Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

BOUNDING AVAILABILITY OF REPAIRABLE COMPUTER SYSTEMS

R. R. Muntz
E. de Souza e Silva
A. Goyal

September 1988
CSD-880070

Bounding Availability of Repairable Computer Systems *

R.R. Muntz † E. de Souza e Silva ‡
A. Goyal §

† UCLA Computer Science Department
‡ Federal University of Rio de Janeiro, NCE
§ IBM T.J. Watson Research Center

February 2, 1989

Abstract

Markov models are widely used for the analysis of availability of computer/communication systems. Realistic models often involve state space cardinalities that are so large that it is impractical to generate the transition rate matrix let alone solve for availability measures. Various state space reduction methods have been developed, particularly for transient analysis. In this paper we present an approximation technique for determining steady state availability. Of particular interest is that the method also provides bounds on the error. Examples are given to illustrate the method.

1 Introduction.

The importance of reliability and availability of computer systems clearly increases with their use in life critical systems and where system failures can cause significant economic loss. Analytic methods for dependability analysis of computer systems has been an active area of research for some time (e.g., [GEIS85,GOYA86,GROS84,DeSO86b,HEID87, TRIV82]) and a number of tools have been built to aid in the specification of system models and make the analytic methods more accessible (e.g., [CARR86,COST81,GOYA85, MAKA82,TRIV84]). Increased complexity of the target systems and the sophistication of the reliability and availability measures that are of interest continue to challenge our ability to analyze these systems.

*The work of R.R. Muntz and E. de Souza e Silva was supported in part by a grant from NSF INT-8514377, CNPq-Brazil and IBM Brazil.

There are two major categories of dependability measures that are of interest. Transient measures relate to dependability over a finite period of time, e.g. the duration of a mission. An example of a transient measure is "mean interval availability" which, for a specified interval length, would be the mean amount of time the system is operational during the interval. This type of measure is more appropriate for the analysis of mission oriented systems, e.g. a space mission. Steady state measures are associated with limiting behavior and therefore are appropriate for systems with a lifetime that for modeling purposes is viewed as being infinite. This is particularly appropriate for repairable systems for which the lifetime of the system is expected to span many failures. We will be concentrating in this paper on the problem of determining steady state availability.

A number of techniques have been used in dependability analysis including combinatorics, Markov or semi-Markov model analysis [TRIV82], and simulation [LEWI84, CONW87]. Recent work on simulation techniques holds promise for broadening the applicability of this technique [CONW87]. Continuous time Markov models are still the most widely used for dependability analysis. This class of models encompasses an extremely broad range of dependability models. Unfortunately the characteristics of dependability models (complex interactions between components, multiple repair facilities and scheduling policies, complex criteria for a system to be operational, etc.) preclude the possibility of closed form solutions in general. Thus numerical solution methods are most widely used. The most pervasive limitation to the use of numerical solution techniques is the size of the state space of realistic dependability models. State space cardinalities of real models can outstrip current memory and processor capabilities by orders of magnitude; i.e. while tens of thousands of states can be effectively handled, real models often have more than tens of millions of states. The natural way to deal with this problem is via state space reduction techniques; the most common of which involve truncation of the state space or aggregation of states. A major difficulty with these methods (as with many approximation methods) is providing a bound on the error that is introduced.

This paper presents an approximation method that not only provides the requisite state space reduction for numerical analysis but also provides error bounds.

The following section provides a more detailed description of the problem domain and discussion of some of the more closely related previous work on analysis of dependability models. Section 3 presents the analytic development of the results. In Section 4 we present several examples to illustrate the application of the results and Section 5 concludes with a summary and discussion.

2 Problem Definition.

We are interested in continuous time, discrete state, homogeneous Markov processes and in particular those which represent models for computer/communication systems reliability analysis. We assume that the Markov processes involved are ergodic. We are not directly concerned in this paper with the problem of translating a system specification into a state transition matrix. A number of systems have been built that take a high level system description and generate the state transition matrix for the underlying Markov process. Among these are those particularly concerned with reliability modeling [MAKA82,TRIV84, GOYA86,CARR86,BERS87]), as well as those for other application areas such as queueing networks [IRAN71] and distributed algorithms [PLAT85]. A major limitation of such tools is the memory and computation requirements as a function of the cardinality of the state space. In this paper we will often refer to the state transition matrix as though it existed although our ultimate aim is to obviate the need to generate the entire state transition rate matrix and thus alleviate this limitation.

For availability analysis the state space $\mathcal{S}$ of a Markov model can be assumed to be partitioned into two sets: the set of operational states $\mathcal{O}$ and the set of failed states $\mathcal{F}$. The availability $\mathcal{A}$ of the model is defined as:

$\mathcal{A} = \sum_{s \in \mathcal{O}}$ P(s)

where $P(s)$ is the steady state probability for state s.

Availability is a special case of a measure that can be expressed in terms of the expected value of a reward function. In general let R[s] be the reward for state s. Then the expected value of the reward is:

$\mathcal{R} = \sum_{s \in \mathcal{S}}$ P(s) R[s] .

Availability is easily expressed in terms of reward function that is 1 for states in $\mathcal{O}$ and 0 otherwise. It will be convenient to use the reward function notion at times.

We present in this paper a method by which the availability can be approximated and moreover the error introduced by the approximation can be bounded. In addition the method provides the means for avoiding generation of the entire state transition matrix; often only a small fraction of the transition rate matrix is required. This method also provides the flexibility to tradeoff computational resources and error; if one can generate more of the state transition matrix then better error bounds can be obtained.

A good introduction to recent work in the area of numeric techniques for large de-

pendability models can be found in [TRIV82,GOYA86,GROS84,DeSO86b,HEID87]. These papers provide discussion of known results on convergence, sensitivity analysis, transient analysis, etc. Our concern in this paper is limited to a specific problem which has up to this point defied solution: determining bounds on steady state availability that are "tight", computationally feasible and require only a fraction of the state transition rate matrix to be generated. The literature on availability modeling is useful background to the general area but is not directly relevant to the technique developed here. More pertinent is the work on decomposition [COUR77]. We assume that the reader is familiar with the basic aggregation/disaggregation approximation procedure as described in [COUR77]. We will also use the fact that exact aggregation is always possible regardless of the form of the transition rate matrix or the states chosen to aggregate.

In [COUR84] Courtois and Semal showed how to compute bounds on the error introduced by aggregation. Part of their results are repeated in the following section. While their main concern was to bound individual state probabilities we will be concerned more with bounds on a global measure, namely availability. Conceptually it is a simple matter to apply their results to bounding a measure like availability. However there are several problems with such a direct application of their results. The computation cost of a direct application of the Courtois and Semal results would be prohibitive in many reliability modeling applications. We leave the detailed justification of this claim to the next section which describes the problems and our solutions.

