Reliability Analysis Techniques Explored Through a Communication Network Example

Kishor S. Trivedi Steve Hunter[*] Sachin Garg[†] Ricardo Fricks
kst shunter sgarg fricks
@ee.duke.edu

Center for Advanced Computing and Communications
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708

Abstract

This paper reviews general methods used to perform dependability analysis on a given system. A communication network example is used in relation to a client/server type of application to illustrate the reliability and availability modeling techniques. We review both non-state space as well as state space based methods and discuss the benefits and limitations of each. The paper assumes a general understanding of probability theory.

1 Introduction

The *reliability* of a system is its ability to maintain operation over a period of time t. Formally, the reliability, $R(t)$, of a system is

$$R(t) = Pr(\textit{the system is operational in } [0,t]).$$

If we define X to be a random variable representing the lifetime of the system and also letting F be the cumulative distribution function (CDF) of X, then the reliability of the system at time t is

$$R(t) = Pr(X > t) = 1 - F(t).$$

It is assumed that a system is working properly at $t = 0$; therefore, $R(0) = 1$.

When modeling a system, it is often but not always assumed that the failure rate is constant; however, this assumption only holds for the normal lifetime of a system and is not true during *burn-in* or *end-of-life*. The importance of this assumption is when the failure rate, λ, is constant, the resulting CDF of the lifetime of the components is exponential. That is

$$F(t) = 1 - e^{-\lambda t}$$

and the reliability is

$$R(t) = e^{-\lambda t}$$

Another measure often used for the analysis of systems is *availability*. The availability of a system is often expressed as the *instantaneous availabilty*, $A(t)$, and/or the *steady-state availability* (i.e., $\lim_{t\to\infty} A(t)$). The instantaneous availabilty, $A(t)$, is defined as the probability that a system is operational at time t. It allows for one or more failures to have occured during the interval $(0,t)$. If a system is not repairable (e.g., a deep space exploring spacecraft), the definition of $A(t)$ is equivalent to $R(t)$. *Dependability* is used as a catch-all phrase for various measures such as reliability, availability etc.

Another measure used to describe a system is its expected life or *mean time to failure* (MTTF). Formally,

$$MTTF = \int_0^\infty R(t)dt.$$

If we continue our assumption of a constant failure rate, λ, then the MTTF of a system is simply $1/\lambda$.

The purpose of this paper is to illustrate methods used for determining the reliability of a system. In particular, we show how to map a given reliability modeling problem into various model types in the SHARPE software tool. The paper is organized as follows: Section 2 describes a client/server application which will be used to demonstrate various techniques, Section 3 presents Non-State Space based models that include Series-Parallel Reliability Block Diagrams, Fault Trees and Reliability Graphs. Section 4 presents State Space based models such as Generalized Stochastic Petri Nets and Stochastic Reward Nets. Section 5 introduces some advanced reliability modeling techniques using *non-Markovian* models. The paper concludes with Section 6.

---

[*] On leave from IBM Corporation, Research Triangle Park, NC.

[†] Supported in part by an IBM Fellowship.

2 Network Example Description

Due to increasing interest in network-centric computing, partly due to advances in high-speed networking technologies and the popularity of client/server applications (e.g., the *World Wide Web*), the reliability of the network for a distributed computing environment is becoming ever more important. The network technology becoming the focus of intense interest is asynchronous transfer mode (ATM). ATM has become the transport mode of choice for broadband integrated-service networks (B-ISDNs)[24].

One difference ATM has with some of the current networking techniques (e.g., bridging, routing) is the concept of establishing a virtual circuit or connection. That is, before a source node can transmit data, it must first set up a connection with a destination node. The source sets up this connection by passing an address of the destination, the amount of bandwidth required and the Quality of Service (QoS) parameters using a signaling protocol. The ATM Forum has standardized Q.2931 as the signaling protocol and is described in the ATM UNI specification [2, 3]. When a connection is established, it is assigned a Virtual Path Identifier/Virtual Circuit Identifier (VPI/VCI) which is unique for every Virtual Channel Connection (VCC).

