that ARIES assumes all active modules and spares to be working at time $t = 0$. Note that $\mathcal{B}$ is the rate matrix formed by excluding the failed state. As described in [Ng76], the solution to the model is not affected by excluding the failed state. Finally, the solution to Equation 2.1 suggested in ARIES is given by the following equation:

$$\underline{P}(t) = \left[\sum_{i=1}^{m} e^{\sigma_i t} \prod_{j=1, j\neq i}^{m} \frac{\mathcal{B} - \sigma_i \mathcal{I}}{\sigma_j - \sigma_i}\right] \underline{P}(0) \tag{2.2}$$

where, $\sigma_i, (1 \leq i \leq m)$ denote the eigenvalues of $\mathcal{B}$. If multiple eigenvalues exist, then, only distinct eigenvalues are used. Summing over all non-failed states, the reliability function $R(t)$ of the non-repairable fault-tolerant system is derived as:

$$R(t) = \sum_{i=1}^{n-1} P_i(t) \tag{2.3}$$

$\mathcal{B}$ is triangular for any non-repairable system because the ARIES model does not consider compensating failures. Hence, the eigenvalues of the $\mathcal{B}$ matrix appear along its diagonal; eigenvalues may repeat. Equation 2.2 is a valid solution method

provided that $\mathcal{B}$ is diagonalizable. In such a case duplicate eigenvalues may be ignored; otherwise, Equation 2.2 is not applicable.

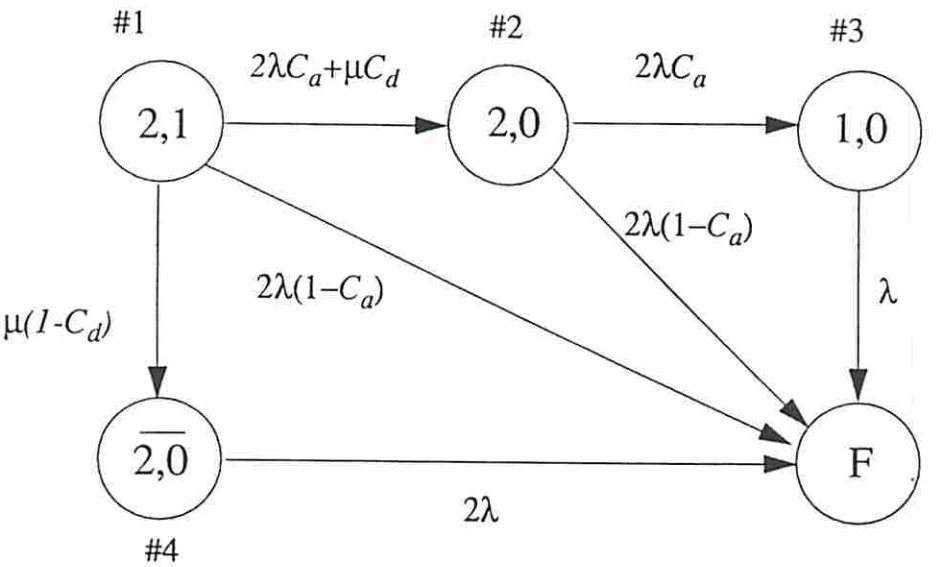
2.2.1 Repeated Eigenvalues

In [MT83], Geist and Trivedi have pointed out that the assumption of distinct eigenvalues and the exponential failure distribution significantly restricts the applicability of the ARIES reliability model. Through an example of a simple system with 2 active units and 1 spare, they illustrate the ease with which systems with multiple eigenvalues can arise in practice. The example is reproduced in Figure 2.4. Further, they show for this example that, although the eigenvalues are not distinct, the eigenvectors are independent and so a standard solution technique for a Continuous Time Markov Chain by diagonalization of $\mathcal{B}$ yields correct results.

In fact, from Figure 2.3 it is clear that for any non-repairable system in which the active configuration is permitted to degrade, i.e., D is greater than zero in the input specification, there are as many pairs of states with identical configurations as there are spares. Thus, a large number of non-repairable fault-tolerant systems represented by the ARIES model have repeated eigenvalues.

2.3 Analysis of the Markov Chain

In this section we prove that $\mathcal{B}$, the rate matrix of any non-repairable system modeled by ARIES is always diagonalizable. Consequently, Equation 2.2 is a valid solution technique for any non-repairable ARIES system. The outline of the proof is as follows. Through a careful study of the structure of the Markov Chain given



Eigenvalues Eigenvectors

$-(2\lambda+\mu)$	$[1, -(2C_a + C_d\mu)/\mu, 2\lambda C_a(2\lambda C_a + \mu C_d)/(\lambda+\mu)\mu, --]$
-2λ	$[0, 1, -2C_a, 0]$
-2λ	$[0, 1, -2C_a, 1]$
$-\lambda$	$[0, 0, 1, 0]$

Figure 2.4: An Example with Repeated Eigenvalues.

in Figure 2.3 and from results in Transform Theory [ZD63], we establish that the Jordan Canonical Form of $\mathcal{B}$ is a diagonal matrix. Since Figure 2.3 represents all non-repairable fault-tolerant systems that ARIES models, we infer that the rate matrices of all non-repairable fault-tolerant systems modeled by ARIES are diagonalizable. If $\mathcal{B}$ is diagonalizable, then, from matrix theory [Str80] we have that the roots of the minimal polynomial of $\mathcal{B}$ are always distinct . Finally, we note that Equation 2.2 is applicable when the roots of the minimal polynomial of $\mathcal{B}$ are distinct [ZD63]. We now present a detailed analysis.

2.3.1 Properties of the Markov Chain Structure

Techniques for finding the transient solution for the state probabilities of arbitrary Markov models are very cumbersome in practice. In some cases, however, the structure of the Markov Chain permits a simple solution. In this chapter we indicate a solution technique which takes advantage of the structure of the Markov Chain for non-repairable ARIES fault-tolerant systems and use it to prove that the rate matrix $\mathcal{B}$ is diagonalizable.

We now list some of the properties of the Markov Chain (see Figures 2.2 and 2.3). The graph theoretic term *node* and the term *state* of the Markov Chain are used interchangeably. From the graphical structure of the Markov Chain we observe the following properties:

1. There is exactly one source node in the graph. This is the initial state (N, S) of the system.

2. The graph is directed and acyclic.

3. Let $(i,j),(k,l) \in I$ where, I denotes the state space of the Markov Chain excluding the failed state. For any two nodes $(i,j),(k,l)$ joined by a directed arc, either $(i = k = N, j > l)$ or $(i > k, j = l = 0)$. In either case, the transition results in a *smaller* configuration.

4. There is at least one path from the source to every node in the graph.

5. The total transition rate at node $(i,j) \in I$ is given by $\sigma_{(i,j)} = i\lambda + j\mu$.

6. Let $\wp(i,j) = \{p_1, p_2, p_3 ... p_q ... p_l\}$ be the set of distinct (not necessarily disjoint) paths from the source to node (i,j). Let $(n_{p1}, n_{p2}, ..., n_{pq}, ..., n_{pl})$ be the number of nodes (inclusive of source and destination nodes) on paths $(p_1, p_2, ..., p_q, ..., p_l)$ respectively. Also, let $\sigma_r, (1 \leq r \leq n_{pq}, 1 \leq q \leq l)$ be the total transition rate at node r on a path p_q. Then, in the sequence $\sigma_1 \sigma_2 ... \sigma_{n_{pq}}$ along any path p_q, no two rates are repeated. This can be established by observing that Properties 3 and 5 above are satisfied at each node on path p_q when starting from the source.

2.3.2 Solution in the Laplace Domain

In this section the first two theorems show that the structure of the Markov Chain described above permits a recursive transient solution for state probabilities in the Laplace Domain. The last theorem establishes that the roots of the minimal polynomial of the transition rate matrix of every non-repairable fault-tolerant system are always distinct. Following is a definition of some of the terms used in these theorems.

I = $\{(i,j) : i =$ number working active modules, $j =$ number of accessible spares $\}$

i.e. the finite set of non-failure states of the Markov Chain

$P_{(i,j)}(t)$ = Probability non-repairable system is in state (i,j) at time t

$\overline{P}_{(i,j)}(s)$ $\Leftrightarrow$ $\mathcal{L}\{P_{(i,j)}(t)\}$ are a Laplace Transform pair

$U(i,j)$ = $\{(k,l) : (k,l)$ is an upstream neighbor of $(i,j)\}$

i.e., the set of upstream neighbors of (i,j)

$f^a{}_{(k,l):(i,j)}$ = fraction of rate $k\lambda$ contributing to the transition rate of the $(k,l) \rightarrow (i,j)$ transition

$f^s{}_{(k,l):(i,j)}$ = fraction of rate $l\mu$ contributing to the transition rate of the $(k,l) \rightarrow (i,j)$ transition

$\wp(i,j)$ = Set of distinct paths from the source to node (i,j)

$\mathcal{N}_p^{(i,j)}$ = Set of nodes (u,v) along a path p from the source to node (i,j), inclusive of source and destination

Theorem 3.2.1 *In the Laplace Domain, the transient solution for the Markov Chain of any non-repairable fault-tolerant system modeled by ARIES is given by the following recurrence relation:*

$$\begin{aligned}\overline{P}_{(i,j)}(s) &= \frac{1}{s+i\lambda+j\mu}\sum_{(k,l)\in U(i,j)}\left[f^{a}{}_{(k,l):(i,j)}k\lambda + f^{s}{}_{(k,l):(i,j)}l\mu\right]\overline{P}_{(k,l)}(s), (i,j)\\ &\neq (N,S)\end{aligned}$$

$$\overline{P}_{(i,j)}(s) = \frac{1}{s+N\lambda+S\mu}, \quad (i,j) = (N,S) \tag{2.4}$$

Proof: Using standard Markov modeling theory for CTMC, the state probability $P_{(i,j)}(t)$ is characterized by the following differential equation [Tri82]:

$$\frac{d}{dt}P_{(i,j)}(t) = -(i\lambda+j\mu)P_{(i,j)}(t) + \sum_{(k,l)\in U(i,j)}\left[f^{a}{}_{(k,l):(i,j)}k\lambda + f^{s}{}_{(k,l):(i,j)}l\mu\right]P_{(k,l)}(t) \tag{2.5}$$

Applying the Laplace Transform to both sides of the above equation, the transient solution in the Laplace Domain is given by :

$$s\overline{P}_{(i,j)}(s) - P_{(i,j)}(0) = -(i\lambda+j\mu)\overline{P}_{(i,j)}(s) +$$

$$\sum_{(k,l)\in U(i,j)}\left[f^{a}{}_{(k,l):(i,j)}k\lambda + f^{s}{}_{(k,l):(i,j)}l\mu\right]\overline{P}_{(k,l)}(s) \tag{2.6}$$

Since

$$P_{(i,j)}(0) = \begin{cases} 1, & (i,j) = (N,S) \\ 0, & \text{otherwise.} \end{cases}$$

and

$$U(N,S) = \phi,$$

the theorem follows from Equation 2.6. ■

Corollary 3.2.1 *If $\wp(i,j)$ is the set of all distinct (not necessarily disjoint) paths from the source to node (i,j), then for $(i,j) \neq (N,S)$, $\overline{P}_{(i,j)}(s)$ can be expressed as*

$$\sum_{p \in \wp(i,j)} \frac{K_p}{\prod_{(u,v)}(s + u\lambda + v\mu)}$$

where, $(u,v) \in \mathcal{N}_p^{(i,j)}$ and K_p is some constant whose value is a function of path $p \in \wp(i,j)$.

Proof: By Property 2 of Section 2.3.1 we know that the reliability model for non-repairable systems can be represented by a directed, acyclic graph. By Property 4 of the same section, we know that there is at least one path from the source to every node in the graph. By expanding the recurrence relation for $\overline{P}_{(i,j)}(s)$ in Equation 2.4, it is possible to obtain an expression for $\overline{P}_{(i,j)}(s)$ exclusively in terms of the Laplace Transforms of nodes that are upstream from (i,j).

The expanded expression for $\overline{P}_{(i,j)}(s)$ is in the sum-of-products form. The number of terms in the expression is equal to the number of distinct paths from the source to node (i,j). Each product term is given by

$$\frac{K_p}{\prod_{(u,v)} s + u\lambda + v\mu}, \quad (u,v) \in \mathcal{N}_p^{(i,j)}, \quad (i,j) \neq (N,S) \tag{2.7}$$

where,

$$K_p = \prod_{(c,d)} \left[f^a{}_{(c,d):(e,f)} c\lambda + f^s{}_{(c,d):(e,f)} d\mu \right],$$
$$(c,d) \in \{\mathcal{N}_p^{(i,j)} - (i,j)\},$$

(e,f) is the downstream neighbor of (c,d) on path p.

Summing Equation 2.7 over all $p \in \wp(i,j)$ gives the required expression. ■

Theorem 3.2.2 *The partial fraction expansion of* $\overline{P}_{(i,j)}(s)$ *is of the form*

$$\overline{P}_{(i,j)}(s) = \sum_{(x,y)} A_{(x,y)}^{(i,j)} \frac{1}{s + \sigma_{(x,y)}}$$

where,

$$\sigma_{(x,y)} = x\lambda + y\mu, \quad (x,y) \in \bigcup_p \mathcal{N}_p^{(i,j)}, \quad p \in \wp(i,j)$$

Proof: Corollary 3.2.1 shows that $\overline{P}_{(i,j)}(s)$ can be expressed as sum-of-products where each product term is as in Equation 2.7. By Property 6 in Section 2.3.1, we

know that no two rates along any path from the source to node (i,j) are repeated. Hence, the partial fraction expansion for each product term is of the form

$$\sum_{(u,v)} \frac{C_{(u,v)}}{s+\sigma_{(u,v)}}, \quad C_{(u,v)} = K_p \frac{s+\sigma_{(u,v)}}{\Pi_{(x',y')}(s+\sigma_{(x',y')})}\Bigg|_{s=-\sigma_{(u,v)}}, \quad (u,v),(x',y') \in \mathcal{N}_p^{(i,j)} \tag{2.8}$$

In the above equation, $C_{(u,v)}$ are the constants obtained by partial fraction expansion of Equation 2.7. Note that every product term corresponds to a distinct path from the source to node (i,j). Therefore, by summing Equation 2.8 over all paths $p \in \wp(i,j)$ we get the expression stated in the theorem.

Corollary 3.2.2 *The solution to the ARIES model for non-repairable fault-tolerant systems is a weighted sum of pure exponentials.*

It is clear that the Inverse Laplace Transform of $\overline{P}_{(i,j)}(s)$ in Theorem 3.2.2 yields this result. ■

Theorem 3.2.3 *For every non-repairable fault-tolerant system modeled by ARIES, the rate matrix $\mathcal{B}$ is diagonalizable so that*

$$\mathcal{B} = \mathcal{S}\Lambda\mathcal{S}^{-1}$$

where $\mathcal{S}$ is a matrix whose columns are the eigenvectors of $\mathcal{B}$ and Λ is a diagonal matrix whose elements are the eigenvalues of $\mathcal{B}$.

Proof: From the form of the partial fraction expansion of $\overline{P}_{(i,j)}(s)$ (or equivalently, from the form of the solution $P_{(i,j)}(t)$), we gain an insight into the Jordan

Canonical Form of the transition rate matrix $\mathcal{B}$ [ZD63]. We note that the partial fraction expansion (see Theorem 3.2.2) contains only terms of the form $\frac{1}{(s+\sigma)}$ in s. This indicates that the Jordan Matrix corresponding to $\mathcal{B}$ is a diagonal matrix of its eigenvalues. From this result we infer that for every non-repairable fault-tolerant system modeled by ARIES, the rate matrix $\mathcal{B}$ has a set of linearly independent eigenvectors. From standard matrix theory, then, it follows that $\mathcal{B} = \mathcal{S}\Lambda\mathcal{S}^{-1}$.

■

From matrix theory, the roots of the minimal polynomial of a diagonalizable matrix are always distinct. Also, from matrix theory [ZD63], the solution technique specified in Equation 2.2 is applicable if the roots of the minimal polynomial of the transition rate matrix $\mathcal{B}$ are distinct. This proves that Equation 2.2 is a valid approach for solving the Markov Chain for non-repairable ARIES fault-tolerant systems.

Theorem 3.2.2 suggests an algorithm to solve the Markov Chain. By this theorem, the state probability $P_{(i,j)}(t)$ is a weighted sum of pure exponentials; the parameters of the exponentials are the poles of the corresponding Laplace Transform. As seen in the proof of Theorem 3.2.2, the parameters and weights associated with the exponential terms can be obtained by a graph traversal of the Markov Chain structure. This is similar to the ACE algorithm [MRT87] which solves any acyclic Markov Chain (i.e., rate matrices need not be diagonalizable). However, here we take advantage of the diagonalizability of the rate matrix to simplify the procedure.

2.4 An Example

In this section we evaluate the reliability function for the non-repairable fault-tolerant system shown in Figure 2.2. The associated rate matrix has repeated eigenvalues. For this example, the ARIES software package produces results which are identical with those obtained by the the Laplace Transform method. The example serves to illustrate the validity of using Equation 2.2 to solve non-repairable ARIES systems.

The solution to the Markov Chain presented in Figure 2.2 and using the Transform method is given by the following set of expressions:

$$p_1(t) = e^{-(2\lambda+2\mu)t}$$

$$p_2(t) = \frac{1_a}{\mu}\left[e^{-(2\lambda+\mu)t} - e^{-(2\lambda+2\mu)t}\right]$$

$$p_3(t) = \frac{1_a.2_a}{2\mu^2}\left[e^{-2\lambda t} - 2e^{-(2\lambda+\mu)t} + e^{-(2\lambda+2\mu)t}\right]$$

$$p_4(t) = \frac{1_a.2_a.3_a}{2\mu^2}\left[\left\{\frac{1}{\lambda} - \frac{2}{\lambda+\mu} + \frac{1}{\lambda+2\mu}\right\}e^{-\lambda t} - \frac{1}{\lambda}e^{-2\lambda t} + \frac{2}{\lambda+\mu}e^{-(2\lambda+\mu)t} - \frac{1}{\lambda+2\mu}e^{-(2\lambda+2\mu)t}\right]$$

$$p_5(t) = \frac{1_d}{\mu}\left[e^{-(2\lambda+\mu)t} - e^{-(2\lambda+2\mu)t}\right]$$

$$p_6(t) = \frac{(1_d.5_a+1_a.2_c)}{2\mu^2}\left[e^{-2\lambda t} - 2e^{-(2\lambda+\mu)t} + e^{-(2\lambda+2\mu)t}\right] + \frac{1_c}{2\mu}\left[e^{-2\lambda t} - e^{-(2\lambda+2\mu)t}\right]$$

The parameter constant pairs[2] obtained from the above expressions for input parameter values as given in Equation 2.9 are $(-1, 2.916)$, $(-2, -3.518)$, $(-3, 2.088)$, $(-4, -0.486)$ so that

$$R(t) = 2.916e^{-t} - 3.518e^{-2t} + 2.088e^{-3t} - 0.486e^{-4t}$$

The input parameter set provided to the ARIES software package is as follows:

$$(N, S, D, C_a, C_d, \lambda, \mu, \underline{Y}, \underline{CY}) = (2, 2, 1, 0.9, 0.9, \underline{1}, \underline{0.9}) \tag{2.9}$$

For this input the ARIES software package generates exactly the same parameter constant pairs as above.

2.5 Conclusions

The main contribution of this chapter is in proving that the state transition rate matrix is diagonalizable for all non-repairable fault-tolerant systems modeled by ARIES. Hence, the Lagrange-Sylvester formula currently used in ARIES is a valid solution approach. We indicate that as a consequence of this proof, the transient solution of the Markov Chain for ARIES non-repairable systems can be efficiently computed, for example, by a simplified version of the ACE algorithm proposed in [MRT87].

[2]The pair (a, b) denotes be^{at} where a is a distinct eigenvalue of the rate matrix.

As mentioned in the introduction, several mathematical models for predicting the behavior of fault-tolerant systems (for example, ARIES, CARE-III, HARP, SAVE and SURF) are currently available. These models have different capabilities and limitations. It is therefore recommended that the system be evaluated by more than one tool. For a discussion of how two or more tools such as ARIES and HARP complement each other in evaluating a fault-tolerant system, we refer the reader to [RL89].

Chapter 3

An Analysis of a Reliability Model for Repairable Fault-Tolerant Systems

3.1 Introduction

While fault-tolerant computer systems for airborne applications are non-repairable, there are several high reliability fault-tolerant systems for which repair is possible. Such applications include real-time systems for telecommunications and telemetry, medical applications and process control. One is tempted to believe that because of the repair facility, these systems should operate forever. But, even in the presence of a perfect repair facility, there is a non-zero probability that the fault-detection mechanism fails to detect a fault in the system or that the system does not reconfigure itself successfully after a failure is detected. In this case, the system fails because no corrective action is possible. One of the dependability measures for repairable fault-tolerant systems, then, is the probability that the system operates continuously in an interval $[0, t]$, when it is known to operate at $t = 0$, that is, the reliability of the repairable fault-tolerant system.

In order to evaluate the reliability of a fault-tolerant system, several researchers have abstracted the system in terms of an analytic or simulation model. A number of prediction tools based on these models have emerged in recent years. Examples of some software packages are ARIES [Ng76], HARP [BDT+87], SAVE [GCdSSL86], SURF [CDLL81] and SURE [But84]. The earliest of these was ARIES which models closed (non-repairable) as well as repairable fault-tolerant systems as a finite homogeneous continuous time Markov Chain. In this chapter we focus on the ARIES model for repairable fault-tolerant systems. For a summary of the other models we refer the reader to [MRT87] [MT83] [MT86].

An important feature of the ARIES model is that it unifies several classical reliability models. Another desirable feature is that the reliability of the system is obtained as an expression in terms of time t so that it is not necessary to solve the model for each value of t. Hence, any improvement in the solution technique in terms of computational complexity, accuracy of results, verification of the correctness of model construction or the applicability of the solution technique would be of practical use. In this chapter we analyze the properties of the ARIES model for repairable systems and, drawing from well established results in matrix theory, suggest an efficient solution for computing the reliability of the repairable system. We also present an example of a repairable system for which the ARIES solution technique is not applicable; this example serves to illustrate that the ARIES solution method is not as general as to solve every repairable fault-tolerant system that it models. Further, we show that the states of the Markov Chain can be graphically represented so that there is a one-to-one correspondence between the

graphical construction of the Markov Chain and the behavior of the fault-tolerant system as abstracted by the ARIES model. With this systematic layout it is very easy to verify the correctness of the Markov Chain construction. Currently, this is a tedious task and much prone to error for all but systems of very small size.

The solution technique for the Markov Chain proposed in the ARIES model consists of finding the eigenvalues of the State Transition Rate Matrix (STRM) of the fault-tolerant system and then applying the Lagrange-Sylvester interpolation formula [Gan77a]. The Markov Chain structure of any non-repairable fault-tolerant system modeled by ARIES is *acyclic* (in a graphical sense). Consequently, the eigenvalues appear along the diagonal of the STRM; for the repairable case, the eigenvalues must be computed since the Markov Chain structure contains cycles. The Lagrange-Sylvester Interpolation formula is applicable only to matrices with an independent set of eigenvectors [Gan77a]. This formula has been suggested as a solution to the ARIES model based on an assumption that in practice (the STRMs of) fault-tolerant systems have distinct eigenvalues [1]. This has been one of the objections to the ARIES method of solution.

It has been pointed out in [MT83] and [BR90] that many closed fault-tolerant systems modeled by ARIES have repeated eigenvalues. However, in [BR90] it has been proved that the eigenvectors of the STRMs of all closed systems modeled by ARIES are independent even in the presence of repeated eigenvalues. Therefore, the Lagrange-Sylvester Interpolation formula is a valid approach for solving ARIES model for any closed systems. Despite its applicability, the computational

[1] For repeated eigenvalues, the eigenvectors are independent if and only if the minimal polynomial has distinct roots.

complexity and numerical stability aspects of this method may render it impractical.

For the repairable case also, ARIES uses this interpolation formula assuming distinct eigenvalues. In Section 3.6 we show that for fault-tolerant systems with a single spare there exists a class of repairable systems with repeated eigenvalues for which the eigenvectors are *not* independent. For these systems, the Lagrange-Sylvester formula is not applicable. We also show that for all non-degradable repairable systems with a single spare, if the eigenvalues are repeated, then, the eigenvectors are automatically dependent. However, as described in Section 3.6, repeated eigenvalues are unlikely to occur in any practical fault-tolerant system with one spare. It is not clear at this stage whether these properties extend to repairable fault-tolerant systems with more than one spare.

In ARIES, the STRM for repairable systems is generated algorithmically and the graphical representation of the Markov Chain appears to be quite complicated [Ng76, page 129]. Also, the numbering rule does not cause the STRM to be in any special form. In this chapter, in Section 3.3 we observe that the Markov Chain for the ARIES repairable system exhibits a very regular structure and that this structure can be translated systematically into graphical form. Then, with a proper numbering of the states, we show that it is possible for the corresponding STRM to take on the Hessenberg form. Further, we observe that the STRM is quasi-triangular and that each diagonal block is tri-diagonal and quasi-symmetric

[2]. These are very desirable properties from the point of view of eigenvalue computation since there are highly efficient numerical methods available for computing the eigenvalues of such matrices. Currently, ARIES uses the well-known QR algorithm for computing the eigenvalues of the STRM of size $n = O(S^3)$, where S is the number of spares in the fault-tolerant system. When preceded by a pre-conditioning step which requires $O(S^9)$ multiplications, the QR algorithm has a complexity of $O(S^6)$ per iteration with (essentially) quadratic convergence. The function of the pre-processing step is to reduce the STRM to the Hessenberg form. In Section 3.4 of this chapter, we show that as a direct consequence of the properties of the STRM, the complexity of the QR algorithm can be reduced to $O(S^2)$ per iteration with cubic convergence. By our numbering rule for the states of the Markov Chain, the STRM is already in Hessenberg form and eliminates the need for the expensive pre-processing step. We however prepare the STRM for the QR algorithm by applying a diagonal similarity transformation to each of the quasi-symmetric tri-diagonal blocks of the STRM. This is an $O(S^2)$ procedure and converts each block into a real symmetric tri-diagonal matrix. In Section 3.4 we show that this pre-conditioning allows us to employ the most efficient version of the QR algorithm available for eigenvalue computation.

The complexity of computing the state probability vector $P(t)$ by using the Lagrange-Sylvester Formula is $O(S^{12})$; in the ARIES implementation this complexity is $O(S^{15})$. As has been pointed out in [ML78] and [MT83], this is a prohibitively expensive approach and can handle only a very small number of spares.

[2]These and other matrix theoretic terms are defined at relevant points in this chapter.

Also, this method could have problems of numerical stability [ML78] [MT83]. In Section 3.5, once again, the properties of the STRM permit an efficient solution. We show that by using the Hessenberg-Schur algorithm for solving the matrix equation $\mathcal{A}\mathcal{X} + \mathcal{X}\mathcal{B} = \mathcal{C}$ [HV79], the complexity of eigenvector computation of the STRM is $O(S^7)$ for the distinct eigenvalue case. In order to achieve this complexity for eigenvector computation, an additional computational effort is required while computing the eigenvalues. This, in effect makes the QR algorithm complexity $O(S^4)$ per iteration and the complexity of the pre-processing step to $O(S^3)$. Finally, if $\mathcal{R}$ is the matrix of eigenvectors of the STRM and Λ the diagonal matrix of eigenvalues, then, using a classical method for solving linear time-invariant systems, the state probability vector $P(t)$ and the reliability function $R(t)$ are obtained by the following expressions: $P(t) = \mathcal{R}e^{\Lambda t}\mathcal{R}^{-1}P(0)$ and $R(t) = \sum_i P_i(t)$. For $P(0) = (1, 0, \cdots, 0)$, it is sufficient to compute the first column of $\mathcal{R}^{-1}$. If the eigenvectors are already computed, we show that $P(t)$ can computed using $O(S^7)$ operations. Thus, by our method, the complexity of computing the reliability of ARIES repairable fault-tolerant systems with distinct eigenvalues is $O(S^7)$. This is an improvement by a factor of S^8 over the existing implementation. For systems with repeated eigenvalues, more general methods for solving Markov Chains must be used.

The chapter is organized as follows. Section 3.2 is a brief description of the ARIES model for repairable fault-tolerant systems. Section 3.3 details the structure and construction of the Markov Chain. In Section 3.4 we present the properties of the eigenvalues of the STRM and a procedure for computing the eigenvalues

efficiently. This is followed in Section 3.5 by an efficient method for eigenvector computation. In Section 3.5.4 we compute the state probability vector $P(t)$ and the reliability function $R(t)$. The class of repairable fault-tolerant systems with repeated eigenvalues and dependent eigenvectors is presented in Section 3.6 and conclusions in Section 3.7. Since all our solution techniques are based on results from matrix theory, we list some of the more important results in the appendix.

3.2 The ARIES Model for Repairable Systems

ARIES models both non-degradable as well as degradable fault-tolerant systems. These may be closed or repairable. In this chapter we restrict ourselves to the repairable systems and here, we present a brief description of the model. The behavior of a repairable fault-tolerant system (degradable or non-degradable) according to the ARIES model is best described by Figure 3.1 taken from [Ng76].

The ARIES model assumes that all modules in the active configuration and therefore, all spares are identical. The number of modules in the active configuration depends on the state of degradation of the system [3]. If K is the number of degradations that the system has undergone, then, the number of modules in the active configuration is specified by the K^{th} element $\underline{Y}[K]$ of the vector $\underline{Y}$. According to this notation, N, the initial number of modules in the active configuration is identical with $\underline{Y}[0]$. For a degradable system, there are *working* states of the system with $\underline{Y}[K]$ number of active modules for $K = 0, 1, \cdots, D$. Here, D is the

[3]The non-degradable system can be viewed as a degradable system with zero degradation.

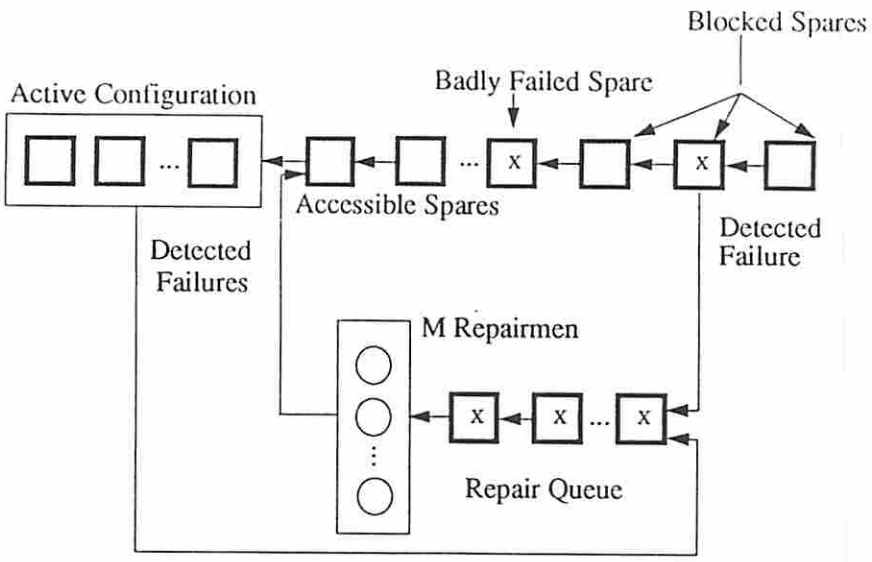


Figure 3.1: The ARIES Model for Repairable Systems

maximum number of permitted degradations for a degradable system; $D = 0$ for a non-degradable system. Clearly, $\underline{Y}[0] > \underline{Y}[1] >, \cdots, > \underline{Y}[D]$.