In more recent work [COUR86a,COUR86b,SEMA87] Courtois and Semal have built upon the [COUR84] results to show how properties of large Markovian models can be estimated/bounded without resorting to generation of the entire transition matrix. This paper is in the same vein as it reports the results of adapting the bounding techniques of Courtois and Semal to availability modeling.

3 Upper/Lower Bounds on Steady State Availability.

In this section we describe how upper and lower bounds on steady state availability can be calculated. The method can be used to bound any steady state measure based on attributing rewards to each state of the Markov chain although the efficacy of the approach depends on the characteristics of the model.

We start by discussing the intuitive motivation behind the method. As mentioned in the introduction, continuous time Markov models encompass an extremely broad range of availability models. Unfortunately, models of real systems can easily require tens of millions of states. This enormous state size precludes not only the use of numerical solution methods but even the generation of the Markov chain itself. However if we consider the common characteristics of these models we find some helpful intuition. Each state of the Markov chain corresponds to a system state in which some components have failed (and so, need some form of repair) and others which are operational. Real systems are designed to have a "high level" of availability, and so it is expected that most of the time the

system operates with the majority of its components operational. This observation simply indicates that most of the probability mass is concentrated on a relatively small number of states in comparison to the total number of states in the model. During its lifetime, the system spends most of the time in this relatively small subset and very rarely reaches other states of the system.

With the above observation in mind, our approach in brief, is to maintain a detailed description of the system model for those states with few components failed (the most popular states) and representing the complement of this subset (the remainder of the system description) in a simplified way. This simplified representation of the complement is achieved by aggregation of states. As will be shown later, the aggregates and transitions between aggregates can be chosen such that we can obtain upper and lower bounds for availability.

3.1 Background.

We will assume in this paper that the model is given in terms of the transition matrix although the underlying model is a continuous time Markov chain. This causes no loss of generality since if G is the generator matrix for an ergodic, continuous time Markov process $\mathcal{G}$, then the equilibrium state probability vector for G is the solution to $xG = 0$. If we let $M = G/\Lambda + I$ where Λ is the largest absolute value of any element of G, then x is the solution to $xM = x$. Thus there is a simple relationship between the generator matrix for a continuous time Markov chain and an underlying discrete time process.

$$\begin{bmatrix} Q_{00} & Q_{01} & Q_{02} & \cdots & Q_{0N} \\ Q_{10} & Q_{11} & Q_{12} & \cdots & Q_{1N} \\ & \ddots & & & \\ & & & & Q_{NN} \end{bmatrix}$$

Figure 1: Transition matrix.

Without loss of generality, we assume that models are described by stochastic matrices. It is convenient to view the transition matrix as being organized as shown in Figure 1. Each principal submatrix $(Q_{00}, Q_{11}, ...Q_{NN})$ corresponds to a particular subset of states. These subsets of states form a partition $\mathcal{P}$, of the state space. Let $\mathcal{F}_i(\mathcal{P})$ denote the subset of states in partition $\mathcal{P}$ corresponding to Q_{ii}. For simplicity of notation we will drop the dependence on $\mathcal{P}$ in the notation $\mathcal{F}_i(\mathcal{P})$ when the state partition is clear from the context and refer to the i^{th} partition as $\mathcal{F}_i$. To motivate what follows, we first consider the special case of a partition of cardinality 2 (i.e. $N = 1$ in Figure 1).

$$\begin{bmatrix} Q_{00} & Q_{01} \\ Q_{10} & Q_{11} \end{bmatrix}$$

In the context of availability modeling, suppose that $\mathcal{F}_0$ corresponds to states that are most "popular", i.e. states with relatively few failed components. $\mathcal{F}_1$ consists of the remaining states. As argued intuitively above we would like the $\mathcal{F}_0$ states to account for almost all of the probability mass but to contain only a small fraction of the total state space. For the size of models that we will be dealing with (up to approximately 100 components) $\mathcal{F}_0$ will contain up to several thousand states while $\mathcal{F}_1$ can contain millions of states. The idea is that Q_{11} and Q_{10} will never actually be generated. Since the matrix is not actually ever explicitly represented the question naturally arises as to what can be computed from only a portion of the transition matrix.

One specific question is "What can be said about the equilibrium state probabilities for the states in $\mathcal{F}_0$ given only the submatrix Q_{00}?" Note that Q_{00} is not a stochastic matrix. The rows of Q_{00} can be "deficient" since they can sum to something less than 1. Below we state some results by Courtois and Semal [COUR84] which provides a partial answer.

Theorem 1 *Let L be any $n \times n$ matrix with $L \geq 0$ such that each row sum is less than or equal to 1.*

Let $\beta(L) = \{B | B$ is an $n \times n$ irreducible stochastic matrix and $B \geq L\}$. [1]

Let $L_i, 0 \leq i \leq n-1$ be the stochastic matrix equal to L except in the i^{th} column. L_i is matrix L with elements in the i^{th} column increased as necessary to make the matrix stochastic.

Let z_i = the vector of steady state probabilities corresponding to L_i.

Let $\mathcal{V}_L = \{v | v$ is the vector of steady state probabilities for some $B \in \beta(L)\}$.

Let $\mathcal{Z}_L = \{v | \exists \beta_i, 0 \leq i \leq n-1$ such that $\sum \beta_i = 1, v = \sum_{i=0}^{n-1} \beta_i z_i\}$ (i.e., $\mathcal{Z}_L$ = is the convex hull of the Perron vectors of $\{L_i\}$).

Then $\mathcal{V}_L = \mathcal{Z}_L$.

With respect to a matrix such as Q_{00} which is a principal submatrix of a larger stochastic matrix, the theorem has a probabilistic interpretation [COUR86a]. Q_{00} plays the role of the matrix L in the above theorem. Let $\mathcal{F}_0$ be the set of states corresponding to Q_{00}. Consider that the system starts in a state in $\mathcal{F}_0$ and as it evolves, every time the original system would have made a transition out of $\mathcal{F}_0$ this is instead made a transition to state i in $\mathcal{F}_0$. For each choice of "return state" i, there is a corresponding matrix L_i formed by incrementing elements of the i^{th} column of Q_{00}. The steady state probability vector of L_i (i.e. z_i) is the conditional state probability vector under the assumption that each time the set of states $\mathcal{F}_0$ is reentered, it is via state i. The theorem says that the conditional state probabilities for states in $\mathcal{F}_0$ (denoted v_0) are the solution (left eigenvector) of some member of $\beta(Q_{00})$. More importantly to us, the theorem states that given the vectors $z_i, 1 \leq i \leq n$ (corresponding to incrementing the i^{th} column), then v_0, the conditional state probability vector, is a linear combination of the z_i.