A possible scenario using this technique for a client/server application is shown in Figure 1. With this application a source/client has a dedicated, switched connection into an access node with redundant backup (nodes $a1$ and $a2$). Other switching nodes (b, c, d, and e) are available to provide a virtual connection to a desired destination or server for this case. As shown, there can be more than one choice for establishing this connection. In fact, with this configuration there are four possible paths that exist and are listed in Table 1.

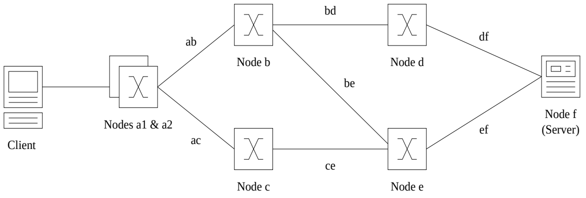


Figure 1: Network Configuration

Table 1: Path Descriptions

Path	Route
1	a-ab-b-bd-d-df
2	a-ac-c-ce-e-ef
3	a-ab-b-be-e-ef
4	a-ac-c-ce-e-be-b-bd-d-df

These path options can be used to improve the overall dependability of the network by providing redundancy. The network is said to be "up" if at least one of the four possible paths is available for communication. For a particular path to be available, all the nodes and links in the corresponding route must be available. Note that failure of a particular link or node may result in unavailability of more than one path. For example, if the node b fails, paths 1, 3 and 4 become unavailable. In the next two sections, we show how all these aspects are captured in various reliability and availability models.

3 Non-State Space Based Models

In this section, reliability block diagram, fault tree and reliability graph techniques are discussed and used to implement the network example. These techniques are concise and efficient and allow the user to capture the relationships between components and the conditions that lead to a system's failure. Their capabilities and limitations are discussed in [20]

3.1 Reliability Block Diagrams

The series-parallel reliability block diagram is the first technique presented for determining a system's dependability. This technique is a subset of other techniques to be shown, such that not all systems can be mapped into a reliability block diagram, but they can be mapped into some of the techniques presented later.

In a block diagram model, components are represented as blocks and are combined with other blocks (i.e., components) in series, parallel, and/or *k-out-of-n* configurations. A diagram that has components connected as a series structure requires that each component must be functioning for the overall system to be operational. A diagram that has components connected as a parallel structure requires only one component to be functional for the overall system to be operational. A *k-out-of-n* structure is a superset of the series and parallel structures and requires k of the n total components to be functional for an operational system. Therefore, parallel and series structures are represented with *k-out-of-n* structures that are *1-out-of-n* and *n-out-of-n*, respectively. The equations for the distribution function of these structures are:

$$F(t) = \begin{cases} 1 - \prod_{i=1}^{N}(1 - F_i(t)) & \text{for a series structure,} \\ \prod_{i=1}^{N} F_i(t) & \text{for a parallel structure.} \end{cases}$$

The distribution function for the k-th order statistic of n independent, identically distributed random variables is

$$F_{k|n}(t) = \sum_{i=k}^{n} \binom{n}{i} F(t)^i (1 - F(t))^{n-i}.$$

Analyzing the network configuration shown in Figure 1, it is realized that a reliability block diagram cannot be generated for this case due to repeated components in the paths. However, to illustrate the concept of reliability block diagrams, our example is simplified by assuming that the link *be* isn't present. Only two alternate paths are possible now and they are listed in Table 2. Figure 2 shows the reliability block diagram model for this modified network. We shall refer to this as the "approximate" model since a simplifying assumption was needed in order to be able to use the series parallel reliability block diagram.

Table 2: Path Descriptions

Path	Route
1	a-ab-b-bd-d-df
2	a-ac-c-ce-e-ef

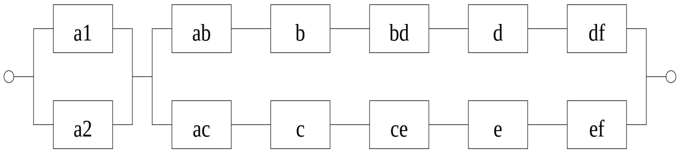


Figure 2: Reliability Block Diagram

3.2 Reliability Graphs

Reliability graph is a directed graph where the edges represent the components of a system being modeled and are assigned a given failure distribution. The graph contains one node with no incoming edges called the *source* and one with no outgoing edges called the *sink*. A system fails when there is no path from the *source* to the *sink* in the reliability graph representation of the system.

