Yale University
Department of Computer Science

PERFORMABILITY ANALYSIS: MEASURES, AN ALGORITHM, AND A CASE STUDY[1]

R. M. Smith[2], K. S. Trivedi[3], and A. V. Ramesh[3]

YALEU/DCS/TR-600
January 1988

[1]This work was supported by the U.S. Air Force Office of Scientific Research under Grant AFOSR-84-0132, the U.S. Army Research Office under Grant DAAG29-84-K-0045 and under DOD Research Instrumentation Grant DAAL03-87-G-0066.
[2]Department of Computer Science, Yale University
[3]Department of Computer Science, Duke University

Abstract

Multiple-processor systems can provide higher performance and higher reliability/availability than single-processor systems. In order to properly assess the effectiveness of multi-processor systems, measures that combine performance and reliability are needed. We describe the behavior of the multi-processor system as a continuous-time Markov chain and associate a reward rate (performance measure) with each state. We evaluate the distribution of performability for analytical models of a multi-processor system using a recently improved polynomial-time algorithm that obtains the distribution of performability for non-repairable as well as repairable systems with heterogeneous components with a substantial speedup over earlier work. The system that we analyze with several Markov reward models is the (C.mmp) multi-processor system developed at Carnegie Mellon University. The example indicates that distributions of cumulative performance measures over finite intervals reveal behavior of multi-processor systems not indicated by either steady-state or mean values alone.

1 Introduction

The proliferation of fault-tolerant multiple processor systems has given rise to the need to develop composite reliability and performance measures. For this purpose, Meyer [20] developed a conceptual framework of *performability*. In this paper, we consider performability models based on Markov Reward Models (MRMs). We obtain a variety of performability measures on several models of a multi-processor system to illustrate the effect of different fault-tolerant mechanisms on the ability of the system to complete useful work in a finite time interval. In the course of this study, we show that the distribution of accumulated reward illuminates effects that are not detected by steady-state values, instantaneous measures, or expected values of cumulative measures. Hence, the performability distribution provides new insight on the behavior of multi-processor computer systems. We describe a new $O(n^3)$ algorithm for the computation of the distribution of accumulated reward in a finite utilization interval where n is the number of states in the MRM.

The evolution of the system through configurations with different sets of operational components is represented by a continuous-time Markov chain (CTMC) which we refer to as a structure-state process. The set of rewards associated with the states of a structure-state process are referred to as the reward structure. Together the structure-state process and the reward structure determine a Markov Reward Model (MRM). Because the time-scale of the performance-related events (*e.g.*, instruction execution, job service) is at least two orders of magnitude less than the the time-scale of the reliability-related events (*i.e.*, component failure, component repair) steady-state values of performance models are used to specify the performance levels or reward rates for each structure state.

We analyze several MRMs of a multi-processor system with 16 processors, 16 memories and a crossbar switch. In Appendix A we describe an improved algorithm to obtain the performability distributions from MRMs with n structure-states that provides an $O(n)$ speedup over the earlier algorithm in [19]. The algorithm may be applied to MRMs constructed for repairable or non-repairable systems. We demonstrate the use of our algorithm on a problem of moderate size. Previously published results on performability distributions for finite time intervals have been carried out only on very small problems. With the multi-processor system, we examine the effect of different modeling assumptions on a number of measures including the distribution of accumulated reward.

The freedom to modify the structure-state process as well as the reward structure allows the modeler to represent a wide variety of situations. In the performability domain, there are two extremes. First we may have a structure-state process with only a single state and a possibly complex performance model to generate the reward associated with the single state. A 'pure' performance model that ignores failure and repair but considers memory contention overestimates the ability of the system to complete useful work. On the other extreme, a 'pure' availability model ignores different levels of performance (other than operational or failed). A model that takes into account both aspects of system behavior by a combined performability measure is more appropriate for the evaluation of computer systems that may undergo a graceful degradation of performance. After completing the introduction, we describe the multi-processor system in section 2. In section 3, we present results for MRMs of the multi-processor system. In Appendix A we describe and analyze the computational cost of the algorithm used to determine the distribution of accumulated reward for cyclic or acyclic MRMs.

1.1 Notation

The evolution of the system in time is represented by the finite-state stochastic process $\{Z(t),\ t \geq 0\}$, which characterizes the dynamics of the system structure and environmental influences. $Z(t) \in\ S = \{1, 2, \ldots, n\}$ is the structure-state of the system at time t. The holding times in the structure-states are exponentially distributed, and hence $Z(t)$ is a homogeneous CTMC. Even in situations where the holding times are generally distributed, they may often be acceptably approximated using a finite number of exponential phases [14]. We let q_{ij} be the transition rate from state i to state j and $Q = [q_{ij}]$ be the n by n generator matrix where

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

Let $p_i(t)$ denote Prob[$Z(t) = i$], the probability that the system is in state i at time t. The column vector $\underline{p}(t)$ of the state probabilities may be computed by solving a matrix differential equation [23]:

$$\frac{d}{dt}\, \underline{p}(t) \quad = \quad Q^T\, \underline{p}(t) \quad . \tag{1}$$

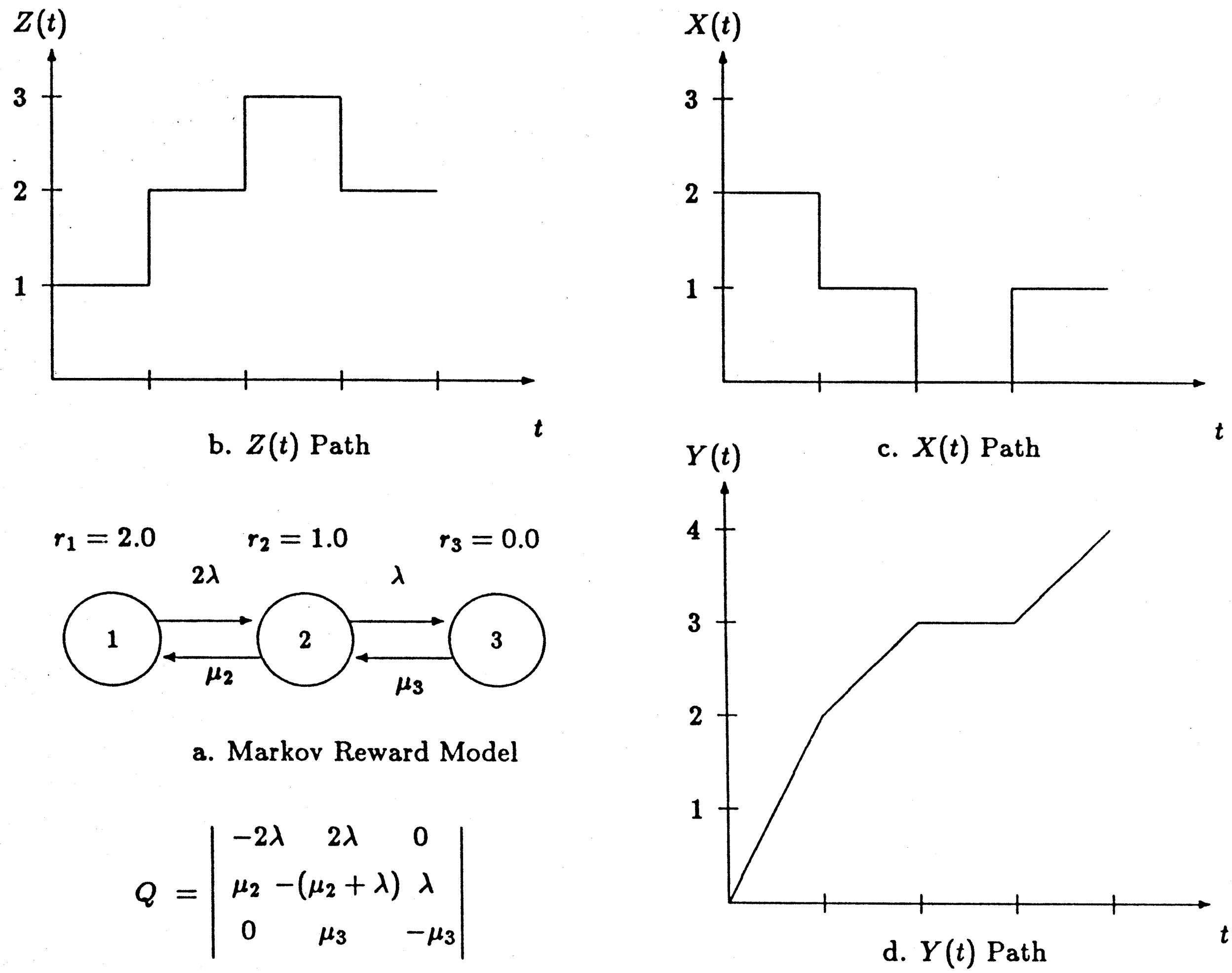
The steady-state probability vector $\underline{\pi}$ of the Markov chain is the solution for the linear system :

$$Q^T\, \underline{\pi} = 0 \qquad \sum_i \pi_i \ = \ 1 \ .$$

Let r_i be the reward rate (or the performance level) associated with structure-state i; then the vector $\underline{r}$ defines the reward structure. The reward rate of the system at time t is defined to be $X(t) = r_{Z(t)}$. We let $Y(t)$ be the accumulated reward until time t, that is, the area under the $X(t)$ curve,

$$Y(t) = \int_0^t X(\tau) d\tau \quad .$$

Consequently, by interpreting rewards as performance levels, we see that the distribution of accumulated reward is at the heart of characterizing systems that evolve through states with different reward rates (*e.g.*, performance levels). In Figure 1 we depict a Markov reward model with a 3-state CTMC



$$Q = \begin{vmatrix} -2\lambda & 2\lambda & 0 \\ \mu_2 & -(\mu_2+\lambda) & \lambda \\ 0 & \mu_3 & -\mu_3 \end{vmatrix}$$

Figure 1: 3-State Markov Reward Model with Sample Paths of Z(t), X(t) and Y(t) Processes.

for the structure-state process and a simple reward structure, the transition matrix of the CTMC, as well as sample paths for the stochastic processes $Z(t)$, $X(t)$ and $Y(t)$. Note that a given sample path of $Z(t)$ determines a unique sample path for $X(t)$ and $Y(t)$.