[1] An expression $A \theta B$ where A and B are matrices means that corresponding elements in A and B satisfy the relationship θ.

Let $\vec{R}$ be the reward vector for the model Q. Let $\vec{R}_0$ be the portion of $\vec{R}$ corresponding to $\mathcal{F}_0$. Then $v_0\vec{R}_0$ is the conditional reward rate, $v_0\vec{R}_0 = \sum_{i\in\mathcal{F}_0} P(i|\mathcal{F}_0)R_i$. In other words, $v_0\vec{R}_0$ is the expected reward per unit time given that the system is in some state of $\mathcal{F}_0$.

Let $v_0\vec{R}_0 = A_{\mathcal{F}_0}$. Then we have the following corollary to Theorem 1 which concerns the extreme values that the conditional reward rate can achieve given the matrix Q_{00}.

Corollary 1 $\min_i\{z_i\vec{R}_0\} \leq A_{\mathcal{F}_0} \leq \max_i\{z_i\vec{R}_0\}$.

Proof:
$v_0 = \sum \beta_i z_i$ for some $\beta_i \geq 0, \sum \beta_i = 1$.
$A_{\mathcal{F}_0} = v_o\vec{R}_0 = \sum \beta_i(z_i\vec{R}_0) = \sum \beta_i r_i$ where $r_i = z_i\vec{R}_0$.
$\min_i\{r_i\} \leq A_{\mathcal{F}_0} \leq \max_i\{r_i\}$. □ .

The above theorem and corollary show how bounds on the conditional reward rate can be calculated. In our case, the reward for a state is either 0 (not operational) or 1 (operational) and $A_{\mathcal{F}_0}$ would be the conditional availability given the system is in some state of $\mathcal{F}_0$.

We need one further result from [COUR84].

Theorem 2 *(Exact Aggregation) Let Q be partitioned as in Figure 1. Consider an $N \times N$ stochastic matrix Q_{Ag} such that*

$$Q_{Ag}[I,J] \;=\; v_I Q_{IJ}\mathbf{1}^T \tag{1}$$

Informally $Q_{Ag}[IJ]$ is the steady state probability of a transition to a state in aggregate J (i.e. $\mathcal{F}_J$) given the system is in some state of aggregate I. If $X = (X_1, X_2, \ldots, X_N)$ is the steady state probability vector for Q_{Ag}, then $X_i, 1 \leq i \leq N$ is the steady state probability of being in some state of $\mathcal{F}_i$ in the original model Q.

To apply the above theorem requires that one first obtain all of the conditional state probabilities. This is not possible with the models we consider due to the cardinality of the state space. We can however make use of the existence of such an aggregation to explain our approach.

3.2 Bounds on Mean Availability.

Different partitions or aggregates will be appropriate for different applications. For our case, we choose to partition the states such that $\mathcal{F}_i$ will contain all of the states corresponding to exactly i failed components. We will assume that the availability model has a transition matrix organized as in Figure 1. The principal submatrices correspond to the partition of the state space as described.

Consider now a (less refined) partition of the state space into two sets: $\mathcal{D} = \bigcup_{i=0}^{K-1} \mathcal{F}_i$ and $\mathcal{R} = \bigcup_{i=K}^{N} \mathcal{F}_i$.

Informally, the intention is that the cardinalities of $\mathcal{F}_0$, $\mathcal{F}_1$, ..., $\mathcal{F}_{K-1}$ are small enough that it is practical to generate the portion of the transition matrix corresponding to these states but that adding $\mathcal{F}_K$ would exceed resource limitations. In the notation, $\mathcal{D}$ stands for "detailed" which is motivated by the fact that the transitions between these states will be represented in detail. The $\mathcal{R}$ stands for "reduced" since we will be using a reduced state space representation of these states. (For ease of exposition we assume that we do not consider "splitting" one of the $\mathcal{F}_i$ between D and R although this is not a problem).

Let $P(\mathcal{D})$ ($P(\mathcal{R})$) be the steady state probability that the system is in a state in $\mathcal{D}$ ($\mathcal{R}$). Let $A_\mathcal{D}$ ($A_\mathcal{R}$) denote the conditional reward rate given the system is in $\mathcal{D}$ ($\mathcal{R}$). Then we can express the availability A as:

$$A = P(\mathcal{D})A_\mathcal{D} + P(\mathcal{R})A_\mathcal{R} \tag{2}$$

Since a lower bound on $A_\mathcal{R}$ is 0 and an upper bound on $A_\mathcal{R}$ is 1 it is easy to see that:

$$P(\mathcal{D})A_\mathcal{D} \leq A \leq P(\mathcal{D})A_\mathcal{D} + P(\mathcal{R}) \tag{3}$$

Since $A_\mathcal{D} \leq 1$ and $P(\mathcal{R}) = 1 - P(\mathcal{D})$ it is easy to see that:

$$[P(\mathcal{D})]_{lb}[A_\mathcal{D}]_{lb} \leq A \leq [P(\mathcal{D})]_{lb}[A_\mathcal{D}]_{ub} + (1 - [P(\mathcal{D})]_{lb}) \tag{4}$$

The subscript "lb" ("ub") on a term in the above equation indicates a lower bound (upper bound) on that term.

Direct application of Theorem 1 and Corollary 1 provides a means of obtaining bounds on $A_\mathcal{D}$. The problem with this approach is the number of models that have to be solved; namely, one for each state in $\mathcal{D}$ to which there is a non-zero transition from a state in $\mathcal{R}$. This may be impractical for most applications, since the number of such states may be in the order of thousands. For example if $\mathcal{D}$ consists of the states in partitions $\mathcal{F}_0, \ldots, \mathcal{F}_{K-1}$ then typically there will be non-zero transitions into all of the states of $\mathcal{F}_{K-1}$. This number may be very large even for small values of K. In the second example presented in the next section, with a total of 36 system components and $K = 4$ (i.e., all states representing up to 3 failures are generated) then 1,532 (i.e., the cardinality of $\mathcal{F}_{K-1}$) models would have to be solved.

There is still the issue of determining a suitable lower bound on $P(\mathcal{D})$. If we had a lower bound matrix for the aggregate transition matrix then we could apply Theorem 1. The tightness of this bound is going to be determined by the tightness of the bounds on the elements of the aggregate transition matrix. The particular properties of availability models can cause the bounds to be too loose for our purposes. This is discussed briefly in the next section.

The above discussion indicates that direct application of Theorem 1 is not sufficient for our problem. The remainder of this section is devoted to showing how these problems can be overcome in the case of availability modeling.

Consider an exact aggregation of partitions $\mathcal{F}_K$, $\mathcal{F}_{K+1}$, ..., $\mathcal{F}_N$. (The individual states in $\mathcal{D}$ can each be considered to be a separate aggregate in which it is the only member state.)