At first glance, we may be tempted to consider the reliability graph shown in Figure 3 for representing the communication network. However, it does not correctly solve the problem. Note that both the dotted arcs from $N11$ to $N2$ and from $N4$ to $N9$ need to represent the exactly the same component where if one fails, so does the other. In the reliability graph, even though both the arcs are labeled *be*, they represent statistically identical, but physically different components. This naturally is a misrepresentation of the problem. Figure 4 shows the correct reliability graph for our network example. Apart from the edges labeled with node names and link names, there are four edges labeled "Inf". These do no represent any component in the real system, but are edges which have a distribution infinity assigned to them which means they can not fail. Their use is necessary for a correct structural relationship between various components in the system. Also, note that both single and bidirectional edges are permissible in a reliability graph.

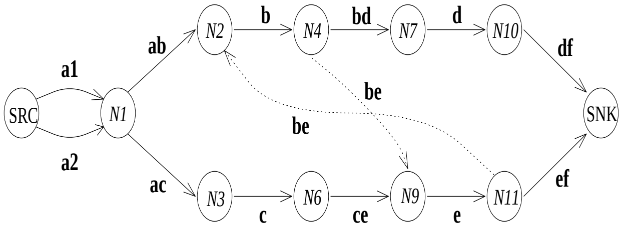


Figure 3: False representation using reliability graph

Reliability graphs are a superset of reliability block diagrams and a subset of fault trees with repeated events [20].

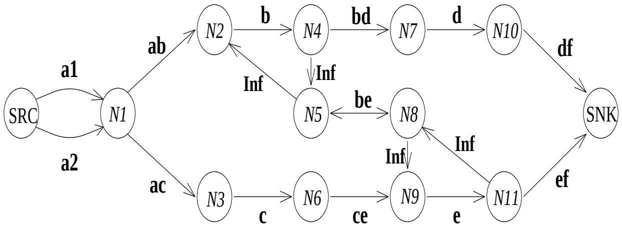


Figure 4: Reliability Graph

3.3 Fault Trees

A fault tree model is a logical structure that aids in the analysis of a system from the perspective of how the reliability of individual components effects the reliability of an overall system. Fault trees are represented as a tree-like structure with the root of the tree being some undesirable event, such as a system failure. The branches of the tree represent the failure of some portion of a system or an individual component. Using this type of pictorial representation can help in focusing on specific combinations of events that lead to system degradation or failure.

Logic gates are used to define the operation at the intersection of branches with the standard logic levels "1" and "0" representing an undesirable event and continued operation, respectively. The logic functions used in fault trees are: *and*, *or* and *k out of n*. A logic 1 at the output of an *and* gate represents a complete or partial system failure when all its inputs are a logic 1. A logic 1 at the output of the *or* gate represents complete or partial system failure when one of its inputs is a logic 1. The output of the *k out of n* gate represents complete or partial system failure when k of its n inputs are a logic 1.

When there are no repeated events in a fault tree, the cumulative distribution, $F(t)$, can be determined with the following equations:

$$\begin{cases} \prod_{i=1}^{n} F_i(t) & \textbf{and} \text{ gate} \\ 1 - \prod_{i=1}^{n}(1 - F_i(t)) & \textbf{or} \text{ gate} \\ \sum_{i=k}^{n} \binom{n}{i} F(t)^i (1 - F(t))^{N-i} & k\textbf{-out-of-}n \text{ gate,} \\ & \text{identically distributed} \end{cases}$$

where $F_i(t)$ denotes the time to failure CDF of component i.

When there is a repeated component, these equations cannot be used, because the failure distributions are no longer independent. For these cases, it is necessary to first obtain cutsets and then use the sum of disjoint products algorithm. Continuing with our networking example, the resulting fault tree for our network is shown in Figure 5. Note that a complete model can be built in this case and there are several repeated components present due to the same links and nodes being used by the four paths defined earlier.