We denote the distribution of accumulated reward by time t evaluated at x as:

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

A fundamental question about any system is simply, "What is the probability of completing a given amount of useful work within a specified time interval?" The answer is provided by the complement of the above distribution:

$$\mathcal{Y}^C(x,t) \equiv \text{Prob}[\ Y(t) > x\ ].$$

The time-averaged accumulated reward, its distribution, and its complementary distribution are denoted as:

$$W(t) \equiv \frac{Y(t)}{t} \equiv \frac{1}{t}\int_0^t X(\tau)d\tau \quad , \quad \mathcal{W}(x,t) \equiv \text{Prob}[\,W(t) \leq x\,] \quad \text{and} \quad \mathcal{W}^C(x,t) \equiv \text{Prob}[\,W(t) > x\,].$$

A special case of $W(t)$ is obtained when we assign a reward rate 1 to operational states and zero to non-operational states. In this case, $W(t)$ is known as the ***interval availability*** $A_I(t)$. The complementary distributions explicitly answer the questions of the modeler and are easily obtained from the results for $\mathcal{Y}(x,t)$ in Appendix A. To complete our notation, we note that we have assumed a distinguished initial state. To explicitly indicate this dependence on the initial state we will use a subscript on cumulative and time-averaged random variables and their distributions. For example, $W_i(t)$ denotes the time-averaged accumulated reward for the interval $(0,t)$ given that the initial state is i (*i.e.*, $Z(0) = i$).

The ability to complete a given amount of work with probability one is a property of some Markov Reward models. An MRM is said to have the ***completion property*** if does not have a reachable closed-set C of states such that $r_i = 0$ for all $i \in C$. As an example of an MRM with the completion property, consider Figure 1a with all parameters greater than zero. Since the probability of remaining in structure state 3 for all but a finite amount of time in an infinite time interval is zero and structure states 1 and 2 have non-zero reward, any finite amount of reward will be accumulated if the time interval is long enough. Because descriptions of fault-tolerant systems almost always include "failed" (zero reward) structure states, we will refer to MRMs of fault-tolerant systems that take repair actions from all "failed" structure states as MRMs with completion. The completion property is a useful distinction because it indicates the most appropriate measures for a model. MRMs with the completion property are appropriately described with $\mathcal{W}(x,t)$, while models without it are readily described with $\mathcal{Y}(x,t)$. Those Markov models without the completion property will be referred to as MRMs with ***imperfect repair***. An MRM in which operational states are assigned reward rate 1 and non-operational states are assigned rate 0 are called ***availability models***. In an availability model, if we further require that all non-operational states are absorbing then we have a ***reliability model***.

1.2 Previous Work

Early attempts to evaluate fault-tolerant computer systems were restricted to transient analysis of the CTMC describing the evolution of the system over time. The immediate result relating the transient probability to the probability of the system operating at a specified reward level,

$$\text{Prob}[\,X(t) = r\,] = \sum_{\{j \mid r_j = r\}} \text{Prob}[\,Z(t) = j\,] = \sum_{\{j \mid r_j = r\}} p_j(t) \quad ,$$

was exploited by Huslende [15] and Wu [28].

Gracefully degrading systems provide useful computation by reconfiguring to adjust to the failure of one or more components. Beaudry used the notion of *computation availability* which in our notation is the expected reward rate at time t:

$$E[\ X(t)\ ] \ = \ \underline{r}^T \underline{p}(t) \ = \ \sum_i r_i p_i(t) \ ,$$

and its limiting value :

$$\lim_{t \to \infty} E[\ X(t)\ ] \ = \ \underline{r}^T \underline{\pi} \ = \ \sum_i r_i \pi_i \ .$$

These two quantities are generalizations of instantaneous and steady-state availability, respectively. Huslende considered performance reliability by assuming a minimum performance threshold:

$$R(\text{threshold}, t) \equiv \text{Prob}[\ X(\tau) \ \geq \ \text{threshold}, \ \forall \tau \leq t\ ] \ ;$$

a generalization of reliability.

Under general assumptions about the stochastic process $\{Z(t),\ t \geq 0\}$ and the reward structure $\underline{r}$, Howard [13] studied the expected accumulated reward $E[\ Y(t)\ ]$ for finite intervals of time and the expected time-averaged accumulated reward over an infinite time interval. It is interesting to note that the limit $t \longrightarrow \infty$ of the expected value of $X(t)$ and $W(t)$ are equal:

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

With our notation we can express $E[\ Y(t)\ ]$ as :

$$E[\ Y(t)\ ] \ = \ E[\ \int_0^t X(\tau) d\tau\ ] \ = \ \int_0^t E[\ X(\tau)\ ] d\tau \ = \ \sum_i r_i \int_0^t \ p_i(\tau) d\tau.$$

To compute $E[Y(t)]$ we define $L_i(t) = \int_0^t \ p_i(\tau) d\tau$ to be an element of $\underline{L}(t)$ and derive a system of ordinary differential equations for $\underline{L}(t)$ by integrating equation (1) :

$$\frac{d}{dt} \underline{L}(t) \ = \ Q^T \underline{L}(t) \ + \ \underline{p}(0) \ .$$

Solutions are readily calculated using methods similar to those used to solve equation (1). Often we are interested in the behavior of $Y(t)$ far from the mean (as is the case when a system is required to deliver a specific reward with high probability), and in this case the central moments do not provide accurate information. Consequently measures that provide a more detailed look at system behavior are needed.

Recently, considerable attention has been given to the problem of evaluating the distribution of accumulated reward, $\mathcal{Y}(x,t)$. The problem is more easily solved if the distribution of accumulated reward is to be evaluated over an infinite time interval. Beaudry [1] has shown that the distribution of accumulated reward until system failure ($\mathcal{Y}(x,\infty)$) for a system with imperfect repair can be obtained as the time-to-failure distribution of an associated CTMC obtained by simply dividing the rates of transitions leaving a given state i by r_i.

For finite time intervals, Meyer [21] obtained the distribution of accumulated reward in acyclic Markov reward models (no loops in the structure-state CTMC) with r_i being a monotonic function of the state labeling. A direct approach that numerically integrated the convolution equations in the time domain for acyclic models was developed and implemented by Furchtgott and Meyer [9]. The computational complexity is exponential in the number of states so the applicability of the direct time-domain approach is limited to problems with a few states over a short time interval. Subsequently, Goyal and Tantawi [10] developed an $O(n^3)$ algorithm to compute the distribution of accumulated reward in general acyclic structure-state processes with monotonic reward rates. Ciciani and Grassi [3] and Donatiello and Iyer [6] proposed algorithms that do not require the rewards to be monotonic.

MRMs that have cyclic structure-state CTMCs are more difficult. By using the central limit theorem, it can be shown that the asymptotic distribution of the accumulated reward over a time interval $(0, t)$ for t sufficiently large is normally distributed with mean $\lim_{\tau\rightarrow\infty} E[X(\tau)]$ multiplied by t and variance $\alpha\sqrt{t}$. Computational methods to determine $\lim_{\tau\rightarrow\infty} E[X(\tau)]$ and α may be found in Hordijk *et al.* [12].

Iyer *et al.* [16] describe a recursive technique for computing moments of the distribution of accumulated reward for cyclic MRMs. With the moments in hand, bounds on the distribution of accumulated reward are available. As noted earlier, because the central moments describe the behavior of the distribution about the mean, the bounds are often too loose to be helpful at the extremes, which are often of interest. The difficulties are similar to those one faces extrapolating the value of a continuous function a distance away from a point where all the derivatives are known.

More recently, Goyal, Tantawi, and Trivedi [11] formulated the interval availability problem (a special instance of $W(t)$, for a reward structure with reward rates $r_i = 1$ if state i is operational and zero else) as a system of first order partial differential equations. The randomization technique has also been applied to the interval availability problem by de Souza e Silva and Gail [8].

Puri [22] derived a linear system in the double Laplace transform of the distribution of accumulated reward for a general CTMC and arbitrary reward structure. The numerical solution of the double transform system was proposed in [19]. In Appendix A we present an improved $O(n^3)$ algorithm to evaluate the distribution of accumulated reward for cyclic and acyclic MRMs with n states. Note that the $O(n)$ speedup over our previous algorithm [19] makes considerably larger MRMs solvable in practice.

In Table 1 we present the measures that we use to examine the behavior of the example multi-processor system. We group the measures by the random variables used in their definition. Each measure's properties are then indicated. The properties that we indicate are whether the quantity measured is instantaneous or cumulative, steady state or transient. We also indicate in Table 1 whether the measure is a distribution or a central moment. We use a column in Table 1 for each measure to indicate the model families each measure is typically applied to. We use rel, av, Imp-rep, and compl as abbreviations for the reliability, availability, imperfect repair, and completion families respectively.

Measure	Notation	Common Model Family	Cumulative or Instantaneous Measure	Steady State or Transient	Distribution or Moment
$p_i(t)$	$P[Z(t) = i]$	av	Z(t) : I	T	pmf
π_i	$\lim_{t\to\infty} P[Z(t) = i]$	av	Z(t) : I	S	pmf
$A(t)$	$\sum_{i \in up} p_i(t)$	av	X(t) : I	T	M
$A(\infty)$	$\lim_{t\to\infty} A(t)$	av	X(t) : I	S	M
Reliability	$P[X(\tau) \geq 1, \forall \tau \leq t]$	rel	X(t) : I	T	cdf
$E[\ X(t)\ ]$	$E[\ X(t)\ ]$	*all*	X(t) : I	T	M
$E[\ Y(t)\ ]$	$E[\ Y(t)\ ]$	Imp-rep	Y(t) : C	T	M
$\mathcal{Y}(x,t)$	$\mathcal{Y}(x,t)$	Imp-rep	Y(t) : C	T	cdf
$\mathcal{Y}(x,\infty)$	$\lim_{t\to\infty} \mathcal{Y}(x,t)$	Imp-rep	Y(t) : C	S	cdf
$P[A_I(t) \leq x]$	$P[W(t) \leq x]$	av	W(t) : C	T	cdf
$E[\ W(t)\ ]$	$E[\ W(t)\ ]$	compl	W(t) : C	T	M
$E[\ W(\infty)\ ]$	$E[\ W(\infty)\ ]$	compl	W(t) : C	S	M
$\mathcal{W}(x,t)$	$\mathcal{W}(x,t)$	compl	W(t) : C	T	cdf
$\mathcal{W}(x,\infty)$	$\lim_{t\to\infty} \mathcal{W}(x,t)$	compl	W(t) : C	S	cdf

Table 1. Measures and Their Characteristics

Measures used to characterize the behavior of Markov reward models of the multi-processor system with imperfect repair (without the completion property) are the reliability, $R(t)$, the distribution of accumulated reward (performability) over a finite interval, $\mathcal{Y}(x,t)$, and $\mathcal{Y}(x,\infty) \equiv \lim_{t\to\infty} \mathcal{Y}(x,t)$. On models with the completion property we use $\mathcal{W}(x,t)$, and $\mathcal{W}(x,\infty) \equiv \lim_{t\to\infty} \mathcal{W}(x,t)$. The effect of changes in the structure-state process, the reward structure and utilization interval on these measures of performability for MRMs of the multi-processor system are investigated in next two sections.