Whenever any of the modules in the active configuration fails, and the failure is detected by the fault-detection mechanism built into the system, the system reconfigures itself by switching in a spare from the head of the queue of spares; the failed active module is placed at the end of the repair queue to successfully complete the reconfiguration operation. If an active module fails but is not detected, or, reconfiguration is unsuccessful, the entire system fails. A module repaired by the repair facility (which has M servers) is placed at the head of the queue of spares. Spares in the *spares-queue* may also fail and their failure may or may not be detected. If detected, the spare is removed from the queue of spares and placed at the end of the repair queue. If not, it remains in the spare queue and is called a

badly failed spare. If this badly failed spare reaches the head of the queue (it may not, if there is another badly failed spare ahead of it), it is switched into the active configuration during reconfiguration and the ARIES model conservatively assumes that it causes the entire system to fail. The first badly failed spare in the spares-queue is said to *block* all spares behind it (See Figure 3.1). All spares in front of the first badly failed spare are said to be accessible. If D is the maximum number of permitted degradations, a degradable system is assumed to be *working* as long as at least $\underline{Y}[D])$ modules in the active configuration are working. Note that a system can only degrade if there are no badly failed spares. This is because the system assumes that the badly failed spare can replace a failed active module and keep the system working with N modules in the active configuration. As explained above, the badly failed spare actually causes the entire system to fail. If the system reconfigures successfully, then, if K is the number of degradations that the system has undergone so far, there are $\underline{Y}[K]$ modules in the active configuration.

ARIES models the dynamics of repairable fault-tolerant systems described above by a finite homogeneous continuous time Markov Chain. The state of the Markov Chain is defined by the number of accessible spares (P), the number of modules in the repair queue (Q), the number of badly failed spares (R) and the number of degradations the system has undergone (K) [Ng76]. The fifth element (B), introduced here merely for clarity, is a dependent variable and represents the number of blocked spares. The initial state is $(P, Q, R, K, B) = (S, 0, 0, 0, 0)$. The system parameters λ, μ, ψ, C_a and C_d are defined as follows [4]:

[4] In some versions of ARIES, C_s is used instead of C_d.

λ = Failure rate of one active module

μ = Failure rate of one spare

ψ = Repair rate of one module

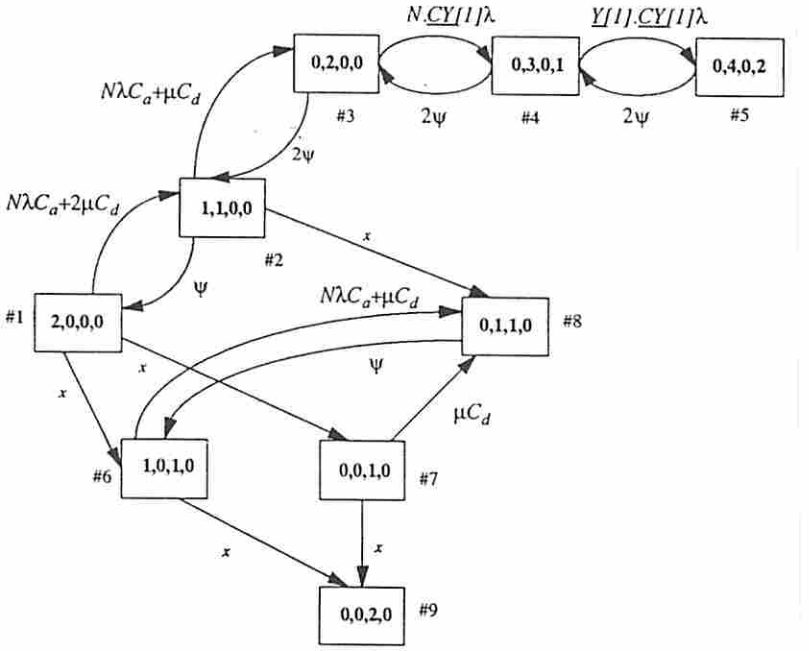
C_a = Coverage factor for recovery from active-module failures (Probability of successful recovery after failure)

C_d = Coverage factor for recovery from spare-module failures

$\underline{Y}$ = An integer vector defining the number of modules in successive active configurations after successfully recovering from failure

$\underline{CY}$ = Coverage Vector for non-degraded and degraded configurations

$\underline{CY}[0] = C_a$. If the coverage factor depends on the state of degradation of the system, $\underline{CY}[i], i = 1, \cdots, D$ can be made different for successive degradations. The transition rates between the states of the Markov Chain can easily be computed from the above information. The number of states in the Markov Chain of the ARIES repairable system model is a function of the number of spares S, and the number of degradations D. The Markov Chain structure for $S = 2$, $D = 2$, $M = 2$ taken from [Ng76] is shown in Figure 3.2. The number of modules in the active configuration (N) and the number of servers at the repair facility (M) only affect the transition rates; they have no effect the number of states in the Markov Chain. Transitions to the failed state F are not shown. All states are numbered according to the ARIES numbering rule.



x = $(1 - C_d)\mu$
state = (P, Q, R, K)
P = number of accessible spares
Q = number of modules in the repair queue
R = number of badly failed spares
K = number of degradations undergone

Figure 3.2: Structure of the Markov Chain for S=2, D=2.

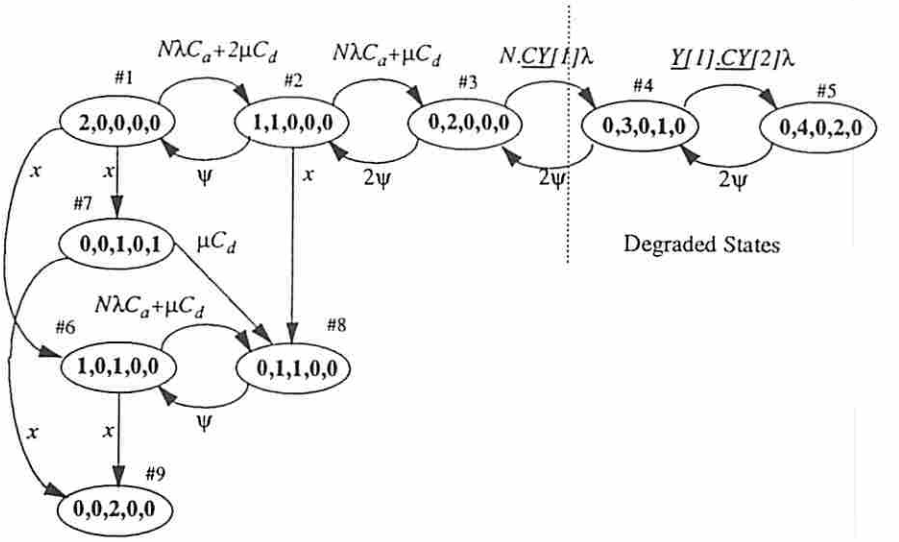
3.3 The Structure of the Markov Chain

As seen in Figure 3.2 taken from [Ng76], the structure of the Markov Chain appears to be quite complicated. We show that by re-arranging the positions of the states relative to each other, the Markov Chain actually exhibits a very regular structure. This structure is presented in Figure 3.3. We extend this structure to $S = 3,\ D = 0$ in Figure 3.4. One immediate advantage of the regularity in structure is that Markov Chains of relatively large sizes can be recursively constructed with a smaller chance of error. In Section 3.3.1 which follows, we describe a method by which the Markov Chain can be constructed systematically. The rules for the generation of states are exactly those detailed in [Ng76]. However, because of the regularity of the Markov Chain structure we can completely characterize the out-going transition rates for every state when given the state (P, Q, R, K) and the system parameters (see also [MAG82, page 23]). Also, the properties of the STRM become apparent when the states are laid out as suggested in the following section.

3.3.1 Construction and Numbering of the Markov Chain

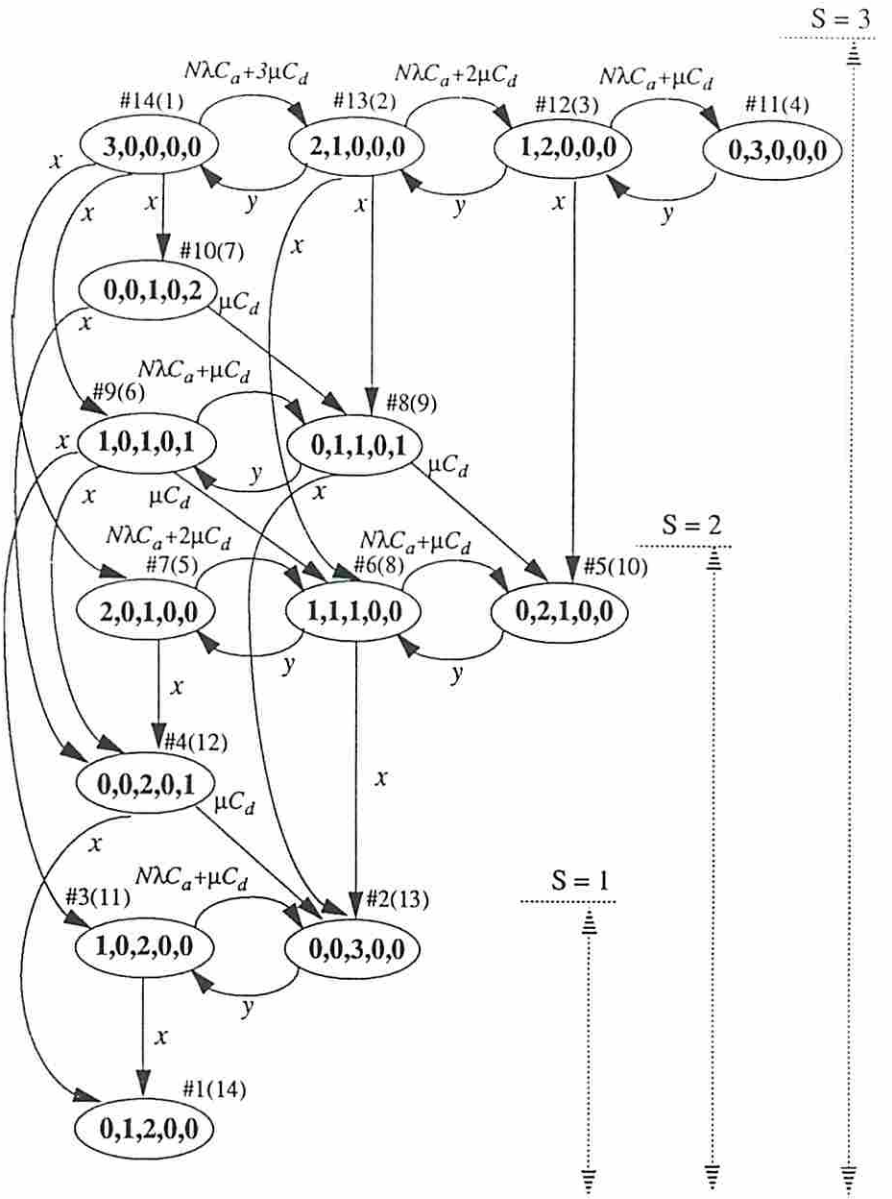
The initial state is $(P, Q, R, K, B) = (S, 0, 0, 0, 0)$. For any state (P, Q, R, K, B) of the Markov Chain, the out-going transitions are specified by the following rules:

- Any failed module (active or spare), if detected takes the system to state $(P - 1, Q + 1, R, K, B)$ at rate $\underline{Y}[K].\lambda C_a + P\mu C_d$ for $P \geq 1$. For $P = 0,\ K < D$, the system goes to $(0, Q+1, R, K+1, B)$. Otherwise, there is no



$x = (1 - C_d)\mu$
state $= (P, Q, R, K, B)$
B = number of blocked spares

Figure 3.3: Rearranged States for S=2, D=2, M = 2.



$$
\begin{aligned}
\text{state} &= (P, Q, R, K, B) \\
x &= (1 - C_d)\mu \\
y &= min(Q\psi,\ M)
\end{aligned}
$$

Figure 3.4: Recursive Markov Chain Structure.

transition. In Figure 3.3 and Figure 3.4, all such transitions are seen along the positive X-axis.

- An undetected failure of an active module causes the system to go to the failed (absorbing) state. Thus, from every state (P, Q, R, K, B) the system goes to the failed state F at rate $\underline{Y}[K].\lambda(1 - C_a)$. These transitions are not shown in figures.

- For $P > 0$, a spare which fails without being detected causes a transition to state $(P', Q, R + 1, B')$ where $P' + B' = P - 1, P' = 0, 1, \cdots, P - 1$. Thus, there are P distinct transitions to P distinct states each at rate $(1 - C_d)\mu$; the value of P' depends on the position of the failed spare relative to the head of the spares-queue. In Figure 3.3 and Figure 3.4, each transition generates a state in a separate *level*. Such transitions are vertical in the figure. The vertical co-ordinate of the state is determined by Q and the horizontal co-ordinate by R and B'. There are no transitions of this type for $P = 0$.

- States for which $B > 0$ have two additional transitions, $(P, Q, R, K, B) \rightarrow (P, Q+1, R, K, B-1)$ at rate $BC_d\mu$ (diagonal transitions) and $(P, Q, R, K, B) \rightarrow (P, Q, R + 1, K, B - 1)$ at rate $(1 - C_d)B\mu$ (vertical transitions).

- For $Q > 0$, a module module repaired by the repair facility takes the system to state $(P+1, Q-1, R, K, B)$ at rate $min(Q\psi, M)$. These transitions appear along the negative X-axis in the figures. There arcs are responsible for the cycles in the Markov Chain.

The Markov Chain in Figure 3.4 can be regarded (from the point of view of the structure) as a collection of birth-death processes with transitions occurring between the processes. These birth-death processes can be ordered so that all inter-process transitions occur strictly in one direction. Since each birth-death process can be represented by a tri-diagonal matrix, and since all inter-process transitions occur strictly in one direction, it is possible to find a numbering rule such that the STRM of the Markov Chain is in Hessenberg form [5]. Transitions to the failed state are ignored since they do not affect the solution. The numbering rule that results in a lower Hessenberg form of the STRM is given by Equation 3.1.

$$I = \begin{cases} N - [Q + K + 1] & \text{if R= 0} \\ N - \left[\sum_{j=0}^{R-1} K_j + \sum_{j=0}^{P+Q-1} (j+1) + (Q+1)\right], & 1 \leq \text{R} \leq \text{D} \end{cases} \tag{3.1}$$

where,

$$K_j = \begin{cases} S + D + 1 & \text{if j=0} \\ \frac{1}{2}(S + 2 - j)(S + 1 - j), & 1 \leq \text{j} \leq \text{R} \end{cases}$$

As seen in Figure 3.4, this formula numbers the states in increasing order from right to left and bottom to top. The numbers within the brackets are as per the

[5] A matrix H is in lower Hessenberg form if for $j > (i+1)$, $h_{ij} = 0, \forall i$.

ARIES numbering system. The following two properties of our numbering rule cause the STRM to be in lower Hessenberg form [6].

1. With the exception of the *repair* transitions (marked by ψ in Figure 3.4), all transitions are from bigger (with respect to numbering) states to smaller states.

2. All *repair* transitions are from smaller to bigger states such that the difference is exactly one in each case.

Table 3.1 shows the lower Hessenberg form of the STRM corresponding to the Markov Chain in Figure 3.4. Notice the recursive structure of the Markov Chain for $D = 0$; in Table 3.1, the STRM for $S = 3,\ D = 0$ completely contains the STRM for $(S = 2,\ D = 0)$ and $(S = 1,\ D = 0)$.

3.4 The Eigenvalues of the STRM

It is clear from the previous section that with a suitable rule for numbering the states of the Markov Chain, it is always possible to obtain an STRM for ARIES repairable systems which is in the Hessenberg form. We now make a more careful study of the structure of the STRM and note that according to our numbering rule, there are never any transitions from the rightmost state of any birth-death process to a state whose number is exactly one less. This property of the rightmost state in every birth-death process causes the STRM to have blocks along the diagonal.

[6] If states are indexed according to the terms within the square brackets in Equation 3.1, the STRM will be in upper Hessenberg form.

S = 1			S = 2										S = 3
-3	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-103	100	0	0	0	0	0	0	0	0	0	0	0
0.1	3.6	-4	0	0	0	0	0	0	0	0	0	0	0
0.1	0.9	0	-4	0	0	0	0	0	0	0	0	0	0
0	0	0	0	-103	100	0	0	0	0	0	0	0	0
0	0.1	0	0	3.6	-104	100	0	0	0	0	0	0	0
0	0	0.1	0.1	0	4.5	-5	0	0	0	0	0	0	0
0	0.1	0	0	0.9	0	0	-104	100	0	0	0	0	0
0	0	0.1	0.1	0	0.9	0	3.6	-5	0	0	0	0	0
0	0	0	0.2	0	0	0	1.8	0	-5	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-103	100	0	0
0	0	0	0	0.1	0	0	0	0	0	3.6	-104	100	0
0	0	0	0	0	0.1	0	0.1	0	0	0	4.5	-105	100
0	0	0	0	0	0	0.1	0	0.1	0.1	0	0	5.4	-6

$N = 3,\ \lambda = \mu = 1,\ C_a = C_d = .9,\ M = 1.$

Table 3.1: Recursive STRM for D = 0 and S = 3, S = 2, S = 1.

Thus, in addition to being in Hessenberg form, the STRM is lower quasi-triangular [7]. Also, every diagonal block is a tri-diagonal matrix. It follows from standard matrix theory that the eigenvalues of a quasi-triangular matrix are exactly the eigenvalues of the individual blocks along the diagonal [8]. Since each block on the diagonal of the STRM is a tri-diagonal matrix, the eigenvalues of the STRM are simply a collection of the eigenvalues of the individual tri-diagonal matrices, a very desirable situation with respect to eigenvalue computation. For a system with S spares and a maximum degradation of D, the STRM contains $(S-i+1)$ diagonal blocks of size $i, (1 \leq i \leq S)$ and one block of size $(S+D+1)$. In the following section we analyze the properties of the eigenvalues of the STRM; many of these properties are inherited from the properties of the tri-diagonal blocks. In Section 3.4.2 we use these properties to develop an efficient algorithm for computing the eigenvalues of the STRM for ARIES repairable fault-tolerant systems.

3.4.1 Properties of the Eigenvalues of the STRM

Property 4.1.1 *Every block on the diagonal of the STRM is similar to a real symmetric tri-diagonal matrix.*

An inspection of the STRM shows that each block on the diagonal is a *quasi-symmetric tri-diagonal* [9] matrix [Wil65]. From matrix theory [Wil65, page 336], there exists a diagonal similarity transformation which transforms the block into a

[7] A lower quasi-triangular matrix has blocks along the diagonal and zeros above the diagonal blocks.

[8] In the special case, when all diagonal blocks of the quasi-triangular matrix are of dimension one, the matrix is triangular and the eigenvalues appear along the diagonal.

[9] For a definition, see Appendix A.

real symmetric tri-diagonal matrix. This transformation is reproduced in Appendix A for easy reference.

Property 4.1.2 *The eigenvalues of each tri-diagonal block of the STRM are real and distinct.*

By the previous property, each tri-diagonal block is similar to a real symmetric tri-diagonal matrix. From matrix theory, (see for example, [Wil65, page 25]) the eigenvalues of a symmetric matrix are all real. Also, the symmetric tri-diagonal matrix obtained by the diagonal similarity transformation satisfies the Sturm Sequence Property [10]; the eigenvalues of each tri-diagonal block are therefore distinct. This proves that the eigenvalues of each tri-diagonal block of the STRM are real and distinct.

Property 4.1.3 *The eigenvalues of the STRM are strictly negative.*

By Property 4.1.2 we know that the eigenvalues of the STRM are all real. We observe that the sum of the absolute values of the non-diagonal elements in each row of the STRM are strictly less than the (absolute value of) the diagonal element, i.e., the STRM is strictly diagonally dominant. By Gerschgorin's circle theorem [Wil65] it follows that the eigenvalues of the STRM are strictly negative [11].

Property 4.1.4 *Any two blocks $\mathcal{B}_{ii}$ and $\mathcal{B}_{jj}$ along diagonal of the STRM denoted by $\mathcal{B}$ have a disjoint set of eigenvalues if and only if the matrix equation $\mathcal{B}_{ii}\mathcal{X} -$*

[10] See Appendix B
[11] See Appendix C.

$\mathcal{X}\mathcal{B}_{jj} = \mathcal{B}_{ij}, i \neq j$ *has a unique solution.* $\mathcal{B}_{ij}$ *denotes the block in the* i^{th} *row and the* j^{th} *column of* $\mathcal{B}$.

This follows from a standard result for the solution of the matrix equation $\mathcal{A}\mathcal{X} - \mathcal{X}\mathcal{B} = \mathcal{C}$. See, for example, [GV84, page 242] [Gan77a, page 225].

Property 4.1.5 *The eigenvalues of any two blocks of a given size differ by an integer multiple of* μ*, the failure rate of a spare.*

As shown in Property 4.1.1, every diagonal block of the STRM is similar to a real symmetric matrix. We observe that any two *transformed* blocks of a given size differ only in the diagonal elements. In addition, each of these diagonal elements differ by exactly the same constant which is an integer multiple of the failure rate μ of a single spare. Thus, if $\mathcal{W}_{ii}$ and $\mathcal{W}_{jj}$ are two *transformed* blocks of the same size in the STRM, then, $\mathcal{W}_{ii} = \mathcal{W}_{jj} + k\mu\mathcal{I}$ for some integer k. From matrix theory for perturbations in symmetric matrices, we have the following result: *If* $\mathcal{C} = \mathcal{A} + \mathcal{B}$ *where,* $\mathcal{A}$ *and* $\mathcal{B}$ *are symmetric matrices, then, when* $\mathcal{B}$ *is added to* $\mathcal{A}$*, its eigenvalues are changed by an amount which lies between the smallest and the largest eigenvalues of* $\mathcal{B}$ [Wil65, page 102].

Property 4.1.6 *For* $D = 0$*, the set of eigenvalues of the STRM for a system of size* $(S-1)$ *are completely contained within the set of eigenvalues for size* S*. For* $D \neq 0$*, except for the eigenvalues of the largest tri-diagonal block matrix, all others are contained in the set of eigenvalues of the STRM for size* S.

For $D = 0$ this property follows from the recursive structure of the Markov Chain and the fact that the eigenvalues of the STRM are a collection of the eigenvalues of the individual diagonal blocks. Since the Markov Chain of a degradable (repairable) system differs from a non-degradable one only in the birth-death process corresponding to $R = 0$ (Figure 3.3), from the point of view of the eigenvalues, the STRMs of the two systems differ only in their largest tri-diagonal blocks. The largest blocks are not of equal size.

Property 4.1.2 and Property 4.1.3 corroborate with our expectation of the behavior of the state probability vector, i.e., with time $t \rightarrow \infty$ the system is in the $up - state$ with probability zero. The other properties allow us to use highly efficient numerical methods to solve the eigenvalue problem. This is the focus of the next section.

3.4.2 The Complexity of Eigenvalue Computation

We begin by characterizing the size of the STRM of a repairable fault-tolerant system with S spares and a maximum permissible degradation of D modules. $D = 0$ corresponds to a non-degradable system. For a system with S spares and a maximum permissible degradation of D, the STRM contains $(S - i + 1)$ diagonal blocks of size $i, (1 \leq i \leq S)$ and one diagonal block of size $(S+1+D)$. As discussed in Section 3.2, the only effect of D on the number of states is to increase the size of the largest diagonal block from $(S + 1)$ to $(S + 1 + D)$; the tri-diagonal form is preserved. The size n of the STRM may be calculated by summing over the sizes of the diagonal blocks as in the following expression:

For $k = 1, 2, \cdots$

Let $\mathcal{A}_{k-1} = \mathcal{U}_k \mathcal{R}_k$ be the Q-R factorization of $\mathcal{A}_{k-1}$.

Set $\mathcal{A}_k = \mathcal{R}_k \mathcal{U}_k$.

Note: $\lim_{k \to \infty} \mathcal{A}_k = (\mathcal{U}_0 \mathcal{U}_1 \cdots \mathcal{U}_k)^H \mathcal{A}_0 (\mathcal{U}_0 \mathcal{U}_1 \cdots \mathcal{U}_k) \rightarrow$ Schur form of $\mathcal{A}_0$

Figure 3.5: The Basic QR Iteration

$$n(S,\ D) = (S+1+D) + \sum_{i=1}^{S} i(S-i+1) = \frac{1}{6}(S^3+3S^2+8S+6) + D, \quad D < N \quad (3.2)$$

Note that the number of states in the Markov Chain is independent of the number of modules, N, in the active configuration. The only dependence is through D which must be less than N.

ARIES uses the iterative QR algorithm to compute the eigenvalues of the STRM. Figure 3.5 shows the basic form of the QR algorithm [GV84]. Any matrix, say, $\mathcal{A}_0$ can be expressed as a product of a unitary matrix, $\mathcal{U}_0$ and an upper triangular matrix $\mathcal{R}_0$ by using the QR decomposition technique ($\mathcal{A}_0 = \mathcal{U}_0 \mathcal{R}_0$). If $\mathcal{A}_1 = \mathcal{R}_0 \mathcal{U}_0$, then, clearly $\mathcal{A}_0$ is unitarily similar to $\mathcal{A}_1$. The above two operations could be applied with $\mathcal{A}_0$ replaced by $\mathcal{A}_1$. Thus, the QR algorithm in Figure 3.5 generates a (theoretically infinite) sequence of unitarily similar matrices. It has been proved in [GV84, page 209] that this sequence actually converges to the Schur decomposition of $\mathcal{A}_0$. The Schur form of $\mathcal{A}_0$ is a quasi-triangular matrix.

In its basic form, the QR algorithm is very expensive. For a square matrix of size n, the computational complexity per iteration is $O(n^3)$ and convergence is linear. Fortunately, this algorithm lends itself to numerous modifications, causing it to

For $k = 1, 2, \cdots$

$$\mathcal{W}_{k-1} - \mu\mathcal{I} = \mathcal{Q}_k\mathcal{R}_k \qquad \text{(Q-R factorization)}$$
$$\mathcal{W}_k = \mathcal{R}_k\mathcal{Q}_k + \mu\mathcal{I}$$

μ is the Wilkinson Shift

Figure 3.6: The QR Algorithm for Symmetric Tri-Diagonal Matrices

be one of the most efficient algorithms for solving the eigenvalue problem. Here, we discuss some of these modifications and the resulting impact on computational complexity and/or convergence.

For a matrix in Hessenberg form, the complexity per iteration of the QR algorithm is only $O(n^2)$. Thus, it is always desirable to apply the QR iterations to a matrix in Hessenberg form. By a standard result from matrix theory [Wil65], it is always possible to reduce an arbitrary real square matrix to Hessenberg form by a unitary similarity transformation with $O(n^3)$ multiplications [12]. ARIES reduces the $n \times n$ STRM to Hessenberg form in the pre-processing step.

The convergence of the QR algorithm can be accelerated by *shifting* the matrix $\mathcal{A}_k$ before the QR-factorization step. The shift operation causes the convergence of the algorithm to be essentially quadratic. For details of the convergence properties, we refer the reader to [GV84, page 228]. Currently, ARIES uses the *Shifted*-QR algorithm for computing the eigenvalues of the STRM of size n after first reducing the STRM to its Hessenberg form. From Equation 3.2, if D is a linear function of S, then, $n = O(S^3)$. Measured in terms of the number of spares (which is a measure

[12]The product of $(n-2)$ Householder transformations results in a unitary matrix. This matrix can be used as a similarity transformation matrix.

of the degree of fault-tolerance), the complexity of the QR algorithm is $O(S^6)$ per iteration; the algorithm to reduce the STRM to Hessenberg form is of complexity $O(S^9)$. In the remainder of this section, we show, that by exploiting the properties of the STRM, the complexity of the QR algorithm can be reduced to $O(S^2)$ with cubic convergence while the reduction to Hessenberg form can be eliminated. This reduction is, however, replaced by a pre-processing step of complexity $O(S^2)$.

We have already observed in Section 3.4 that the eigenvalues of the STRM are exactly those of the individual tri-diagonal block matrices on the diagonal of the STRM. Thus, the problem of finding the eigenvalues of an $n \times n$ reduces to finding the eigenvalues of $(S - i + 1)$ tri-diagonal matrices of size $i, (1 \leq i \leq n)$ and one matrix of size $(S + 1 + D)$. Property 4.1.5 permits a further savings in computational effort. Since the eigenvalues of a block of a size i, $1 \leq i \leq S$ can be trivially derived from the eigenvalues of any other block of the same size, it is sufficient to compute the eigenvalues of exactly one block of a given size i. Since the tri-diagonal matrices are already in Hessenberg form, we proceed with the pre-conditioning step as follows. By Property 4.1.1, each diagonal block of size i can be transformed by a diagonal similarity transformation to a real symmetric tri-diagonal matrix with $O(i)$ multiplications. Thus, the S matrices whose eigenvalues are to be computed can be pre-conditioned with $O(S^2)$ multiplications. We now show that with this pre-conditioning step, the most efficient version of the QR algorithm, applicable only to real symmetric tri-diagonal matrices can be used to compute the eigenvalues of the STRM.

For a real symmetric matrix, $\mathcal{W}_0$, the QR algorithm can be refined as follows (Figure 3.6). Since $\mathcal{W}_0$ is real, all unitary matrices in Figure 3.5 are real and so are orthogonal matrices (denoted by $\mathcal{Q}_k$). Since $\mathcal{W}_0$ is symmetric, all its eigenvalues are real so that its Schur form is a triangular matrix (a special case of the quasi-triangular matrix). Further, since $\mathcal{W}_0$ is a tri-diagonal matrix of size i, the complexity of the QR algorithm is $O(i)$ per iteration. For a fault-tolerant system with S spares and a maximum number of D degradations, the matrix sizes are $1, 2, \cdots, S$ and $(S + D + 1)$. This translates to an overall complexity of $O(S^2)$ per iteration for the eigenvalue problem. The QR algorithm is particularly suited to real symmetric tridiagonal matrices; in addition to a reduction in computational complexity, it offers cubic convergence for these matrices [GV84, page 278]. The $Wilkinson - Shift$ [GV84, page 279] is recommended for accelerating the convergence of the the algorithm. Experimental results show [DBA74, page 216] that with these modifications, the number of QR iterations per eigenvalue is less than two. In effect, the iterative QR algorithm is a non-iterative algorithm of complexity $O(S^3)$.

It has been shown in [Osb60] (see also [PR69]) that algorithms which find the eigenvalues or eigenvectors of a matrix, say, $\mathcal{A}$, do so with an error at least of order $\epsilon\|\mathcal{A}\|_E$, where ϵ is the machine precision and $\|\mathcal{A}\|_E$ the Euclidean norm of $\mathcal{A}$. It is therefore recommended that the matrix be pre-conditioned by means of a sequence of (theoretically infinite) diagonal similarity transformations so that $\mathcal{B}$ is diagonally similar to $\mathcal{A}$ and $\|b^i\|^p = \|b_i\|^p$ (i.e, the p-norm of the i^{th} row equals the p-norm of the i^{th} column of matrix $\mathcal{B}$). Matrix $\mathcal{B}$ is then said to be *balanced* and

the norm of $\mathcal{B}$ is less than the norm of $\mathcal{A}$. Clearly, the similarity transformation which makes the quasi-symmetric tri-diagonal blocks, symmetric, also balances the matrix for the QR iterations.

We conclude our analysis of the eigenvalues of the STRM by discussing the advantage offered by Property 4.1.6. Oftentimes, in designing fault-tolerant systems it is useful to evaluate the effect of adding a spare to a system whose reliability has already been computed. The design process can be considerably speeded up if the effect of the additional spare on the reliability of the system can be computed incrementally. In the next paragraph we show that Property 4.1.6 offers such a facility for computing the eigenvalues of the larger system incrementally.