3.4 Results

All of the above non-state space based models were solved using the SHARPE (Symbolic Hierarchi-

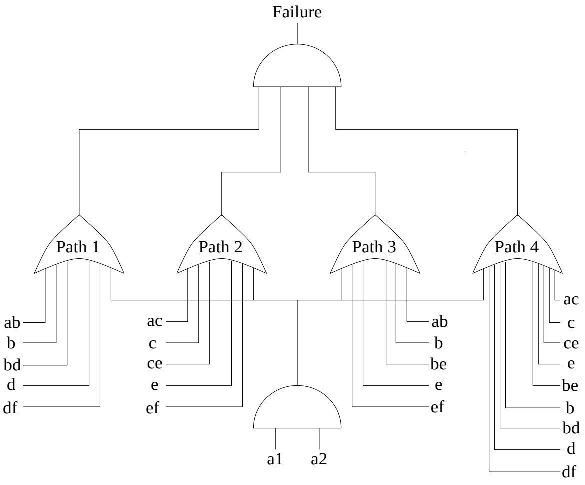


Figure 5: Fault Tree of Network

cal Automated Reliability and Performance Evaluator) software package. SHARPE [26] can solve a variety of model types including series parallel reliability block diagrams, fault trees, reliability graphs, Markov chains, semi-Markov chains, series parallel directed graphs, generalized stochastic Petri nets and product form queueing networks.

Within *SHARPE*, each component in a non-state space model can have exactly one of the following attributes attached to it:

1. A probability of failure
2. A failure rate
3. A Coxian (exponomial) distribution function
4. An instantaneous unavailability function

Reliability, $R(t)$ of the system can be obtained if the individual components have one of the attributes $1-3$ whereas instantaneous availability, $A(t)$, can be determined from attributes $1-4$. If the failure rate of a component is λ and the repair rate is μ, then the instantaneous unavailability of the component is given by:

$$U(t) = \frac{\lambda}{\lambda+\mu}\left(1 - e^{-(\lambda+\mu)t}\right).$$

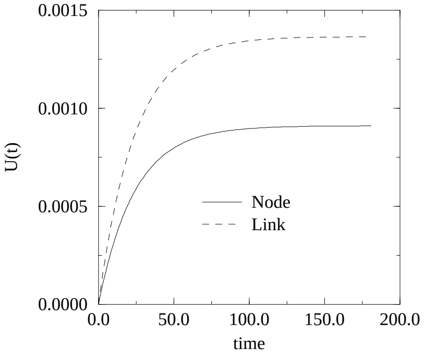


Figure 6: Instantaneous unavailability of a component

Figure 6 plots $U(t)$ for the node as well as the link. $U(t)$ is a defective distribution with mass at infinity. The MTTF of the system can be determined from attributes 2 and 3. Steady state availability can be determined by associating a probability equal to its steady state unavailability given by $\lambda/(\lambda+\mu)$, with each individual component and then solving the model.

For the example network, let the failure rates for the nodes and links be $\lambda_{node} = 0.000038$ failures/hour (i.e., 1 failure in 3 years) and $\lambda_{link} = 0.000057$ failures/hour (i.e., 1 failure in 2 years) respectively. Figure 7 shows the reliability of the network when no repair is allowed.

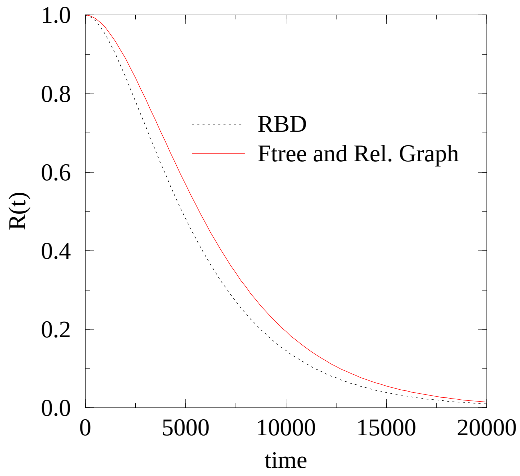


Figure 7: Reliability of the Network

The curve corresponding to reliability block diagram model is only an approximation and yields a lower reliability than the case when the link *be* is present which is modeled by a fault tree as well as by a reliability graph. The mean time to failure obtained is 5838.1 hrs. for the approximate case model and 6669.4 hrs for the exact model.