2 Multi-processor Model Description

We begin with a basic Markov reward model of the multi-processor system and then indicate a set of changes in the structure-state process and reward structure. The measures obtained for the various models of the multi-processor system are listed in Table 1. In the following section, each graph plots measures for a sequence of illustrative models.

Determining the way changes in the reward structure and the structure-state process affect measures of interest is crucial to using MRMs effectively in the system design process. Efforts to change system behavior in a favorable way must use the appropriate model and measure or they will be ineffective. For example, consider adding a repair facility to a high reliability non-repairable system (failure rate of $\sim 10^{-5}$). The steady-state behavior will change radically. However, if the utilization interval is short ($\sim$ 10 hours) then the repair facility will not substantially change the availability over the 10 hour interval. We wish to indicate some situations where the distribution of accumulated reward or its time average will indicate behavior not captured by other measures. We briefly describe the types of failure and repair behavior of the multi-processor system modeled with structure-state processes. The system consists of 16 processors, 16 memories, and an interconnection network (*i.e.*, crossbar switch) that allows a processor to access any memory. Since the system we analyze is similar to the Carnegie-Mellon multi-processor system, C.mmp, we use the failure data from that system. Siewiorek in [24] determined the failure rates per hour for the components to be:

$$\begin{array}{lccc} & \text{Processor} & \text{Memory} & \text{Switch} \\ \text{Failure Rates}: & \lambda = 0.0000689 & \gamma = 0.0002241 & \delta = 0.0002024 \end{array} .$$

Viewing the network as a single switch and modeling the system at the processor-memory-switch (PMS) level, we see that the interconnection network is essential for system operation. It is also clear that a minimum number of processors and memories are necessary for the system to be operational. We follow Siewiorek's choice of 4 processors, 4 memories and 1 interconnection network (switch) as the minimal operating configuration required for handling a task. Each state is specified by a triple (i, j, k) indicating the number of operational processors, memories, and networks, respectively. We augment the states with a non-operational state F. Events that decrease the number of operational components are associated with failure, and events that increase the number of operational elements are associated with repair. We assume that failures do not occur when the system is not operational. When a component of the multi-processor system fails, a recovery action must be taken (*e.g.*, shutting down a failed processor, so that it does not fill memories with spurious data), or the whole system will fail and enter state F. The probability that the recovery action is successfully completed is known as the coverage.

We consider two kinds of repair actions, global repair which restores the system to state $(16, 16, 1)$ with rate $\mu = 0.2$ per hour from state F and local repair, which can be thought of as a repair person beginning to fix a component of the system as soon as a component failure occurs. Our model of local repair assumes that there is only one repair person for each component type. We let the local repair rates per hour be:

$$\begin{array}{lccc} & \text{Processor} & \text{Memory} & \text{Switch} \\ \text{Local Repair Rates}: & \nu = 2.0 & \eta = 1.0 & \epsilon = 0.5 \end{array} .$$

A further refinement of the structure-state process can be made with respect to the interconnection network. Siewiorek in [25] notes that the C.mmp interconnection network is actually implemented as a set of 16 fan-out switches for each processor and memory port. In this case the failure rate of the interconnection system with respect to some operational configuration $(i, j, 1)$

is simply $\delta(i,j) = (i+j) *$ (fanout switch failure rate + line failure rate). Since the cause of failure is uniformly distributed over the fanout switches and their lines, we will simply let the failure rate associated with each fanout switch and line pair be 1/32 of the lumped failure rate of the switch. Thus $\delta(10,10) = 20 * 0.000006325 = 0.0001265$. We are pessimistic in that we assume that the failure of one fan-out switch and line brings the system to a non-operational state (*i.e.*, $(i,j,0)$). The single or "lumped" network with failure rate δ is more pessimistic than the "distributed" network with failure rate $\delta(i,j)$. A Markov model of the structure-state process for the C.mmp system with a "lumped" network and global repair has 170 states.

The two variations of the structure-state process we consider for the failure transitions are imperfect coverage (*i.e.*, leakage to state F), and the network failure rate ("lumped" or "distributed"). Local or global repair actions are the two kinds of repair strategies investigated. The substantial increase in model complexity that results from adding a local repair capability is evident in Figure 2, which depicts the structure-state process of a model with a "lumped" interconnection network, local and global repair, and imperfect coverage (365 states). The lower plane in Figure 2 contains the set of states where component exhaustion has occurred. Most of the states (169) in the lower level are the result of the interconnection network failing. Thirteen states represent system failure due to the exhaustion of operational memories and thirteen more states represent system failure due to the exhaustion of operational processors. The local repair models will include both local and global repair. When we speak of a model with only global repair, we set all local repair rates (ν, η, ϵ) in Figure 2 to zero and merge all non-operational states with state F. The structure-state processes of the MRMs of the multi-processor system thus can be characterized by their failure type (coverage), interconnection network type ("lumped" or "distributed"), and repair type (global or local) .

It remains to present the reward structures we use to characterize the performance behavior of the multi-processor system when it is in a given structure state. The simplest reward structure is obtained by dividing the structure states into two classes, operational and non-operational, and assigning the reward rate 1.0 to the operational states and 0 to the rest. A more accurate measure of system performance is more closely related to the system's ability to do useful work. Because memory is the slowest resource in the C.mmp system, the effectiveness of the system is limited by the number of available memories. Thus if there are more memories than processors, performance will still be limited by the memory bandwidth needed by the processors, while if there are more processors than memories the performance will be limited by the number of memories. A simple capacity-based performance model of an operational structure-state $(i,j,1)$ is to let the associated reward rate be $min\ \{i,j\}$. This performance model is optimistic because it does not consider processors contending for the memories.

When we consider contention for the memories, we use a model developed by Bhandarkar [2] to obtain the average number of busy memories or memory bandwidth. Bhandarkar found the average number of busy memories, and hence the reward rate in an operational state $(i,j,1)$ to be:

$$r_{i,j,1} = m(1-(1-1/m)^l), \tag{2}$$

where $l = min\{i,j\}$ and $m = max\{i,j\}$. We assign a zero reward rate to each non-operational state. Hence, in addition to a variety of structure-state processes we also have three reward structures of interest, the availability-based reward structure (0, 1), the capacity-based reward structure ($min\{i,j\}$), and the contention-based reward structure (equation (2)).

The initial state of the system in all our models will be (16,16,1) except in section 3.3 where $\underline{p}(0)$, the initial state probability vector, is equal to the steady-state probability vector, $\underline{\pi}$. The effect of changes in utilization-interval length, structure-state process, and reward structure for the multi-processor MRMs are examined in the next section.

3 Multi-processor Performability Results

3.1 The effects of coverage and utilization interval on $E[X(t)]$ and $E[W(t)]$, functions of $\underline{p}(t)$.

First, we use a sequence of models that illustrate the way the completion property affects $E[X(t)]$ and $E[W(t)]$ as a function of time in Figures 3 and 4, respectively. In both Figure 3 and Figure 4, we use our contention-based performance model to obtain the reward structure. $E[X(t)]$ is the expected instantaneous reward at time t and has been called the *computation availability* in [1]. This measure answers the question, "What is the expected performance of the system at time t?". $E[W(t)]$ is the expected time-averaged accumulated reward over the interval $(0,t)$. $E[W(t)]$ answers the question, "What is the time-averaged performance of the system over the interval $(0,t)$?". In Figures 3 and 4 we let curve I be a 'pure' performance model of the state $(16,16,1)$. The 'pure' performance model does not have any failures so the system performance is independent of time. With memory contention but no failure, the reward rate is 10.303 and both $E[X(t)]$ and $E[W(t)]$ are 10.303 for all time t. In curve II, only component failures occur ($c = 1$ for coverage), and we see that the expected performance level has been halved at time $t = 2000$. At time $t = 2000$ in Figure 4, the expected time-averaged accumulated reward has decreased by only one quarter because $E[W(t)]$ is the time average of $E[X(t)]$ over $(0,t)$. Thus $E[W(t)]$ is relatively insensitive, for large t, to the state of the system at a particular instant, $\tau < t$. Both Figure 3 and Figure 4 show the importance of the completion property. Models with the completion property (curves I, IV and V) strongly dominate those without it (curves II and III) indicating the value of global repair for long utilization intervals.

In curves III and V the coverage is reduced to 0.9. Curve III like curve II has no repair, and the expected performance level of curve III deteriorates more rapidly than that of curve II. Curve IV has only component failures ($c = 1$), and global repair as well. Consequently, the expected performance level of curve IV is much improved over that of curve II, especially for large

t. One might expect curve IV to dominate curve V, which uses the same model as curve IV with $c = 0.9$, just as curve II dominates curve III. However, for large t it is better on the average to experience a coverage failure and rapidly return to the highest reward state $(16, 16, 1)$ rather than spend a long interval in the relatively low reward states before returning to structure state $(16, 16, 1)$. Both of these measures indicate the importance of global repair for longer time intervals.

Unfortunately $E[X(t)]$ and $E[W(t)]$ do not address the likelihood of completing a given amount of work in a specified interval. $E[W(t)]$ merely gives an indication of the average behavior over a utilization interval. We use $\mathcal{Y}^C(x,t)$ to examine the behavior of a non-repairable system over different length utilization intervals in the next section.

3.2 The effect of utilization interval on $\mathcal{Y}^C(x,t)$ for non-repairable models.

We consider a model of the C.mmp system with a "lumped" interconnection network, $c = 0.90$ for the coverage, and no repair in Figure 5. The CTMC of the structure-state process is depicted in Figure 2 with all repair rates μ, ν, η, ϵ set equal to zero. The reward structure is based on the contention-based performance model. Curves I, II, III, IV plot the value of $\mathcal{Y}^C(x,t)$ for $t = 100$, 1000, 10000, and ∞ respectively.