The effect of adding a spare in a non-degradable system with $(S-1)$ spares, is to introduce a set of tri-diagonal blocks of sizes $1, 2, \cdots, (S-1)$ and $(S+1)$ in the STRM (See Figure 7). Since the eigenvalues of a system with $(S-1)$ spares are entirely contained in the set of eigenvalues of a system with S spares, these eigenvalues need not be re-computed for the bigger system. Also, by Property 4.1.5 it is possible to derive the eigenvalues of all newly introduced blocks of size i, $1 \leq i \leq (S-1)$ from the eigenvalues of the smaller system (the information about the block size associated with an eigenvalue is assumed to be available). Thus, for non-degradable systems it is sufficient to compute the eigenvalues of a single tri-diagonal matrix of size $(S+1)$ when the number of spares is increased from $(S-1)$ to S.

A degrabable system with $(S-1)$ spares contains blocks ranging in size from 1 to $(S-1)$ and one block of size $(S+D)$ (for a non-degradable system, the size of

the corresponding largest block is S because $D = 0$). By adding a spare to this degradable system, as in the non-degradable case, blocks of sizes i, $1 \leq i \leq (S-1)$ are introduced. As before, the eigenvalues of these blocks can be derived from the eigenvalues of the system with $(S-1)$ spares. In addition, a block of size $(S+D+1)$ is introduced for degradable systems; the corresponding block the non-degradable system is of size $(S+1)$. The eigenvalues of this block must be computed. Further (this does not happen for degradable systems), the block of size $(S+D)$ of the system with $(S-1)$ spares is reduced to a block of size S; the eigenvalues of this block must, therefore, be re-computed.

3.5 The Eigenvectors of the STRM

Let $\mathcal{B}$ denote the STRM for ARIES repairable systems. In the previous section, we performed the following two steps on each individual tri-diagonal block $\mathcal{B}_{ii}$ in order to compute the eigenvalues of the STRM. In the first step we applied a diagonal similarity transformation to convert the block into a real symmetric tri-diagonal matrix ($\mathcal{W}_{ii} = \mathcal{D}_i^{-1}\mathcal{B}_{ii}\mathcal{D}_i$). We then applied the QR iterations to the symmetric matrix so that the eigenvalues of the block appeared on the diagonal of the matrix $\lim_{k\to\infty} \mathcal{W}_{ii}^k$. As has been proved in [GV84], when the QR iterations are applied to a matrix, the matrix converges to its Schur decomposition. From matrix theory, the Schur decomposition of a symmetric matrix is a diagonal matrix of its eigenvalues [13]. This implies that the similarity transformation matrix (a

[13] A symmetric matrix is *normal*; the Schur decomposition of a normal matrix is diagonal.

theoretically infinite product of orthogonal matrices) which reduces $\mathcal{W}_{ii}$ to its approximate Schur decomposition is infact the (approximate) matrix of eigenvectors of $\mathcal{W}_{ii}$. If we were to accumulate the orthogonal matrices in successive iterations of the QR algorithm, we could compute the eigenvectors of each $\mathcal{W}_{ii}$ along with its eigenvalues. We show later in this section that the eigenvectors of the STRM can be obtained by combining the eigenvectors of the individual symmetric blocks by means of a transformation. The accumulation of the orthogonal matrices requires additional computational effort and in effect increases the complexity of eigenvalue computation. In the next paragraph we estimate the increase in this complexity.

If we were to compute only the eigenvalues using the QR algorithm, it would be un-necessary to accumulate the orthogonal matrices; the computational complexity per iteration for a tri-diagonal matrix of size i is then $O(i)$. With the accumulation of the orthogonal matrices, however, the complexity per iteration is $O(i^2)$ [GV84]. Also, by virtue of Property 4.1.5 it was sufficient to calculate the eigenvalues of exactly one tri-diagonal matrix of a given size i. Now, since we require the orthogonal matrices of each tri-diagonal matrix, one (obvious) solution is to apply the QR algorithm to every block of the STRM. The complexity of the combined computation of the eigenvalues and the eigenvectors of the tri-diagonal matrices is therefore $O(S^4)$ per iteration; convergence remains cubic. The complexity of the pre-conditioning step increases to $O(S^3)$.

Assume, then, that the matrix of eigenvectors of individual symmetric matrices have been computed with the above complexity. From standard matrix theory [Wil65, page 299], if $\mathcal{Q}_{ii}$ is the matrix of eigenvectors of the symmetric matrix

$\mathcal{W}_{ii}$, then, the matrix of eigenvectors of $diag(\mathcal{W}_{11}, \cdots, \mathcal{W}_{jj}, \cdots, \mathcal{W}_{qq})$ is the matrix $diag(\mathcal{Q}_{11}, \cdots, \mathcal{Q}_{jj}, \cdots, \mathcal{Q}_{qq})$. Thus, if $\mathcal{X}$ is a similarity transform matrix which transforms the STRM $\mathcal{B}$ to $diag(\mathcal{W}_{ii})$, then $\mathcal{R}$, the matrix of eigenvectors of $\mathcal{B}$ is equal to $\mathcal{X}diag(\mathcal{Q}_{ii})$. Section 3.5.1 is entirely devoted to constructing the similarity transform matrix $\mathcal{X}$ and computing it by methods that are numerically stable. However, before proceeding in this direction, we discuss the computational complexity associated with the ARIES solution technique.

ARIES does not compute the eigenvectors of the STRM; it computes the state probability vector of the Markov Chain by direct computation of the matrix function $e^{\mathcal{B}t}$. Then, $P(t) = e^{\mathcal{B}t}P(0)$, where $\mathcal{B}$ is the STRM. By the Lagrange-Sylvester interpolation formula, $e^{\mathcal{B}t} = \sum_{j=0}^{n-1} e^{\lambda_j t}\left[\Pi_{k=0,k\neq j}^{n-1}(\mathcal{B} - \lambda_k\mathcal{I})/(\lambda_j - \lambda_k)\right]$, where λ_i are the eigenvalues of $\mathcal{B}$ assuming that for practical systems, these are always distinct. The complexity for computing $e^{\mathcal{B}t}$ is $O(n^4)$ [ML78] which translates to $O(S^{12})$ in terms of the number of spares, S; for the ARIES implementation, this complexity is $O(S^{15})$. With $P(0) = (1, \cdots, 0, 0)$, the first column of the matrix $e^{\mathcal{B}t}$ is equal to $P(t)$. By our method so far, we have computed the eigenvalues and the eigenvectors of the individual tri-diagonal blocks with a computational complexity of $O(S^4)$ per iteration and cubic convergence. In effect, we have reduced the complexity of the eigenvalue computation by a factor of S^2 per iteration and improved upon the convergence properties such that the QR iterative algorithm is in effect a non-iterative algorithm of complexity $O(S^7)$. We show that infact the state probability vector can be computed with a complexity of $O(S^7)$.

3.5.1 Constructing the Similarity Transform

Our objective in this section is to construct a transformation which will combine the matrix of eigenvectors of the symmetric blocks computed by the QR algorithm to give the eigenvectors of the STRM. As before, let $\mathcal{B}$ be denote the STRM of the Markov Chain for ARIES repairable systems and let us *assume that the eigenvalues of* $\mathcal{B}$ *are distinct.* Also, let us denote the symmetric tri-diagonal blocks by $\mathcal{W}_{ii}, 1 \leq i \leq q$ and the corresponding matrix of eigenvectors by $\mathcal{Q}_{ii}$. Recall that before applying the QR iterations, each quasi-symmetric tri-diagonal block of the STRM was transformed to a symmetric tri-diagonal matrix by means of a diagonal similarity transformation $\mathcal{D}_i$ (See Appendix A). We use the following two operations to construct a similarity transformation between $\mathcal{B}$ and $diag(\mathcal{W}_{ii})$. In the first step we apply the diagonal similarity transformation $\mathcal{D} = diag(\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_q)$ to $\mathcal{B}$; i.e., $\mathcal{W} = \mathcal{D}^{-1}\mathcal{B}\mathcal{D}$. In the next step, assuming that such a transform exists, we *block diagonalize* $\mathcal{W}$ by a similarity transformation to get a quasi-diagonal matrix $diag(\mathcal{W}_{ii})$, i.e., $\mathcal{W} = diag(\mathcal{W}_{ii}) + \mathcal{L}$ is transformed to $diag(\mathcal{W}_{ii})$. Here, $\mathcal{L}$ denotes all elements of $\mathcal{W}$ other than the diagonal blocks. If $\mathcal{Y}$ denotes the block-diagonalizing transformation, then $(\mathcal{D}\mathcal{Y})^{-1}\mathcal{B}(\mathcal{D}\mathcal{Y}) = diag(\mathcal{W}_{ii})$ so that $\mathcal{B}$ is similar to $diag(\mathcal{W}_{ii})$. $\mathcal{Y}$ is a lower triangular matrix and is of the form shown in Equation 3.3.

$$\begin{pmatrix} \mathcal{I} & 0 & 0 & . & 0 \\ \mathcal{Y}_{21} & \mathcal{I} & 0 & . & 0 \\ \mathcal{Y}_{31} & \mathcal{Y}_{32} & \mathcal{I} & . & 0 \\ . & . & . & . & . \\ \mathcal{Y}_{q1} & \mathcal{Y}_{q2} & \mathcal{Y}_{q3} & . & \mathcal{I} \end{pmatrix}^{-1} \begin{pmatrix} \mathcal{W}_{11} & 0 & 0 & . & 0 \\ \mathcal{W}_{21} & \mathcal{W}_{22} & 0 & . & 0 \\ \mathcal{W}_{31} & \mathcal{W}_{32} & \mathcal{W}_{33} & . & 0 \\ . & . & . & . & . \\ \mathcal{W}_{q1} & \mathcal{W}_{q2} & \mathcal{W}_{q3} & . & \mathcal{W}_{qq} \end{pmatrix} \begin{pmatrix} \mathcal{I} & 0 & 0 & . & 0 \\ \mathcal{Y}_{21} & \mathcal{I} & 0 & . & 0 \\ \mathcal{Y}_{31} & \mathcal{Y}_{32} & \mathcal{I} & . & 0 \\ . & . & . & . & . \\ \mathcal{Y}_{q1} & \mathcal{Y}_{q2} & \mathcal{Y}_{q3} & . & \mathcal{I} \end{pmatrix}$$

$$= \begin{pmatrix} \mathcal{W}_{11} & 0 & 0 & . & 0 \\ 0 & \mathcal{W}_{22} & 0 & . & 0 \\ 0 & 0 & \mathcal{W}_{33} & . & 0 \\ . & . & . & . & . \\ 0 & 0 & 0 & . & \mathcal{W}_{qq} \end{pmatrix} \tag{3.3}$$

The matrix $\mathcal{Y}$ can be constructed as a product of a sequence of elementary similarity transformations of a single type referred to in [Wil65] as the $N-type$. Each elementary matrix is a unit lower triangular matrix (and therefore invertible) and performs a similarity transformation by operating on exactly one row and the corresponding column. Clearly, a sequence of similarity transformations of this type leaves the diagonal and all elements above the diagonal unaltered. We refer the reader to [Wil65] for further details on synthesizing the matrix $\mathcal{Y}$. We now analyze the effect of the similarity transformation $\mathcal{W} = \mathcal{D}^{-1}\mathcal{B}\mathcal{D}$ on the elements of $\mathcal{B}$.

Clearly, the diagonal blocks are transformed to symmetric matrices $\mathcal{W}_{ii}$. A non-diagonal block $\mathcal{B}_{k,j}, k > j$ is transformed to $\mathcal{W}_{kj} = \mathcal{D}_k^{-1}\mathcal{B}_{kj}\mathcal{D}_j$. Due to the structure of the Markov Chain for repairable systems, $\mathcal{B}_{kj}$ is either of the form

$[\alpha\mathcal{I}]$, $[\alpha\mathcal{I}, 0]$, or of the form $[0, \alpha\mathcal{I}]^T$ for some real value of α, or else, it is the zero matrix (See Figure 7). The matrix $\mathcal{D}_i$ by construction is such that the elements vary widely in their magnitude. The effect of the transformation on $\mathcal{B}_{kj}$, however, is to scale its non-zero elements by real scalars of similar orders of magnitude to give $\mathcal{W}_{kj}$. This does not imply that the elements of $\mathcal{W}_{kj}$ are of the same order across blocks. It does imply, however, that the elements within a given block are of the same order of magnitude.

Having computed the matrix $\mathcal{W}$, we compute the unknown variables $\mathcal{Y}_{kj}, k > j$ by equating $\mathcal{W}\mathcal{Y} = \mathcal{Y}diag(\mathcal{W}_{ii})$ on a block-by-block basis. This gives rise to the following set of matrix equations for $k > j$:

$$\begin{aligned} \mathcal{W}_{kk}\mathcal{Y}_{kj} - \mathcal{Y}_{kj}\mathcal{W}_{jj} &= -\mathcal{W}_{kj} - \sum_{l=j+1}^{k-1} \mathcal{W}_{kl}\mathcal{X}_{lj} \\ & 2 \leq k \leq (q = (S+1)S/2 + 1), 1 \leq j \leq (k-1) \end{aligned} \tag{3.4}$$

From Equation 3.4, it is clear that for a given column j, $\mathcal{Y}_{kj}$ can be computed provided that $\mathcal{Y}_{ij}$ is known for every row i,$(\ (j+1) \leq i \leq k-1), k > j+1)$. We are then faced with the task of solving $S(S+1)/2$ matrix equations of the type $\mathcal{A}\mathcal{X} + \mathcal{X}\mathcal{B} = \mathcal{C}$. This equation is referred to as the Lyapunov Equation in Control Theory and the Sylvester Equation in [Gan77a] and has been very well studied, both from the theoretical as well as the numerical aspects of the solution [Gan77a], [BS72], [HV79]. In the next section, we discuss issues related to the solution of $\mathcal{A}\mathcal{X} + \mathcal{X}\mathcal{B} = \mathcal{C}$.

3.5.2 The Equation $\mathcal{A}\mathcal{X} + \mathcal{X}\mathcal{B} = \mathcal{C}$

It is well-known that the equation $\mathcal{A}\mathcal{X} + \mathcal{X}\mathcal{B} = \mathcal{C}$ has a unique solution if and only if $\mathcal{A}$ and $-\mathcal{B}$ have no common eigenvalues [Gan77a],[Bar79]. In the case of common eigenvalues, it has infinitely many solutions if $c \in$ *column space* of $\mathcal{D} = \mathcal{A} \otimes \mathcal{I}_n + \mathcal{I}_m \otimes \mathcal{B}^T$ [14] [Bar79]. One of the best techniques for solving the equation numerically when $\mathcal{A}$ and $\mathcal{B}$ have a disjoint set of eigenvalues, is the Bartels-Stewart Algorithm [BS72]. This algorithm reduces both matrices $\mathcal{A}$ and $\mathcal{B}$ to their respective Schur form and solves the transformed equation. The Hessenberg-Schur Algorithm proposed by Golub, Nash and Van Loan [HV79] differs from the Bartels-Stewart Algorithm in that $\mathcal{A}$ is reduced to Hessenberg form instead of the Schur form. We briefly describe the Hessenberg-Schur Algorithm since it is more applicable to solving the system of equations described by Equation 3.4. Following is a description taken from [HV79].

1. Transform $\mathcal{A}$ to Hessenberg form and $\mathcal{B}$ to Schur form through similarity transformations. $\mathcal{A}_1 = \mathcal{U}^{-1}\mathcal{A}\mathcal{U}, \mathcal{B}_1 = \mathcal{V}^{-1}\mathcal{B}\mathcal{V}$.

2. Solve $\mathcal{U}\mathcal{F} = \mathcal{C}\mathcal{V}$ for $\mathcal{F}$.

3. Solve the transformed system $\mathcal{A}_1\mathcal{Y} + \mathcal{Y}\mathcal{B}_1 = \mathcal{F}$ for $\mathcal{Y}$.

4. Solve $\mathcal{X}\mathcal{V} = \mathcal{U}\mathcal{Y}$ for $\mathcal{X}$.

We now apply this algorithm to Equation 3.4. Since we have assumed that the STRM denoted by $\mathcal{B}$ has distinct eigenvalues, $\mathcal{Y}_{kj}$ is unique for every equation

[14] $\otimes$ denotes Kronecker multiplication. c is a column vector formed from rows of $\mathcal{C}$. $\mathcal{A}$ and $\mathcal{B}$ are assumed to be $m \times m$ and $n \times n$ respectively.

described by Equation 3.4. In Equation 3.4, $\mathcal{W}_{kk}$ is already in Hessenberg form, being tri-diagonal. The Schur's form of $\mathcal{W}_{jj}$ is a diagonal matrix of eigenvalues Λ_{jj}. The orthogonal matrix $\mathcal{Q}_{jj}$ obtained from the QR algorithm corresponds to $\mathcal{V}$ of the Hessenberg-Schur Algorithm. The Hessenberg-Schur Algorithm as applied to Equation 3.4 is as follows:

1. Compute $\mathcal{F}_{kj} = \left[-\mathcal{W}_{kj} - \sum_{l=j+1}^{k-1} \mathcal{W}_{kl}\mathcal{Y}_{lj}\right] \mathcal{Q}_{jj}$.

2. Solve $\mathcal{W}_{kk}\mathcal{Y}'_{kj} - \mathcal{Y}'_{kj}\Lambda_{jj} = \mathcal{F}_{kj}$ for $\mathcal{Y}'_{kj}$.

3. Compute $\mathcal{Y}_{kj} = \mathcal{Y}'_{kj}\mathcal{Q}^T_{jj}$.

We now estimate the computational complexity of each of the above three steps.

3.5.3 The Complexity of Eigenvector Computation

As discussed earlier, $\mathcal{W}_{kl}$ is obtained from $\mathcal{B}_{kl}$ by scaling the non-zero elements of $\mathcal{B}_{kl}$. Hence, the *form* of $\mathcal{W}_{kl}$ is exactly that of $\mathcal{B}_{kl}$. The sum-of-products term to compute $\mathcal{F}_{kj}$ in Step 1, therefore, reduces to a scale-and-add operation of matrices $\mathcal{Y}_{lj}, j+1 \leq l \leq k-1$. Since, for a system with S spares, l and j are each less than or equal to $(S+1)$, each product term $\mathcal{W}_{kl}T_{lj}$ can be computed with complexity $O(S^2)$. Further, in any row k, the number of non-zero matrices $\mathcal{W}_{kl}$ never exceeds the number of spares, S (Figure 7). We conclude that each $\mathcal{F}_{kj}$ can be computed with complexity $O(S^3)$.

To simplify notation, let $\mathcal{W}_{kk} = \mathcal{H}$, $\Lambda_{jj} = \mathcal{T}$, $\mathcal{Y}'_{kj} = \mathcal{Z}$ and $\mathcal{F}_{kj} = \mathcal{K}$. Then Step 2 can be rewritten as $\mathcal{HZ} - \mathcal{ZT} = \mathcal{K}$. As per the procedure in [HV79], the k^{th} column of $\mathcal{Z}$ denoted by z_k can be computed by solving the Hessenberg system

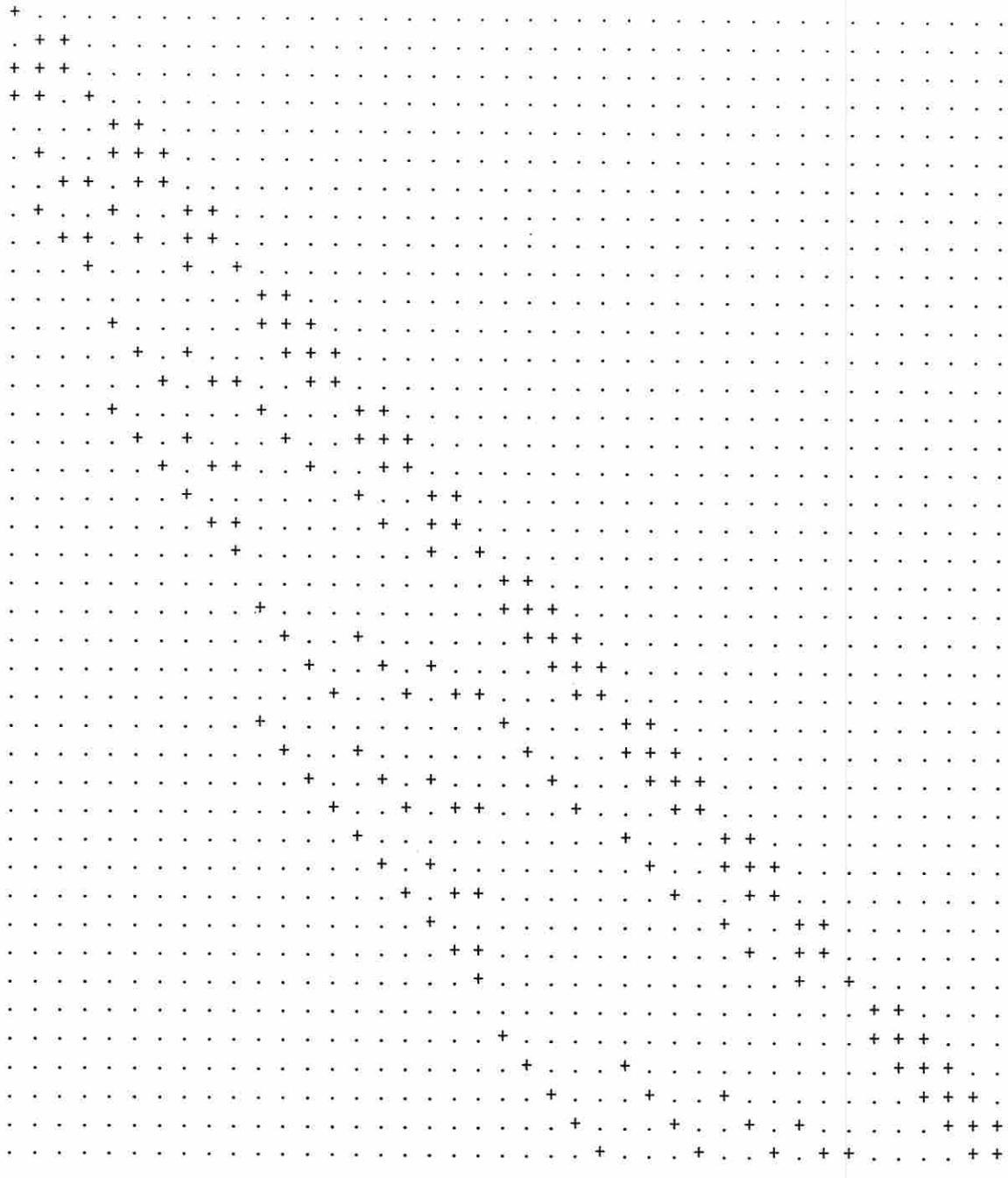
$$(\mathcal{H} + t_{kk}\mathcal{I})z_k = k_k \tag{3.5}$$

Thus, the complexity of computing each column of $\mathcal{Y}'_{kj}$ is $O(S^2)$ and hence, Step 2 can be computed with complexity $O(S^3)$. Clearly, Step 3 can also be computed using $O(S^3)$ operations. $\mathcal{Y}_{kj}$ can be computed by an algorithm of complexity $O(S^3)$. Therefore, the matrix $\mathcal{Y}$ can be computed using $O(S^7)$ operations. Finally, the matrix of eigenvectors of the STRM, $\mathcal{R} = \mathcal{D}\mathcal{Y}\mathcal{Q}$. Since $\mathcal{D}$ is a diagonal matrix, $\mathcal{Y}$ is lower triangular and $\mathcal{Q}$ is quasi-diagonal, the computation of the product of $\mathcal{D}$,$\mathcal{Y}$ and $\mathcal{Q}$ requires $O(S^7)$ operations. Thus, the eigenvectors of the STRM can be computed by an algorithm of complexity $O(S^7)$. We defer the computation of $P(t)$ to the next section; here we show that as in the case of eigenvalues, the eigenvectors can also be computed incrementally if a spare is added to the system.

The STRM for S spares completely contains the STRM for $(S-1)$ spares. The addition of a spare to a system with $(S-1)$ spares introduces S additional diagonal blocks. The matrix of eigenvectors of these diagonal can be computed by the QR algorithm. The number, $N(S)$ of non-diagonal blocks in the STRM for a system with S spares is given by the following recursive formula

$$N(S) = N(S-1) + \frac{S(S^2+1)}{2}, \qquad N(0) = 1. \tag{3.6}$$

As seen in Figure 7, $S(S^2-S+2)/2$ of the $S(S^2+1)/2$ additional (non-diagonal) blocks are extensions of the columns of the STRM for $(S-1)$ spares. From Step 2 of the Hessenberg-Schur Algorithm, it is clear that each block of matrix $\mathcal{Y}$ is dependent only on the column of blocks (of $\mathcal{Y}$) directly above it. Hence, the matrix



+ = non-zero element, . = zero

Figure 3.7: Block Structure of the STRM for S = 5 or less.

$\mathcal{Y}$ for a system with $(S-1)$ spares is completely contained in the corresponding matrix for S spares. If the matrix $\mathcal{Y}$ is available from previous computations, it is sufficient to compute the $\mathcal{Y}_{kj}$ s corresponding to the additional blocks. This can be done with complexity $O(S^6)$. In the absence of the incremental facility, the complexity is $O(S^7)$.

3.5.4 Computing the Reliability Function

The state probability vector of the Markov Chain in terms of the matrices computed so far is:

$$P(t) = (\mathcal{D}\mathcal{Y}\mathcal{Q})e^{\Lambda t}(\mathcal{D}\mathcal{Y}\mathcal{Q})^{-1}P(0), \quad where, \ \mathcal{D}\mathcal{Y}\mathcal{Q} = \mathcal{R} \tag{3.7}$$

Since $\mathcal{D}$ is diagonal, $\mathcal{Y}$ is lower triangular and $\mathcal{Q}$ is quasi-diagonal, the complexity of computing $\mathcal{D}\mathcal{Y}\mathcal{Q}$ is $O(S^7)$. If $P(0)$ is $(1, 0, \cdots, 0)$, then it is sufficient to compute the first column of $(\mathcal{D}\mathcal{Y}\mathcal{Q})^{-1}$. This first column can be computed by an algorithm of complexity $O(S^6)$. Thus, we conclude that the overall complexity of computing $P(t)$ is $O(S^7)$. The reliability function is derived by summing over the n elements of the vector $P(t)$.

3.5.5 Numerical Aspects

In this section we highlight points in the computation of the reliability function which could possibly lead to numerical instability if not carefully implemented. We do not provide the solutions to these problems here.

The eigenvalues can be computed in a numerically stable manner if the relative magnitudes of the repair rate ψ and the failure rate μ are such that the square root of their product can be computed accurately. In the case of the eigenvector computation, the diagonal similarity transformation, $\mathcal{D}$, which transforms the diagonal blocks of the STRM to symmetric matrices has as many orders of magnitude as there are spares in the system. However, the effect of the transformation on the non-diagonal blocks is such that the orders of magnitude in the transformed STRM are exactly those appearing in the original STRM (this can be achieved with proper implementation). In computing $\mathcal{F}_{ij}$ in Step 1 of the Hessenberg-Schur Algorithm, care must be taken when adding two matrices of different orders of magnitude (we know that within any matrix, the elements are of the same order of magnitude). Once the $\mathcal{F}_{ij}$ are computed accurately, the Hessenberg-Schur Algorithm computes the $\mathcal{Y}_{ij}$ in a numerically stable way [HV79].

The computation of $P(t)$ involves the computation of exponentials of the eigenvalues where the magnitude of the eigenvalues can be as large as $S\psi$ and as small as μ. Also, the elements of matrices $\mathcal{D}$ and $\mathcal{D}^{-1}$ have varying magnitudes. Thus, unless carefully computed, the accuracy of $P(t)$ in Equation 3.7 will be poor. If the relative magnitudes of $\mathcal{D}$ vary too widely (this variation increases with the number of spares), it might be necessary to decompose the fault-tolerant system into two systems based on the relative magnitudes of the eigenvalues of the STRM to handle this situation.

3.6 Repeated Eigenvalues

Finally, we present an example of a repairable fault-tolerant system modeled by ARIES for which the Lagrange-Sylvester interpolation formula is not applicable. Consider a repairable fault-tolerant system with $S = 1$, $D = 0$, $M = 1$ and $C_d \neq 1$. Figure 3.4 contains the Markov Chain for this example. With the states numbered according to Equation 3.1, the STRM for this system is :

$$\mathcal{B} = \begin{pmatrix} -N\lambda & 0 & 0 \\ 0 & -(N\lambda + \psi) & \psi \\ \mu(1 - C_d) & (N\lambda C_a + \mu C_d) & -(N\lambda + \mu) \end{pmatrix} \tag{3.8}$$

Since the eigenvalues of $\mathcal{B}$ are the eigenvalues of the individual diagonal blocks, $-N\lambda$ is an eigenvalue of this STRM. Therefore, $(\mathcal{B} + N\lambda\mathcal{I})$ is a singular matrix. Rank$(\mathcal{B} + N\lambda\mathcal{I})$ is two as seen in Equation 3.9.

$$\mathcal{B} + N\lambda\mathcal{I} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\psi & \psi \\ \mu(1 - C_d) & (N\lambda C_a + \mu C_d) & -\mu \end{pmatrix} \tag{3.9}$$

In order for eigenvalues to repeat, $-N\lambda$ must be an eigenvalue of the 2×2 diagonal block of $\mathcal{B}$ [15]. Then, it is easy to verify that $-N\lambda$ satisfies the characteristic equation of the diagonal block only if $N = \mu(1 - C_d)/\lambda C_a$. Under this condition, $-N\lambda$ has a multiplicity of two ; the third eigenvalue is $-(N\lambda + \mu + \psi)$. The rank of $(\mathcal{B} + N\lambda\mathcal{I})$ is two even when $N = \mu(1 - C_d)/\lambda C_a$. From matrix theory the number

[15]The two eigenvalues of the 2×2 diagonal block cannot repeat since this matrix obeys the Sturm Sequence property.

of independent eigenvectors corresponding to the eigenvalue $-N\lambda$ is equal to the dimension of the null space of the matrix $(\mathcal{B} + N\lambda\mathcal{I})$. Hence, we infer that there is only one independent eigenvector corresponding to the repeated eigenvalue $-N\lambda$. The Lagrange Interpolation formula is not applicable as a solution method for a system with dependent eigenvectors. Note that, if $C_d = 1$, there can be no repeated eigenvalues. Based on the Sturm Sequence property of the diagonal blocks of the STRM, we conclude that the only eigenvalue which can possibly repeat is $-N\lambda$ and the multiplicity of this eigenvalue can never exceed two. Thus, for all non-degradable repairable fault-tolerant systems with $S = 1$, *the eigenvectors of the STRM are dependent if and only if the eigenvalues are repeated.* In the next paragraph we show that systems with repeated eigenvalues (for $S = 1$) are not likely to occur in practice.

We rewrite the condition for repeated eigenvalues as follows:

$$N\left(C_a\frac{\lambda}{\mu}\right) + C_d = 1 \tag{3.10}$$

In any practical system, C_d , the coverage factor for a spare is close to one. Suppose that $\lambda = \mu$ and $N = 1$. Then by Equation 3.10, C_a, the coverage factor for an active module must be close to zero. Further, for practical fault-tolerant systems, it is reasonable to assume that $\lambda \geq \mu$, i.e., the failure rate of an active module is greater than or equal to that of a spare [MA70]. Also, the number of modules in the active configuration is generally greater than one. Under these practical assumptions, C_a must be even closer to zero than just discussed. Systems with such poor values of active module coverage are of no practical use and there is

no reason to analyze them for reliability. Hence, we conclude that the eigenvalues of practical repairable fault-tolerant systems with one spare modeled by ARIES have distinct eigenvalues.