The matrix Q' obtained is illustrated below. In this figure $p_{K,K} \ldots p_{N,N}$ are each single transition probabilities corresponding to the aggregate states while the $Q_{i,i}$ are matrices.

$$\begin{bmatrix}
Q_{00} & & & \cdots & & & & \\
Q_{10} & Q_{11} & & & \cdots & & & \\
0 & Q_{21} & \ddots & & & \cdots & & \\
. & \ddots & \ddots & Q_{K-1,K-1} & & & \cdots & \\
0 & & 0 & Q_{K,K-1} & p_{K,K} & & & \\
. & \ddots & \ddots & 0 & p_{K-1,K} & \ddots & & \\
. & & & & 0 & \ddots & p_{N-1,N-1} & \\
0 & 0 & & \cdots & & 0 & p_{N,N-1} & p_{N,N}
\end{bmatrix}$$

We note the following properties of this matrix which we will find useful.

(a) The only non-zero transition probabilities to states in $\mathcal{D}$ are from aggregate state K.

(b) We assume that there is zero probability of more than one component being repaired at exactly the same time. (Note that this does not preclude representing multiple repair facilities or components that are "unusable" due to failure of other components. In the latter case, the unusable component should simply not be considered as "failed" for the purpose of aggregation the states.) This assumption implies that the transitions between aggregate states are "nearest neighbor" in one direction, i.e. the transition probability $p_{J,K}$ is 0 for $J < K - 1$. Note that this also implies that the only transitions from aggregate state K to $\mathcal{D}$ are to states in $\mathcal{F}_{K-1}$.

(c) Since we assume that Q' was formed by exact aggregation it follows that the conditional state probabilities for states in $\mathcal{D}$ are the same as in the original model and that $P(\mathcal{D})$ is the same as in the original model. It follows from this observation and Equation 3 that upper and lower bounds can be obtained by assigning reward 1 or 0 respectively to the aggregate states while maintaining the same rewards for states in $\mathcal{D}$.

Q' as defined above gives us a "simpler" model to think about but is not a practical solution to our problem since it requires exact aggregation (which in turn requires too much computation). The following two lemmas provide part of the basis for computing bounds on mean availability. The first lemma indicates that the exact transition probabilities from $\mathcal{F}_K$ to $\mathcal{D}$ do not have to be known but only the total of all transitions probabilities from $\mathcal{F}_K$ to $\mathcal{D}$. The second lemma shows that the transition rates between aggregates (and between $\mathcal{F}_K$ and $\mathcal{D}$) do not have to be known exactly but rather bounds on these rates can be used. (The former is a simple application of Theorem 1. The latter is different than the direct application of Theorem 1 which uses only lower bounds on transition rates. This difference is vital to the computation of reasonable bounds as is discussed in the next section.)

Lemma 1 *Consider a stochastic matrix of the form Q' as above. Then for each state s_n in $\mathcal{F}_{K-1}$ to which there is a non-zero transition probability from aggregate state S_K, define the stochastic matrix Q'_n as being equal to Q' except that:*

(a) the only non-zero transition from aggregate S_K to states in $\mathcal{F}_{K-1}$ is to state s_n.

(b) the transition probability from aggregate state S_K to s_n is equal to the sum of the transition probabilites from aggregate S_K to states in $\mathcal{F}_{K-1}$ in Q'.

Let the rewards for states in $\mathcal{D}$ be the same as in the original model.

Let the reward rate conditioned on being in $\mathcal{D}$ corresponding to matrix Q'_n be denoted $A(n)$.

Then the conditional availability of the the original model is bounded by:

$$\min_n\{A(n)\} \leq A_{\mathcal{D}} \leq \max_n\{A(n)\} \tag{5}$$

Proof: This is a straightforward application of Corollary 1.□

The significance of this lemma is that we can (theoretically) determine extreme values (upper and lower bounds) on the availability conditioned on being in a state of $\mathcal{D}$ by solving each of the Q'_n models. A major difficulty with actually doing this is that the transition rates between the aggregate states have not been specified. Obtaining these is not practical for the sizes of the models we are considering. The following lemma shows that bounds on the transition rates between aggregate states can be used in the Q'_n matrices rather than the actual values and that $P(\mathcal{D})$ is minimized (as required in equation(4)).

In the lemma below, for convenience, we use generator matrices instead of stochastic matrices. Recall that the matrix notation G represents a generator matrix for a continuous time Markov chain.

Lemma 2 Maximum Holding Time

Consider a Markov process with generator G in which the states are partitioned into subsets as indicated above and states in subset $\mathcal{F}_k$, $k \geq K$, are exactly aggregated. Furthermore, the partition into subsets satisfies the "nearest neighbor" requirement. From this process let us construct a new process with generator G' as shown in Figure 2. In this figure, the "+" signs indicate that the upper triangular elements (except rows above the horizontal line) are replaced by upper bounds and the "−" signs indicate that the transition rates from aggregate $\mathcal{F}_k$ to $\mathcal{F}_{k-1}$ are replaced by lower bounds. Note also that there is only a single transition from aggregate $\mathcal{F}_K$ to a state in $\mathcal{F}_{\mathcal{D}}$. More formally we have:

$$\begin{array}{lcll} G'_{\mathcal{D}\mathcal{D}} & = & G_{\mathcal{D}\mathcal{D}} & \\ G'_{\mathcal{D}k} & = & G_{\mathcal{D}k} & K \leq k \leq N \\ g'_{k,k-1} & \leq & g_{k,k-1} & K < k \leq N \\ g'_{K\mathcal{D}_i} & \leq & g_{K\mathcal{D}_i} & \\ g'_{k,l} & \geq & g_{k,l} & K \leq k < l \leq N \end{array}$$

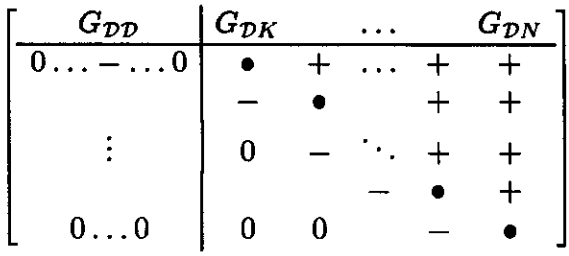

Figure 2: Matrix G': holding time lemma.

where g (g') is an element of matrix G (G') and the index $\mathcal{D}_i$ indicates the i^{th} state in subset $\mathcal{F}_\mathcal{D}$. Let $P(\mathcal{F}_\mathcal{D})$ ($P(\mathcal{F'}_\mathcal{D})$) be the steady state probability that the system is in any state of set $\mathcal{F}_\mathcal{D}$ ($\mathcal{F'}_\mathcal{D}$). Then

$$P(\mathcal{F'}_\mathcal{D}) \leq P(\mathcal{F}_\mathcal{D}) \tag{6}$$

Proof:
The proof is in the appendix. □ .