Loosely speaking, $\mathcal{Y}^C(x,t)$ answers the question, " What is the probability that x units of work is completed by time t?" Because the model does not have the completion property, $\mathcal{Y}^C(x,t)$ is substantially less than 1.0 for moderate amounts of accumulated reward even if $t \to \infty$. It is interesting to note that $\mathcal{Y}^C(x,t)$ for moderate t only falls below $\lim_{t\to\infty} \mathcal{Y}(x,t)$ as $x \to \sim 9t$.

The non-repairable system performs near its asymptotic limit, $\mathcal{Y}^C(x,\infty)$ for moderate t. However, systems that satisfy the completion property will complete any finite amount of work in an arbitrarily long utilization interval. When comparing different systems for the same utilization interval, $\mathcal{Y}^C(x,t)$ is quite satisfactory, whether the system satisfies the completion property or not. If we wish to compare the behavior of systems that satisfy the completion property over different utilization intervals, then we need to normalize the curves of the different complementary distributions of accumulated reward so that they can be compared over the same interval. The natural approach is to time average the accumulated reward and use $W(t)$ as the random variable rather than $Y(t)$. In the next section we examine the behavior of a system that satisfies the completion property over different utilization intervals. The results are rather surprising.

3.3 The effect of utilization interval on $\mathcal{W}^C(x,t)$ for models with the completion property.

As noted in section 3.1, both $E[X(t)]$ and $E[W(t)]$ are functions of the instantaneous probability vector, $\underline{p}(t)$. If we let the initial probability vector, $\underline{p}(0)$, of the system equal the steady-state probability vector, $\underline{\pi}$, then neither $E[X(t)]$ nor $E[W(t)]$ will change since then $\underline{p}(t) = \underline{\pi}$ for all t. We show the presence of behavior not detected by these measures in Figure 6. In Figure 6 we use a structure-state process with global repair and coverage $= 0.95$ to model the failure and repair activity of the C.mmp system and the contention-based performance model to obtain the reward structure. The measure $\mathcal{W}^C(x,t)$ can be used to answer the question, "What is the probability that the reward accumulated in the interval $(0,t)$ is at least xt?"

We examine the distribution of time-averaged memory bandwidth (performance) for utilization intervals of length 10, 100, 1000, and 10000 in curves I, II, III and IV of Figure 6. We indicate the steady-state expected reward rate, $\sum_i \pi_i r_i$, with a vertical line labeled V (+ + +). We can see the way the curve smooths out and approaches a jump at the steady-state, time-averaged reward rate as t increases. The dynamic behavior of the system in steady state is indicated in Figure 6. Measures such as $E[X(t)]$ and $E[W(t)]$ are unable to capture the steady-state system dynamics since both these measures are invariant with respect to time for the Markov reward model with $\underline{p}(0) = \underline{\pi}$.

3.4 The effect of reward structure, and model "family" on $\mathcal{W}^C(x,t)$ for models with the completion property.

Insight into the way the structure-state process and the reward structure affect the ability of the multi-processor system to complete a fixed amount of work in a given time interval $(0,t)$ is obtained from the complementary distribution of time-averaged accumulated reward. We plot the complementary distribution of time-averaged accumulated reward (in this case the time-averaged memory bandwidth) for a basic Markov reward model with a "lumped" interconnection network, perfect coverage ($c = 1$), and global repair. We use "pure" performance models to provide an optimistic upper bound for MRMs comparing the capacity-based and contention-based reward structures resulting from the different performance assumptions about the way memory is accessed. We examine the distribution of $W(t)$ and the distribution of the interval availability, $A_I(t)$, in Figure 7. First we consider the system without failure and repair in curves I and II. The result of this modeling assumption is that no degradation of performance takes place and the state of the system is always $(16,16,1)$. Consequently, curves I and II of the complementary distribution of time-averaged accumulated reward are step functions. If we ignore memory contention, then there are 16 processors and 16 memories and the memory bandwidth is 16. It follows that the system performance level (reward rate) is constant and $W(t) = 16$. Curve I in Figure 7 depicts this unit step form of the complementary distribution of time-averaged accumulated reward. For curve II, we assume that there is contention at the memories. The result of modeling the contention is to lower the ability of the system to deliver useful work. Therefore, the step for curve II occurs at a smaller value of accumulated reward per unit time than the step for curve I. We use the work of Bhandarkar [2] to estimate the effect of contention on the performance of state $(16, 16, 1)$. Hence, with memory contention but no failure, the reward rate is 10.303 for all time t. Thus, in curve II the complementary distribution of $W(t)$ is a unit step, though at 10.303 instead of 16 as in curve I.

In curves III and IV, we examine the effect of modeling failure and repair on the complementary distribution of time-averaged accumulated memory bandwidth. For curve III, we assume there is no memory contention and use the capacity-based performance model for each structure state in Figure 2 (assuming no local repair and merging all non-operational states in F). The performance (reward rate or memory bandwidth) of operational state $(i, j, 1)$ is set to $min\ \{i, j\}$. The gap between curve I and curve III reflects the fact that when failure and repair are taken into account, memory bandwidth varies with time thus lowering the time-averaged accumulated memory bandwidth. Without the occurrence of failures, the memory bandwidth stays constant at 16. Another way of stating the situation is to say that curve III will asymptotically approach curve I as the maximum of all the failure rates tends to zero. Curve IV has a similar relationship to curve II. In curve IV, we use our most detailed performance model, and take into account failure and repair. Thus the performance level (reward rate) of each operational state $(i, j, 1)$ is determined by equation (2). Because the performance degradation due to component failure is smaller with Bhandarkar's performance estimates than with the capacity-based performance estimates, curve IV more closely approaches curve II than curve III approaches curve I. The relationship of the 4 curves discussed indicates that the performance model assumptions show an upper limit of the system's ability to complete work. The magnitude of the failure and repair rates effect the rate at which the complementary distribution of time-averaged memory bandwidth declines below the step function defined by the performance model.

We see that 'pure' performance models overestimate the ability of the system to complete useful work. For example, using curve IV, we see that the probability that the time-average memory bandwidth is greater than or equal to 9.5 is 0.989, whereas using the 'pure' performance-based model of curve II, this probability is 1. It is also true that 'pure' failure/repair (availability) models in which the reward rates for operational states are set to 1.0 and non-operational states are assigned reward 0.0 underestimate the ability of a system to complete useful work when the performance levels are scaled in such way that the minimum reward operation state has a reward rate ≥ 1.0. Using this reward structure, $W(t)$ is the interval availability $A_I(t)$. To complete the set of reward structures considered for performability models of the multi-processor system, with curve V we display the complementary distribution of interval availability. We see that curve V is nearly a step function at 1.0 because only a network failure will cause the system to immediately enter state F (13 processor or 13 memory failures must occur before the system will enter state F).

3.5 The effect of coverage and utilization interval on $\mathcal{W}^C(x,t)$ for models with the completion property.

In this section we continue examining a model of the multi-processor system with a "lumped" interconnection network, global repair, and different coverage values. We will use the most accurate performance model, namely Bhandarkar's, to obtain the reward structure for the operational states. In Figure 8 we show the effect of coverage and of the observation period on the chosen measure of effectiveness. As $t \rightarrow 0$, independent of c, $W(t)$ approaches the 'pure' performance behavior shown in curve I, a step at 10.303. Curves II, III, and IV ($c = 1.0$, 0.95, and 0.90, respectively) show that for larger observation intervals ($t = 100$), the higher coverage curves dominate the lower coverage curves illustrating the effect of coverage on the complementary distribution of time-averaged accumulated reward. In curves V-VII of Figure 8, we plot the steady-state computation availability, $\lim_{t\rightarrow\infty} E[W(t)] = \sum_i r_i \pi_i$, for the different coverage values where π_i is the steady state probability of being in state i. We can see that as the length of the utilization interval increases the probability of accumulating a given amount of reward becomes more pessimistic. One cause of this effect is that repair takes place only when the whole system has become inoperable and the failure rates are small enough to make the occurrence of more than one failure in a relatively small interval (100 hours) extremely unlikely. States in which a significant number of failures have occurred become more likely as time passes. Allowing repair only when the system has failed yields the relative position of curves V-VII ($c = 1.0$, 0.95, and 0.90, respectively). As the coverage probability decreases, the steady-state computation availability actually increases!

The anomalous behavior of the steady-state computation availability is caused by several factors: the disparity in the reward rates of the operational states; the relatively large global repair rate in relation to failure rates; and the assumption that the global repair rate is independent of the number of failed components. If we set the reward rates for all the operational states to be 1.0 and 0 otherwise, then the availability ($\sum_i r_i \pi_i$) decreases as the coverage decreases (the anomaly disappears). Similarly, the anomaly disappears if we make the global repair rate comparable to the failure rates or make it dependent on the actual number of components that have failed. Also, local repair causes the anomaly to disappear. The point is that extrapolating from steady-state values and expected values can be misleading.

3.6 The effect of interconnection network type and repair capabilities on $\mathcal{W}^C(x,t)$ for models with the completion property.

We examine the effect of adding a local repair facility for each component type to the multi-processor system in this section. In Figure 9 we obtain $\mathcal{W}^C(x,t)$ for two pairs of models with a utilization interval of 100 hours. In curve I we plot $\mathcal{W}^C(x,t)$ for a model of the multi-processor system with both global and local repair, $c = 0.90$ for coverage and a distributed interconnection network (+ + +). The model used to obtain curve II (solid) is the same as that used for curve I except for a "lumped" instead of distributed interconnection network. The effect of the slightly lower failure rate of the distributed interconnection network is small but discernible.

Curve III is the plot of $\mathcal{W}^C(x,t)$ for the same model as curve I without local repair. Replacing the distributed interconnection network in curve III with the slightly more failure prone "lumped " interconnection network model produces Curve IV. The difference between curves III and IV is also quite small. The effect of the local repair facility is sizable for time-averaged memory bandwidth requirements greater than 9.2. This result indicates the value of local repair for the multi-processor system over even moderately sized utilization intervals. Another way of expressing the situation is to observe that as the time-averaged workload requirement increases, the size of the utilization interval where the local repair facility will substantially increase $\mathcal{W}^C(x,t)$ becomes smaller. Roughly speaking, we can conclude that local repair is worthwhile for systems expected to operate at nearly full capacity (maximum reward rate), even if the utilization interval is of only moderate size.