3.7 Conclusions and Future Work

In this chapter we have emphasized on the computational complexity aspect of the solution to the ARIES reliability model for repairable fault-tolerant systems. We have reduced the complexity of computing the reliability function from $O(S^{15})$ to $O(S^7)$ for a repairable fault-tolerant system with S spares when the eigenvalues of the STRM are distinct. If it is true that most practical fault-tolerant systems have distinct eigenvalues, then, our method offers a satisfactory solution. We have shown in Section 3.6 of this chapter that for non-degradable repairable systems with a single spare, all practical systems have distinct eigenvalues. It is not clear whether this property holds for systems with more than one spare.

It might be the case for the ARIES model that the STRM is non-defective although the eigenvalues are repeated. In such a case it is theoretically possible to find a *block-diagonalizing* similarity transformation for the STRM but, the Hessenberg-Schur algorithm may not be applicable to equations described by Equation 3.4 for which the diagonal blocks involved have common eigenvalues. However, such a case of non-defective STRMs cannot occur if we can establish that repeated eigenvalues imply the dependence of eigenvectors for all repairable systems (this implies that the block diagonalizing transformation does not exist and our method is therefore inapplicable). In Section 3.6, we have shown that for non-degradable

repairable systems with one spare, the existence of repeated eigenvalues imply the dependence of eigenvectors. Further work needs to be done to see if this result can be extended to repairable systems with more than one spare. Finally, the numerical consequences of the computation outlined in Section 3.5.5 also need attention.

Chapter 4

Reliability Models for Fault-Tolerant Private Networks

4.1 Introduction

Traffic networks such as telephone networks, data networks and private networks are built on an underlying *facilities network*. The facilities offered by the facilities network are switches and circuits. In a public (telephone) network, *trunks* or circuits are logical connections between locations; circuits are carried over a sequence of physical links. See Figure 4.1. (A good discussion of this physical and logical architecture can be found in the introduction of [CLV+91].) In public switched telephone networks (which are logical networks) call switches are connected using these trunks.

A private network is one of the many logical traffic networks built on a facilities network. It is along distance network meant for the use of a single organization such as a bank to interconnect its branches or an airlines company to interconnect reservation systems. While physical transmission facilities are shared, the set of

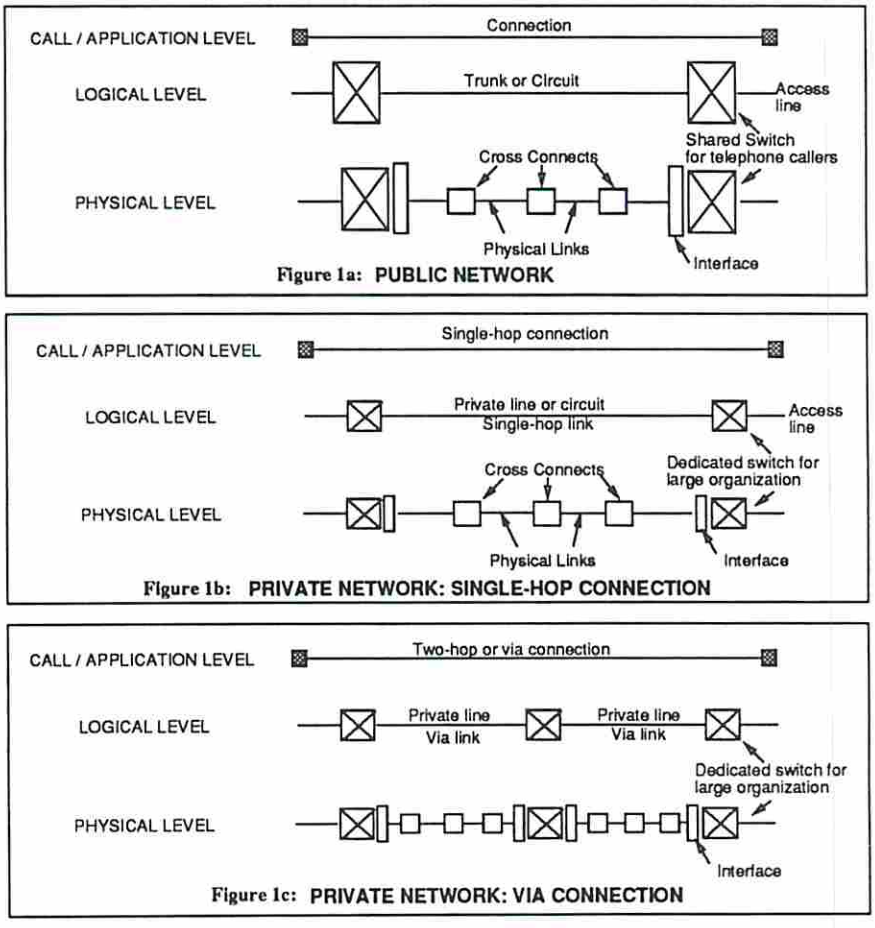


Figure 4.1: Simplified Comparison of a Public Network with a Private Network

circuits used by various traffic networks are disjoint. Circuits assigned to private networks are called *private lines*. They do not pass through the call switches of the public network; switches are not shared across networks (Figure 4.1). A private network customer can lease a circuit between two fixed locations at a range of possible transmission rates. The nodes of the private network are small switches or multiplexors owned or leased by the organization. To the end user, the private network looks like the public network, allowing them to make voice calls or establish data connections.

Although the underlying facilities network is not fully connected, for many applications, the traffic network is potentially fully-connected, because a logical circuit can be set up at the desired rate between any two locations. In a graph model of a private network where edges represent leased lines and nodes represent switches, the graph is potentially complete (although the actual implementation will not usually be complete). Because the topology and switching equipment owned by the organization are generally simpler than in the public network, connections in the private network are usually implemented using a single private network link (leased line), or perhaps a pair of links tied together by a dedicated switch (Figure 4.1). Going to more than two-hop routing increases complexity and network resources used without significantly improving network performance [oTSABL83, pages 89,90,114].

In a public network, fault-tolerance is provided at many levels [Mor89] and most of these network fault-tolerance methods are for protection against link failures; these are an order of magnitude more common than failures of nodes or switches,

which have their own inherent fault-tolerant architectures. Individual links have protection circuits for the replacement of a single failed circuit. Recently, re-routing of circuits in response to link cuts has been proposed [CLV+91] [Gro87] [GVsM90]. Finally, the most dramatic reliability improvements are made at the call or application level, where alternate routing algorithms allow calls to be sent on alternate routes when circuits or groups of circuits fail. For switched services such as the telephone service, when a new call set-up is attempted and is blocked on the first route, the network's routing algorithm tries alternate routes from a list kept for this purpose (the routing algorithm does not seek out routes exhaustively). An example is dynamic non-hierarchical routing (DNHR), the call routing scheme used in AT&Ts' long distance network [ACM81]. At the network level, little can be done to save calls in progress when a serious failure occurs; these calls will probably will be lost. This is undesirable, but in terms of the total number of calls potentially affected by a severe outage, the number of interrupted calls is small. The combination of different fault-tolerance techniques at different levels of the network makes the call processing network reliable; many types of network component failures are almost invisible to network users.

Private lines do not pass through the call switches and thus do not benefit from the switches' alternate routing capability. Also, while a telephone caller uses a circuit for a short period of time, a private line uses one or more circuits for an indefinite period. So, although cutoffs are not as serious a problem with individual calls, they are more likely, and more of a problem with private lines. Fault-tolerance features can be added to private networks to improve the network's reliability and

protect against circuit failures. The details of such a fault-tolerance scheme are discussed in the next section.

Much has been published on reliability modeling of networks and distributed system applications [RA90a] [RA90b] [tro89] [tro91]. In this chapter we develop a state-space model for fault-tolerant private networks and analyze it to evaluate reliability of private networks. In Section 4.3 we describe our model in the general framework of network reliability models. In Section 4.2 we describe the operation of a fault-tolerant private network focusing on the features and behavior that we need to represent in our reliability model. In Section 3.2 we develop a state space description of the network's behavior, including both failure and repair processes. The state space gets unmanageably large, so we replace it by a lumped, but approximate model. One objective is to construct approximate reliability models for cases where the original problem is much too big to be solved by analytical or numerical methods and to evaluate the solution to the smaller model. The approximate models are constructed to be of reasonable size. This is not the same as solving sub-models by decomposition and combining the solutions; we solve one reduced model through *lumping* similar states together. Aupperle and Meyer [AM91] also lump states of degradable multiprocessor models generated automatically from a high-level specification, but, they do not allow the lumped process to be non-Markovian.

In Section 4.5, we compare results obtained by solving the lumped model by numerical methods with those obtained by solving the original model by Monte Carlo Simulation. In the twenty four cases we experimented with, results from

the lumped model tracked those of the original model very closely. Further, we use the model to examine the impact of an alternate repair and reconfiguration method on the network's reliability. The results of this analysis indicate that repair and reconfiguration strategies have a major impact on the reliability of a private network. Our results also indicate that the network reliability is highly dependent on the set of office-pairs that need to be connected over the specified interval.

4.2 Fault-Tolerant Private Network Operation Model

In this section, we introduce some terminology and list the simplifications we have made in defining the operation of a particular fault-tolerant private network, the *Non-Reconfiguring-on-Repair* (NROR) Network. The reliability models developed in this chapter are for the NROR network presented in Section 4.2.1 and for a variation of this, the *Reconfiguring-on-Repair* (ROR) Network outlined in Section 4.5.

As mentioned in the introduction, the private network connects offices of a single corporation through dedicated leased lines. We refer to the set of office pairs requiring continuous connection for a given session as the *traffic pattern.* In defining the NROR network operation, we place restrictions on the traffic pattern, leased line capacity and demand. These simplifications are so that we can focus attention on the effectiveness of using lumped approximations to solve a network reliability problem. In the future, we plan to generalize our models to include

a more realistic representation of traffic, capacity and demand. For the NROR network, we restrict the traffic pattern to be L single-hop connections and refer to each connection as single-hop *link*. A single-hop link employs exactly one leased line to realize the connection. Further, we assume that each connection always demands a bandwidth equal to that of a single circuit. Also, that the leased line capacity is equal to that of a circuit's bandwidth. For applications such as teleconferencing, the traffic pattern is likely to be static and so we assume that it is unaltered for a given session. The NROR model does not place any restriction on the topology of the private network, however. This is because of our assumption that a single-hop link uses up all the capacity on a leased line. So, fictitious single-hop links can be introduced into the traffic pattern to account for the topology of the network. A fully connected network is used as a template and any topology can be accommodated by appropriately placing single-hop links to account for the topology. As the reader may infer later, the private network topology only affects the method of parameter estimation (in Section 4.4.3) but makes no difference to the analytic expressions derived in Sections 4.4.1 and 4.4.2. Since the former is done through simulation, the *missing links* in the fully connected template can be specified in the input to the program.

To complete our definition of the NROR network, we restrict alternate connections to be two-hop connections called *vias* and refer to each hop in the via as a *via link* (Figure 4.1). This simplifies the derivation of lumped transition rates; for

larger lengths, some analytic expressions would be more complicated, although *doable* in principle. With these simplifications, we discuss the fault-tolerant operation of the NROR network, next.

4.2.1 Failure, Repair, and Reconfiguration

In this section, we discuss the mechanisms of failure, reconfiguration and repair through the example shown in Figure 4.2. In the figure, solid lines are links carrying connection, dotted lines are free links, and dashed lines are reserved single-hop links.

When an active single-hop link fails, the single-hop connection is replaced by a via by tieing two free links at a user's dedicated switch. We assume that if an alternate connection is available, the network recovers perfectly, i.e., that the switching mechanism never fails when an alternate route is available. Imperfect coverage in fault tolerant networks is considered in detail in [RZ90]. Although we do not include reconfiguration processes here, any overheads during reconfiguration can be modeled through the coverage parameter; this is because the processes of reconfiguration operate on time scales which are typically six orders of magnitude smaller than that of failure processes [MST85] [DT89].

To continue description of the network operation, failed vias cannot be re-routed over vias; if a via fails before the corresponding single-hop link is repaired, the network does not seek out alternate vias. To achieve a reasonable level of overall network reliability, repair rates are usually significantly higher (by three orders of magnitude) than the failure rate of a via. Thus, we do not expect this

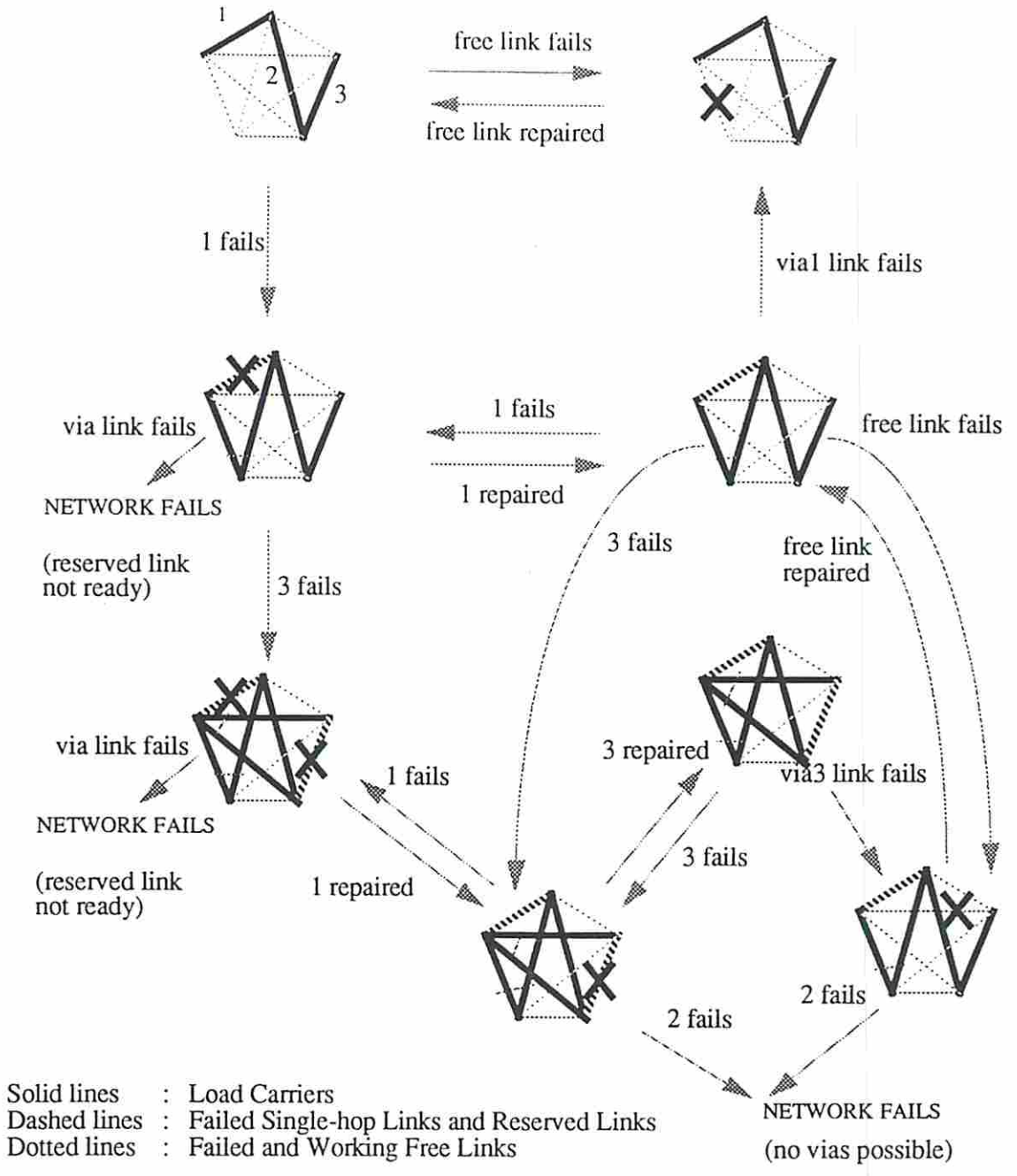


Figure 4.2: Failure, Repair and Reconfiguration.

assumption to have a serious impact on the results. On the other hand, it greatly simplifies the model.

When a link is repaired, it becomes available for providing a connection for the private network. The NROR network does not restore on repair of a single-hop link. When a single-hop link is repaired, it is placed in a wait-state. Only when *its* via link fails is the corresponding reserved single-hop link used, i.e., the repaired single-hop link is *reserved* to carry the connection for one specific office pair in the traffic pattern. Such a situation occurs if, in the interest of simplicity, the list of alterate routes for each single-hop connection is never updated. Then, the initial list of alternate routes cannot include the set of single-hop links realizing connections in the traffic pattern initially.

Given that the network is operational initially, it is considered to fail at the instant when a connection in the traffic pattern fails and this connection cannot be re-established through an alternate route. In the next section, we construct a formal model for the reliability of NROR networks and use the model to compare alternate repair and reconfiguration strategies. In Section 4.5, we consider the effect of making the network restore on repair. In this case, when a single-hop link is repaired, its via is torn down and the connection is restored through the single-hop link just repaired.

4.3 Network Reliability Modeling

Network reliability has traditionally been evaluated by combinatorial methods and a significant amount of research in this area is based on the theory of combinatorial

enumeration [Col87]. Several papers on implicit enumeration, reliability bounds and approximations under various network operation criteria have been published [RA90a] [RA90b] [tro89] [tro91]. However, the problem of reliability evaluation is not always amenable to solution by combinatorial methods and an alternate model such as a state space model is often used to evaluate network reliability for the general case. Thus, in Section 4.3.1 we outline the general procedure for network-reliability computation in the framework of a state space model for the private network introduced in Section 4.2.1 of this chapter and give an example where the problem reduces to one of combinatorial enumeration; in this description we use terminology from the combinatorial modeling approach [Col87] wherever possible. In Section 4.3.2, we define a Markov reliability model for our private network.

4.3.1 Background

As is usual in the combinatorial approach, we first model the private network by a *probabilistic graph* whose n vertices represent the offices of the organization owning the private network and $|E|$ edges, leased lines or *links*. We assume that links fail independently and that their times-to-failure are identically distributed. We also assume an independent but identical repair facility for each link. Further, we define each combination of failed/working links to be a *state* in the state space model. As in any specification for network reliability, we have a *Network Operation Criterion* and declare each state to be a *pathset* or *cutset* depending on whether or not the arrangement of failed and working links in the state satisfies the specified criterion

for network operation. Standard examples of network operation criteria are *two-terminal reliability*, *k-terminal reliability* and *all-terminal reliability* [Col87]. The operation criterion for our private network is that connections must be maintained continuously between a specified set of offices pairs for the duration of several hours (or a few days) either by means of a direct connection or by an alternate connection employing exactly two links. It is assumed that the network begins operation in a pathset. Network reliability is the probability that starting from a pathset at time zero, the network operates continuously over an interval $(0, t]$. That is, the network should not visit any cutset in this interval, for a cutset is a state of network failure. In the next paragraph we calculate reliability from state space representation and give an example where the combinatorial method is useful.

Let $\mathcal{A}$ denote the set of all pathsets under a given network operation criterion and let $P_s(X_t = j, X_u \in \mathcal{A}; 0 \leq u < t)$, denoted by $P_s(X_t = j, B)$ be the probability that the network is in pathset j at time t without ever having failed in the interval $[0, t)$. Then, network reliability is given by $R(t) = \sum_{j \in \mathcal{A}} P_s(X_t = j, B)$, where s is a given initial pathset. In some cases, $P_s(X_t = j, B)$ is available in closed form. An example is when failed links cannot be repaired and and the lifetime of links are independent and identically distributed. In this case, $P_s(X_t = j, B) = P_s(X_t = j) = p(t)^i(1 - p(t))^{|E|-i}$ for a pathset j having exactly i working links; $p(t)$ is the probability that a link is working at time t and is computed from a single independent link failure process. Of the set of states with exactly i working links, only $N_i \leq \binom{|E|}{i}$ arrangements are pathsets. It is clear then, that for this case, the only information necessary for reliability computation is a count of the

number of pathsets having exactly i working links for $0 \leq i \leq |E|$. However, the number of pathsets in any network model is usually exponential in the number of links $|E|$ and so exhaustive enumeration is out of the question. The problem of network reliability evaluation is computationally difficult even for this special case where it is unnecessary to solve for $P_s(X_t = j, B)$. The task only gets harder if this probability must be computed for each pathset and summed together.

For the networks we are modeling, the irregular topology (which translates to arbitrary state dependent coverages) combined with the need to model repair processes precludes any easy method for computing $P_s(X_t = j, B)$ (despite our assumptions of independent link repair and failure). A possible solution approach is to model the failure and repair processes by a continuous-time Markov Chain and solve it to obtain $P_s(X_t = j, B)$. Accordingly, in the next section, we present the state space of our Markov reliability model.

4.3.2 A Markov Reliability Model

As discussed in the previous section, we model the failure and repair processes of fault-tolerant private networks using a state space model assuming independent failure and repair of individual links. We further assume that the time-to-failure and time-for-repair are exponentially distributed with rates λ and μ respectively. In this section we evaluate these assumptions for a private network and define a state space for the NROR network described in Section 4.1.

In public networks, circuits do not normally fail independently since two or more circuits may share a physical link. But, in private networks, although it is

not always easy, network providers can often engineer the private network so that most of the circuits are realized from physically diverse links [MHLK87]. This is easier if the underlying network is large relative to the size of the private network, providing more possible routes. There is no guarantee of independent failures, however. Also, in reality, outage and repair times are not exponentially distributed. But, these assumptions greatly simplify exploratory modeling, so we accept them in our model. Under these assumptions, we can model the fault-tolerant behavior of the private network as a continuous-time time-homogeneous absorbing Markov Chain with a finite state space consisting of pathsets and a state F. F denotes all cutsets.

Figure 4.3 shows an example of some pathsets resulting from single-hop link failures in a NROR network with five nodes and three office-pair connections (A-B, B-C and C-D). The figure displays all possible arrangements of single-hop links and vias found in any pathset of the network. Notice that the three connections A-B, B-C and C-D are maintained in every state. The figure only shows transitions caused by single-hop link failure; free links and failed single-hop links are ignored. When free link failures are added, then, for each specific arrangement of single-hop and via connections there are several possible arrangements of failed and working free links. Each of these combinations is a valid pathset. Also, since the network does not restore on repair, there is an inactive single-hop link (either failed or reserved) corresponding to each via present in the network; all combinations of failed and reserved links also yield pathsets.

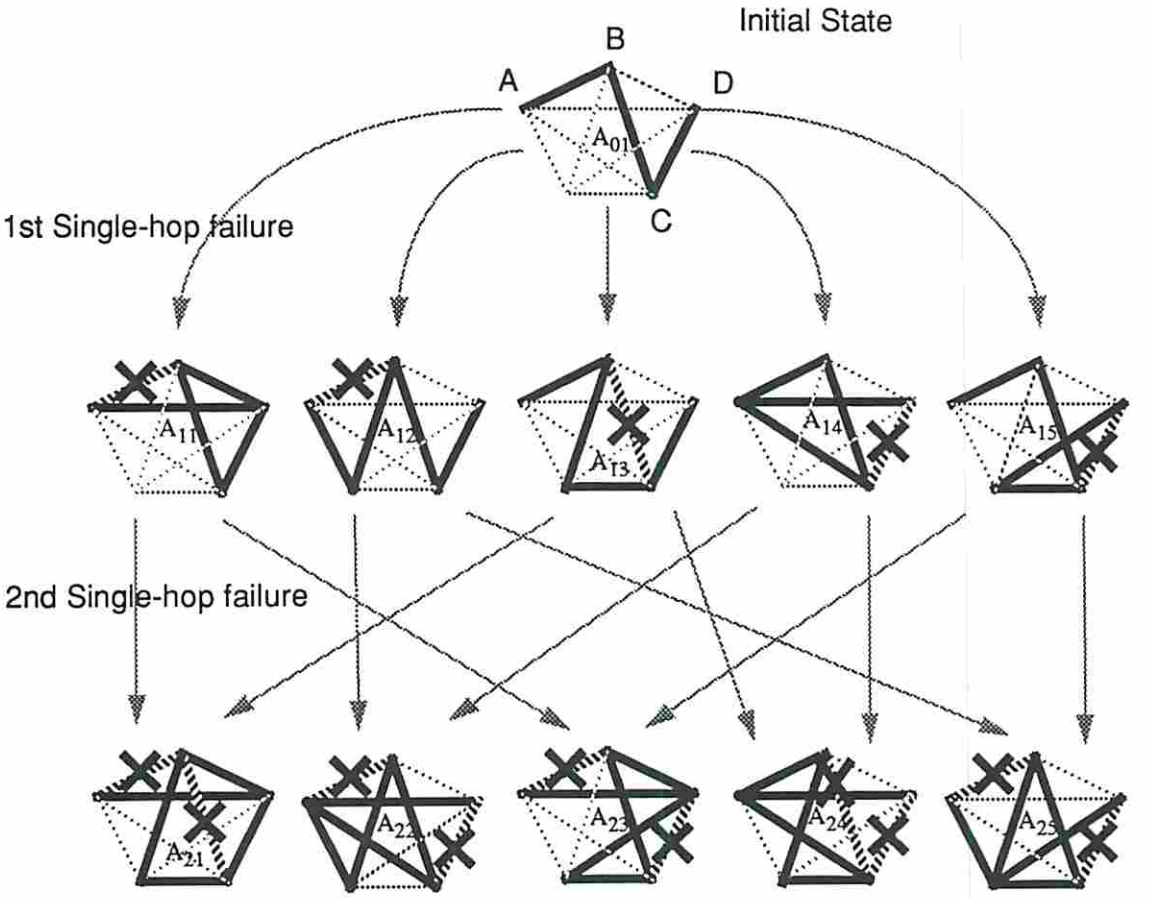


Figure 4.3: Partial State Space for a 5 Node Network.

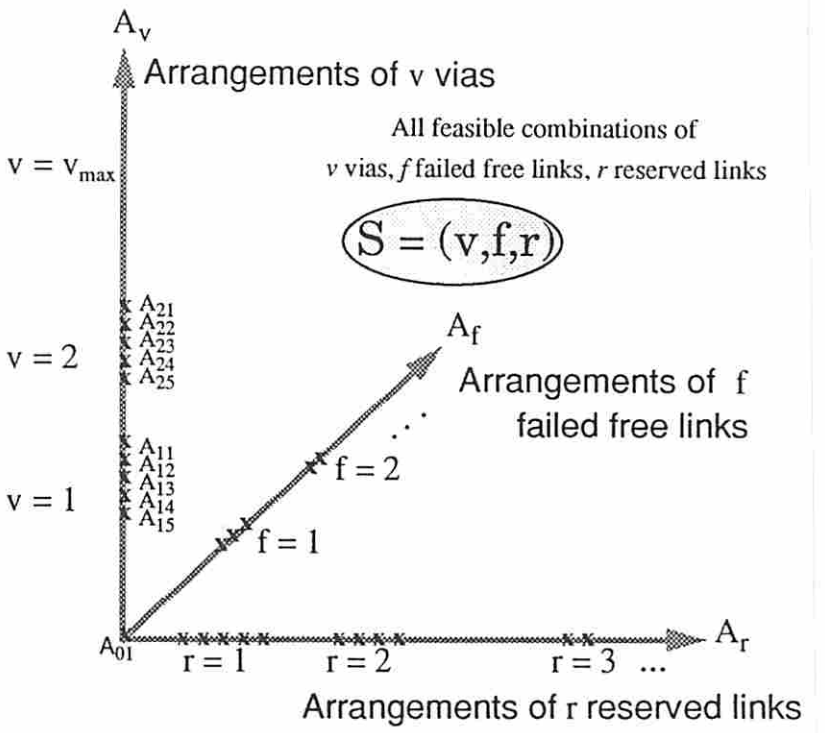


Figure 4.4: The 3D State Space of the Original Markov Chain.

Figure 4.4 shows the three dimensional state space of the Markov reliability model. All states represented in the figure are pathsets. Each state (a feasible point) in the space is a combination of a specific arrangement of v vias, a specific arrangement of f failed free links and a specific arrangement of r reserved links such that $v + d = L$ (implying that all connections are maintained). We conclude this section with some observations about the state space.

In Figure 4.4, the vertical axis A_v represents the arrangement of vias, which is also the arrangement of single-hop links. The A_f and A_r axes represent the arrangement of failed free links and reserved single-hop links respectively. The states shown in Figure 4.3 are via arrangements corresponding to the A_v axis in Figure 4.4 for $f = 0$ and $r = 0$. V_{max}, the maximum number of vias that the network can support for the given traffic pattern is 2 for this example.

Given a fixed via arrangement of v vias, there are $\mathcal{N}_f = \binom{N-L-2v}{f}$ arrangements of failed/working free links for each $0 \leq f \leq N-L-2v$ and $\mathcal{N}_r = \binom{v}{r}$ arrangements of failed/reserved single-hop links for each $0 \leq r \leq v$. In general, the exact value of $\mathcal{N}_v$, the number of via arrangements having exactly v vias can only be found by enumeration. Once the $\mathcal{N}_v$ arrangements are enumerated for $0 \leq v \leq V_{max}$, the states of the Markov Chain are completely specified. Note that V_{max} is an unknown.

The state space described in this section is clearly too large even for specification of the generator matrix. Thus, solution methods requiring specification of the generator matrix are ruled out. One may solve the Markov Chain by Monte-Carlo simulation or alternately by a state space reduction method such as *lumping*

[KS76]. Our approach is to construct a new state space model with a reasonable number of states by lumping together similar states and to solve the resulting model numerically. We present this approach in the next section.

4.4 Construction of a Lumped Markov Model

In the previous section we saw that the state space of the reliability model for the private networks was too large even to input to any Markov solver. In this section, we simplify the model by *lumping* [KS76] [Nic90] the Markov Chain. The process of lumping is achieved in two steps. The first step is to partition the state space into a convenient set of subsets and replace each subset by a single state. The second, to derive transition rates for the smaller process. Thus, in this section we define a partition on the the state space shown in Figure 4.4 and derive lumped transition rates. The model constructed by the procedure (a heuristic) described here is a Markov model which can be solved by standard Markov solvers. However, the constructed model is, in general, only an approximation because the solution of this model is not the same as that of the original Markov Chain summed over (original) states in each subset; that is, in general, the smaller process is non-Markovian. When the smaller process is also Markovian, the original Markov Chain is said to be *lumpable with respect to the given partition.* In this context, Aupperle and Meyer present a group-theoretic approach for automatically identifying such lumpable partitions in state spaces occurring in degradable multiprocessor performability models [AM91]; our goal is to construct a smaller model even when the Markov Chain is not lumpable.

In the next paragraph we define a partition on the state space. In Sections 4.4.1 and 4.4.2 we describe the derivation of lumped transitions in detail. Our model is hybrid in that these lumped rates are derived in parametric form. The parameters require estimation and for this we use an efficient simulation (which is different from solving the original Markov Chain by Monte-Carlo simulation). In Section 4.4.3 we explain the parameter estimation method. In Section 4.5 we present some experiments to demonstrate that the constructed reliability model yields acceptable results. For this, we simulate the original Markov Chain by Monte-Carlo simulation and compare with the solution of the constructed Markov model using the software package SHARPE [ST87].

We first partition the state space shown in Figure 4.4 by grouping together pathsets having a common arrangement of single-hop and via connections. The resulting state space can still be very large. So we further lump together lumps with exactly v vias. Under this partition, we can characterize each lumped state $\mathcal{S}$ as follows:

- d : number of active single-hop links. $d = L - v$.
- u: number of working free links. $u = N - L - 2v - f$.
- v : number of vias.
- r : number of links that have undergone repair but are currently not used (reserved links).
- f : number of failed free links.
- w: number of failed single-hop links. $w = v - r$.