The two previous lemmas indicate a method for finding the bounds described in Equation 4. To summarize, the "recipe" is as follows:

1. Generate the portion of the transition matrix corresponding to transitions between states in $\mathcal{D}$ and from states in $\mathcal{D}$ to states in $\mathcal{R}$.

2. Find the upper and lower bounds on the transition rates between aggregate states as defined in Lemma 2. Normally the minimum repair rate of all components would suffice for the lower bounds and the sum of all failure rates would suffice for the upper bounds. (These are the bounds used in our examples discussed in the next section.)

3. For each state in $\mathcal{F}_{K-1}$ to which there is a non-zero transition rate from $\mathcal{F}_K$, construct and solve the matrix G'_n. For each such model, find the mean reward assuming first a reward of 0 for the aggregate states and then a reward of 1 for these states. A lower bound on the mean availability is the minimum of these values and an upper bound is the maximum.

The bounding method just described can require a large number of submodels to be solved and is often impractical. Fortunately, we can transform the original model so that the number of submodels to be solved can be drastically reduced. First we partition the states of a Markov chain model into subsets $\mathcal{F}_i$, as before and we assume that all states that will be generated belong to $\mathcal{D} = \{\bigcup_{i=0}^{K-1} \mathcal{F}_i\}$.

Let F be any integer $0 \leq F < K$. Now form three sets of states:

$\mathcal{G}_0 = \{\bigcup_{i=0}^{F-1} \mathcal{F}_i\}$
$\mathcal{G}_1 = \{\bigcup_{i=F}^{K-1} \mathcal{F}_i\}$
$\mathcal{G}_2 = \{\bigcup_{i=K}^{N} \mathcal{F}_i\}$

Figure 3 illustrates the transition matrix G in which G_{ii} is the principal submatrix corresponding to states in $\mathcal{G}_i$. (The submatrix shown as 0 is a consequence of the "nearest neighbor" property discussed previously.) Now construct a new transition matrix as shown in Figure 4. It is clear that G' is a stochastic matrix if G is stochastic. The relationship between the process defined by G and that defined by G' is illustrated in Figure 5. Basically, in the new process there are two sets of states corresponding to the states $\mathcal{G}_1$ of the original model. Let us call them $\mathcal{G}'_{1u}$ and $\mathcal{G}'_{1d}$ as shown in Figure 4. The idea behind this transformation can be explained as follows. Assume the system starts in the "all components up" state, i.e. $\mathcal{F}_0$. As components fail and are repaired the system will stay in states in $\mathcal{G}'_0$ and $\mathcal{G}'_{1u}$ until the first time that there are K or more failed components. At this point the system is in a state of $\mathcal{G}'_2$. However when the number of failed components falls below K, the system now enters a state in $\mathcal{G}'_{1d}$. (Now the notation is explainable; "u" stands for "going Up" and "d" stands for "going Down". As the number of failed components goes up the system visits states in $\mathcal{G}'_{1u}$ and after K failures have been reached, it visits the states in $\mathcal{G}'_{1d}$ as the number of failed components goes down.)

From the construction it is easy to show that the two transition matrices are such that the steady state probabilities of the original process can be calculated from the steady state probabilities of the second process. In other words, if $[\pi_0, \pi_1, \pi_2]$ is the solution of $\pi G = \pi$ and $[\pi'_0, \pi'_{1_1}, \pi'_{1_2}, \pi'_2]$ is the solution of $\pi' G' = \pi'$ then: $\pi_0 = \pi'_0$; $\pi_1 = \pi'_{1_1} + \pi'_{1_2}$; $\pi_0 = \pi'_0$.

There is a natural mapping of the states in G' to states in G. In terms of rewards, the reward function for states in G' is simply to assign the same reward as the corresponding state in G. It is clear then that the mean availabilities of the two systems are identical.

$$\begin{bmatrix} G_{00} & G_{01} & G_{02} \\ G_{10} & G_{11} & G_{12} \\ 0 & G_{21} & G_{22} \end{bmatrix}$$

Figure 3: Matrix G.

Given a model G, the procedure would be to choose a value for F, construct the model G' and apply the "recipe" given above. In applying the recipe, the states corresponding to $\mathcal{D}$ (whose detailed transitions are represented) are the states in $\mathcal{G}'_0$ and $\mathcal{G}'_{1u}$. The remainder of the states correspond to $\mathcal{R}$. As before the states in $\mathcal{R}$ are aggregated with each aggregate containing states with the same number of failed components.

In summary, the sum of the cardinalities of $\mathcal{G}'_0$ and $\mathcal{G}'_{1u}$ (which is also the cardinality of $\{\mathcal{F}_0 \bigcup \ldots \bigcup \mathcal{F}_{K-1}\}$) determines the size of the matrix used in the calculation and the

$$\left[\begin{array}{cc|cc} G_{00} & G_{01} & 0 & G_{02} \\ G_{10} & G_{11} & 0 & G_{12} \\ \hline G_{10} & 0 & G_{11} & G_{12} \\ 0 & 0 & G_{21} & G_{22} \end{array}\right]$$

Figure 4: Duplication of states. Matrix G'.

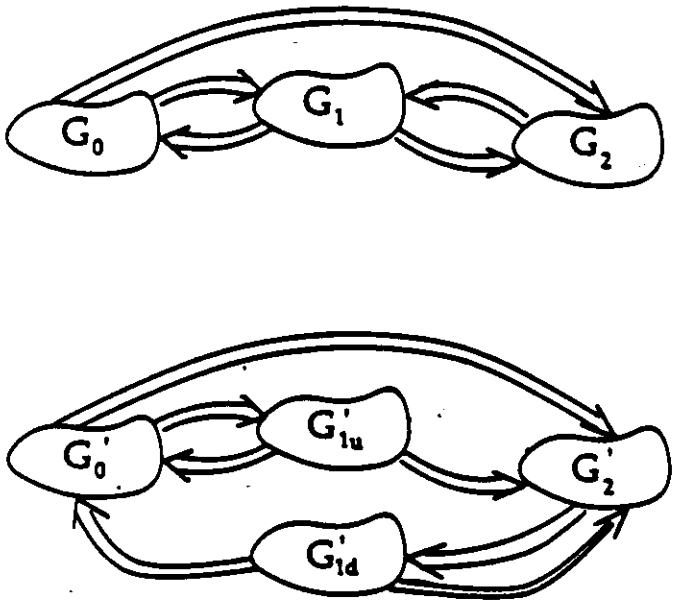

Figure 5: Relationship of G and G'.

cardinality of $\{\mathcal{F}_{F-1}\}$ determines the number of availability values to be calculated for upper and lower bounds. For example, if we choose $F = 1$, since the cardinality of $\mathcal{F}_0$ is one, only one submodel has to be solved. As a consequence, we can trade off computational complexity with some accuracy of the bound.