4 Conclusion

The ability to determine the distribution of accumulated reward and its time average for moderate size problems is a recent development. We presented a systematic study of a complex multi-processor system and an $O(n^3)$ algorithm for the computation of the distribution of accumulated reward of general Markov reward models.

The study of Markov reward models of the multi-processor system points to a number of interesting facts about different performability measures. The first three examples indicate that instantaneous measures do not show the dynamic behavior of the system while $\mathcal{W}^C(x,t)$ does. The next two examples show that steady-state values are deceptive in some circumstances, and the final example indicates the importance a local repair facility may have on the distribution of performability for moderate-size utilization intervals. Furthermore the study indicates how changes in the failure/repair behavior of the system such as the interconnection network failure rate, repair strategy, and coverage probability affect the complementary distribution of accumulated reward. We also examine the way changing the reward structure and utilization interval affects the distribution of time-averaged accumulated reward. Thus some inadequacies of steady-state values and expected values are illustrated and an examination of how changes in Markov reward models effect the performability distribution is made. The new algorithm presented in the paper can thus aid the system designer in exploring detailed dynamic behavior of multi-processor systems.

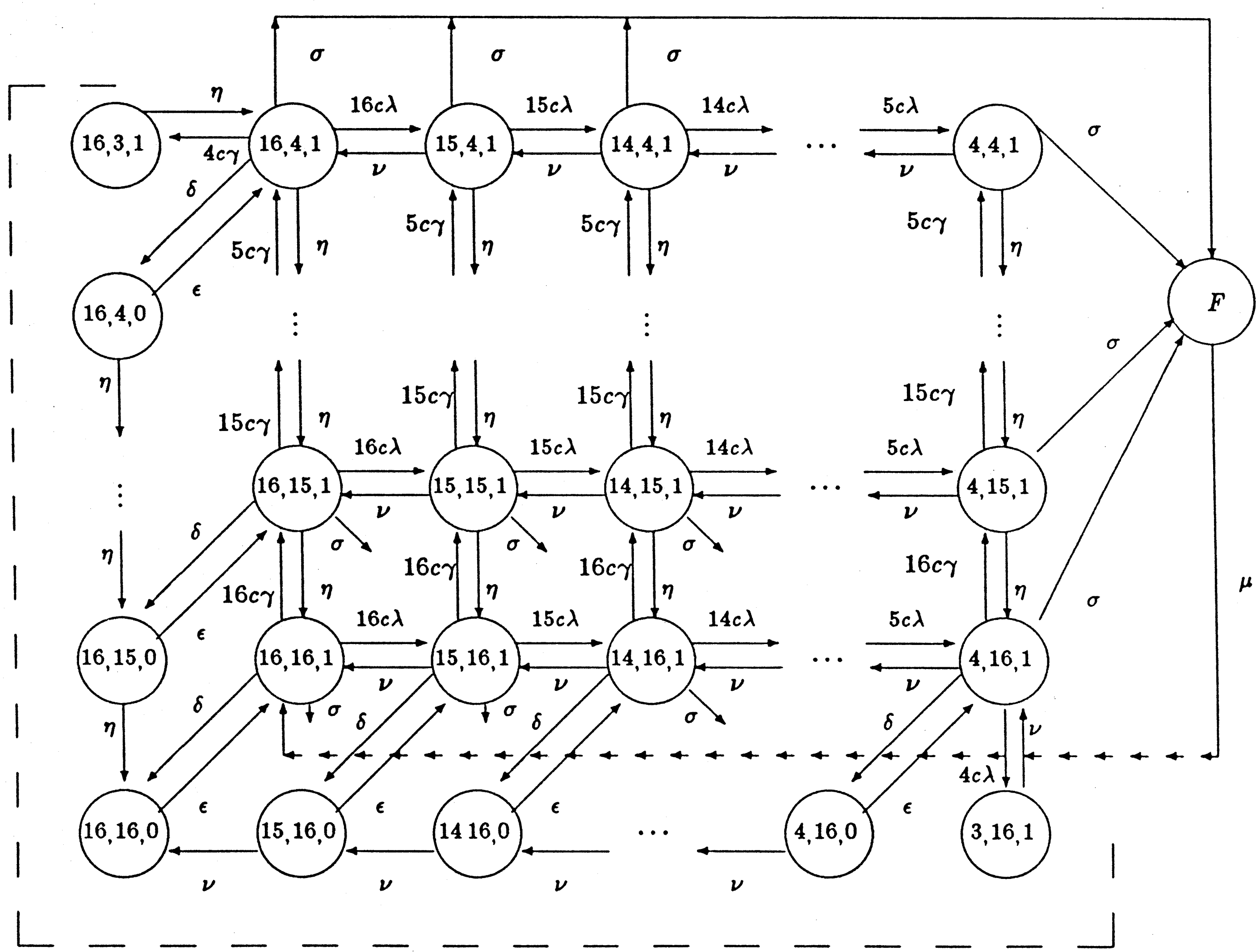


Figure 2: Markov Chain for the Multi-processor System (each state (i, j, k) has a transition with rate $\sigma = ((1-c)(i\lambda + j\gamma))$ to state F).

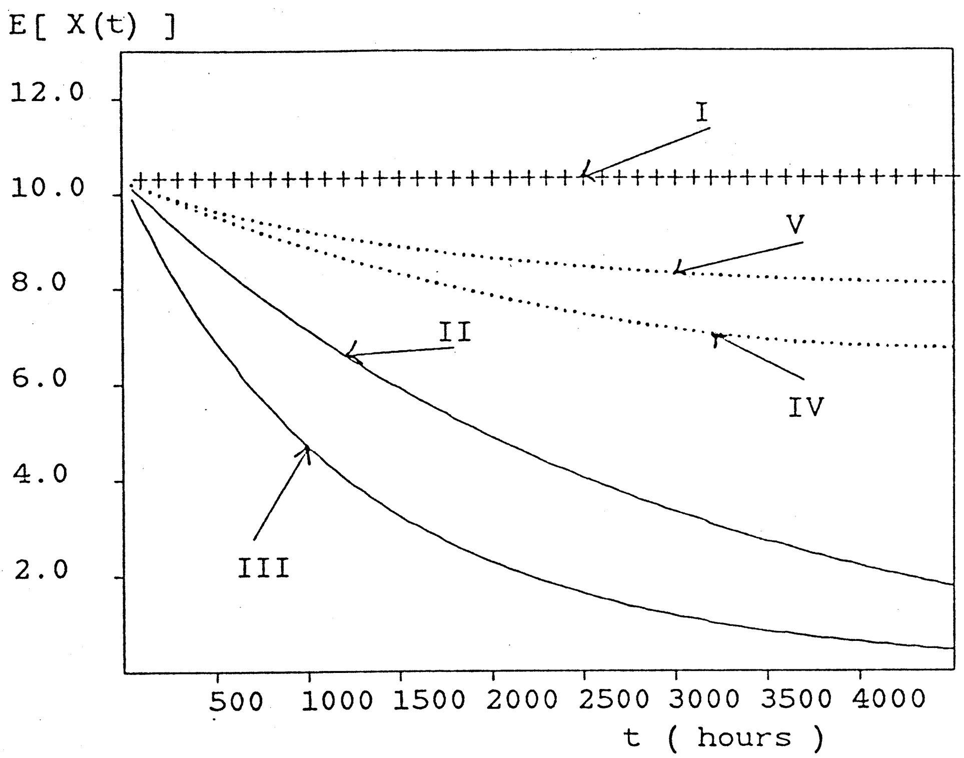


Figure 3: Expected Instantaneous Memory Bandwidth $E[X(t)]$ Vs. time t.

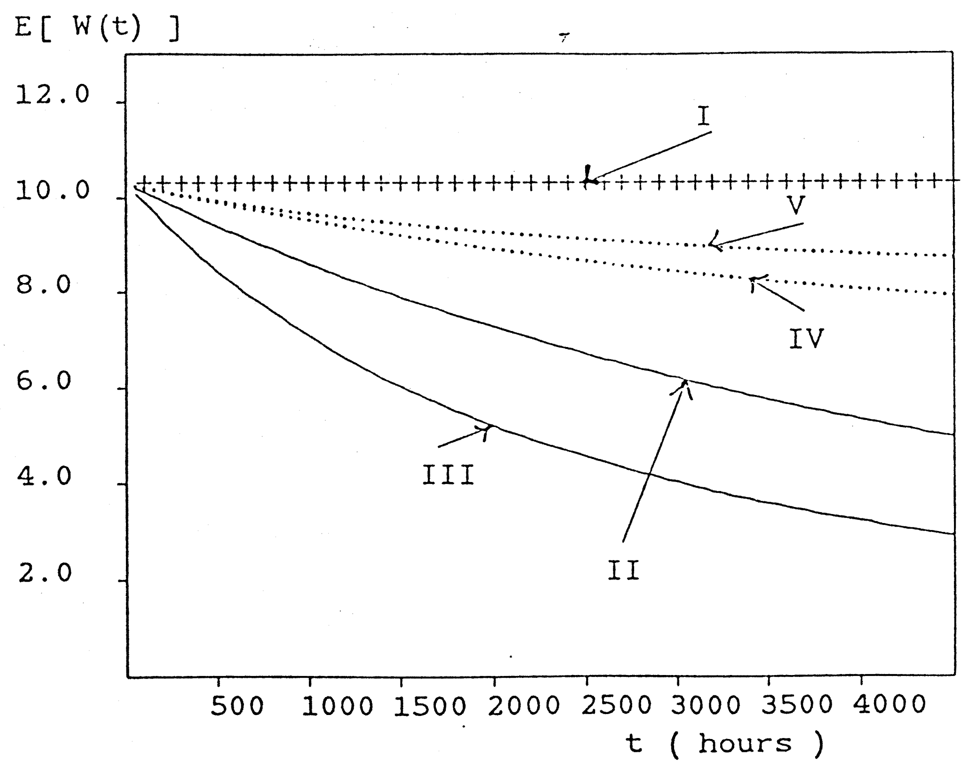


Figure 4: Time-Averaged Expected Accumulated Bandwidth $E[W(t)]$ Vs. time t.