Of this list, the number of vias, the number of failed free links and the number of reserved links are sufficient to characterize the state - the others are dependent variables so that $\mathcal{S} = (v, f, r)$.

The following observation about this partition is used in Section 4.4.3 for reducing computational effort. Consider the set of via arrangements with exactly v vias. Each via arrangement A_{vi} in this set is common to lumped states differing only in the f and r parameters. Thus, if the set of via arrangements is specified, the assignment of original states to $(N - L - 2v + 1) \times (v + 1)$ different lumps is specified. Next we derive lumped transition rates.

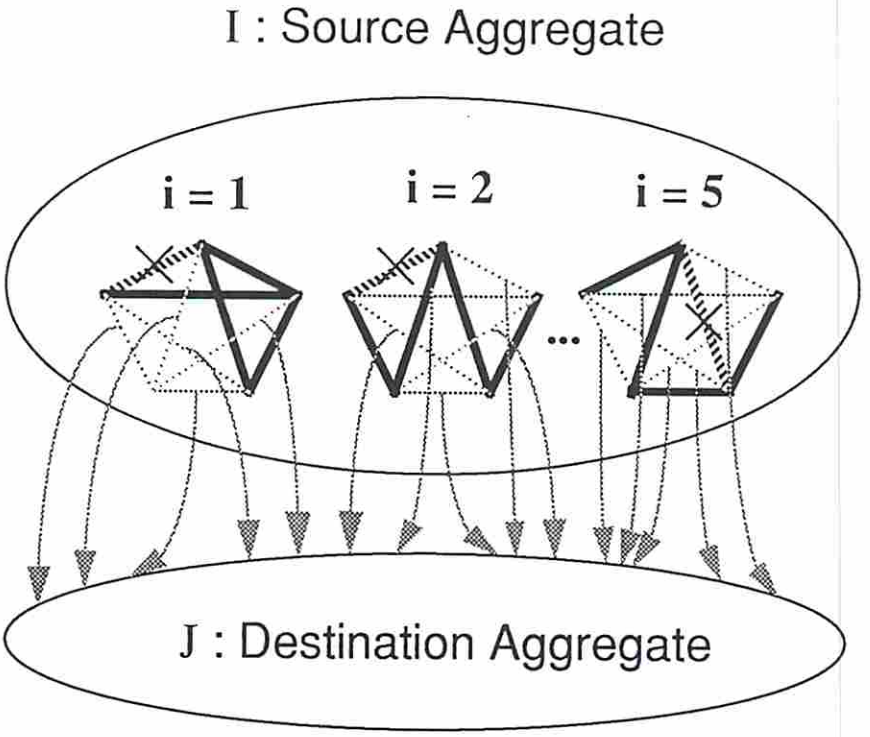
4.4.1 Transition Rates of the Lumped Model

The transition rates for the lumped model are obtained by *Fixed-Weight Aggregation* [HS84] described as follows. Let $\{1, 2, \ldots, V, F\}$ be the subsets of the partition of the original state space. Let I, J be two distinct lumped states and q^e_{IJ} be the transition rate from I to J due to event e. Then, by the Fixed-Weight Aggregation formula,

$$q^e_{IJ} = \sum_{i \in I} w^i \sum_{j \in J} r^e_{ij} \tag{4.1}$$

where, w^i are weights assigned to each state $i \in I$ and r^e_{ij} are rates corresponding to event e. Each original state $i \in I$ is assigned a weight w^i and the *lumped* transition rate is obtained as a weighted sum (Figure 4.5). If all states in I are equally likely sources of the transition, then, $w^i = \frac{1}{|I|}$.

Six events cause state transitions in our network model :

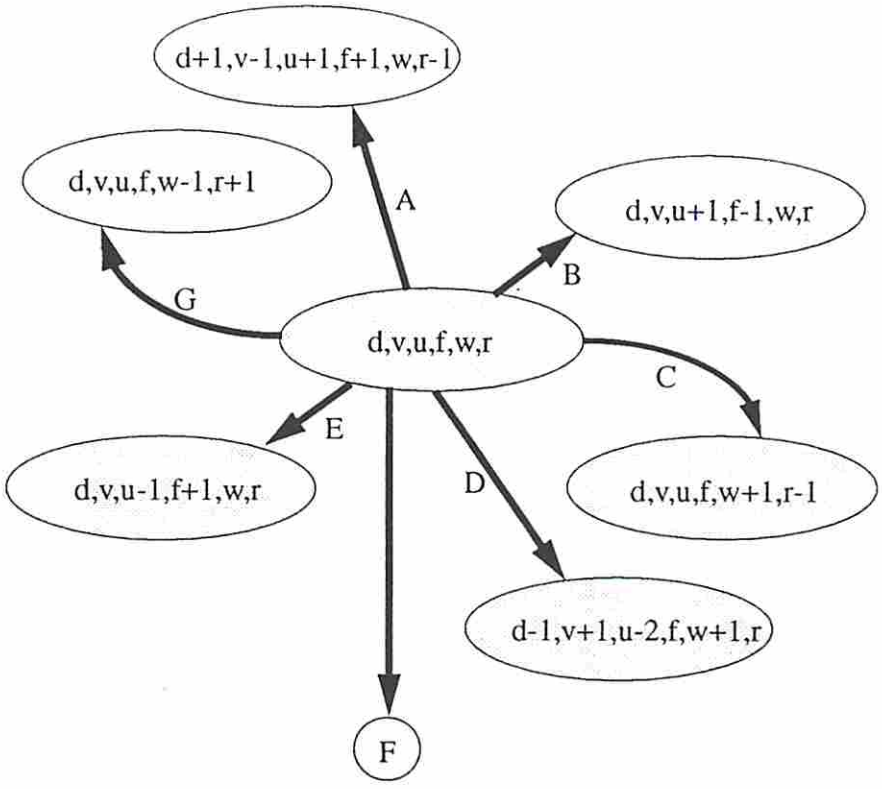


e : Free link failure.
λ : Free link failure rate.
$\sum_{j \in J} r_{ij}^{e} = 5\lambda$.
$w^{i} = 1/5, \forall i$.
$q_{IJ}^{e} = 5\lambda$.

Figure 4.5: Deriving Aggregate Transition Rates.

- **single-hop link failure** - a currently working single-hop link fails.
- **via link failure** - one of the via links fails.
- **free link failure.**
- **single-hop link repair.**
- **free link repair.**

These are displayed in Figure 4.6. Note that except for single-hop and via link failures, transitions originating from a lump due to a given event have destinations in exactly one lump. Also, each original state in the source lump has exactly the same number of (original) transitions to the destination lump so that each original state contributes exactly the same rate to the flow into the destination state. Applying Equation 4.1 to this case with equal weights, the transition rates in the lumped model for events (B,C,E,G) are $(\mu f, \lambda r, \lambda u, \mu w)$ as shown in Figure 4.6. For single-hop and via failures, transitions are to exactly two other lumps, one of which is the failed state F. The rates contributed by an original state to flow into the destination lump on via link failure are are $2\lambda v(r/v)$ to the working lump and $2\lambda v(1 - r/v)$ to F. Here, r/v is the probability of network recovery when given that a via link fails in an original state belonging to the lumped state $\mathcal{S}$. Since each original state contributes equally to the transition rate of the lumped model, r/v is also the probability of network recovery when given that the network is in lumped state $\mathcal{S}$ and a via link fails. This probability denoted by $C_2(\mathcal{S})$ is the *coverage* on via link failure for the network in lumped state $\mathcal{S}$.



v : Number of Vias.
f : Number of Failed free links.
r : Number of Reserved links.
d : Number of Single-hop links.
u : Number of Working free link failure.
w : Number of Failed Single-hop links.

A : Via link failure, $2v\lambda C_2(\mathcal{S})$
B : Free link repair, $f\mu$
C : Reserved link failure, $r\lambda$
D : Single-hop link failure, $d\lambda C_1(\mathcal{S})$
E : Free link failure, $u\lambda$
G : Single-hop link repair, $w\mu$
F : $2v\lambda(1 - C_2(\mathcal{S})) + d\lambda(1 - C_1(\mathcal{S}))$

Figure 4.6: State Transitions in the Aggregate Model.

So far, the only information required to compute the transition rates for the lumped model are the failure and repair rates of individual links and the lumped state information $\mathcal{S} = (v, f, r)$. Also, so far, no information has been lost through lumping. In using fixed-weight aggregation for single-hop link failure, however, there will be a loss of information because of the pattern sensitiveness of the network's recovery (we cannot enumerate all patterns due to the size of the state space). This is explained in the next paragraph.

Consider the failure of a single-hop link in an original state of lumped state $\mathcal{S}$. The number of original transitions due to single-hop failure to the working lump is *not the same* for *every* original state in $\mathcal{S}$. It depends on the arrangement of the initial traffic pattern, vias and working free links in the network, and on the particular single-hop link that fails (Figure 4.7). In general, the fraction of the d transitions from an original state in the source lump $\mathcal{S}$ to the common destination lump is different for different original states in $\mathcal{S}$. The fixed weight aggregation method replaces the original transitions by a single transition to the destination lump with rate equal to a weighted average of the original transition rates. This is an approximation (unless we can prove that the Markov Chain is lumpable). In the next section we derive Equation 4.2 for calculating the *exact* coverage on single-hop link failure for arbitrary traffic patterns when the network is in lumped state $\mathcal{S}$. Such an approach of accounting for pattern sensitivity by coverages in a Markov model has been used by Najjar and Gaudiot [NG87] and Abraham and Padmanabhan [AP88] in the context of hypercube reliability and by Smith and Trivedi for a class of local area networks [ST89]. They can take advantage of the

regular topologies because they assume that the traffic *looks* the same from every node (statistically) and so it is sufficient to consider one node. Since the traffic on private networks over the interval of interest is virtually static, we cannot base our results upon the view from one node. We can, however, parameterize the coverages as follows.

$$C_1(\mathcal{S}) = \sum_{i=1}^{\mathcal{N}_v} \sum_{l=1}^{d} \left\{ 1 - \sum_{t=0}^{\min\{u, M_l(A_{vi})\}} \frac{2^t \binom{M_l(A_{vi})}{t} \binom{N-L-2v-2M_l(A_{vi})}{u-t}}{\binom{N-L-2v}{u}} \right\} \frac{1}{d} w^i \qquad (4.2)$$

For each state $\mathcal{S}$, the unknowns in the above expression are $M_l(A_{vi})$, the maximum number of alternate two-hop paths as seen by failing single-hop link l in the i^{th} arrangement of v vias (see Figure 4.7), $\mathcal{N}_v$, the number of distinct arrangements of v vias in the state space and $w^i, 1 \leq i \leq \mathcal{N}_v$, the *weights* for each via arrangement required for fixed weight aggregation. We must bear in mind that since the original Markov Chain is, in general, not *lumpable*, the lumped model is an approximation even if these unknown parameters are exactly found, say, by exhaustive enumeration. Next we derive Equation 4.2.

4.4.2 Recovery Probability on Single-hop Link Failure

The problem we address in this section is as follows. Given that the network is in aggregate state $\mathcal{S}$ and a single-hop link fails, find the probability $C_1(\mathcal{S})$ that the network recovers from the failure. The network cannot recover if there is no alternate route to replace the currently failed single-hop link.

Consider an lumped state $\mathcal{S}$. Partition this state $\mathcal{S}$ into $\mathcal{N}_v$ subsets where $\mathcal{N}_v$ is the number of distinct arrangements of v vias in a working state $\mathcal{S} = (v, f, r)$. These correspond to arrangements $A_{vi}, 1 \leq i \leq \mathcal{N}_v$ in Figure 4.4. Let $S_i, 1 \leq i \leq \mathcal{N}_v$ be the subsets obtained by partitioning $\mathcal{S}$. Applying the law of total probability, $C_1(\mathcal{S})$ is given by :

$$C_1(\mathcal{S}) = \sum_{i=1}^{\mathcal{N}_v} C_1(S_i).w^i$$

where w^i is the probability of occurrence of via arrangement A_{vi} when restricted to lump $\mathcal{S}$.

Observe (Figure 4.3) that, each arrangement of v vias identifies a set of d working single-hop links. Denote the d working single-hop links in a given arrangement by $\mathcal{D} = \{l, 1 \leq l \leq d\}$. For each lumped state $\mathcal{S}$ for which A_{vi} is a feasible arrangement, these d links are uniquely identified and are the same for every original state $s \in S_i$. Clearly, the network's recovery on single-hop failure (given that it is in some state $s \in S_i$) depends on which particular single-hop link in $\mathcal{D}$ fails. $C_1(S_i)$ is the probability that the network recovers given it is in some state $s \in S_i$ when the single-hop failure occurs, so that,

$$C_1(\mathcal{S}_i) = \text{P(Network recovers} \mid \mathcal{S}_i) = \sum_{l=1}^{d} \text{P(Network recovers} \mid \mathcal{S}_i \text{ and } l).\text{P}(l \text{ fails} \mid \mathcal{S}_i)$$

re-written as,

$$C_1(\mathcal{S}_i) = \sum_{l=1}^{d} C_1(\mathcal{S}_i, l).\frac{1}{d} \tag{4.3}$$

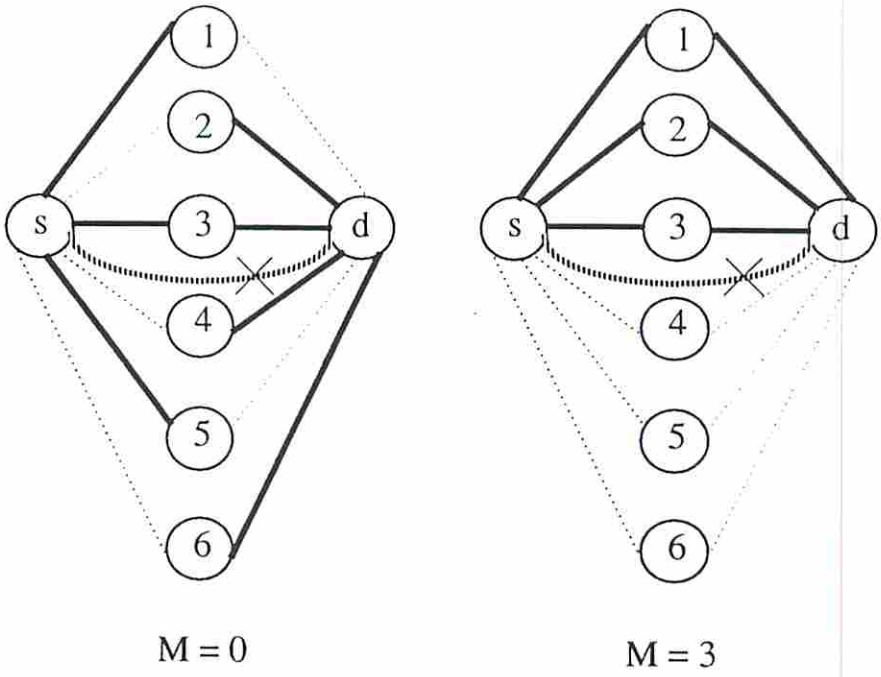


Figure 4.7: Effect of Traffic-Pattern on Network Recovery.

Since $\mathcal{S}_i \subseteq \mathcal{S}$, it is identified by v, u, r and A_{vi}. However, the recovery probability is independent of the reserved links so that $C_1(\mathcal{S}_i) = C_1(A_{vi}, v, u)$. In the rest of this section we derive an expression for $C_1(A_{vi}, v, u, l)$.

Recall that states $s \in S_i$ differ only in the arrangements of free links and reserved links; they have exactly the same arrangement of vias. When a given single-hop link, l, fails, the network recovers with a probability equal to zero or one depending on whether or not a 2-link path is available to replace failed link l. The availability of the 2-link path depends on the arrangement of the free links adjacent to the failed single-hop. The only links which are *useful* to via formation are those which are adjacent to the currently failed link; see Figure 4.7.

If, however, these links are already single-hop links in use between other node pairs or are marked *permanently used*, they cannot participate in via formation. (That is, vias may be formed only of initially free links). Figure 4.7 shows two examples to illustrate that some links adjacent to the failed link can never serve as vias in the event of failure. In the first example, the traffic pattern excludes all possibility of successful recovery on failure of the single-hop link indicated. In the second example showing a different traffic pattern, three alternate routes are available for that same single-hop link. Define $M_l(A_{vi})$ to be the maximum number of 2-link paths as seen by single-hop link l in the presence of v vias in the i^{th} arrangement, and when every free link in the network is working. In Figure 4.7, the values of M are zero and three, respectively. When a via replaces a failing single-hop link l in arrangement A_{vi}, the arrangement changes to $A_{(v+1)k}$ with $M_l(A_{(v+1)k}) = M_l(A_{vi}) - 1$, for some via arrangement k, as expected. However, the via-links consumed by the failing single-hop link may decrement the $M-$values of other single-hop links in the network, even making some of them equal to zero. This interference can inhibit future recoveries from single-hop link failure.

Thus, given the traffic pattern and the via arrangement A_{vi}, a necessary condition for the network to recover on failure of l is for $M_l(A_{vi})$ to be greater than zero. For each working single-hop link that this is true, the network is guaranteed to recover if the arrangement of working and failed free links is favorable in that it permits an alternate 2-link path for l; otherwise, the network fails. In summary, the conditions for network recovery from failure of single-hop link l are as follows.

$$C_1(M_l(A_{vi}), A_{uj}, v, u) = \begin{cases} 1, & M_l(A_{vi}) > 0, A_{uj} \text{ is favorable.} \\ 0, & \text{otherwise.} \end{cases}$$

The above equation is the conditional probability $C_1(S_i, l, A_{uj})$.

Let us consider only those single-hop links for which $M_l(A_{vi}) > 0$ and uncondition on the set of arrangements with exactly u working free links. Let $\mathcal{N}_u$ be the count of all possible arrangements of u working free links in the presence of the via arrangement A_{vi}. Assuming that each arrangement $A_{uj}, 1 \leq j \leq \mathcal{N}_u$ is equally likely, after summing $C_1(M_l(A_{vi}), A_{uj}, v, u)$ over j, the result is a function of $M_l(A_{vi})$, v and u. Since $C_1(M_l(A_{vi}), A_{uj}, v, u)$ is either zero or one, the resulting unconditional probability

$$C_1(M_l(A_{vi}), v, u) = \sum_{j=1}^{\mathcal{N}_u} C_1(M_l(A_{vi}), A_{uj}, v, u).\frac{1}{\mathcal{N}_u}$$

is simply the number of favorable (to recovery) arrangements of u free working links divided by the total number of arrangements of the u free working links for a fixed l and A_{vi}. The exact expression for $C_1(M_l(A_{vi}), v, u)$ is derived in the following result.

Result: *If a single-hop link l fails when the network is in a set of states $\mathcal{S}_i = \{s : A_{vi}, v, u, r$ are fixed, $M_l(A_{vi}) > 0\}$, then, the probability that a via is available as an alternate path for l is given by*

$$C_1(M_l(A_{vi}), v, u) = 1 - \sum_{t=0}^{\min\{u, M_l(A_{vi})\}} U_t P_t$$

where

$$P_t = \frac{\binom{2M_l(A_{vi})}{t}\binom{N - L - 2v - 2M_l(A_{vi})}{u-t}}{\binom{N - L - 2v}{u}}$$

$$U_t = \frac{2^t \binom{M_l(A_{vi})}{t}}{\binom{2M_l(A_{vi})}{t}}$$

Proof: u is the number of working free links in state $s \in \mathcal{S}_i$ and $M_l(A_{vi})$ is the number of 2-link paths available to the single-hop link l after taking into account the initial traffic and interference due to the currently existing vias, but assuming that all free links are working. We will refer to the set of $2M_l(A_{vi})$ free links adjacent to single-hop link l as its $M - vicinity$. Given the value of $M_l(A_{vi})$, the expression for $C_1(M_l(A_{vi}), v, u)$ is obtained by conditioning on t, the number of working free links in the $M-$vicinity; the remaining $M_l(A_{vi}) - t$ free links in the $M-$vicinity are failed free links.

In the derivation, U_t is the conditional probability that no via is formed although t working free links are in the $M-$vicinity of the single-hop link ($u - t$ are scattered over the rest of the network). P_t is the probability that t working free links are in the $M-$vicinity. The expressions for P_t and U_t are derived in Lemma 1 and Lemma 2 respectively.

Lemma 1: We determine P_t by using a simple counting argument. There are $\binom{N-L-2v}{u}$ possible arrangements of u working free links in the network. If t of

these are given to be in the $M-$vicinity, then, there are $\binom{2M_l(A_{vi})}{t}\binom{N-L-2v-2M_l(A_{vi})}{u-t}$ arrangements of u links such that exactly t are in the $M-$vicinity. This accounts for the expression for P_t in the statement. ■

Lemma 2: U_t is equal to the number of arrangements of t links in the $M-$vicinity that result in *no alternate route*, divided by the total number of arrangements of t links in the $M-$vicinity. First, partition the set of $(u+f)$ free links in the network into two sets of size $2M_l(A_{vi})$ and $(N-L-2v-2M_l(A_{vi}))$. The number of ways that t working free links can be distributed over $M_l(A_{vi})$ 2-link paths with no more than one link per path is $\binom{M_l(A_{vi})}{t}$. This is exactly the number of ways that t identical objects can be assigned to $M_l(A_{vi})$ locations such that there is no more than one object per location. (Working free links correspond to objects and locations to 2-link paths). Within any particular 2-link path there are two ways that one working link can be assigned to it. Thus, the number of non-via-forming arrangements of t working free links in the vicinity is $2^t\binom{M_l(A_{vi})}{t}$. P_t and U_t evaluate to zero if $y > x$ in any term of the $\binom{x}{y}$ form. ■

The expression

$$C_1(M_l(A_{vi}), v, u) = 1 - \sum_{t=0}^{\min\{u, M_l(A_{vi})\}} U_t P_t$$

subtracts from 1 the fraction of arrangements of u working free links that are unfavorable to via formation. To see this, substitute for P_t and U_t and notice that $2^t\binom{M_l(A_{vi})}{t}\binom{N-L-2v-2M_l(A_{vi})}{u-t}$ is the number of arrangements that do not form a via for failed single-hop l when exactly t working free links are in its $M-$vicinity. The summation over all possible values of t gives the total number of unfavorable

arrangements of free links with exactly u of them working. This is divided by $\binom{N-L-2v}{u}$, the total number of arrangements of u free working links. This completes our derivation of Equation 4.2. ■

The expression in curly brackets in Equation 4.2 is $C_1(M_l(A_{vi}), v, u)$ $(= C_1(\mathcal{S}_i, l))$, the network's recovery probability for a currently working single-hop link l given that the network is in $\mathcal{S}_i$ at the time of failure. This expression is summed over the d single-hop links to give $C_1(\mathcal{S}_i)$ and over the $\mathcal{N}_v$ via arrangements to give $C_1(\mathcal{S})$. We emphasize at this point that the coverage parameters are sensitive to a function M of the via arrangement as seen by the failing single-hop link, not on the arrangement itself. Thus, the via arrangements need not be stored for coverage computations. A $d-$vector of the $M-$values will suffice.

Equation 4.2 assumes that $M_l(A_{vi})$ is available for each via arrangement A_{vi}. Also assumed is knowledge of $\mathcal{N}_v$, the total number of via arrangements with exactly v vias. Since we do not plan to generate the entire original state space, these quantities are not available. So, we propose to estimate these quantities through simulation and in turn, use estimates of $C_1(\mathcal{S})$ in the lumped Markov model.

4.4.3 Estimation of Coverages on Single-Hop Link Failure

The reason that the coverage parameter $C_1(\mathcal{S})$ cannot be expressed in terms of just the state information, $\mathcal{S}$, is because coverage depends on $M_l(A_{vi})$, a function of the via *arrangement* occurring in $\mathcal{S}$ and on the set of working single-hop links

identified by the via arrangement. Thus, an exact evaluation of $C_1(\mathcal{S})$ requires the generation of all via arrangements occurring in states assigned to $\mathcal{S}$.

Suppose, we could algorithmically generate all possible arrangements A_{vi} of v vias in lump $\mathcal{S}$ and determine $M_l(A_{vi})$ for each arrangement. Also, suppose that $\mathcal{N}_v$ is found by keeping a count of the arrangements. Then, $C_1(\mathcal{S})$ can be calculated exactly. For the network model, $C_1(\mathcal{S})$ is required for each lump $\mathcal{S}$. Thus, we must generate each possible via arrangement in the network. If this number is small, then, the exact determination of the coverage parameters is computationally feasible. Otherwise, for arbitrary traffic patterns, it becomes necessary to resort to estimation through simulation. Before we describe the simulation procedure, we consider some special traffic patterns. For these special cases, the simulation effort is negligible.

Suppose a traffic pattern has the characteristic that the maximum number of potential vias $M_l(A_{vi}) = M$ for all single-hop links and for every possible via arrangement in the network. It is enough to know $M_l(A_{vi})$ for exactly one via arrangement and one single-hop in that arrangement. We do not require to know $\mathcal{N}_v$, the number of possible via arrangements having exactly v vias either since the two summations in Equation 4.2 are unnecessary. Likewise, it is possible that a particular traffic pattern is independent of A_{vi}. In this case, the only information required is a d-vector $M_l, 1 \leq l \leq d$ and there is no need to sum over the via arrangements or know $\mathcal{N}_v$. Similarly, a traffic pattern may be independent of l but not of A_{vi}; for this case $\mathcal{N}_v$ is required.

In the first example, information required to calculate $C_1(\mathcal{S})$ can be obtained by generating exactly one state of the original state space in $\mathcal{S}$. In fact, we need to find M for exactly one single-hop link in this state. In the second example, once again, exactly one state is required to gather the necessary information. However, we need to determine M_l for each of the d single-hop links in that state; then, we compute the weighted average of $C_1(M_l, v, u)$ over the d values of l with weight $\frac{1}{d}$ to obtain the coverage for lump $\mathcal{S}$. Furthermore, by simply varying the value of v, the number of vias and u, the number of working free links, we get the coverage for every state $\mathcal{S}$ in the lumped model. Since the dependence of coverages on u is expressible in analytical form (see Equation 4.2), we have, for all practical purposes, a closed form expression for the coverage parameters of the lumped model for these special cases. For an arbitrary traffic pattern and for an exact value of the coverages, $M_l(A_{vi})$ must be calculated for each via pattern and for each single-hop link in the via pattern.

For an arbitrary traffic pattern, one could think of generating not all, but only some via arrangements corresponding to each value of v, the number of vias in the network. Based on these, conditional coverages $C_1(S_i)$ can approximately be found for $(N - L - 2v + 1) \times (v + 1)$ different lumps as observed at the end of Section 4.4. In the following paragraph, we propose one possible way of estimating the coverages. An estimate of V_{max} which decides the number of states in the lumped model is obtained as a by-product.

Starting from the initial state of the unaggregated model, a sequence of single-hop link failures and recovery are simulated using Monte-Carlo simulation until

there is no via to replace the most recently failed single-hop link. The number of single-hop links that are replaced by vias during the simulation run is $\hat{V}_{max}^{(1)}$, a first estimate of V_{max}. This one simulation run generates one via arrangement for each value of v, $0 \leq v \leq \hat{V}_{max}^{(1)}$. For each via arrangement A_{vx} generated in this simulation run, we record the values of $M_l(A_{vx})$, $1 \leq l \leq d$ and calculate $C_1(\mathcal{S}_x)$ and accumulate in $\hat{A}_1(\mathcal{S})$.

We repeat the simulation described above K times and base our estimate of the coverages on the K or fewer via arrangements generated per lump as follows. Let $K(\mathcal{S})$ be the number of arrangements generated for state $\mathcal{S}$. Then, an estimate of the coverage $\hat{C}_1(\mathcal{S}) = \hat{A}_1(\mathcal{S})/K(\mathcal{S})$. For K simulation runs, the estimate of V_{max} is $max\ \{\hat{V}_{max}^{(k)}, 1 \leq k \leq K\}$. In simulating for the coverages in this way, we have implicitly performed a summation over the most probable via arrangements in the $\mathcal{S}$. Further, since the via arrangements can repeat, the coverages for the most probable via arrangements are accumulated in the ratio of their frequency of occurrence. The w^is are thus estimated implicitly. Note that this is a simulation meant for estimating coverage parameters on failure of single-hop links. Sample paths are restricted to those resulting from sequences of single-hop link failures only. No repair or via link failure is simulated and this makes the simulation effort much less than that for the entire process. We will simulate the entire original process for an example in Section 4.5 to compare with the results of our approximate lumped model.

Figure 4.8 shows the lumped state space and state transitions for a network with 8 nodes and a load of 12. The lumped model is a three dimensional Markov

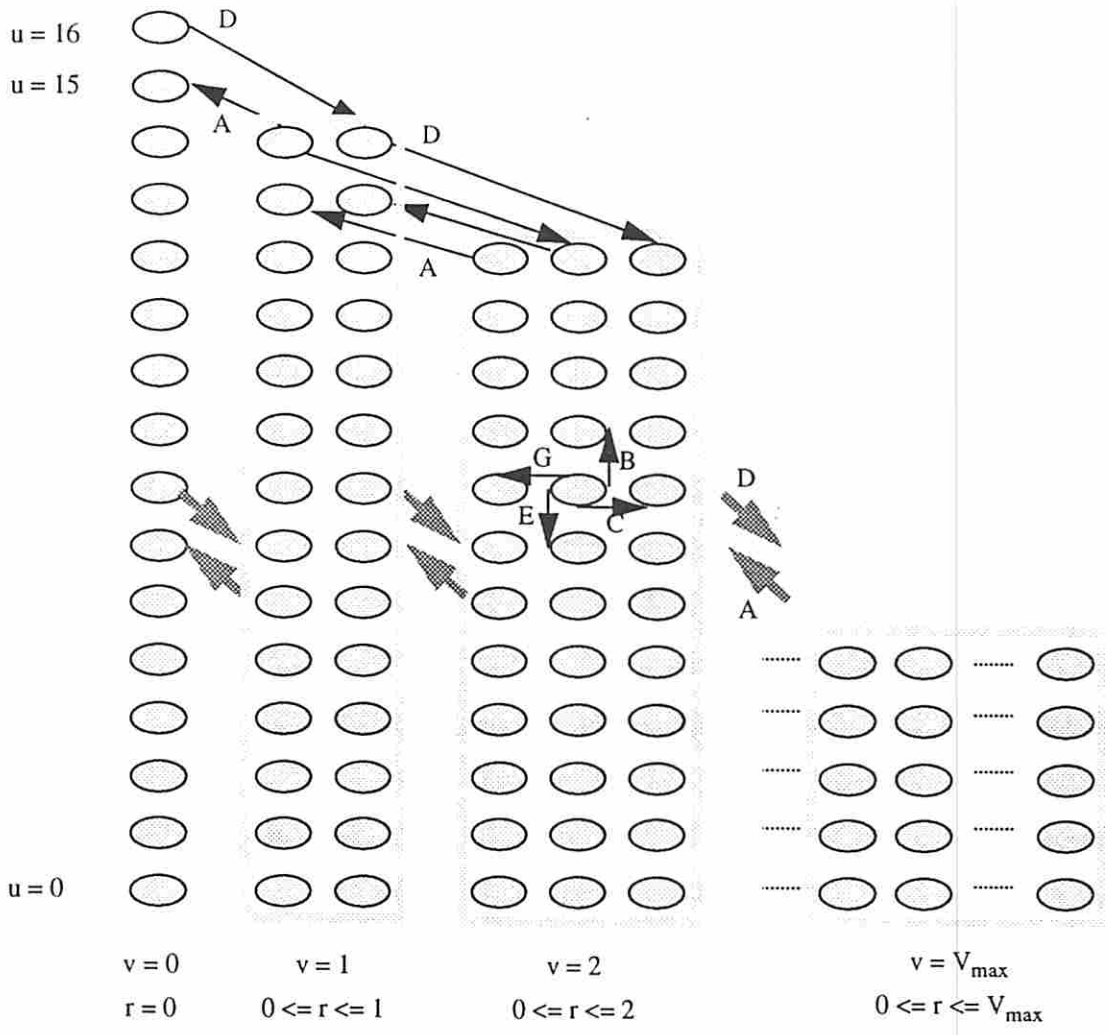


Figure 4.8: Aggregate Model for an 8 Node Network with Load 12.