In the next section we present two examples which will show the usefulness of the approach. It is also easy to prove that this "state duplication" procedure although it reduces computational costs to calculate the bounds on steady state probability, also gives less tight bounds. (Note that the state probabilities associated with states in $\mathcal{G}_1$ are split between $\mathcal{G}'_{1u}$ and $\mathcal{G}'_{id}$ in G' and that the states in $\mathcal{G}'_{1d}$ are lumped with the aggregates.) However, as we will show in the examples, the loss in accuracy is insignificant for typical availability models.

4 Examples.

In this section we present two examples to illustrate our method. The first is a simple example taken from the SAVE manual [GOYA87]. (Although the state space of this model is too small to require the state space reduction technique, it is a simple example that is easily understood and illustrates the concept.)

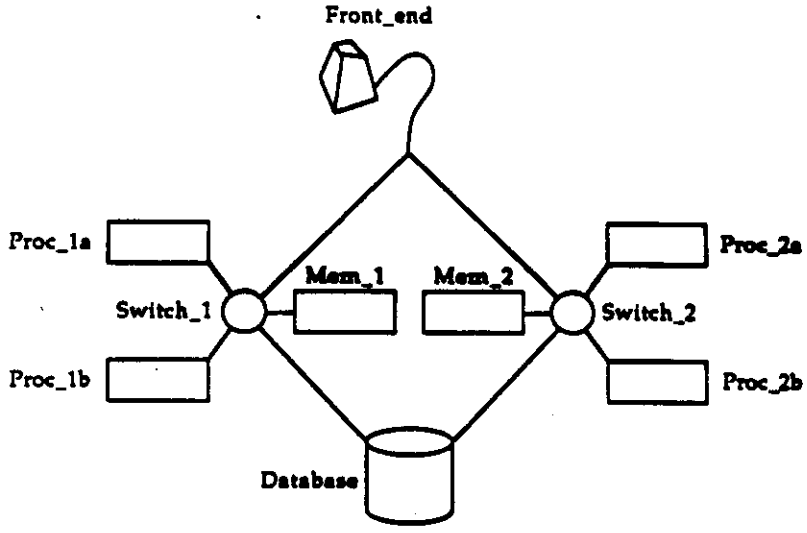

Figure 6: A fault-tolerant database system.

It is a model of a fault-tolerant database system (see Figure 6). The components of this system are: a front-end, a database, and two processing subsystems formed by a switch, a memory and two processors. Components may fail and are repaired according to the rates given in Table 1. Components are repaired by a single repairman who gives priority to the front-end and database, followed by the switches and memory units, followed by the processors. The repairman chooses components with the same priority level at random. As in the SAVE manual, in this model no unit can fail once the system is down. Furthermore, if a processor fails it contaminates the database with probability 0.01.

The complete model has 226 states, and the maximum number of concurrent failures is 7. Solving this model one can obtain : steady state availability = 0.998835336 (unavailability = 0.0011646636).

We can apply our approach to limit the number of states generated and yet obtain bounds on the result. Table 2 presents the bounds when the states are generated in detail up to k failures $0 \leq k \leq 4$, and the other states are aggregated. Furthermore, the detailed states are duplicated so that only one model is needed to obtain the bounds. We can observe that the results are accurate to three decimal places for $k = 2$.

The bottom part of the table also shows the results when we generate detailed transitions of the model between states of up to two failures and duplicate states with two failures

Component	Mean Failure Rate	Mean Repair Rate
Database	1/2400	1
Frontend	1/2400	1
Switch	1/2400	1
Memory	1/2400	1
Processor	1/120	1

Table 1: Failure and repair rates (per hour) for the first example.

only. Since there are 8 states with one failure, a search has to be made to determine the bounds. In this case, 8 models have to be solved. As we observe, the duplication of states does not significantly enlarge the bounds (only the sixth decimal place is affected), and so computational savings are obtained in this case without losing much accuracy.

The second example is a model of a distributed architecture for a database system, as shown in Figure 7.

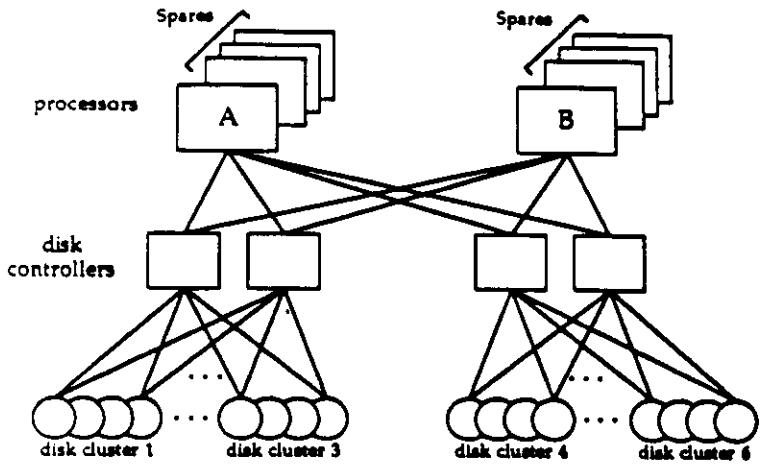

Figure 7: A distributed architecture for a database system.

In this model there are 2 processor types (A and B), 2 sets of dual ported controllers with 2 controllers per set and 6 clusters of disks, each consisting of 4 dual ported disks units. Each set of controllers controls half of the 6 disk clusters. In a disk cluster, data is replicated so that one disk can fail without affecting the system. (The "primary" data on a disk is replicated such that one third is on each of the other three disks in the same cluster. Thus one disk in each cluster can be inaccessible without losing access to the data.) Each processor is linked to both sets of controllers so that each one has access to

Detailed states up to k failures	Availability (upper bound lower bound)	Unavailability (lower bound upper bound)	Number of states
0	1.0 0.963821274	0.0 0.0361787261	11
1	0.999196564 0.996639239	8.03435669e-04 0.00336076077	19
2	0.998845751 0.998763641	0.00115424919 0.00123635926	48
3	0.998835667 0.998834152	0.00116433264 0.00116584751	104
4	0.99883534 0.998835329	0.0011646597 0.00116467095	171
Bounds when detailed states with 1 failure are NOT duplicated			
states generated to 2 failures	0.998845729 0.998764637	0.0011542714 0.0012353634	

Table 2: Bounds with varying number of states generated. First example.

all data stored in the system. If processor A fails, it has a 0.10 probability of affecting processor B. There are 3 spare units for each processor. On occurrence of a failure of a processor, the unit is immediately replaced by a "hot standby" if one is available. Each unit in the system has two failure modes which occur with equal probability. (The failure modes model can be used to model hyperexponential repair distribution for a unit since each failure mode can have a different exponential distribution.) Components may fail and are repaired according to the rates given in Table 3. The different failure rates for the controllers and disk units can be motivated as modeling the different usage of the units according to the type of data stored. (Furthermore, different rates preclude the use of "lumping techniques" [GOYA86] for reducing the state space of the model.)