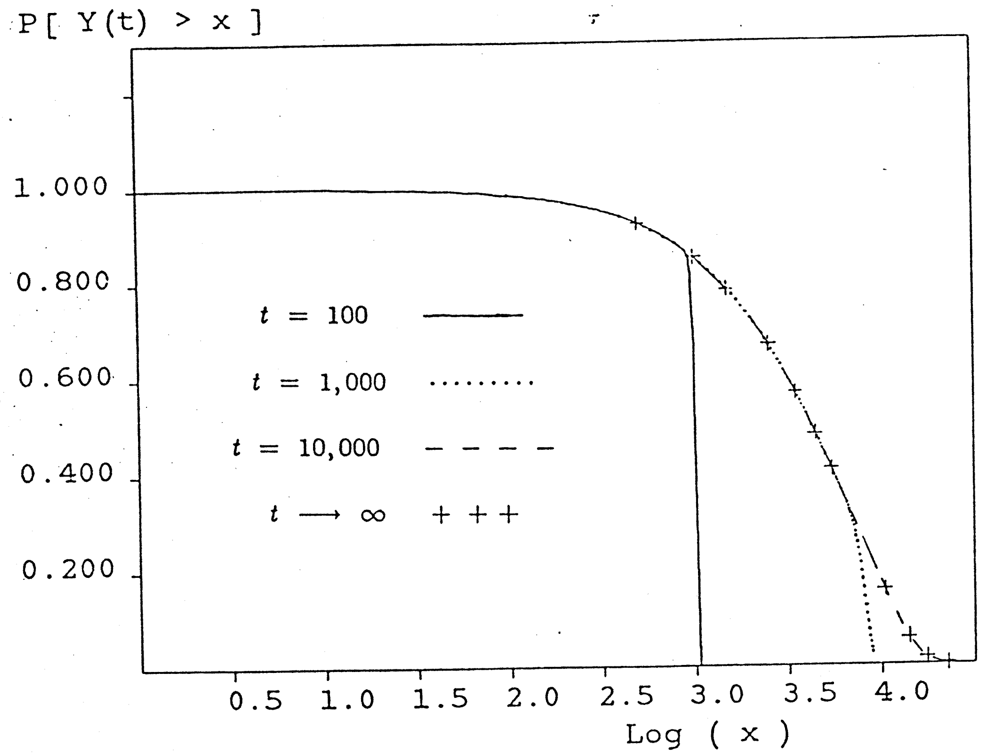


Figure 5: Complementary Distribution of Accumulated Bandwidth $\mathcal{Y}^C(x,t)$ for Different t Values.

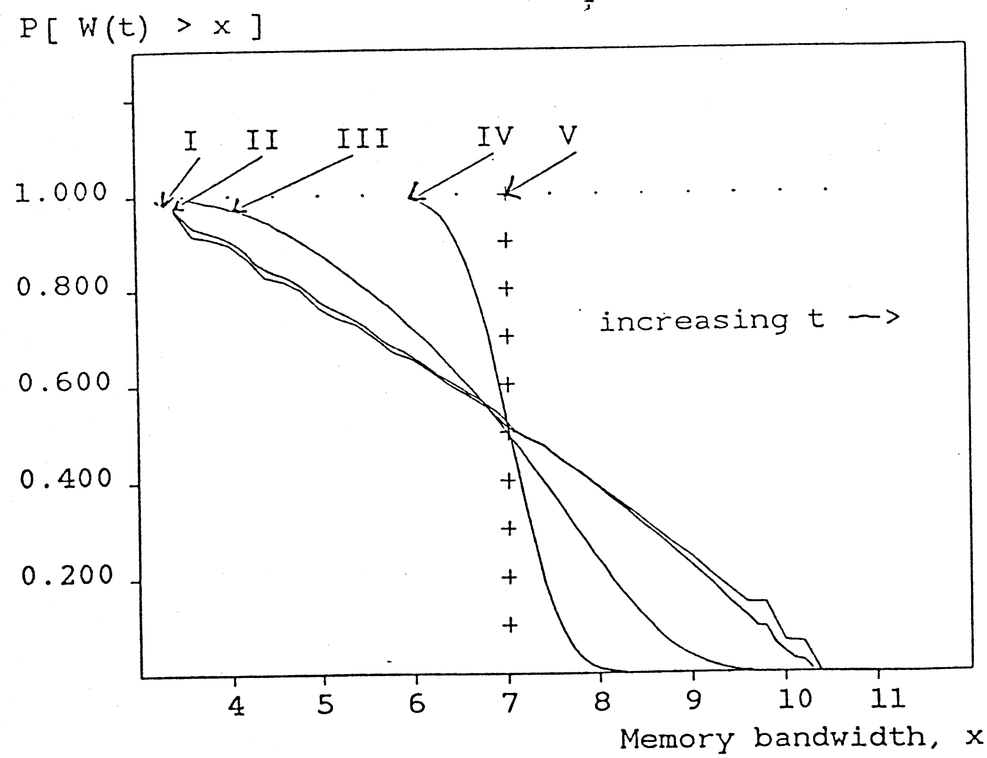


Figure 6: Complementary Distribution of Time-Averaged Accumulated Bandwidth $\mathcal{W}^C(x,t)$ for Different t Values.

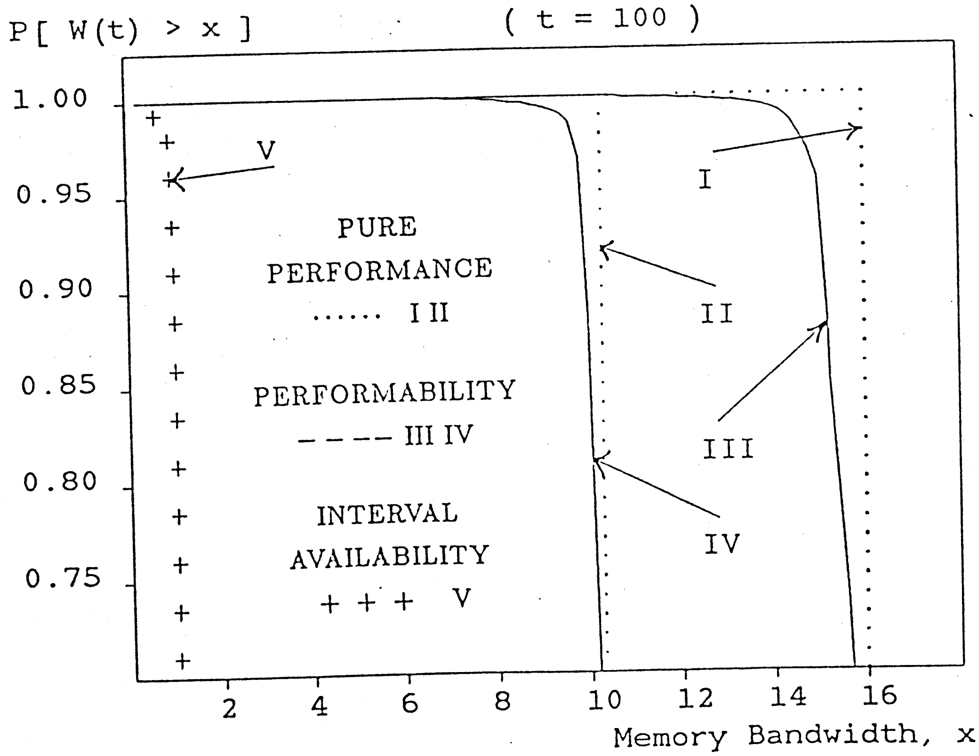


Figure 7: Complementary Distribution of Time-Averaged Accumulated Bandwidth $\mathcal{W}^C(x,t)$ for Different Reward Structures.

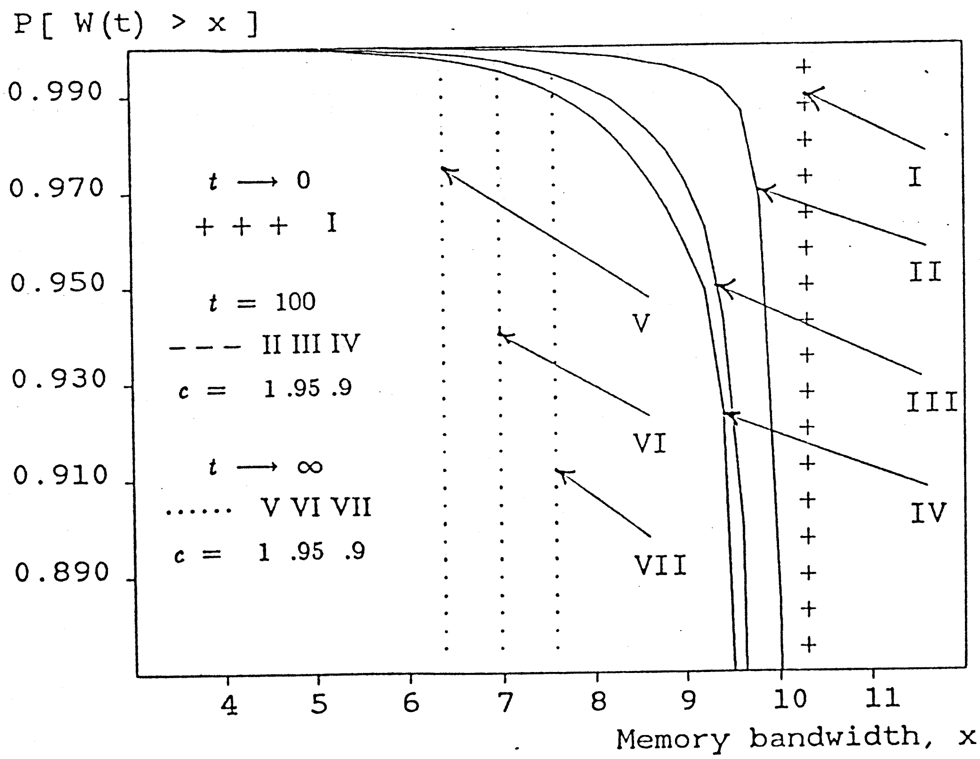


Figure 8: Complementary Distribution of Time-Averaged Accumulated Bandwidth $\mathcal{W}^C(x,t)$ for Different Coverage Values.