Chain. The three independent variables used to describe each lumped state are v, r, and u, the number of vias, working reserved links and working free links, respectively. Transitions marked A,B,C,D,E and G correspond to those in Figure 4.6. Transitions to the network's failure state are not shown. In the figure, the first column corresponds to states with zero vias. Proceeding rightward, columns of a given size correspond to a fixed number of vias. The size of the rightmost column is clearly dependent on the maximum number of vias that the network can support. $V_{max} \leq L$ is an unknown, but is estimated through simulation.

4.5 Example Application

One application of the model we have developed is in evaluating alternative reconfiguration strategies. For example, with a simple modification, we can model the effect of reconfiguring on repair. In this alternative approach, when a failed single-hop link is repaired, the via carrying the corresponding connection is immediately torn down and the connection is returned to the original single-hop link. This frees up links. There are no reserved links. Figure 4.9 gives a comparison of the two reconfiguration mechanisms, namely, *Non-Reconfiguring on Repair* (NROR) and *Reconfiguring on Repair* (ROR). In the ROR mechanism, a via link failure always causes the network to fail.

The state transitions for a typical lumped state of the ROR network are shown in Figure 4.10 and the lumped Markov Chain is shown in Figure 4.11. The NROR network's Markov Chain is three dimensional whereas the ROR's chain is two dimensional due to the absence of the *reserved link dimension.* Thus, the state

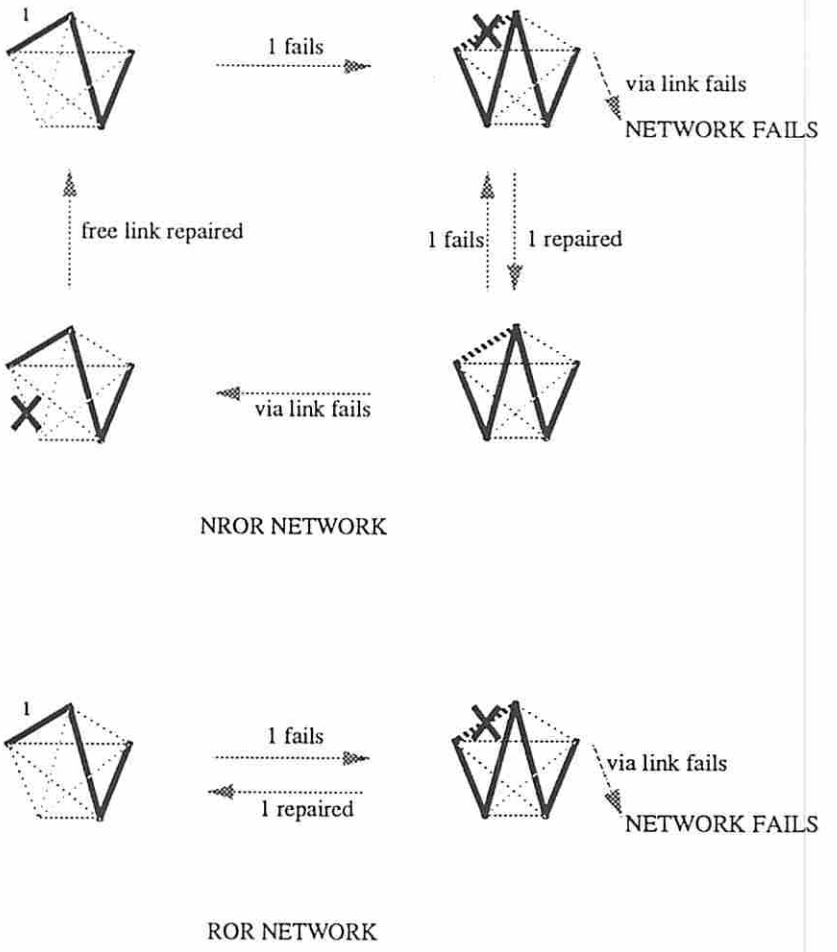
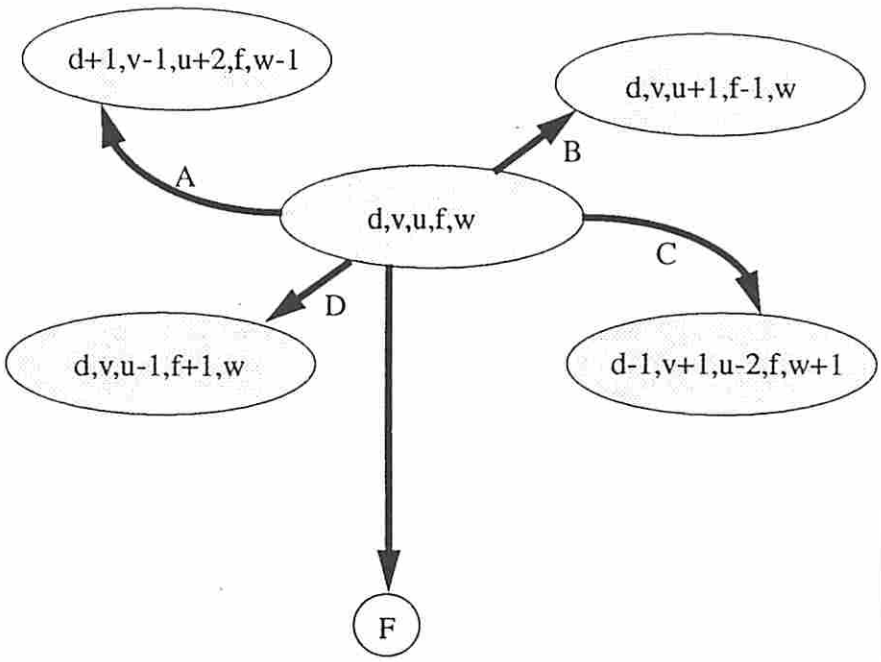


Figure 4.9: Responses to Single-hop Link Repair in NROR and ROR Networks.



w : Number of Failed Single-hop links.
f : Number of Failed free links.
d : Number of Single-hop links.
u : Number of Working free link failure.

A : Single-hop link repair,$w\mu$
B : Free link repair, $f\mu$
C : Single-hop link failure, $d\lambda C_1(\mathcal{S})$
D : Free link failure, $u\lambda$
F : $2v\lambda + d\lambda(1 - C_1(\mathcal{S}))$

Figure 4.10: State Transitions in the ROR Network Lumped Model.

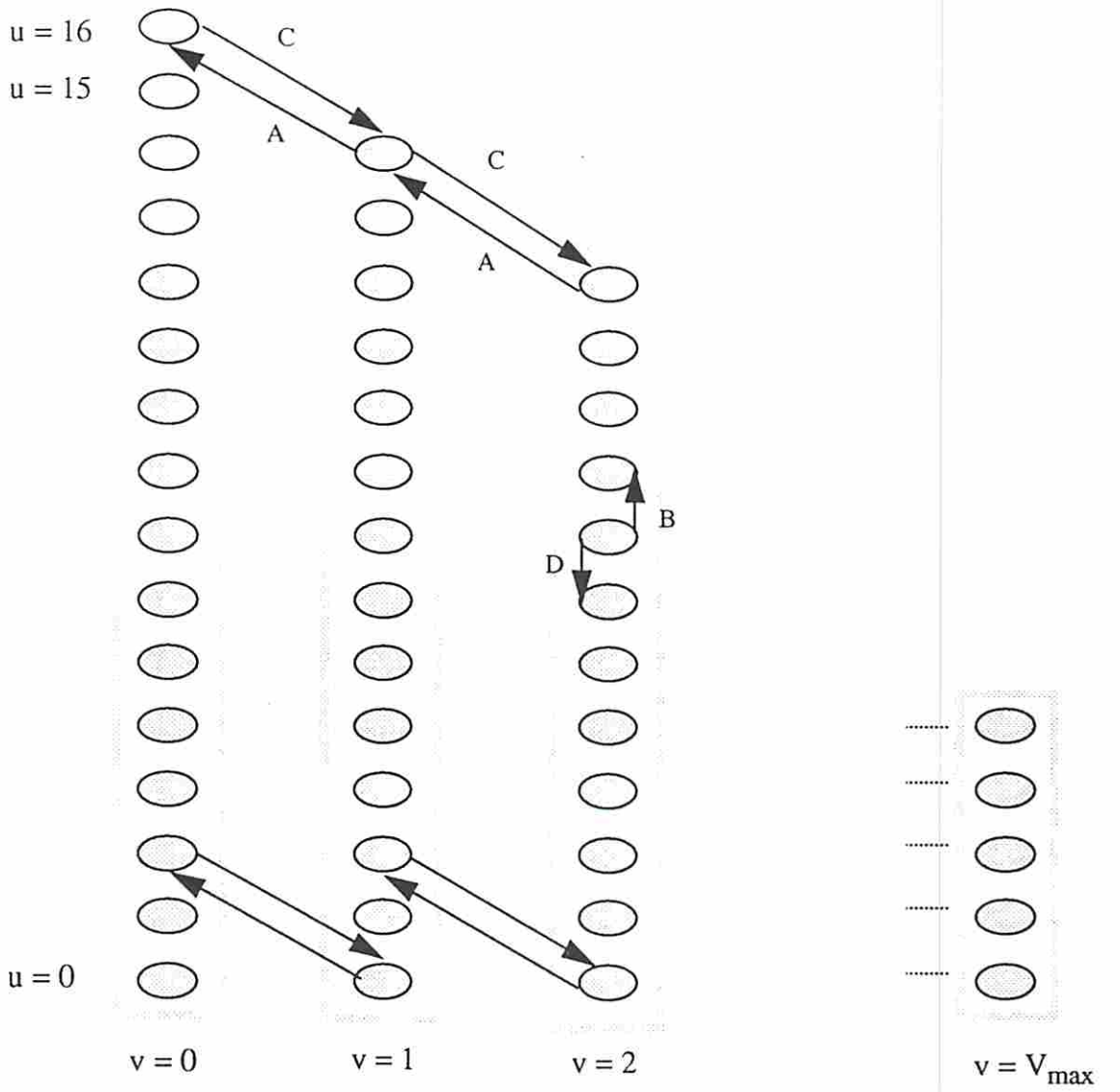


Figure 4.11: Lumped Model for an 8 Node ROR Network with Load 12.

space of the ROR network is a subset of the state space of the corresponding NROR network. *For a given lump*, the coverage parameters on single-hop failure are exactly the same as for the NROR network state with the reserved link dimension ignored. The number of states in the lumped model of an ROR network is $O((N-L)V_{max})$ as compared to $O((N-L)V_{max}^2)$ of the corresponding NROR network.

For a numerical example, we consider an eight node private network with a randomly generated load of twelve (Figure 4.12). The link failure and repair rates are $\lambda = .0005$ failures per hour and $\mu = .4995$ repairs per hour, respectively. We construct two lumped models corresponding to the two reconfiguration mechanisms, NROR and ROR after estimating coverages through simulation as in Section 4.4.3. Since the two networks differ only in their responses to repair, these coverage estimates (on single-hop link *failure*) are used for both models. The estimates are based on 1000 sample paths. The coverage values for the lumped states are tabulated in Table 4.1. For this example, $\hat{V}_{max}$ was 7. The computer run time for estimating coverages and V_{max} was less than a minute and a half on a SUN 3/50 workstation. The following observations are made from Table 4.1. The highest value for the coverage parameter occurs when there are no vias in the network. As the number of vias increases, the highest coverage (over all values of u) decreases. For example, $C_1(0,16) < C_1(1,14) < C_1(2,12)$. For a fixed number of vias, the coverage decreases with the number of free working links available in the network. If there are no free working links or if there is exactly one, there can be no alternate paths. Accordingly, coverages are zero for these values of u. The lumped Markov Chain for the NROR and ROR networks are now completely specified.

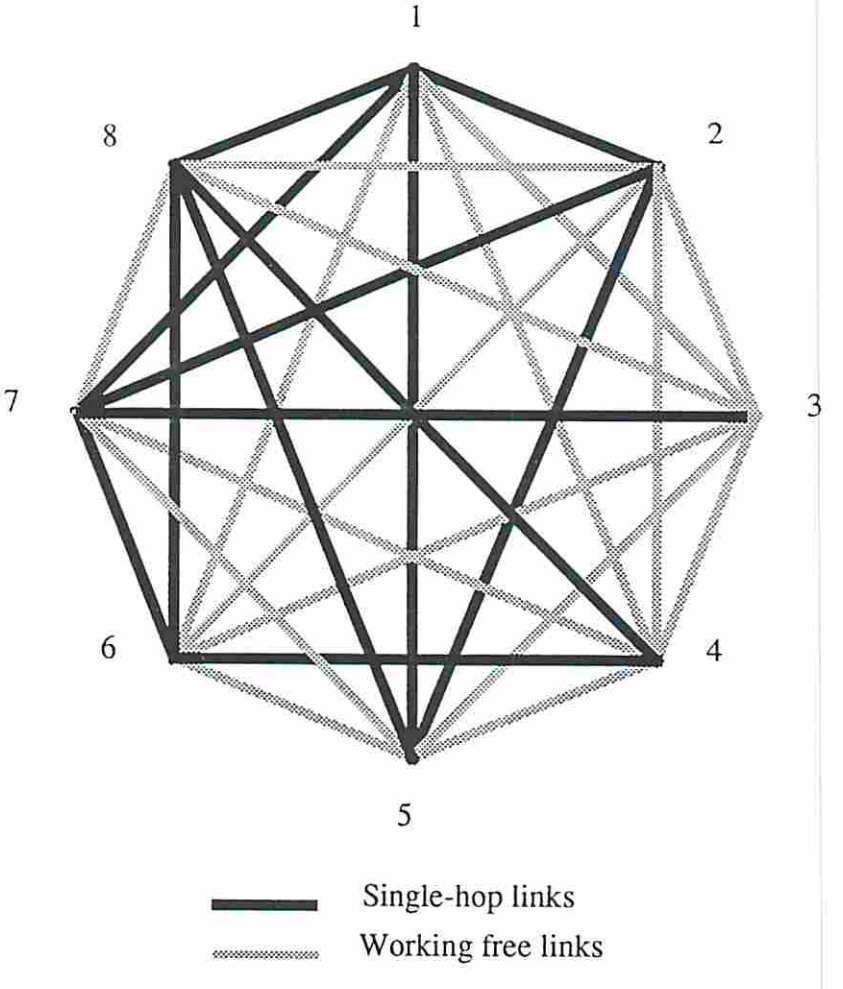


Figure 4.12: Traffic-Pattern used for the Example

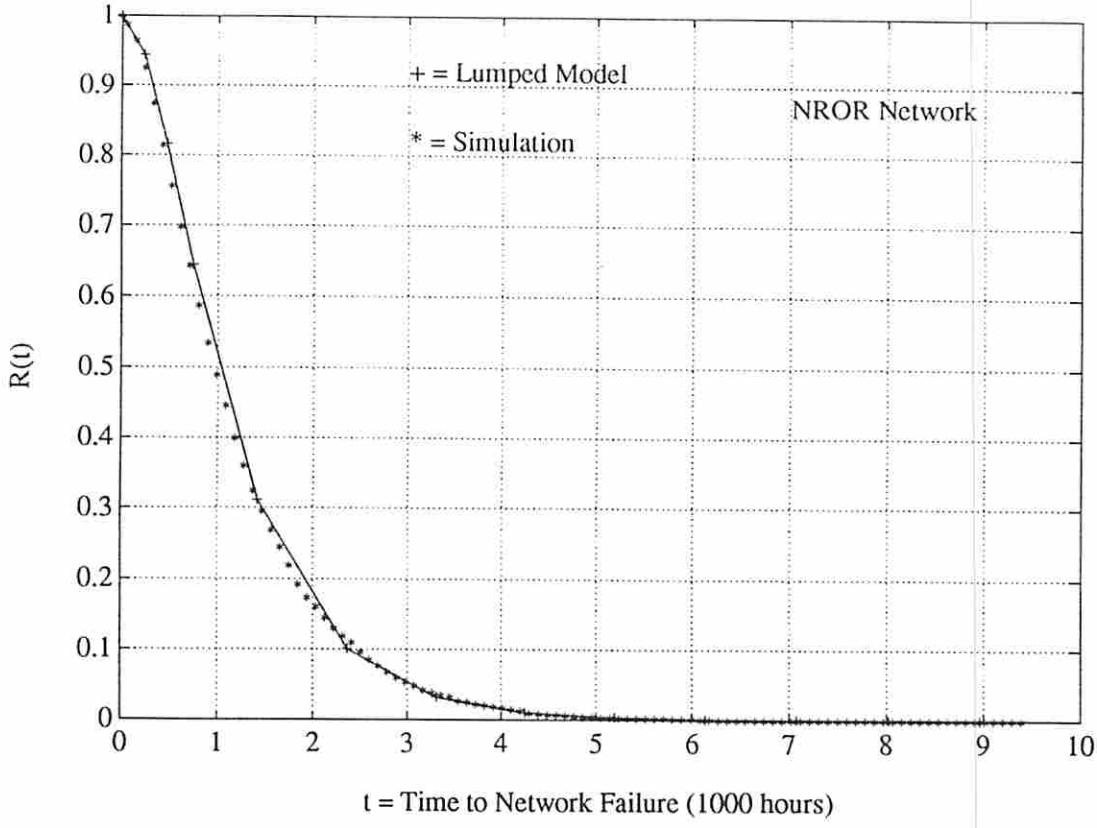


Figure 4.13: Reliability plot for the NROR Network.

We solve these approximate Markov Chain models using SHARPE [ST87]. The resulting network reliability functions are shown by + in Figure 4.13 and Figure 4.14 respectively.

To see the error introduced in the approximation, we solve the original Markov model by simulation. The number of simulation runs is 2500. The resulting reliability functions are shown by $*$ for NROR and ROR networks in Figure 4.13 and Figure 4.14, respectively. While the NROR network took only about 12 minutes to simulate, the ROR network took over 10 hours for these 2500 runs on a SUN 3/50

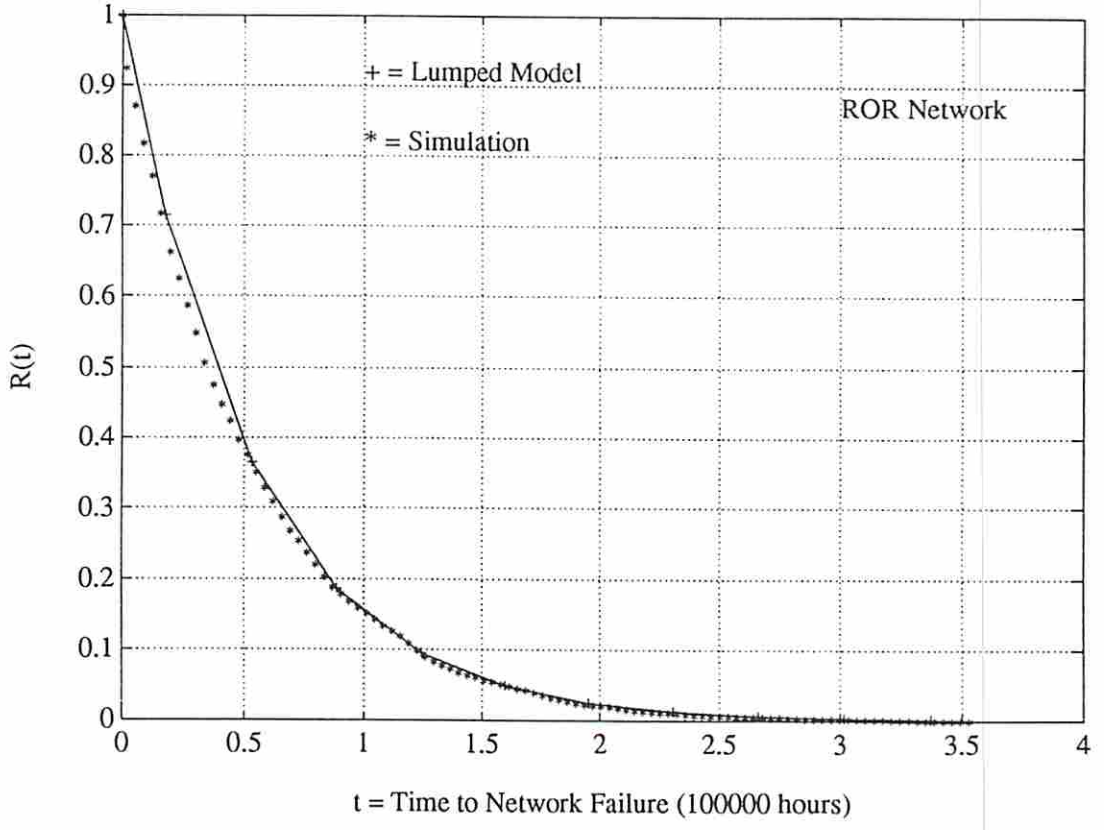


Figure 4.14: Reliability plot for the ROR Network.

workstation with 4MB of memory – we should expect the runtime to increase with the network's reliability. An analysis of the simulation data showed the mean-time-to-failure (MTTF) of the ROR network to be two orders of magnitude higher than that of the NROR network. The 95% confidence interval for the MTTF was .029% of the estimated MTTF for the NROR network and .038% of the estimated MTTF for the ROR network. These were obtained as follows. For the NROR network, 2500 values of the network's time-to-failure were recorded. From this data, the estimated MTTF was 1259 hours and estimated variance of the time-to-failure was 890756. Let $\hat{a}$ denote the standard estimator of the MTTF. Then, an estimate of the variance of $\hat{a}$ for 2500 runs is $\hat{\sigma}^2 = 890756/2500$. The t-distribution with 2499 degrees of freedom was used to place a confidence interval on the MTTF [Wel83] as follows. For a 95% confidence interval, $Prob\{|\hat{a} - MTTF| \leq \epsilon\} \approx .95$, where $\epsilon = t_{2499}(1-0.95/2)\hat{\sigma}$. $t_{2499}(1-0.95/2)$ is seen from the tables to be approximately 1.96. Using the estimated 1259 hours for the MTTF, $\epsilon = 37.0$; this is .029% of the estimated MTTF. For the ROR network 53327 and 1.0937×10^6 were the estimates of the MTTF and of the variance of $\hat{a}$; the corresponding ϵ was 2049.5.

The results (note the time scales on Figures 4.13 and 4.14) indicate that there is much to be gained in terms of reliability if the network is forced to reconfigure as soon as a failed single-hop link is repaired. From the plots in Figures 4.13 and 4.14, it can be seen (at least for this 8 node network example) that the reliability of the ROR network is well above that of the NROR network for both the lumped and original models. In comparing the simulation results with those of the lumped Markov model, the plots of the reliability functions are seen to follow each other

very closely in each case. This is due to the coarseness of the scale of the plots - the differences in reliability of any model and its approximation is likely to be in the third or fourth place of the decimal. Recall, that the main objective of the analysis is to compare the NROR and ROR reconfiguration mechanisms. Since there is a clear indication of the superiority of the ROR mechanism, we believe this degree of coarseness to be adequate for comparing the reliabilities of the two networks.

For assessing the quality of the approximation of the lumped model, we expand in the $0.93 - 1.0$ region of reliability. The magnified plots are shown in Figure 4.15 and 4.16, respectively, along with 95% confidence intervals of some of the percentiles of the failure-time distribution, $1 - R(t)$ [Wel83].

For this experiment, we repeated the simulation 7500 times. As expected, discrepancies due to approximating a non-Markovian process by a Markovian process, estimating of coverages for the approximate model and simulation errors in solving the lumped model show up on this expanded scale.

Following is a summary of our other experiments. We experimented with eleven other randomly generated traffic patterns for the eight node network each with a load of twelve. In each case, the approximation tracked the simulation results (for both NROR and ROR networks) as closely as for the example presented here. These examples are an indication that the hybrid model (the lumped Markov model with parameter estimation by simulation) is an approximating method worthy of further study. Further experimentation/analysis is required to characterize the nature of the approximation and the range of its applicability.

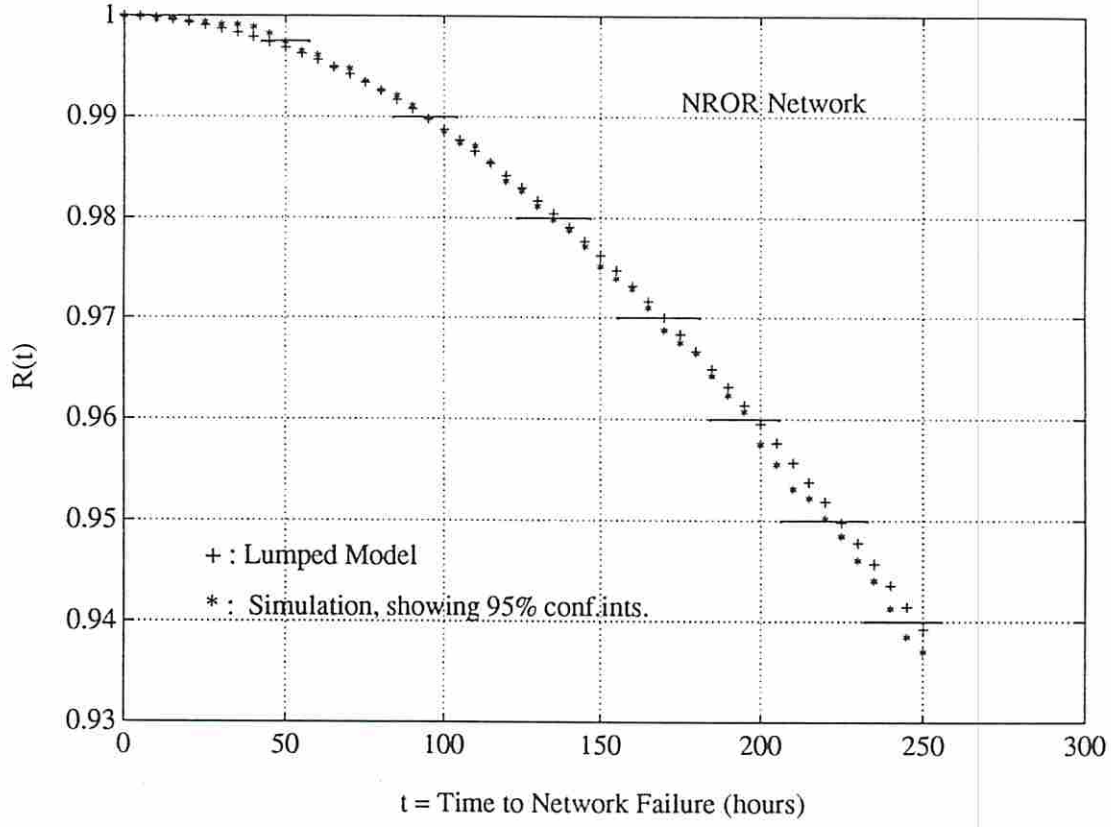


Figure 4.15: Magnified plot for the NROR Network.

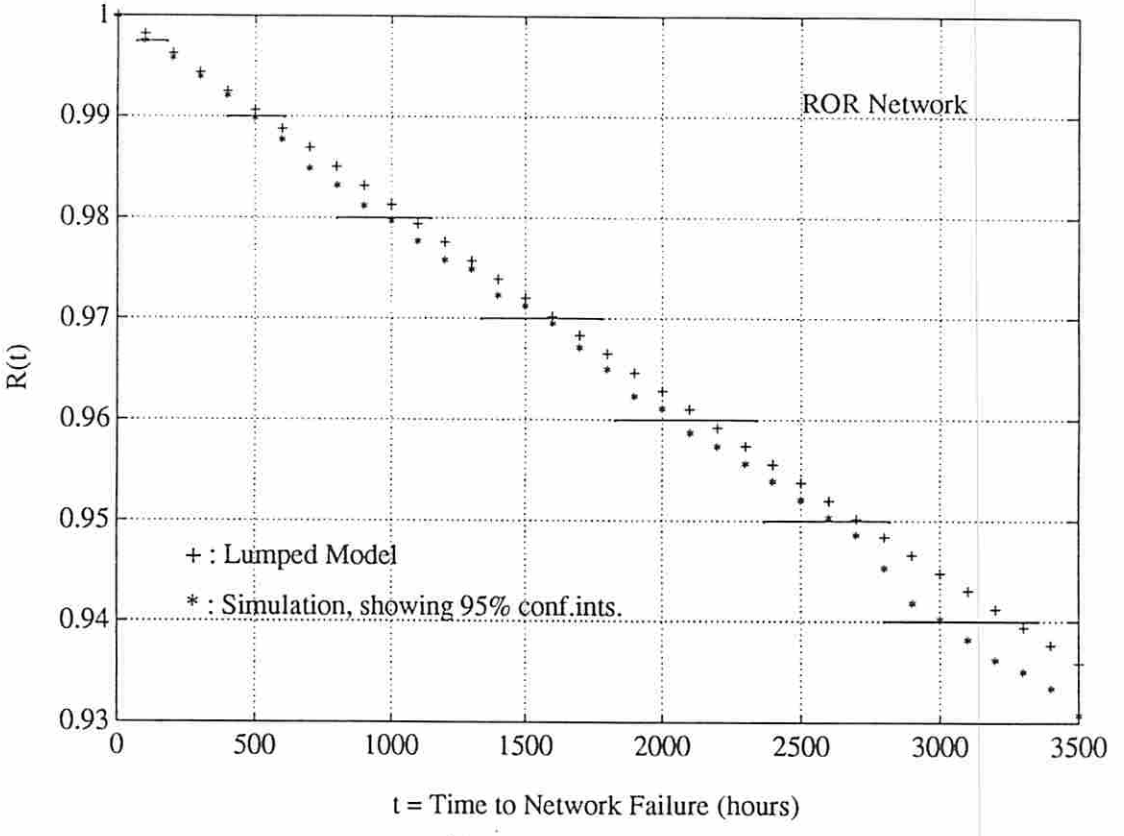


Figure 4.16: Magnified plot for the ROR Network.

Experiments using the models suggest that the reliability function of the private network is highly sensitive to changes in the traffic pattern. The ROR network, however, was seen to be more sensitive than the NROR network. While the MTTF for the NROR network remained within the same order of magnitude for all the twelve cases, the MTTF of the ROR network spanned three orders of magnitude. However, in each case, the ROR network was seen to be more reliable than the NROR network. The networks, thus, exhibit the following typical fault-tolerant system behavior: higher sensitivity to coverages in systems having higher reliability.

Finally, for the example eight node network in Figure 4.12, the number of states in the original model for NROR and ROR networks was algorithmically found to be 60726×2^9 and 60726×2^2 respectively. The maximum number of vias that the example network could support (exact value of V_{max}) was seven. In this case, the estimate of V_{max} happens to be equal to the true value and is one less than the bound which is eight – there are only sixteen free links to begin with. The number of feasible arrangements of vias for $0 \leq v \leq 7$ is given in Table 4.2. To give an idea of the number of possibilities that need to be examined in order to select the feasible arrangements, we have included the number of infeasible arrangements in Table 4.2. The lumped models for this example have 276 states for the NROR and 80 states for the ROR network. A typical state in the lumped model would have 28 transitions (one per link). In the lumped model, this is 7 and 5 for the NROR and ROR respectively.