Components are repaired by a single repairman which chooses components at random from the set of failed units. The system is defined to be operational if all data is accessible to at least one of the two processors, which means that at least the processors, one controller in each set and 3 out of 4 disk units in each of the 6 disk clusters are operational. We also assume that operational components continue to fail at the given rates when the system is down.

The complete model has approximately 9×10^{10} states, which precludes both its generation or solution. The maximum number of concurrent failures is 36. However, we may expect that most of the time the system will be in a small subset of the states of the

Component	Mean Failure Rate	Mean Repair Rate: mode 1	Mean Repair Rate: mode 2
Processors	1/2000	1	1/2
Controller 1	1/2000	1	1/2
Controller 2	1/2000	1	1/2
Disk set 1	1/6000	1	1/2
Disk set 2	1/8000	1	1/2
Disk set 3	1/10000	1	1/2
Disk set 4	1/12000	1	1/2
Disk set 5	1/14000	1	1/2
Disk set 6	1/16000	1	1/2

Table 3: Failure and repair rates (per hour) for the second example.

system, representing operational states and states with a few components failed. Table 4 presents the bounds when the transitions are generated in detail for states up to k failures $0 \leq k \leq 3$. As in the previous example the detailed states are duplicated so that only one model is needed to obtain the bounds. The tightness of the bounds is clear.

The bottom part of the table also shows the results when we generate the model up to three failures and do not duplicate states with one failure. Since there are 20 states with one failure, a search has to be made to determine the bounds. In this case, 20 models have to be solved. Again in this example, the duplication of states does not enlarge the bounds, and so computational savings are obtained without losing much accuracy. Tighter bounds would be obtained if state duplication is not used, but the number of states to be searched would be infeasible in this case.

Earlier we indicated that the straight forward use of the results in [COUR84] to determine bounds on the aggregate state probabilities would give less tight bounds than we obtain via Lemma 2. We have solved models using both methods and verified this claim. Informally, the bounds computed via Theorem 1 require that one use lower bounds for all of the transition probabilities between aggregate states. The matrices L_i in Theorem 1 correspond to all the possible states to which the "unknown" transition probabilities can be assigned. One such state (column) corresponds to the state with all components failed. It is intuitively clear that this choice will result in the minimum value for $P(\mathcal{D})$. We have found that when the repair rates (the largest rates involved) vary much (e.g. by a factor of 5 or 10) then this method can give poor bounds. In contrast the result of Lemma 2 serves to limit the manner in which the "unknown" transition probabilities can be distributed.

States generated up to k failures	Availability (upper bound lower bound)	Unavailability (lower bound upper bound)	Number of states
0	1 0.990027574	0 0.00997242573	37
1	1 0.999660234	0 0.000339765689	57
2	0.999996769 0.999991011	3.23132725e-06 8.9886018e-06	267
3	0.999996683 0.999996582	3.31669666e-06 3.41823502e-06	1799
Bounds when detailed states with 1 failure are NOT duplicated			
states generated up to 3 failures	0.999996683 0.999996607	3.31669676e-06 3.39291503e-06	

Table 4: Bounds with varying number of states generated. Second example.

5 Conclusions.

We have developed a method for determining bounds on the steady state availability from the Markov model of a repairable computer system. The results are based on an adaptation of bounding techniques borrowed from Courtois and Semal. Direct application of the Courtois/Semal results can be prohibitively expensive for availability models. We showed that by modification of the original model, bounds can be obtained with much less cost with minor loosensing of the bounds. On the other hand certain properties of availability models were used to tighten the bounds (thus confusing the issue). Examples were given to illustrate the method and the tightness of the bounds that can be expected.

The development in the paper is couched in terms of models of repairable computer systems and determining bounds on availability. However the methods appear to have promise in other applications. The important relevent property of availability models is that the equilibrium state probabilities are concentrated in very few states. It is reasonable to expect that this same property will hold for example in modeling load balancing. In this case there is presumably a policy for balancing the load on the resources in the system. Thus we expect that a large number of possible states of the system will have "small" equilibrium probability since the scheduler will be biasing the system toward a small number of preferred states. Research into such applications is ongoing.

Acknowledgement: The authors gratefully acknowledge the help of Steven Berson in preparing and running the examples.

Appendix

Proof of Lemma 2:

For each process, aggregate the states in $\mathcal{F}_{\mathcal{D}}$ ($\mathcal{F}'_{\mathcal{D}}$). Since there is a "single return" state in $\mathcal{F}_{\mathcal{D}} \equiv \mathcal{F}'_{\mathcal{D}}$ and $G'_{\mathcal{D}\mathcal{D}} = G_{\mathcal{D}\mathcal{D}}$, then

$$g'_{\mathcal{D},k} = g_{\mathcal{D},k} \qquad k \geq K$$

Figure 8 illustrates the state transition diagram for any of the processes.

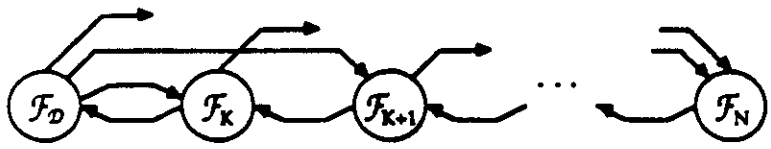

Figure 8: Transition diagram for the two aggregated processes.

From the flow equations for the process with generator G we have:

$$P(\mathcal{F}_{\mathcal{D}}) = \frac{P(\mathcal{F}_K) g_{K,\mathcal{D}}}{\sum_{k \geq K} g_{\mathcal{D},k}} \tag{7}$$

$$P(\mathcal{F}_k) = \frac{\sum_{l<k} P(\mathcal{F}_l) \sum_{j \geq k} g_{l,j}}{g_{k,k-1}} \tag{8}$$

Equivalent formulas also hold for process with generator G'.