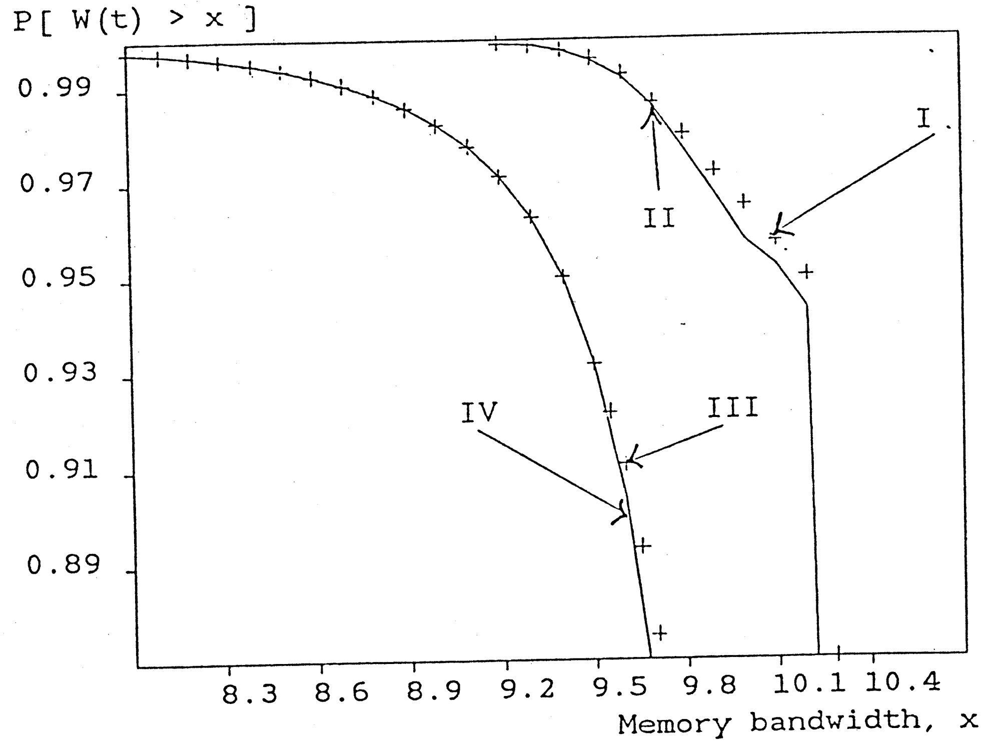


Figure 9: Complementary Distribution of Time-Averaged Accumulated Bandwidth $\mathcal{W}^C(x,t)$ for Different Networks and Different Repair Policies.

A An Algorithm and Its Analysis

In this section, we detail the double-transform inversion method for the distribution of accumulated reward. We begin with:

$$\mathcal{Y}_i(x,t) \equiv P[\, Y(t) \le x \mid Z(0) = i \,].$$

First, we apply the LST, (*i.e.*, $\int_0^\infty e^{-ux} d\mathcal{Y}_i(x,t)$) , signified by $\sim$, to $\mathcal{Y}_i(x,t)$ with respect to the work requirement x (transform variable u), and then apply the Laplace Transform signified by $*$ with respect to time t (transform variable s). The following linear system has been derived for $\underline{\mathcal{Y}}^{\sim *}(u,s)$ in [22] and [18]:

$$(sI + uR - Q)\ \underline{\mathcal{Y}}^{\sim *}(u,s) \;=\; \underline{e}\,. \tag{3}$$

The matrix of reward rates is $R = diag\ [r_1,\ r_2,\ \ldots,\ r_i,\ \ldots,\ r_n]$, Q is the generator matrix of the CTMC, and $\underline{e}$ is a column vector of size n with all elements equal to 1.

Using Cramer's rule, we can see that $\mathcal{Y}_i^{\sim *}(u,s)$ is a rational function in s . Hence, it has a partial fraction expansion:

$$\mathcal{Y}_i^{\sim *}(u,s) \;=\; \sum_{j=1}^{d}\sum_{k=1}^{m_j} a_{ijk}(u)(\, s - \lambda_j(u)\,)^{-k} \tag{4}$$

where the $\lambda_j(u)$, 1, 2, ..., j, ..., d are the d distinct eigenvalues of $[\,Q \;-\; uR\,]$, each with algebraic multiplicity m_j. The QR algorithm [27] is used to numerically determine eigenvalues of $[Q - uR]$ in $O(n^3)$ time. Using (4) we can invert analytically with respect to s and obtain:

$$\mathcal{Y}_i^{\sim}(u,t) \;=\; \sum_{j=1}^{d}\sum_{k=1}^{m_j} \frac{a_{ijk}(u)}{(k-1)!} t^{k-1} e^{\lambda_j(u)t}\,.$$

We define the column vector $\underline{A_i}$ by:

$$\underline{A_i} = \begin{vmatrix} a_{i11} \\ a_{i12} \\ \vdots \\ a_{i1m_1} \\ \vdots \\ a_{idm_d} \end{vmatrix}$$

and the row vector $\underline{E}^T(s)$ by:

$$\underline{E}^T(s) = \left| (s-\lambda_1(u))^{-1} \quad \cdots \quad (s-\lambda_1(u))^{-m_1} \quad \cdots \quad (s-\lambda_d(u))^{-m_d} \right| .$$

Now equation (4) can be written in vector notation as:

$$\mathcal{Y}_i^{\sim *}(u,s) \;=\; \underline{E}^T(s)\underline{A_i} \;.$$

In order to determine $\underline{A_i}$ we need n linearly independent equations. For this purpose we choose n distinct values of s, denoted by $s_1,\ s_2,\ \ldots,\ s_n$ sufficiently separated from the eigenvalues of $[Q-uR]$. The matrix E is constructed from the eigenvalues of $[Q-uR]$ and the n values of s:

$$E = \begin{vmatrix} \underline{E}^T(s_1) \\ \underline{E}^T(s_2) \\ \vdots \\ \underline{E}^T(s_n) \end{vmatrix} .$$

A way of choosing the n values of s so that E has a reasonably small condition number is to choose s_j so that the j^{th} element of$\underline{E}^T$ $(s_j) \sim 1.0$. This causes the diagonal elements of E to be reasonably large, although it does not guarantee E is non-singular. If E is found to be singular, then a new s value can easily be chosen at the time. We then solve the linear system

$$E\underline{A_i} \;=\; \underline{\mathcal{Y}_i}^{\sim *}(u,\underline{s}) \;=\; \begin{vmatrix} \mathcal{Y}_i^{\sim *}(u,s_1) \\ \mathcal{Y}_i^{\sim *}(u,s_2) \\ \vdots \\ \mathcal{Y}_i^{\sim *}(u,s_n) \end{vmatrix} \tag{5}$$

for the unknown vector $\underline{A_i}$ once the right-hand side has been determined. Since the problems we consider are of small size (for the solution of a linear system) we use a direct method (LU factorization) on E. The $O(n^3)$ LU factorization must be done only once and an $O(n^2)$ backsolve must be done for each of the n possible right-hand sides. Thus we may solve $E\,\underline{A_i} \;=\; \underline{\mathcal{Y}_i}^{\sim *}(u,\underline{s})$ for all n values of i for a cost that is $O(n^3)$. The practical implication of this fact is that the $\underline{\mathcal{Y}}^{\sim}(u,t)$ may be obtained at $O(n^3)$ cost if the n different right-hand sides can be obtained at an $O(n^3)$ cost as well.

Since we are interested in the n vectors of partial fraction coefficients $\underline{A_i}$, $1 \le i \le n$, let us define

$$A = [\ \underline{A_1}\ \underline{A_2}\ \cdots\ \underline{A_n}\ ] \quad \text{and} \quad Y = [\ \underline{\mathcal{Y}_1}^{\sim *}(u,\underline{s})\ \underline{\mathcal{Y}_2}^{\sim *}(u,\underline{s})\ \ \cdots\ \underline{\mathcal{Y}_n}^{\sim *}(u,\underline{s})\ ] \;.$$

Hence the problem of determining the n^2 partial fraction coefficients can be written in matrix form as:

$$EA = Y \quad . \tag{6}$$

Because of the economical way direct methods handle multiple right-hand sides we need only address the problem of determining all the $\underline{\mathcal{Y}_i}^{\sim *}(u, \underline{s})$ that make up Y in $O(n^3)$ time.

We first transform the linear system (3) to a simpler one. Any matrix can be put into upper Hessenberg form using a sequence of Householder unitary similarity transformations [27]. Therefore we can write

$$U^{\dagger} = U^{-1} \quad \text{and} \quad U^{\dagger}(Q - uR)U = H \quad ,$$

where † denotes conjugate transpose and H is an upper Hessenberg matrix. By making the transformation $U^{\dagger}\underline{\mathcal{Y}^{\sim *}}(u, s) = \underline{\mathcal{M}^{\sim *}}(u, s)$ the linear system (3) can be rewritten as

$$(sI - H)\underline{\mathcal{M}^{\sim *}}(u, s) = U^{\dagger}\underline{e}$$

This upper Hessenberg linear system requires only $O(n^2)$ time to determine $\underline{\mathcal{M}^{\sim *}}(u, s)$. $\underline{\mathcal{Y}^{\sim *}}(u, s)$ can be regained from $\underline{\mathcal{M}^{\sim *}}(u, s)$ by a matrix vector product:

$$\underline{\mathcal{Y}^{\sim *}}(u, s) = U\underline{\mathcal{M}^{\sim *}}(u, s) \quad ,$$

which also costs $O(n^2)$. An important observation here is that the required sequence of unitary transformations (U) and the matrix H are already available from the QR algorithm that solves the eigenvalue problem for $(Q - uR)$ and hence does not add any cost. Thus the complexity to obtain $\underline{\mathcal{Y}^{\sim *}}(u, s)$ is now only $O(n^2)$ for every value of s for each u.