$C_1(\mathcal{S})$	v = 0	v = 1	v = 2	v = 3	v = 4	v = 5	v = 6
u = 16	1.000	*	*	*	*	*	*
u = 15	0.969	*	*	*	*	*	*
u = 14	0.931	0.939	*	*	*	*	*
u = 13	0.883	0.892	*	*	*	*	*
u = 12	0.823	0.832	0.842	*	*	*	*
u = 11	0.749	0.759	0.770	*	*	*	*
u = 10	0.664	0.674	0.685	0.692	*	*	*
u = 9	0.569	0.580	0.591	0.599	*	*	*
u = 8	0.470	0.480	0.491	0.499	0.497	*	*
u = 7	0.371	0.380	0.389	0.396	0.395	*	*
u = 6	0.275	0.283	0.290	0.296	0.296	0.308	*
u = 5	0.189	0.194	0.200	0.204	0.204	0.212	*
u = 4	0.116	0.119	0.123	0.125	0.125	0.130	0.124
u = 3	0.058	0.060	0.062	0.063	0.063	0.065	0.063
u = 2	0.019	0.020	0.021	0.021	0.021	0.022	0.021
u = 1	0.000	0.000	0.000	0.000	0.000	0.000	0.000
u = 0	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 4.1: Coverage Estimates $C_1(\mathcal{S})$ for the Example.

Via Count	Number Feasible	Number infeasible
0	1	0
1	28	0
2	278	76
3	1221	1449
4	2474	10894
5	2229	44529
6	791	116235
7	84	210830

Table 4.2: Number of Feasible and Infeasible Arrangements when given v vias

4.6 Conclusions

In this chapter, we developed a state space reliability model for fault-tolerant private networks with repair. Our model is also applicable to other networks with dedicated circuits, like simple digital facility networks. Using the model we developed we considered two alternative network reconfiguration mechanisms. The first approach takes no action when a link is repaired. The second restores re-routed traffic to its original route, as soon as the route is repaired. Within the limits of the modeling assumptions, we have shown that the reliability of the network is significantly improved by restoring the traffic immediately upon completion of repair.

We have focused attention on the reduction of the state space of the reliability model. Our approach is to construct a lumped Markov model to replace the original model. After reducing the state space, we obtain transition rates for the lumped model in parametric form. The parameters are estimated by an efficient Monte Carlo simulation. We thought it essential to estimate by simulation because the parameters are *coverage parameters* of the lumped model and are known to have a major effect on the reliability function [Arn73] [BCS69]. Experimental results suggest that model is a good approximation that can replace an expensive solution by simulation.

The focus of our analysis has been the construction and evaluation of the approximate models. Consequently, we have made some assumptions which are unrealistic of a real network. In our future work, we plan to relax these assumptions as far as possible.

Chapter 5

An Analysis of a Lumping Heuristic for a Markov Network Reliability Model

5.1 Introduction

The purpose of this chapter is to analyze the lumping heuristic we used in the previous chapter for reducing the state space of the Non-Reconfiguring-on-Repair (NROR) and Reconfiguring-on-Repair (ROR) network reliability models and explain the cause for the good approximations: some typical examples of this approximation are presented in Figures 4.15 and 4.16. In the heuristic described in the previous chapter, we first defined a partition on the state space of a Markov reliability model and replaced each subset in the partition by a single state. The set of subsets of the partition or *lumped* states served as the state space for a smaller Markov Chain, SM. We then obtained the transition rates for SM using a heuristic based on the *fixed-weight aggregation* method [HS84] and solved the smaller Markov Chain, say SM, to obtain reliability of the NROR and ROR networks.

To analyze the heuristic, we begin Section 5.2 by deriving the transition probabilities of the stochastic process (NM) resulting from partitioning the state space of a time-homogeneous Markov Chain (M). This lumped stochastic process, NM, is in general non-time-homogeneous and non-Markovian, as we shall see, because the transition probabilities of the lumped process are functions of the path traversed form the starting state and they vary with time. Thus, the heuristically constructed process SM is a time-homogeneous Markov approximation to the process NM. Based on the expression for transition rates of NM, we give an interpretation of the fixed-weight aggregation method used in the heuristic (given by Equation 4.1) in Section 5.2.1. Then, using this interpretation, we present a procedure for obtaining the rates of an approximate Markov Chain in Section 5.2.2. This procedure has the characteristic that if the original Markov Chain (M) is *lumpable* [KS76], that is, NM is time-homogeneous and Markovian under the given partition, then, the procedure constructs a Markov Chain which is identical to NM We conjecture that the Markov Chain constructed by this procedure deviates from the "best" Markovian approximation to NM as the original Markov Chain, M deviates from being lumpable for the given partition. Next, we interpret the heuristic used in the previous chapter in terms of the procedure detailed in Section 5.2.2 as follows. In Section 5.3.1, we show that this heuristic constructs the approximate Markov Chain SM by systematically lumping the original state space and deriving lumped transition rates after each lumping step. Then, in Section 5.3.2 we show that this

heuristic constructs lumped network reliability models in accordance with the procedure prescribed in Section 5.2.2 provided the time-scales of operation between the repair and failure processes are taken into account.

5.2 Mathematical Preliminaries

In this section, we first derive an expression for the state transition rates of a stochastic process (NM) constructed by specifying a partition on the state space of a Markov Chain M $= \{X_t; t \geq 0\}$ for a fixed starting state s. Here, the state space of M corresponds to the state space of the original Markov Chain while the state space of the lumped process is the set of lumped states under the given partition. If I and J are two lumped states and there are transitions between original states in I to states in J, then, $q_{IJ}\triangle t$ is the probability that a transition occurs from subset I to subset J in an interval $[t, t + \triangle t]$. This probability can also be written as $P(X_{t+\triangle t} \in I \mid X_t \in J, X_u \in A_{f(u)}; 0 < u < t, X_0 = s)$, assuming that s is the given starting state and $\{s\}$ is a subset under the given partition. $A_{f(u)}; 0 < u < t$ denotes the path followed by lumped process before reaching state I, while $f : u \rightarrow$ *set of lumped states.* Clearly, if the lumped process is Markovian, then, $q_{IJ}\triangle t$ will be a function only of the lumped states I and J. In the result in the next paragraph, we obtain the transition probabilities for the lumped process in terms of the original process. This helps in understanding the effect of lumping a Markov Chain. This result, along with results from the theory of lumpability of Markov Chains [KS76], gives a useful interpretation of the fixed-weight aggregation formula reproduced in Equation 5.4 (Section 5.2.1). In Section 5.2.2, we show that

this interpretation translates, under certain conditions, to finding the expected number of visits to the states of M.

Result: Let $\{X_t; t \geq 0\}$ be a time-homogeneous continuous-time Markov Chain with a finite state space and let $A = \{A_1, \ldots, A_r\}$ be a partition specified on the state space. Let s be the fixed starting state for the process and let $A_i = \{s\}$ for some i. Then,

$$P_s(X_{t+\Delta t} \in A_n | X_t \in A_m, X_u \in A_{f(u)}; 0 < u < t)$$
$$= \sum_{i=1}^{|A_m|} \left\{ \sum_{j=1}^{|A_n|} q(m_i, n_j)\Delta t \right\} \left\{ \frac{P_s(X_t = m_i | X_u \in A_{f(u)}; 0 < u < t)}{\sum_{i=1}^{|A_m|} P_s(X_t = m_i | X_u \in A_{f(u)}; 0 < u < t)} \right\}$$

where, $f : u \to \{1, \ldots, r\}$, $A_n, A_m, A_{f(u)} \in A$ and $m_i \in A_m, 1 \leq i \leq |A_m|$,$n_j \in A_n, 1 \leq j \leq |A_n|$ are states of the Markov Chain belonging to lumped states A_m and A_n respectively.

Proof: Define the following events.

- $B = \{X_u = A_{f(u)}; 0 < u < t\}$, denotes the past.

- $C_j = \{X_{t+\Delta t} = n_j\}, 1 \leq j \leq |A_n|$; $C = \{X_{t+\Delta t} \in A_n\} = \bigcup_j C_j$, denotes the future.

- $D_i = \{X_t = m_i\}, 1 \leq i \leq |A_m|$; $D = \{X_t \in A_m\} = \bigcup_i D_i$, denotes the present.

With this notation,

$$\begin{aligned} P_s(X_{t+\Delta t} \in A_n | X_t \in A_m, X_u \in A_{f(u)}; 0 < u < t)n \\ = P_s(C | \bigcup_i D_i, B) \\ = P_s(C | \bigcup_i (D_i \cap B)) \end{aligned}$$

Let $E_i = D_i \cap B$ so that E_i are a set of mutually disjoint events.

That is, $E_i = \{X_t = m_i, X_u \in A_{f(u)}; 0 < u < t\}$, $m_i \in A_m$. It follows that

$$\begin{aligned} P_s(X_{t+\Delta t} \in A_n | X_t \in A_m, X_u \in A_{f(u)}; 0 < u < t) = P_s(C | \bigcup_i E_i) \\ = \frac{\sum_i P_s(C \cap E_i)}{\sum_i P_s(E_i)} \\ = \frac{\sum_i P_s(C|E_i) P_s(E_i)}{\sum_i P_s(E_i)} \end{aligned} \tag{5.1}$$

Next, consider the probability $P_s(C|E_i)$ occurring in Equation 5.1.

$$P_s(C|E_i) = P_s(\cup_j C_j | E_i) = \frac{\sum_j P_s(C_j|E_i) P_s(E_i)}{P_s(E_i)} = \sum_j P_s(C_j|E_i)$$

Substituting for C_j and E_i and remembering that $\{X_t; t \geq 0\}$ is a time-homogeneous Markov Chain,

$$P_s(C|E_i) = \sum_{j \in A_n} P_s(X_{t+\Delta t} = n_j | X_t = m_i, B) = \sum_{j \in A_n} q(m_i, n_j) \Delta t \tag{5.2}$$

Let $q(m_i, A_n)$ denote the sum of the transition rates from state $m_i \in A_m$ to lumped state A_n in Equation 5.2. Then, Equation 5.1 can be written as follows:

$$
\begin{aligned}
& P_s(X_{t+\Delta t} \in A_n | X_t \in A_m, B) \\
& \qquad = \frac{\sum_i q(m_i, A_n) \Delta t . P_s(X_t = m_i, B)}{\sum_i P_s(X_t = m_i, B)} \\
& \qquad = \sum_i \frac{q(m_i, A_n) \Delta t . P_s(X_t = m_i | B) P_s(B)}{\sum_i P_s(X_t = m_i | B) P_s(B)} \\
& \qquad = \sum_i \underbrace{q(m_i, A_n) \Delta t}_{f_1(i)} \underbrace{\left\{ \frac{P_s(X_t = m_i | B)}{\sum_i P_s(X_t = m_i | B)} \right\}}_{f_2(i,B,s,t)}
\end{aligned}
\tag{5.3}
$$

This completes the proof. ■

To summarize the result in Equation 5.3, the transition rates to lumped state A_n are accumulated for each (original) state m_i in the source A_m and multiplied by Δt. The lumped transition rate is obtained as a convex combination of these accumulated sums. In the next section, we analyze Equation 5.3 and relate it to the fixed-weight formula.

5.2.1 An Interpretation of the Fixed-Weight Aggregation Method

In this section, we first present conditions under which Equation 5.3 describes a Markov Chain (that is, NM is itself Markovian) and discuss these conditions in terms of the lumpability theorems given in [KS76]. As we shall see later in this section, this enables us to use the fixed-weight formula for developing a procedure

for constructing a lumped Markov Chain. This Markov Chain is an approximation to NM if NM is non-Markovian and the same as NM if the original Markov Chain M happens to be lumpable for the given partition and starting state.

The lumped process defined by Equation 5.3 is Markovian iff $P_s(X_{t+\Delta t} \in A_n | X_t \in A_m, X_u \in A_{f(u)}; 0 < u < t)$ is independent of $\{X_u \in A_{f(u)}; 0 < u < t\}$ and it is time-homogeneous if $P_s(X_{t+\Delta t} \in A_n | X_t \in A_m)$ is invariant with t for every pair of subsets A_m and A_n. Since the LHS inherits the properties of the RHS, we focus on the RHS of Equation 5.3. Notice that the LHS is dependent on $B = \{X_u \in A_{f(u)}; 0 < u < t\}$ only through the f_2 term. Also, that the f_2 term is less than or equal to one for each state in A_m, and summed over all $m_i, 1 \leq i \leq |A_m|$, is exactly equal to one. There are two ways that the dependence on $\{X_u \in A_{f(u)}; 0 < u < t\}$ can be removed.

1. Make the RHS independent of f_2.

2. Make f_2 independent of B for each i. Note that this does not imply that $P_s(X_t = m_i|B)$ must be independent of B, for f_2 can be independent of B even when $P_s(X_t = m_i|B)$ is not.

An obvious way of making the LHS independent of f_2 is to make $q(m_i, A_n)$ independent of i. The LHS is then equal to the common transition probability $q(m_i, A_n)\Delta t$ of going from any state $m_i \in A_m$ to A_n. This equality of accumulated rates is exactly the condition for the (original) Markov Chain to be *strongly lumpable* with respect to a given partition in Theorem 6.3.2 by Kemeny and Snell [KS76] where they have shown this to be a necessary and sufficient condition for strong lumpability. Thus, strong lumpability requires that the lumped transition

rates be independent of the values assigned to the f_2 term for every pair of lumped states for which a transition is defined, for every possible starting probability vector.

If the LHS is Markovian on account of f_2 being independent of B for at least one starting vector, the original Markov Chain is said to be *weakly lumpable* [KS76] with respect to the given partition and the specified starting vector (or starting state). Here, we assume that the process always starts from the state s because it is so in our application. In general, it is difficult to establish whether or not a Markov Chain is weakly lumpable for a given partition and starting state. Some tests for sufficiency are available [KS76] [BT77] [TB77] but none that are both necessary and sufficient. For some special cases, however, it is easy to establish weak lumpability. For example, when the original Markov Chain is ergodic and the starting vector is equal to the steady-state vector, the Markov Chain is lumpable for any partition [KS76]. The states of interest in reliability analysis are all transient, and so this case is not relevant to the present application. In general, weak lumpability is difficult to establish for Markov Chains with large state spaces. Most computation-oriented tests would require the specifying of transition rates of a large Markov Chain, a task we wish to avoid altogether. Our goal is as follows: To prescribe a procedure for constructing a time-homogeneous Markovian approximation to a lumped process (NM) obtained through a specific partition of the original state space and a given starting state, assuming that this state is a singleton in the lumped state space. If weakly lumpable, the procedure should yield the lumped Markov Chain defined by Equation 5.3; otherwise, the procedure should construct

a meaningful approximation. With this discussion, we give an interpretation of the fixed-weight aggregation formula [HS84] given below.

Let $\{1, 2, \ldots, V, F\}$ be subsets of a partition on the original state space. Let I, J be two distinct lumped states and q_{IJ} be the lumped transition rate from I to J. Then, the fixed-weight formula is given by:

$$q_{IJ} = \sum_{i \in I} \left(\sum_{j \in J} r_{ij} \right) w^i \tag{5.4}$$

where, w^i are *weights* assigned to each state $i \in I$, r_{ij} transitions rates in the original Markov Chain and $\sum_i w^i = 1$. Clearly, this formula is the same as Equation 5.3 if $f_2(i, t, B)$ is replaced by w^i (actually $w^i(s, A_m)$ to indicate that there is a weight associated with each original state and varying with the starting state s). In case the original Markov Chain is *weakly lumpable*, $w^i(s, A_m)$ can be obtained as the ratio $P_s(X_t = m_i|B)/P_s(X_t \in A_m|B)$ for any one set of values of t and B. This is because weak lumpability guarantees this ratio to be invariant with t and B even when the numerator and denominator vary with t and B. Thus, any heuristic which obtains w^i as this ratio will construct the Markov Chain NM in the case of weak lumpability. Next, we describe a procedure which yields NM if the *ratio is invariant* with t and B, but constructs an approximation otherwise. Note that for the strongly lumpable case, q_{IJ} is independent of w^i since $\sum_i w^i = 1$ and the lumped transition rate is equal to the common transition rate $\sum_{j \in J} r_{ij}$ to the destination.

5.2.2 Relation to the Expected Number of Visits

From the discussion in the previous section, it is clear that, in general, there do not exist a set of weights for use in Equation 5.4 which are independent of t and B. In this situation, a *good* approximation would be to average the weights $P_s(X_t = m_i|B)/P_s(X_t \in A_m|B)$ over all possible paths B and over all possible times t for each state in the original Markov Chain and to use this average for w^i. While this is a reasonable approach to take, there appears to be no simple way to obtain w^i procedurally. In this section, we suggest a method for obtaining w^i approximately in terms of known attributes of a Markov Chain.

The approximation is to average t and B over the numerator $N(t, B) = P_s(X_t = m_i|B)$ and the denominator $D(t, B) = P_s(X_t \in A_m)$ separately and to use the ratio of the averages as w^i. This is analogous to replacing the average of ratios by the ratio of the averages of the numerator and the denominator. Note that one may interchange the average and the ratio if all ratios are equal to each other. Thus, we can use this approximation without penalty in the case of weak lumpability because the ratios $P_s(X_t = m_i|B)/P_s(X_t \in A_m|B)$ are all equal to each other. Any procedure using this approximation constructs NM *exactly* if the the original process is lumpable. Otherwise, it constructs an approximation which we conjecture will get better as the original process gets closer to being weakly lumpable. We now show that the weights are expressible in terms of the expected number of visits for an absorbing Markov process which always starts at a fixed state s.

In the first step toward obtaining w^i, we find the mean of $P_s(X_t = m_i|B)$ over all possible paths B starting from state s and reaching state m_i at time t. To

achieve this, multiply the numerator and denominator of $P_s(X_t = m_i|B)/P_s(X_t \in A_m|B)$ by $P_s(B)$, and sum each individually over all B. This yields unconditional probabilities $P_s(X_t = m_i)$ and $P_s(X_t \in A_m)$ respectively. In the next step we show that by integrating over t, these probabilities translate to the expected time spent by the original Markov Chain in state m_i.

Let $T_i(\tau)$ be a random variable denoting the time spent by the process in a non-absorbing state m_i when observed over a time interval $[0, \tau)$. Assume that $X_0 = s$. Then,

$$T_i(\tau) = \int_{t=0}^{\tau} I_{\{X_t = m_i | X_0 = s\}} dt$$

Taking expectations on both sides,

$$\begin{aligned} E[T_i(\tau)] &= E[\int_{u=0}^{\tau} I_{\{X_u = m_i | X_0 = s\}} du] \\ &= \int_{u=0}^{\tau} E[I_{\{X_u = m_i | X_0 = s\}}] du \\ &= \int_{u=0}^{\tau} P_s(X_t = m_i) du \end{aligned} \tag{5.5}$$

Assuming limits exist,

$$\lim_{\tau \to \infty} E[T_i(\tau)] = \int_{u=0}^{\infty} P_s(X_u = m_i) du \tag{5.6}$$

The integral with respect to t in Equation 5.6 is interpreted as the *expected time spent in state* m_i before absorption and so w^i is obtained as the expected

time spent by the process M in state m_i relative to the expected time spent in the subset A_m. Thus, w^i can be estimated by simulating Markov Chain M as follows.

Using Monte-Carlo simulation and starting from state s, simulate the original Markov Chain for K runs, terminating each run at the instant that the process hits an absorbing state. In each simulation run, record the time spent in state m_i, $1 \leq i \leq |A_m|$. Actually, it is sufficient to record the *number of visits* to state m_k in each simulation run because the expected time spent in each state m_k during any visit is exponentially distributed with a known mean. Thus, if $N_s^K(m_k), 1 \leq k \leq |A_m|$ are the number of visits to states $m_k \in A_m$ accumulated over K simulations runs, an estimate of $w^i(s, A_m)$ is given by:

$$w^i(s, A_m) \approx \frac{E[S_i]N_s^K(m_i)}{\sum_i^{|A_m|} E[S_k]N_s^K(m_k)} \tag{5.7}$$

where $E[S_k]$ is the expected sojourn time for state m_k. In the special case when the expected sojourn times are the same for all $m_i \in A_m$, w_i is approximately equal to the number of visits to m_i relative to the number of visits to A_m before absorption.

While the analysis so far has resulted in establishing a method for obtaining a good set of weights to construct an approximate Markov Chain, it has not solved the problem of size - the reason for lumping in the first place. In general, it is necessary to obtain the relative number of visits to each original state. However, in some applications, the contribution of some states may be negligible as compared to others. Consequently, it may be possible in some cases to focus the simulation effort towards obtaining accurate estimates of weights of the important states. Further,

symmetries offered by the specific application can also sometimes be exploited. Fortunately, for fault-tolerant network reliability models, the difference in the time scales of operation of the failure and repair process allows us to focus on a relatively small set of states. We discuss this in detail in Section 5.3.2.

Before concluding this section, we describe another method for obtaining the weights w^i due to the following observation. The Monte-Carlo simulation experiment described in this section re-starts the process as soon as it reaches an absorbing state. The same effect can be achieved by directing all transitions to the absorbing state to state s and eliminating all absorbing states in the original Markov Chain. By solving the modified (original) Markov Chain for its steady-state probabilities, the fraction of time that the process spends in each state of the original Markov Chain can be found. The steady-state probabilities can be used to calculate w_i. This, again, involves the solving of a very large Markov Chain for steady-state probabilities. With this preliminaries, we analyze our heuristic method for constructing lumped reliability models NROR networks in the next section.

5.3 Lumping in the NROR Network Reliability Model

In the previous section, we characterized a heuristic for constructing an approximate Markov Chain with a small number of states to replace an absorbing Markov Chain with a large state space. In this section, we establish a correspondence of

the heuristic method used to lump NROR and ROR network reliability models in the previous chapter with the procedure developed in Section 5.2. In Section 5.3.1 we show that the lumping heuristic used in the previous chapter is a sequence of lumpings of the original Markov Chain and discuss the characteristics of each lumping step. Then, in Section 5.3.2 we show that this heuristic essentially has the characteristics of the one detailed in Section 5.2.

Recall that in Section 4.4 the original state space of the network reliability model is the set of pathsets and a partition is defined such that all pathsets having the same *number* of vias, failed/working free links, failed/reserved single-hop links are assigned to the same subset/lumped state $\mathcal{S}$. That is, each pathset is characterized by the following parameters:

- d : number of single-hop links currently in use. $d = L - v$.
- u: number of working free links in the network. $u = N - L - 2v - f$.
- v : number of vias in the network.
- r : number of links that have undergone repair but are currently not used (working reserved links).
- f : number of failed free links in the network.
- w: number of failed single-hop. $w = v - r$.

All pathsets with the same set of parameters are assigned to the same subset of the partition and $\mathcal{S} = (v, f, r)$, the other parameters beings dependent quantities.

In Section 4.4.1, the transition rates for the lumped model are obtained according to Equation 4.1: For lumped states I and J, each original state $i \in I$

is assigned a weight w^i and the lumped transition rate is obtained as a weighted sum (Figure 4.5); Indeed, it is this lumped transition rate for which we derived an expression in Equation 5.3. For example, if all states in I are equally likely sources of the transition, then, $w^i = \frac{1}{|I|}$. Note that in the lumped transition rates shown in Figure 4.6, the only lumped transition dependent on the choice of weights is the single-hop failure transition. Thus, consider the failure of a single-hop link in an original state (pathset) of lumped state $\mathcal{S}$. We have seen in the previous chapter that the number of original transitions due to single-hop failure to the working lump is not the same for every original state in $\mathcal{S}$. It depends on the arrangement of the initial traffic pattern, vias and working free links in the network, and on the particular single-hop link that fails (Figure 4.4). In general, the fraction of the d transitions from an original state in the source lump $\mathcal{S}$ to the common destination lump is different for different original states in $\mathcal{S}$. The fixed weight aggregation method replaces the original transitions by a single transition to the destination lump with rate equal to a weighted average of the original transition rates. Clearly, different interpretations of the weights will lead to different approximating Markov Chains. By using the interpretation from Section 5.2.1, we will ensure that the set of weights are not arbitrary. In Section 4.4.2, we specified the lumped transition rate on single-hop failure as $C_1(\mathcal{S})d\lambda$, where $C_1(\mathcal{S})$ is given by the expression below. Once the $C_1(\mathcal{S})$ are available for each lumped state, the approximate Markov model's transitions are defined.

$$C_1(\mathcal{S}) = \sum_{i=1}^{\mathcal{N}_v} \left\{ \frac{\sum_{l=1}^{d} \left[\binom{N-L-2v}{u} - \sum_{t=0}^{\min\{u, M_l(A_{vi})\}} 2^t \binom{M_l(A_{vi})}{t} \binom{N-L-2v-2M_l(A_{vi})}{u-t} \right]^{\frac{1}{d}}}{\binom{N-L-2v}{u}} \right\} w^i \tag{5.8}$$

For each state $\mathcal{S}$, the unknowns in the above expression are $M_l(A_{vi})$, the maximum number of alternate two-hop paths as seen by failing single-hop link l in the i^{th} arrangement of v vias (see Figure 4.7), $\mathcal{N}_v$, the number of distinct arrangements of v vias in the state space and $w^i(\mathcal{S}), 1 \leq i \leq \mathcal{N}_v$, the weights for each via arrangement required for fixed weight aggregation. We must bear in mind that since the original Markov Chain is, in general, not lumpable, the lumped model is an approximation even if these unknown parameters are exactly found, say, by exhaustive enumeration. The next section is an analysis of Equation 5.8 to show that we have obtained the lumped transition rates $C_1(\mathcal{S})d\lambda$ for NROR/ROR networks through a *succession* of partitions on the original state space such that the resulting partition is the one described earlier in this section. Accordingly, the lumped transition rate $C_1(\mathcal{S})d\lambda$ is derived by successive application of Equation 5.4.

5.3.1 Correspondence to the Fixed-Weight Method

In this section we show that Equation 5.8 is obtained through three successive lumpings of the original Markov Chain by identifying the *weights* and *accumulated rates* for each lumping in a recursive manner. In Section 4.4.2, we obtained

$$C_1(\mathcal{S}) = \sum_{i=1}^{\mathcal{N}_v} C_1(S_i).w^i(\mathcal{S})$$

by the law of total probability as follows. Consider a lumped state $\mathcal{S}$ having $\mathcal{N}_v$ distinct arrangements of v vias. These arrangements correspond to arrangements $A_{vi}, 1 \leq i \leq \mathcal{N}_v$ in Figure 4.4. Let $S_i, 1 \leq i \leq \mathcal{N}_v$ be the subsets obtained by partitioning $\mathcal{S}$ according to its via arrangements. $C_1(\mathcal{S}_i)$ is interpreted as the probability of recovery given that the network is in subset $\mathcal{S}_i$ and $w^i(\mathcal{S})$ is interpreted as the probability of occurrence of via arrangement A_{vi} when restricted to lump $\mathcal{S}$. As discussed in Section 5.2, $w^i(\mathcal{S})$ is in general different for different paths by which the via arrangements are reached and for different times t. Thus, according to our interpretation in Section 5.2.2, $w^i(\mathcal{S})$ can be approximated by the expected number of visits to *via arrangement* A_{vi} relative to the total number of visits to the $\mathcal{N}_v$ via arrangements in lumped state $\mathcal{S}$. These weights are estimated by simulation in the previous chapter. An analysis of the simulation method is presented in Section 5.3.2. Comparing with Equation 5.4, $C_1(\mathcal{S}_i)d\lambda$ must be the accumulated outgoing rate from via arrangement A_{vi} (having exactly v vias) as a result of previous lumpings.

Thus, let each subset $\mathcal{S}_i$ (with states having a fixed via arrangement) be further partitioned into subsets $\mathcal{S}_{ij}$, $1 \leq j \leq \binom{N-L-2v}{u}$, where each subset contains states having a fixed free link arrangement A_{uj}. The number of such subsets is $\binom{N-L-2v}{u}$ (see Section 4.3.2). *Let us assume* that the weights associated with each subset $\mathcal{S}_{ij}$ are equal to each other and each is equal to $\frac{1}{|\mathcal{S}_{ij}|}$. Then, the numerator of the expression in curly brackets in Equation 5.8 times $d\lambda$ must be the rate resulting from earlier lumpings to working destinations. Notice that, since the weights are all equal to each other for this lumping by assumption, it is sufficient to know the total number of transitions to *any* working state for a given via arrangement, that is, over all free link arrangements keeping the via arrangement fixed. There is no need to determine the transition rates out of each free link arrangement. In Section 4.4.2, we have proved using Lemma 1 and Lemma 2 that this numerator (times $d\lambda$) is exactly equal to the total rate resulting from the d possible active-single hop link failures for a given via arrangement from all possible arrangements of failed and working free links. As regards lumpability, our assumption of equiprobability of the occurrence of $\mathcal{S}_{ij}$ causes the Markov Chain to be weakly lumpable with respect to the partitions whose subsets are $\mathcal{S}_{ij}$.

Now, each subset $\mathcal{S}_{ij}$ defined above contains original states which differ only in their arrangement of inactive single-hop links. Recall that recovery of the NROR network on an active single-hop link failure is independent of the arrangement of the inactive single-hop links in the network (see Section 4.2.1). This independence implies that the recovery probabilities are the same for all original states in each subset $\mathcal{S}_{ij}$. So the numerator in curly brackets in Equation 5.8 is the common

transition rate out of original states in $\mathcal{S}_{ij}$. Thus the original Markov Chain is strongly lumpable under partitioning into subsets $\mathcal{S}_{ij}$.

In summary, the three lumping steps used in constructing lumped reliability models (Figure 5.1) in the previous section are as follows. The first step is to partition the original state space into subsets $\mathcal{S}_{ij}$ - subsets having the same number of vias, failed free links and reserved single-hop links and the same via arrangement and free link arrangement. The original Markov Chain is strongly lumpable under this partition so that the result is an exact Markov Chain. The second lumping step is to merge subsets $\mathcal{S}_{ij}$ which have the same via arrangement A_{vi}. For this lumping step, we assumed weak lumpability by imposing equiprobability on occurrence of sets $\mathcal{S}_{ij}$. The final lumping step is to merge subsets $\mathcal{S}_i$ to get state $\mathcal{S}$. For this lumping step, we estimate weights $w^i(\mathcal{S})$ through simulation. In the next section, we show that our assumption of equiprobability of $\mathcal{S}_{ij}$ is acceptable for this application due to the time-scales of operation of failure and repair processes. Also, in the next section we estimate weights for the final lumping step through a simplified simulation; this simplification is also a result of the different time-scales.