We will show that $P(\mathcal{F}'_{\mathcal{D}}) \leq P(\mathcal{F}_{\mathcal{D}})$ by contradiction. Assume that $P(\mathcal{F}'_{\mathcal{D}}) > P(\mathcal{F}_{\mathcal{D}})$. From (7)

$$P(\mathcal{F}_K) = \frac{P(\mathcal{F}_{\mathcal{D}}) \sum_{k \geq K} g_{\mathcal{D},k}}{g_{K,\mathcal{D}}} \tag{9}$$

$$P(\mathcal{F}'_K) = \frac{P(\mathcal{F}'_{\mathcal{D}}) \sum_{k \geq K} g'_{\mathcal{D},k}}{g'_{K,\mathcal{D}}} \tag{10}$$

Since $g'_{\mathcal{D},k} = g_{\mathcal{D},k}\ \forall k \geq K$ and $g'_{K,\mathcal{D}} \leq g_{K,\mathcal{D}}$ then $P(\mathcal{F}'_K) \geq P(\mathcal{F}_K)$. From (8) it is easy to see that $P(\mathcal{F}'_{K+1}) \geq P(\mathcal{F}_{K+1})$, since

$$\begin{array}{lcll} g'_{D,l} & = & g_{D,l} & l > K \\ g'_{K,l} & \geq & g_{K,l} & l > K \\ g'_{K-1,K-2} & \leq & g_{K-1,K-2} & \end{array}$$

Proceeding in this way we can show that

$P(\mathcal{F}'_k) \geq P(\mathcal{F}_k)\ K \leq k \leq N.$
However, since
$P(\mathcal{F}_{\mathcal{D}}) + \sum_{k \geq K} P(\mathcal{F}_k) = 1,$
if $P(\mathcal{F}'_{\mathcal{D}}) > P(\mathcal{F}_{\mathcal{D}})$ and $P(\mathcal{F}'_k) \geq P(\mathcal{F}_k)\ \forall k \geq K,$
then $P(\mathcal{F}'_{\mathcal{D}}) + \sum_{k \geq K} P(\mathcal{F}'_k) > 1$
which is a contradiction. □

References

[BERS87] S. Berson, E. de Souza e Silva and R.R. Muntz, "An Object Oriented Methodology for the Specification of Markov Models", *UCLA Tech. Report CSD-870030*, June 1987 (revised February 1988).

[CARR86] J.A. Carrasco, J. Figueras, "METFAC: Design and Implementation of a Software Toll for Modeling and Evaluation of Complex Fault-Tolerant Computing Systems," *Proceedings of FTCS-16* pp. 424-429, July 1986

[CONW87] A. E. Conway and A. Goyal "Monte Carlo Simulation of Computer System Availability/Reliability Models", *Proceedings of FTCS-17* 1987.

[COST81] A. Costes, J.E. Doucet, C. Landrault, and J.C. Laprie, "SURF: A Program for Dependability Evaluation of Complex Fault-Tolerant Computing Systems," *Proceedings of FTCS-11* pp. 72-78, June 1981.

[COUR77] P.-J. Courtois, "Decomposability: Queueing and Computer System Application", *New York: Academic*, 1977.

[COUR84] P - J. Courtois and P. Semal, "Bounds for the Positive Eigenvectors of Nonnegative Matrices and for Their Approximations" *JACM"* vol. 31, No. 4, pp. 804-825, October 1984.

[COUR86a] P - J. Courtois and P Semal, "Computable Bounds for Conditional Steady-State Probabilities in Large Markov Chanins and Queueing Models" *IEEE JSAC*, vol SAC-4, No. 6, September 1986.

[COUR86b] P.-J. Courtois and P. Semal, "Bounds on Conditional Steady-State Distributions in Large Markovian and Queueing Models", *Teletraffic Anal. and Computer Perf. Eval.*, O.J. Boxma, J.W. Cohen and H.C. Tijms editors, North Holland, 1986.

[DeSO86b] E. de Souza e Silva and H.R. Gail, "Calculating Cumulative Operational Time Distributions of Repairable Computer Systems", *IEEE-TC* vol. C-35, no. 4, pp. 322-332, April 1986.

[DIMI88] Dragomir D. Dimitrijevic and Mon-Song Chen, "An Integrated Algorithm for Probabilistic Protocol Verification and Evaluation", *IBM Res. Rep. RC 13901 (#62470)* 19 p., 8/4/88.

[GEIS85] R. Geist and K. S. Trivedi, "Ultra-High Reliability Prediction for Fault-Tolerant Computer Systems", *IEEE-TC*, C-32, 12, pp. 1118-1127, Dec. 1985.

[GOYA87] A. Goyal, "System Availability Estimator (SAVE)" *IBM Res. Rep. RC 12517 (#56267)* 37 p., 2/18/87.

[GOYA85] A. Goyal, W.C. Carter, E. de Souza e Silva, S.S. Lavenberg and K.S. Trivedi, "The System Availability Estimator", *Proceedings of FTCS-16*, Vienna, pp. 84-89, July 1986.

[GOYA86] A. Goyal, S.S. Lavenberg and K.S. Trivedi, "Probabilistic Modeling of Computer System Availability", *Ann. of Oper. Res.*, vol. 8, pp. 285-306, 1986.

[GROS84] D. Gross and D.R. Miller, "The Randomization Technique as a Modeling Tool and Solution Procedure for Transient Markov Processes", *Oper. Res.*, vol.32, no. 2, pp.343-361, 1984.

[HEID87] P. Heidelberger and A. Goyal, "Sensitivity Analysis of Continuous Time Markov Chains Using Uniformization", *Proc. of the 2nd Intl. Workshop on Applied Math. and Perf./Reliability Models of Computer/Comm. Systems*, Rome, Italy, May 1987.

[IRAN71] K.B. Irani and V.L. Wallace, "On Network Linguistics and the Conversational Design of Queueing Networks," *JACM*, vol. 18, no. 4, pp. 616-629, October 1971.

[LEWI84] E. E. Lewis and F. Bohm, "Monte Carlo Simulation of Markov Unreliability Models", *Nuclear Engineering and Design"*, 77, 1, pp.49-62, 1984.

[MAKA82] S.V. Makam, and A. Avizienis, "ARIES 81: A Reliability and Life-Cycle Evaluation Tool for Fault Tolerant Systems," *Proceedings of FTCS-12* pp. 276-274, June 1982.

[PLAT85] B. Plateau, "On the Stochastic Structure of Parallelism and Synchronization Models for Distributed Algorithms", *Proc. of Sigmetrics Conference*, 1985.

[SEMA87] P. Semal and P - J. Courtois, "Stability Analysis of Large Markov Chains", *PERFORMANCE '87*, pp.363-382, 1988.

[STEW83] G. W. Stewart, "Computable Error Bounds for Aggregated Markov Chains", *JACM*, vol. 30, No. 2, pp. 271-285, April 1983.

[TRIV82] K.S. Trivedi, "Probability & Statistics with Reliability, Queuing and Computer Science Applications", *Prentice Hall*, 1982.

[TRIV84] K.S. Trivedi, J.B. Dugan, R.R. Geist, and M.K. Smotherman, "Hybrid Reliability Modeling of Fault-Tolerant Computer Systems," *Comput. Elec. Eng.*, Vol. 11, pp. 87-108 1984.