It remains to invert $\mathcal{Y}_i^{\sim}(u, t)$ with respect to u. A number of methods to numerically invert the Laplace transform have been developed. Orthogonal polynomials [26] and Fourier series [4] [5] have been the most commonly used tools for inverting the Laplace Transform. To avoid unnecessary notational complexity we define $V(u) \equiv \mathcal{Y}_i^{\sim}(u, t)/u \equiv \mathcal{Y}_i^{*}(u, t)$ and to follow standard notation let $i = \sqrt{-1}$ in the next two equations. We employ the following method to numerically obtain $v(x)$, the inverse Laplace Transform of $V(u)$ using the well known complex inversion formula

$$v(x) = \int_{a-i\infty}^{a+i\infty} e^{ux} V(u) du \; = \frac{e^{ax}}{\pi} \int_0^{\infty} \Re\{V(u)e^{i\omega x}\} d\omega$$

where $u = a + i\omega$. If the above integral is now discretized using the trapezoidal rule with step size π/T, the following Fourier series approximation $\hat{v}(x)$, of period $2T$, is obtained:

$$\hat{v}(x) = \frac{e^{ax}}{T}[\frac{V(a)}{2} + \sum_{k=1}^{\infty}\{\Re(V(a + \frac{k\pi i}{T}))cos(\frac{k\pi x}{T}) - \Im(V(a + \frac{k\pi i}{T}))sin(\frac{k\pi x}{T})\}] . \tag{7}$$

The discretization error declines exponentially as aT increases [7]:

$$\hat{v}(x) - v(x) = \sum_{k=1}^{\infty} e^{-2kaT} v(2kT + x); \quad 0 \leq x \leq 2T \; .$$

Since $v(x) \leq 1.0 \; \forall x$, the discretization error is easily made very small. Therefore, the bulk of the error in the numerical inversion procedure accrues from truncating the Fourier series. The Fourier series exhibits characteristically slow convergence. However, acceleration methods that allow accurate estimates of the series from the first m terms are known. We use the quotient-difference algorithm of Rutishauser with a remainder estimate suggested by DeHoog *et al.* in [5] to accelerate the convergence of the Fourier series. Cooley *et al.* in [4] use the cosine transform to approximate a series very similar to (7), and Jagerman [17] obtains an expression similar in form to (7) by considering the generating function of a sequence of functionals that converge in the limit to $v(x)$. Because of the $O(n^3)$ cost of computing each function value, the method that reliably yields accurate results with the fewest evaluations is best. We have been pleased with the results obtained when the Fourier series is evaluated to the first $m = 80$ terms with the DeHoog remainder estimate (even when the desired distribution has jumps at various values of x). The structure of the overall algorithm is as follows:

```
A:  Determine Y~(u,t)
for( m values of u) {
   determine the eigenvalues of (uR - Q)                      O(n^3)
   for( d unique eigenvalues of (uR - Q) ) {
      solve transformed Hessenberg system                     O(n^2)
   }
    evaluate partial fraction coefficients                    O(n^3)
}
B:  Numerical Laplace Transform Inversion
for( n states ) {
  for( p desired values of t ) {
     for( m values of u ) {
        sum partial fraction coefficients to evaluate V(u)    O(n)
     }
     for(q values of x) {
        sum Fourier series approximation to evaluate v(x)     O(m)
     }
  }
}
```

In the worst case, the inner loop of phase A of the computation is executed $O(n)$ times. Since each iteration of the inner loop has a computational cost of $O(n^2)$, phase A has a computational complexity of $O(mn^3)$. The computational cost of phase B is primarily a function of the p different values of time t at which $\mathcal{Y}(x,t)$ is to be evaluated and the m terms in the Fourier series approximation. Phase B has a computational complexity of

$O(\, pmn \, (n+q) \,)$. Therefore, phase A comprises the principal computational burden of the algorithm. The total computational effort to obtain $\mathcal{Y}(x,t)$ for q values of x at each of p values of t is $O(pmn(n+q)+mn^3)$. The practical implication is that once the computationally expensive phase A has been used to determine $\mathcal{Y}^{\sim}(u,t)$, evaluating $\mathcal{Y}(x,t)$ at other (x,t) points can be done very cheaply.

Often the constants that are brushed under the rug by the $O()$ notation are important. The computational cost of the algorithm is approximately $16m(1+\alpha)n^3$ where α is a difficulty factor for the QR algorithm that depends on the spectrum of $(Q-uR)$. Since for most matrices $1 \leq \alpha \leq 2$, the computational cost should be between $32mn^3$ and $48mn^3$. We present in Table 2 the operation counts (flops) and approximate computation times for determining $\mathcal{Y}_i(x,t)$ on a CONVEX C-1 XP. The operation count values are the median of a small sample, and the time values are the maximum of the same small sample. The order estimates of the previous paragraph indicate the importance of n to the asymptotic behavior of the computation time. Consequently, we fix the number of terms in the Fourier series expansion m at 80 and the number of time values p at 1. The number of values of x (amounts of accumulated reward) is fixed at 100. These are typical values we used to examine $\mathcal{Y}_i(x,t)$ for the examples in this paper.

	$n \rightarrow$ 4	10	40	170	365
flops	3.2x10^5	4.0x10^6	1.8x10^8	1.1x10^{10}	1.0x10^{11}
time	6 sec.	15 sec.	320 sec.	3.1 hr.	25 hr.

Table 2. Computation Time and Flops with Different Values of n

The effect of the $pmn(n+q)$ term is unimportant when $p < n$ and $pq < n^2$. Therefore, the increase in flops becomes approximately n^3 for values of $n \geq 6$. The computation time for the larger state problems is approximate because the jobs were run at a low priority and the CONVEX C-1 was not dedicated to solving these problems.

References

[1] Beaudry, M., "Performance Related Reliability for Computer Systems," *IEEE Trans. Comput.,* Vol. C-27, No. 6, June 1978, 540-547.

[2] Bhandarkar, D., "Analysis of Memory Interference in Multiprocessors," *IEEE Trans. Comput.,* Vol. C-24, No. 11, November 1975, 897-908.

[3] Ciciani, B. and Grassi, V., "Performability Evaluation of Fault-Tolerant Satellite Systems," *IEEE Trans. Commun.,* Vol. COM-35, No. 4, April 1987, 403-409.

[4] Cooley, J., Lewis, A. and Welch, P., "The Fast Fourier Transform: Programming Considerations in the Calculation of the Sine, Cosine and Laplace Transforms," *J. Sound. Vib.*, Vol.12, 1970, 315-337.

[5] DeHoog, F., Knight, J. and Stokes, A. N., "An Improved Method for Numerical Inversion of Laplace Transforms," *SIAM J. Sci. & Stat. Comput.* , Vol.3, No.3, 1983, 357-366.

[6] Donatiello, L. and Iyer, B., "Analysis of a Composite Performance Reliability Measure for Fault-Tolerant Systems," *JACM*, Vol. 34, No. 1, 1987, 179-199.

[7] Durbin, F., "Numerical Inversion of Laplace Transforms: An Efficient Improvement to Dubner and Abate's Method," *Comput. J.*, 17, 1974, 371-376.

[8] de Souza e Silva, E. and Gail R., "Calculating Cumulative Operational Time Distributions of Repairable Computer Systems," *IEEE Trans. Comput.*, Vol. C-35, No. 4, April 1986, 322-332.

[9] Furchtgott, D. and Meyer, J., "A Performability Solution Method for Degradable Nonrepairable Systems," *IEEE Trans. Comput.*, Vol. C-33, No.6, June 1984, 550-554.

[10] Goyal, A. and Tantawi, A., "Evaluation of Performability for Degradable Computer Systems," *IEEE Trans. Comput.*, Vol. C-36, No. 6, June 1987, 738-744.

[11] Goyal, A., Tantawi, A. N. and Trivedi, K. S., "A Measure of Guaranteed Availability," IBM Research Report RC 11341, August 1985.

[12] Hordijk, A., Iglehart, D. and Schassberger, R., "Discrete Time Methods for Simulating Continuous Time Markov Chains," *Adv. Appl. Prob.*, Vol. 8, 1976, 772-788.

[13] Howard, R. A., *Dynamic Probabilistic Systems, Vol.II: Semi-Markov and Decision Processes,* John Wiley and Sons, New York, 1971.

[14] Hsueh, M. C., Iyer, R. K. and Trivedi, K. S., "A Measurement-Based Reliability/Performability Model for a Multiprocessor System," *IEEE Trans. Comput.,* this issue.

[15] Huslende, R. "A Combined Evaluation of Performance and Reliability for Degradable Systems," *ACM/SIGMETRICS Conf. on Measurement and Modeling of Computer Systems,* 1981, 157-164.

[16] Iyer, B. R., Donatiello, L. and Heidelberger, P., "Analysis of Performability for Stochastic Models of Fault-Tolerant Systems," *IEEE Trans. Comput.*, Vol. C-35, No. 10, October 1986, 902-907.

[17] Jagerman, D. L., "An Inversion Technique for the Laplace Transforms," *Bell System Tech. J.*, Vol.61, No.8, 1982, 1995-2002.

[18] Kulkarni, V., Nicola, V. and Trivedi, K., "On Modeling the Performance and Reliability of Multi-Mode Computer Systems," *The Journal of Systems and Software* , Vol. 6, Nos. 1 & 2, May 1986, 175-183.

[19] Kulkarni, V., Nicola, V., Smith R. and Trivedi, K., "Numerical Evaluation of Performability and Job Completion Time in Repairable Fault-Tolerant Systems," *FTCS Proceedings*, Vol.16, 1986.

[20] Meyer, J., "On Evaluating the Performability of Degradable Computer Systems," *IEEE Trans. Comput.*, Vol. C-29, No. 8, Aug. 1980, 720-731.

[21] Meyer, J., "Closed-form Solutions of Performability," *IEEE Trans. Comput.* , July 1982, 648-657.

[22] Puri, P. S., "A Method for Studying the Integral Functionals of Stochastic Processes with Applications: I. The Markov Chain Case," *J. Appl. Prob.,* Vol. 8, 1971, 331-343.

[23] Reibman, A. R. and Trivedi, K. S., "Numerical Transient Analysis of Markov Dependability Models," to appear, *Computers and Operat. Res.*

[24] Siewiorek, D. P., "Multiprocessors: Reliability Modeling and Graceful Degradation," *Infotech State of the Art Conference on System Reliability,* Infotech International, London, 1977.

[25] Siewiorek, D. P., Kini, V., Joobbani, R., and Bellis, H., "A Case Study of C.mmp, Cm*, and C.vmp: Part II- Predicting and Calibrating Reliability of Multiprocessor Systems," *Proceedings of the IEEE* , Vol. 66, No. 10, October 1978, 1200-1220.

[26] Weeks, W. T., "Numerical Inversion of Laplace Transforms Using Laguerre Functions," *JACM,* V.13, No. 3, July 1966, 419-426.

[27] Wilkinson, J. H. and Reinsch, C., *Handbook for Automatic Computation, Vol. II: Linear Algebra,* Springer-Verlag, 1971.

[28] Wu, L. T., "Operational Modes for the Evaluation of Degradable Computing Systems," *ACM/SIGMETRICS Conf. on Measurement and Modeling of Computer Systems,* 1982, 179-185.