5.3.2 Taking Advantage of the Time-Scales

In the previous section, we partitioned the state space of the original Markov Chain by grouping together pathsets having the same number of the various link *types*. We identified two levels of sub-partitions and used strong and weak lumpability conditions to reduce the number of states in each subset of the original partition. In this section we present an approximate but efficient simulation to estimate weights

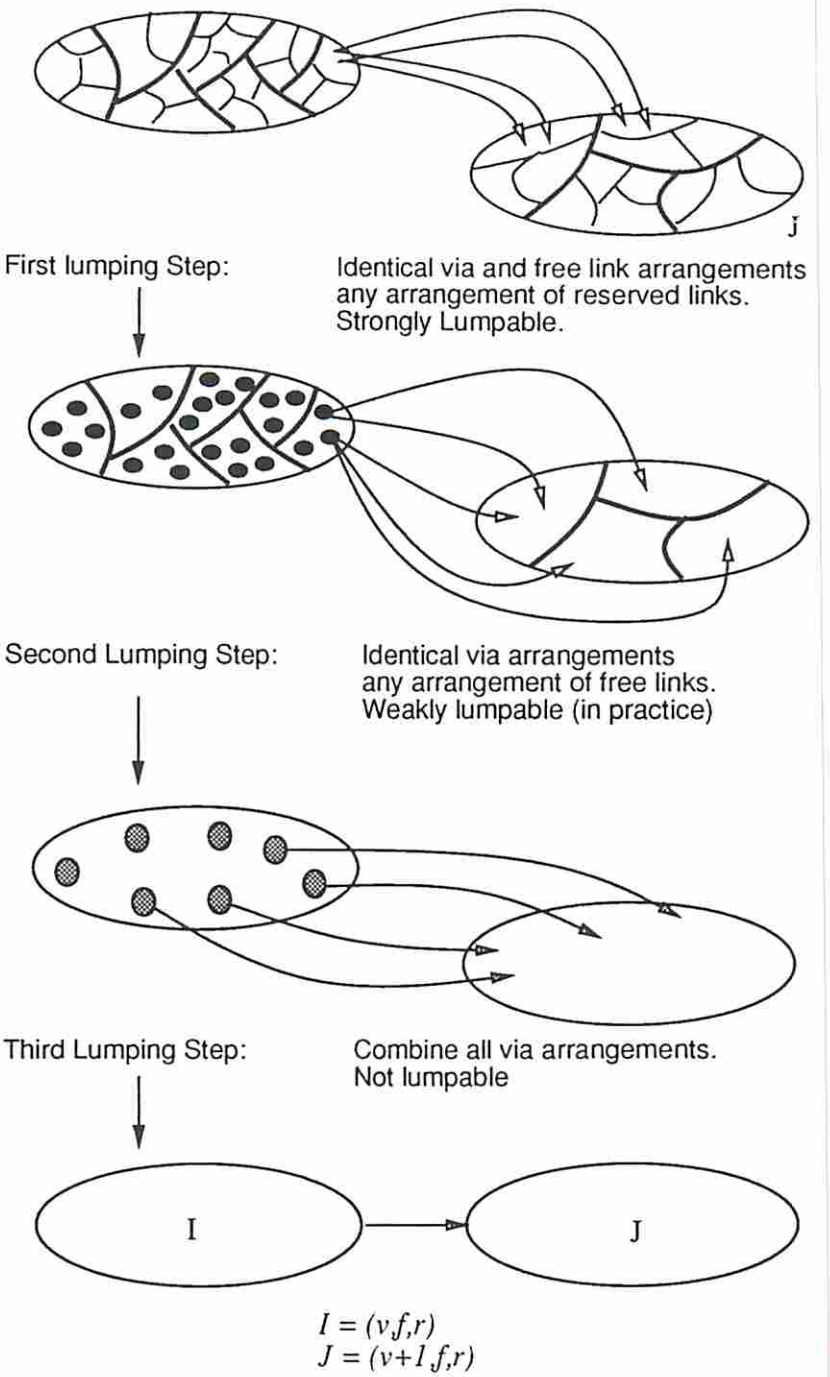


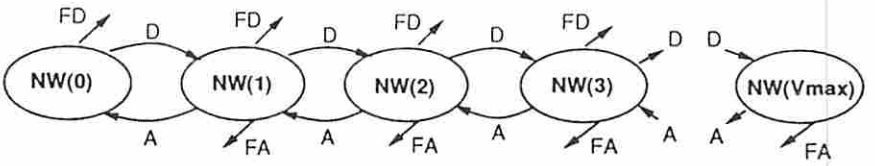
Figure 5.1: Sub-Partitioning of a Lumped State

for the final lumping step and show that the coverage parameters in the lumped model are derived as per the procedure developed in Section 5.2.2.

At this juncture, it is convenient to describe the original NROR Markov Chain in terms of the lumped Markov Chain shown pictorially in Figure 4.8. Each rectangular block in the figure corresponds to a fixed number of v vias in the network. The vertical axis within each block denotes the number of free failed links (equivalently, u, the number of free working links) in the network, while the horizontal axis denotes the number of reserved links in the network. In the figure, the north-west state *of each block* has the following properties. It has exactly as many original states as via arrangements because there is only one arrangement possible with all free links working and all reserved links repaired. Also, there is no transition due to repair in this state. Consequently, there is a vast difference in the sojourn times of states in this lump and other states in the block. Notice also that within any block, the transition flow is towards the north-west corner lumped state. Further, because all transition-causing events change the number of links of certain types (single-hop, free etc.), there are no transitions internal to any lumped state in the entire model. Starting from the north-west corner state (which is exactly the set of $\mathcal{N}_v$ via arrangements in that particular block), we can divide the original states within a block into non-interacting two-dimensional planes. Each plane corresponds to one via arrangement in the north-west corner and all original states in the plane have the same fixed via arrangement; no intra-block transitions can change the plane. Thus, once a plane is chosen upon entry, the network must remain within the plane until it exits. Now, the block may be entered at any state

in the plane. However, since the network is most likely to be in the north-west corner of the block from which the transition is made, the point of entry is most likely to be the lumped state immediately to the right of each north-west corner if the block is entered due to an active single-hop link failure or the lump directly below the north-west corner if due to via-link failure. In each case, a repair takes place in a time which is small as compared to the time-to-failure of each link and in all likelihood the network is instantaneously in the north-west corner state (where it stays for a relatively long interval of time).

Now consider any arbitrary original state in the model. It must belong to some block and to some lumped state within the block; further, it belongs to a plane in that block. This (original) state is most likely to be entered through a sequence of intra-block transitions starting from the north-west because every original state can be reached from its north-west state in the plane. In fact, this was the observation implicitly used to generate the state space of the original Markov Chain in Figure 4.4. All other paths from the initial state may or may not exist, depending on the pattern of free link failures at the time of single-hop link failure. We shall call the path from the north-west state as the primary path. Suppose we ignore all other paths leading to this state, then, it is true that the arrangements of free links are *equally likely* for a given via arrangement (or plane) in a given state $\mathcal{S}$. As the repair process gets faster than the failure process, the primary path gets to be more and more dominant and our *equally likely* assumption of free-link arrangements in the previous section is valid for all practical purposes. Let us assume then, that without combining the $\mathcal{N}_v$ via arrangements together, the



v : Number of Vias.
d : Number of Single-hop links.

A : Via link failure, $2v\lambda C_2(\mathcal{S})$
D : Single-hop link failure, $d\lambda C_1(\mathcal{S})$
FA : $2v\lambda(1 - C_2(\mathcal{S}))$
FD : $d\lambda(1 - C_1(\mathcal{S}))$

NW(0),...,NW(Vmax): North-West corner lumped states in Figure 4.7.

Figure 5.2: Single-hop and Via-Link failure in North-West Lumped States.

Markov Chain is essentially lumpable. We must now estimate the weights $w^i(\mathcal{S})$ for each $\mathcal{S}$.

The network spends most of its time in the north-west corners of each block. Also, the transitions to other states are almost always followed by transitions back to the north-west corner. Recall (see above) that once a via arrangement is chosen, the network remains in the same via arrangement for its stay in the block. To compute $w^i(\mathcal{S})$ efficiently and in an acceptable way, we first consider only transitions between north-west states and transitions (from each) to the failed state. This is shown in Figure 5.2. We propose to obtain the number of visits to each via arrangement in the north-west lumped states by simulating this Markov Chain. Then, we shall use the same weights for all lumped states in each block. Again,

this approximation is reasonable if the primary path is dominant for non-NW corner states are reached from NW corners with approximately equal probability in this case.

Now we describe the method by which we derived the weights for the north-west lumped states in Section 4.4.3. The sojourn times of each via arrangement in any given lump is exponentially distributed with the same parameter. Thus, Equation 5.7 reduces to a ratio of the expected number of visits to each via arrangement. Clearly, it is possible to run K simulation runs and gather the necessary counts. However, in Section 4.4.3, we make one further approximation: starting from the initial state s, where all links are working, we simulate sequences of single-hop links only, ignoring completely all via-link failures. For this approximation, we provide the following argument.

At the end of Section 5.2.2, we suggested finding the weights by converting the absorbing Markov Chain into an ergodic Markov Chain and obtaining the steady state probabilities. Now, consider Figure 5.2 when the coverages are perfect, i.e., there is no transition to the failed state. On a via-link failure, there is only one possibility in the NROR network (see network operation section) - restoring the traffic to the reserved single-hop link. For active-single hop link failures, transitions are each with rate λ divided by the number of alternate vias that can carry the traffic for the failed single hop link. We may conclude therefore, that the steady-state probability of states which are closer to the starting state s are higher than those away from the starting state. In the special case when each single-hop link failure has only one via to carry its traffic, we have a birth-death Chain and the

steady-state probability drops geometrically with parameter 1/2. In the case of more than one via on single-hop failure, the steady-state probability gradient (as we move away from the starting state) will be *steeper* than geometric. Added to this is the fact that coverages are not perfect. Since failure due to imperfect coverage feeds into the starting state, it adds to the probability mass of states closer to the staring state, further increasing the gradient. Thus, it is reasonable to assume that the bulk of the steady-state probability mass for any original state in Figure 5.2 is provided by states which are closer to the starting state relative to its position. Thus, as a first approximation we can delete all transitions due to via-link failures. So we may simulate the resulting Markov chain (now, with transitions only due to single-hop failures and imperfect coverage) for estimating the weights. This indeed was our approach in the previous section and Figures 4.15 and 4.16 demonstrate the effectiveness of this approach for solving reliability models for fault-tolerant private networks. Finally, another source of efficiency in our simulation for coverage parameters is that we automatically eliminate all such via arrangements having relatively small weights.

5.4 Conclusions

The main contribution of this chapter is an analysis of the fixed weight aggregation method of lumping Markov Chains for the purpose of designing efficient lumping heuristics. Based on the analysis, we have developed a heuristic method to lump Markov reliability models in fault-tolerant private network networks; network models usually have very large state spaces. In our analysis of the fixed-weight method,

we show that the lumpability of a Markov Chain is reflected in the *weights* (and only in the weights) associated with each state of the (original) Markov Chain; see Equation 5.4. The weights are in general a function of the starting state, the history of the process and of time. The results of our analysis are as follows. If the original Markov Chain is strongly lumpable with respect to the given partition (of the state space), the lumped transition rates are independent of weights so that the lumped process is time-homogeneous and Markovian. If the Markov Chain is weakly lumpable under the partition, it is Markovian and time-homogeneous if the weights are invariant with history and with time. For example, if states in a lumped state are equiprobable, the Markov Chain is weakly lumpable. If the Markov Chain is not lumpable, then, an approximate lumped Markov Chain can be constructed by *ridding* the weights of history and time according to some rule. In this chapter, we used an approximation to the average of weights over all possible paths from the starting state and all possible times for a fixed starting state. This enabled us to interpret the weights as a ratio of visits to a state to visits to the lumped state it belongs to. We did not really get the average because a straightforward computation of the averages amounts to solving the original Markov Chain (see Equation 5.3) and defeats the purpose of lumping. We show that the approximations to the averages become exact if the original Markov Chain is lumpable, and conjecture that the deviations of the averages from the true averages are proportional to the *degree of lumpability* of the original Markov Chain. We conjecture that our heuristic for solving fault-tolerant network reliability models through lumping gets closer to the best Markovian approximation possible as the original Markov Chain is close

to being lumpable. Thus, we expect our heuristic to apply to Nearly-Lumpable Markov Chains.

From a network reliability point of view, we can see that the pattern-sensitive recovery after failure which is modeled as original-state dependent coverage parameters is not expressible a coverage parameter depending on lumped states alone. The original states associated with the lumped state must be inspected and is responsible for the the difficulties in computation. In practice, differences in the time scales of operation between various processes (such as repair and failure) permit good approximate solutions for fault-tolerant systems and not all lumped states need to be inspected. The model presented here is a simplified model of a fault-tolerant private network because of the restrictions listed in Section 4.2 and further work is needed to accommodate more realistic traffic patterns, capacity and demand models.

In this thesis, we have considered only one measure, namely, reliability. Lumpability measures for other measures [Nic90] also need to be analyzed in order for mathematical modeling to play a useful in the design of fault-tolerant computing systems.

Chapter 6

Conclusions and Future Work

In this thesis, we have analyzed and developed reliability models for traditional fault-tolerant systems which typically have small state spaces and for fault-tolerant private networks which have very large state spaces even for networks of modest size. Accordingly, we have considered the solution aspect of aerospace models and model construction aspect of network models. In each case, we took advantage of properties in the model to simplify construction and solution methods. It is possible, that the properties seen in isolated systems are in fact exhibited by a bigger class of systems.

A systematic study of dependability models from the point of view of behavioral symmetry could help characterize classes of systems for which certain classes of solution methods might be efficient. For example, it seems reasonable to believe that all fault-tolerant reliability (Markov) models have acyclic graph representations at some level of aggregation since the system should eventually fail, albeit after a sequence of lifetimes in different levels of degradation. Also, it appears reasonable to believe that parallel and distributed algorithms would use regular

topologies even if the system's topology is irregular and after failure would reconfigure so as to preserve these topologies as far as possible. Also, these strategies tend to generate recursive topologies. Thus, for parallel algorithm implementations we are likely to find spatial symmetries which should and have been exploited by many researchers. For applications with arbitrary topologies and pattern-sensitive failures, approximation using hybrid lumping methods appears to be promising, But for any practical application, lumped model parameters must be estimated from a sample of the entire population of states. Thus, methods of picking a representative sample appear to be important.

An important step in making solution efficiency available to the non-expert is to automate the process of design. Further research needs to be done to translate the ideas contributing to efficiency at the solution level into commands or features at a high-level representation level. This translation is likely to be a non-trivial task and might require non-conventional approaches of looking at a model construction or solution method.

Appendix A

Quasi-Symmetric Tri-Diagonal Matrices

Let $\mathcal{C}$ be any $n \times n$ tri-diagonal matrix whose elements are $c_{ii} = \alpha_i$, $c_{i,i+1} = \beta_{i+1}$ and $c_{i+1,i} = \gamma_{i+1}$. If α_i are real and $\beta_i \gamma_i > 0$, for $1 \leq i \leq n$, then $\mathcal{C}$ is a *quasi-symmetric* tri-diagonal matrix.

Let $\mathcal{D}$ be a diagonal matrix such that $d_{11} = 1$ and $d_{ii} = (\Pi_{k=2}^{i} \gamma_k / \beta_k)^{\frac{1}{2}}$. Then $\mathcal{D}^{-1}\mathcal{C}\mathcal{D} = \mathcal{T}$, where $\mathcal{T}$ is a real symmetric tri-diagonal matrix such that, $t_{ii} = \alpha_i$ and $t_{i,i+1} = t_{i+1,i} = (\beta_{i+1}\gamma_{i+1})^{\frac{1}{2}}$.

Appendix B

The Sturm Sequence Property

Let $\mathcal{T}$ be an unreduced (i.e., none of the off-diagonal elements are zero in the case of a tri-diagonal matrix. See also [Gan77b, page 50].) symmetric tri-diagonal matrix and $\mathcal{T}_r$ the leading principal submatrix of order r, $1 \leq r \leq n$. Then, the eigenvalues of $\mathcal{T}_r$ *strictly separate* the eigenvalues of $\mathcal{T}_{r-1}$.

Using the minmax characterization of the eigenvalues of a matrix, it has been established that for any matrix $\mathcal{A}$, the eigenvalues of the the principal submatrix of order $(r-1)$ weakly separate the eigenvalues of the principal submatrix of order r [Wil65]. For an unreduced symmetric tri-diagonal matrix, this separation is strict [GV84], [Wil65]. Let $p_r(\lambda)$ be the principal minor of order r of the matrix $(\mathcal{T} - \lambda\mathcal{I})$. Let $t_{ii} = \alpha_i$ and $t_{i+1,i} = t_{i,i+1} = \beta_i > 0$ so that $\mathcal{T}$ is an unreduced symmetric tri-diagonal matrix. Then, the following recursion holds:

$$\begin{aligned} p_r(\lambda) &= (\alpha_i - \lambda)p_{i-1}(\lambda) - \beta^2 p_{i-2}(\lambda), \quad (i = 2, \cdots, n) \\ p_1(\lambda) &= \alpha_1 - \lambda \end{aligned}$$

$$p_0(\lambda) = 1$$

The technique to prove that the eigenvalues of $\mathcal{T}$ are distinct is by assuming that they are non-distinct and reaching a contradiction (see [Wil65, page 300]).

Appendix C

Gerschgorin's Circle Theorem

Every eigenvalue of the matrix $\mathcal{A}$ lies in at least one of the circular discs (in the complex plane) with centers a_{ii} and radii $\sum_{j \neq i} |a_{ij}|$ [Wil65].

Bibliography

[ACM81] G. R. Ash, R. H. Cardwell, and R. P. Murray. Design and Optimization of Networks with Dynamic Routing. *Bell System Technical Journal*, 60(8):1821–1845, October 1981.

[AM91] B. E. Aupperle and J. F. Meyer. State Space Generation for Degradable Systems. In *Proceedings of 21st Fault-Tolerant Computing Symposium*, 1991.

[AP88] S. Abraham and K. Padmanabhan. Reliability of the Hypercube. In *Proceedings of 12th International Conference on Parallel Processing*, 1988.

[Arn73] 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), March 1973.

[Bar79] S. Barnett. *Matrix Methods for Engineers and Scientists*. McGraw Hill Book Company, New York, 1979.

[BCS69] W. G. Bouricius, W. C. Carter, and P. R. Schneider. Reliability Modeling Techniques for Self-Repairing Computer Systems. In *Proceedings of 24th National Conference of the ACM*, 1969.

[BDT+87] S. J. Bavuso, J. B. Dugan, K. S. Trivedi, E. M. Rothman, and W. E. Smith. Analysis of Typical Fault-Tolerant Architectures using HARP. *IEEE Transactions on Reliability*, R-36(2), June 1987.

[Bea78] M. D. Beaudry. Performance-Related Reliability Measures for Computing Systems. *IEEE Transactions on Computers*, C-27(6), June 1978.

[BR90] M. Balakrishnan and C. S. Raghavendra. On Reliability Modeling of Closed Fault-Tolerant Computer Systems. *IEEE Transactions on Computers*, 39(4), April 1990.

[BS72] R. H. Bartels and G. W. Stewart. Solution of the Matrix Equation AX+XB = C. *Communications of the ACM*, 15(9), September 1972.

[BT77] D. R. Barr and M. U. Thomas. An Eigenvector Condition for Markov Chain Lumpability. *Operations Research*, 25(6):1028–1031, December 1977.

[BT86] A. Bobbio and K. S. Trivedi. An Aggregation Technique for the Transient Analysis of Stiff Markov Chains. *IEEE Transactions on Computers*, C-35(9), September 1986.

[But84] R. W. Butler. The Semi-Markov Unreliability Range Evaluator (SURE) Program. Technical report, NASA Langley Research Center, Hampton, Virginia, July 1984.

[CDLL81] A. Costes, J. E. Doucet, C. Landrault, and J. C. Laprie. SURF: A Program for Dependability Evaluation of Complex Fault-Tolerant Computing Systems. In *Proceedings of 11th Fault-Tolerant Computing Symposium*, 1981.

[CLV+91] B. A. Coan, W. E. Leland, M.P. Vecchi, A. Weinrib, and L. T. Wu. Using Distributed Topology Update and Preplanned Configurations to Achieve Trunk Network Survivability. *IEEE Transactions on Reliability*, 40(4), October 1991.

[Col87] C.J. Colbourn. *The Combinatorics of Network Reliability.* Oxford University Press, Oxford, 1987.

[Cou77] P.-J. Courtois. *Decomposability: Queueing and Computer System Applications.* Academic Press, New York, 1977.

[CS84] P.-J. Courtois and P. Semal. Bounds for the Positive Eigenvectors of Nonnegative Matrices and for Their Approximations by Decomposition. *J. Assoc. Comp. Mach.*, 31, 1984.

[DBA74] G. Dahlquist, A. Bjorck, and N. Anderson. *Numerical Methods.* Prentice Hall, Inc., Englewood Cliffs, N. J., 1974.

[DT89] J. B. Dugan and K. S. Trivedi. Coverage Modeling for Dependability Analysis of Fault-Tolerant Systems. *IEEE Transactions on Computers*, 38(6), August 1989.

[DTGN84] J. B. Dugan, K.S. Trivedi, R. Geist, and V. F. Nicola. Extended Stochastic petri-Nets: Applications and Analysis. In E. Gelenbe, editor, *Performance'84*. North Holland, Amsterdam, 1984.

[DTSG86] J. B. Dugan, K. S. Trivedi, M. Smotherman, and R. Geist. The hybrid automated reliability predictor. *AIAA J. Guidance, Contr., Dynam.*, 9, 1986.

[Fle71] J. L. Fleming. RELCOMP: A Computer Program for Calculating System REliability and MTBF. *IEEE Transactions on Reliability*, R-20(3), August 1971.

[Gan77a] F. R. Gantmacher. *Matrix Theory: Vol. I.* Chelsea Publishing Company, New York, 1977.

[Gan77b] F. R. Gantmacher. *Matrix Theory: Vol. II.* Chelsea Publishing Company, New York, 1977.

[GCdSSL86] A. Goyal, W. C. Carter, E. de Souza e Silva, and K. S. Trivedi S. S. Lavenberg. A System Availability Estimator. In *Proceedings of IEEE Int. Symposium on Fault-Tolerant Computing*, 1986.

[GM84] D. Gross and D. Miller. The Randomization Technique as a Modeling Tool and Solution Procedure for Transient Markov Processes. *Operations Research*, 32(2), 1984.

[Gra77] W. K. Grassman. Transient Solution in Markovian Queueing Systems. *Computers and Operations Research*, 4, 1977.

[Gro87] W. D. Grover. The Self-Healing Network: A Fast Distributed Restoration Technique for Networks Using Digital Cross Connect Machines. In *Proceedings of IEEE GLOBECOM*, pages 28.2.1–28.2.6, December 1987.

[GSB89] R. Geist, M. Smotherman, and M. Brown. Ultrahigh Reliability Estimates for Systems Exhibiting Globally Time-Dependent Failure Processes. In *Proceedings of 19th Fault-Tolerant Computing Symposium*, 1989.

[GSH+92] A. Goyal, P. Shahabuddin, P. Heidelberger, V. F. Nicola, and P. W. Glynn. A Unified Framework for Simulating Markovian Models of Highly Dependable Systems. *IEEE Trans. on Computers*, 41(1), 1992.

[GV84] G. H. Golub and C. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, Maryland, 1984.

[GVsM90] W. D. Grover, B. D. Venables, J. H. stidham, and A. F. Milne. Performance Studies of a Self-Healing Network Protocol in Telecom Canada Long Haul Networks. In *Proceedings of IEEE GLOBECOM*, pages 403.3.1–403.3.7, December 1990.

[HS84] D. P. Heyman and M. J. Sobel. *Stochastic Models in Operations Research, Vol. II.* McGraw Hill, Inc., New York, 1984.

[HV79] Golub G. H. and C. Van Loan. A Hessenberg-Schur Method for the Problem AX+XB = C. *IEEE Transactions on Automatic Control*, 24, 1979.

[Joh89] B. W. Johnson. *Design and Analysis of Fault-Tolerant Digital Systems.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[KR86] J. G. Kuhl and S. M. Reddy. Fault-Tolerance Considerations in Large, Multiple-Processor Systems. *IEEE Computer*, pages 56–67, March 1986.

[KS76] J. Kemeny and L. J. Snell. *Finite Markov Chains.* Springer-Verlag, Berlin, 1976.

[Lap85] J. C. Laprie. Dependable Computing and Fault-Tolerance: Concepts and Terminology. In *Proceedings of 15th Fault-Tolerant Computing Symposium*, 1985.

[LSP82] L. Lamport, R. Shostak, and M. Pease. The Byzantine General's Problem. *ACM Trans. Programming Language Systems*, 4(3), July 1982.

[MA70] F. P. Mathur and A. Avizienis. Reliability Analysis and Architecture of a hybrid-redundant Digital System: Generalized Triple Modular Redundancy with Repair. In *Proceedings of the AFIPS SJCC*, 1970.

[MAG82] S. V. Makam, A. A. Avizienis, and G. Grusas. UCLA ARIES 82 Users' Guide. Technical report, Computer Science Department, University of California, Los Angeles, August 1982.

[Mat72] F. P. Mathur. Automation of Reliability Evaluation Procedures through CARE - The Computer-Aided Reliability Estimation Program. In *Proceedings of the AFIPS, Fall Joint Computer Conference*, 1972.

[McG83] J. McGough. Effects of Near-Coincident Faults in Multiprocessor Systems. In *Proceedings of IEEE/AIAADigital Systems Avionics Conference*, November 1983.

[MdG89] R. R. Muntz, E. de Souza e Silva, and A. Goyal. Bounding Availability of Repairable Computer Systems. *IEEE Transations on Computers*, 38(12), December 1989.

[Mey80] J. F. Meyer. On Evaluating the Performability of degradable Computing Systems. *IEEE Transactions on Computers*, 29, August 1980.

[Mey82] J. F. Meyer. Closed-Form Solutions of Performability. *IEEE Transactions on Computers*, 31(7), July 1982.

[MHLK87] J. Meunier, J. Hopkins, S. Leepard, and M. Koch. Network Planning Tools for Flexible, Survivable Networks. In *ISS '87*, pages C6.1.1–C6.1.8, 1987.

[ML78] C. Moler and C. Van Loan. Nineteen Dubious ways of Computing the Exponential of a Matrix. *SIAM Review*, 20(4), 1978.

[Mol85] M. Molloy. Discrete Time Stochastic Petri Nets. *IEEE Transactions on Software Reliability*, SE-11(4), April 1985.

[Mor89] M. Morganti. F-T in Telecommunication Networks: State, Perspectives and Trends. In *Proceedings of the 19th International Symposium on Fault-Tolerant Computing*, 1989.

[MRT87] R. A. Marie, A. L. Reibman, and K. S. Trivedi. Transient Analysis of Acyclic Markov Chains. *Performance Evaluation*, 7, 1987.

[MST85] J. McGough, M. Smotherman, and K. S. Trivedi. The Conservativeness of Reliability Estimates Based on Instantaneous Coverage. *IEEE Transactions on Computers*, C-34(7), July 1985.

[MT83] Geist R. M. and K. S. Trivedi. Ultra-Reliability Prediction for Fault-Tolerant Computers. *IEEE Transactions on Computers*, C-32(12), December 1983.

[MT86] M. Mulazzani and K. S. Trivedi. Dependability Prediction: Comparison of Tools and Techniques. In *Proceedings of IFAC SAFECOMP '86, Sarlat, France*, 1986.

[Mur89] T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4), April 1989.

[MWT91] K. J. Muppala, S. P. Woolet, and K. S. Trivedi. Real-Time-Systems Performance in the Presence of Failures. *IEEE Computer*, May 1991.

[NA80] Y. W. Ng and A. A. Avizienis. A Unified Reliability Model for Fault-Tolerant Computers. *IEEE Transations on Computers*, C-29(11), November 1980.

[Nel90] V. P. Nelson. Fault-tolerant computing: Fundamental concepts. *IEEE Computer*, 23(7), July 1990.

[Ng76] Y. W. Ng. *Reliability Analysis and Modeling for Fault-Tolerant Computers.* PhD thesis, Department of Computer Science, University of California, Los Angeles, September 1976.

[NG87] W. Najjar and J. L. Gaudiot. Reliability and Performance Modeling of Hypercube-Based Multiprocessors. In *Proceedings of the 2nd International Workshop on Applied Mathematics and Performance/Reliability Models of Computer/Communication Systems, Rome (Italy)*, May 1987.

[Nic90] V. F. Nicola. Lumping in Markov Reward Processes. In *First International Conference on the Numerical Solution of Markov Chains, IFIP W.G. 7.3*, January 1990.

[Osb60] E. E. Osborne. On Pre-Conditioning of Matrices. *J. Assoc. Comput. Mach.*, 7, 1960.

[oTSABL83] Members of Tech. Staff AT&T Bell Laboratories. In R. F. Rey, editor, *Engineering and Operations in the BELL SYSTEM.* Bell Telephone Laboratories, Inc., NJ, 1983.

[PR69] B. N. Parlett and C. Reinsch. Balancing A Matrix for Calculations of Eigenvalues and Eigenvectors. *Numer. Math.*, 13, 1969.

[Pra86] D. K. Pradhan. *Fault-Tolerant Computing*, volume I and II. Prentice Hall, first edition, 1986.

[PSL80] M. Pease, R. Shostak, and L. Lamport. Reaching Agreement in the presence of Faults. *JACM*, 27(4), April 1980.

[RA73] D. A. Rennels and A. Avizienis. RMS: A Reliability Modeling System for Self-Repairing Computers. In *Digest of the IEEE 3rd Symposium of Fault-Tolerant Computing*, 1973.

[RA90a] S. Rai and D.P. Agrawal, editors. *Advances in Distributed Reliability.* IEEE Computer Society Press, Los Alamitos, CA, 1990.

[RA90b] S. Rai and D.P. Agrawal, editors. *Distributed Computing Network Reliability.* IEEE Computer Society Press, Los Alamitos, CA, 1990.

[Rei87] A. L. Reibman. *Transient Analysis of Large Stiff Markov Models: Numerical and Approximate Solution Techniques.* PhD thesis, Department of Computer Science, Duke University, 1987.

[RL89] C. S. Raghavendra and D.A. Lee. Reliability Evaluation of Fault-Tolerant Computers for Aerospace Applications. In *AIAA Computers in Aerospace VII Conference and Exhibit, Monterey, Calif.*, 1989.

[RST89] A. Reibman, R. Smith, and K. Trivedi. Markov and Markov Reward Model Transient Analysis: An Overview of Numerical Approaches. *European Journal of Operations Research*, 40, 1989.

[RZ90] A. Reibman and H. Zaretsky. Modeling Fault Coverage and Reliability in a Fault-Tolerant Network. In *Globecom '90*, 1990.

[SBG79] J. J. Stiffler, L. A. Bryant, and L. Guccione. CARE III Final Report. Phase 1. Technical report, NASA Contractor Report 159122, November 1979.

[Ser84] O. Serlin. Fault-Tolerant Systems in Commercial Applications. *IEEE Computer*, August 1984.

[Sie82] D. P. Siewiorek. *The Theory and Practice of Reliable System Design.* Digital Press, Bedford, Massachusetts, 1982.

[Sie84] D. P. Siewiorek. Architecture of Fault-Tolerant Computers. *IEEE Computer*, August 1984.

[Skl76] J. R. Sklaroff. Redundancy Management Technique for Space Shuttle Computers. *IBM Journal of Research and Development*, January 1976.

[ST87] R. A. Sahner and K. S. Trivedi. Reliability Modeling using SHARPE. *IEEE Transactions on Reliability*, R-36(2), June 1987.

[ST89] W. E. Smith and K. S. Trivedi. Dependability Evaluation of a Class of Multi-Loop Topologies for Local Area Networks. *IBM Journal of Research and Development*, 33, September 1989.

[Str80] G. Strang. *Linear Algebra and its Applications.* Harcourt Brace Jovanovich, New York, 1980.

[TB77] M. U. Thomas and D. R. Barr. An Approximate Test of Markov Chain Lumpability. *Journal of the American Statistical Association*, 72(357), 1977.

[TM84] W. N. Toy and M. Morganti. Guest Editor's Introduction: Fault-Tolerant Computing. *IEEE Computer*, August 1984.

[Tri82] K. S. Trivedi. *Probability and Statistics with Reliability, Queueing and Computer Applications.* Prentice Hall, Englewood Cliffs, N. J., 1982.

[tro89] *IEEE Transactions on Reliability: Special Issue on Reliability of Parallel and Distributed Computing Networks*, 38(1), 1989.

[tro91] *IEEE Transactions on Reliability: Special Issue of Telecommunication System and Services*, 40(4), 1991.

[Wel83] P. D. Welch. The Statistical Analysis of Simulation Results. In S. S. Lavenberg, editor, *Computer Performance Modeling Handbook.* Academic Press, NY, 1983.

[Wil65] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, Oxford, 1965.

[ZD63] L. A. Zadeh and C. A Desoer. *Linear System Theory, A State Space Approach.* McGraw-Hill Book Company, Inc., New York, 1963.