Let the mean time to repair either a link or a node be 24 hrs. The repair rate, therefore, is 1/24 per hr. Figure 8 shows the instantaneous availability of the network.

As for the reliability, the instantaneous availability predicted by the approximate model is lower than that predicted by the exact model. The steady state availability was obtained as .99996427 and .99997929 for the RBD and FTREE models respectively.

4 State Space Based Models

So far, we have discussed reliability and availability models which did not involve an explicit state space generation. The applicability of such models, however is limited. They assume stochastic independence among failure and repair of various components. In

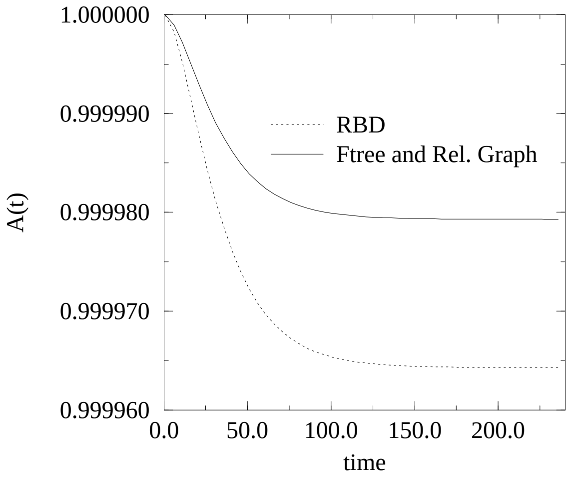


Figure 8: Instantaneous Availability of the network

practical situations, this assumption does not hold. A single repair facility shared among all components, different priorities assigned to repair of different components (i.e., it may be more important to repair a particular node rather than a failed link in our network) are examples which introduce dependencies among the failure/repair behavior of the components. None of RBDs, fault trees or reliability graphs can model this and one needs to use state space based models. The most common of these is the Continuous Time Markov Chain (CTMC). We note that it is possible to formulate and approximately solve models with dependence using non-state space techniques [4].

Since for large and complex systems, manual synthesis of the infinitesimal generator matrix is very tedious, automated methods to specify the system and generate the underlying CTMC are needed and stochastic Petri nets prove to be very useful in this respect.

4.1 Stochastic Petri Net

A Petri net [25] is a directed bipartite graph with two disjoint and finite sets of nodes: *places* and *transitions*. In a graphical representation, the places are depicted by circles and the transitions by rectangles (bars). A place is an *input* to a transition if there is a directed edge called an *input arc* from the place to the transition. Similarly, a place is an *output* to a transition if there is a directed edge called an *output arc* from the transition to the place. *tokens*, depicted by dots, are associated with places and the movement of these tokens represents the dynamic behavior of the system. The tokens move based upon the *firing* of transitions. A transition is *enabled* to fire if each of its input places contains at least one token. Upon firing, one token from each input place is removed and in each of the output places, one token is deposited.

A *marking* of a Petri net is the distribution of tokens in the set of places. Thus, firing of a transition results in a new marking. Each marking defines a state of the system. If the number of tokens in the net is bounded, then the number of markings is finite. A marking is said to be *reachable* from an original marking if there is a sequence of firings starting from the original firing which result in that marking. The *reachability set(graph)* of a Petri net is the set of markings that are reachable from the initial marking.

A stochastic Petri net (SPN) is a Petri net with an exponentially distributed delay associated with firing of each of its transitions. A SPN can be used to model the temporal behaviour of a system along with its functional behavior.

4.2 Generalized Stochastic Petri Nets

Generalized stochastic Petri nets are an extension to SPNs which allow transitions to have zero firing delay or exponentially distributed firing delay [1]. Both SPNs and GSPNs have been shown to be equivalent to CTMCs.

Figure 9 shows the GSPN model for our example network when no repairs are performed. The left most

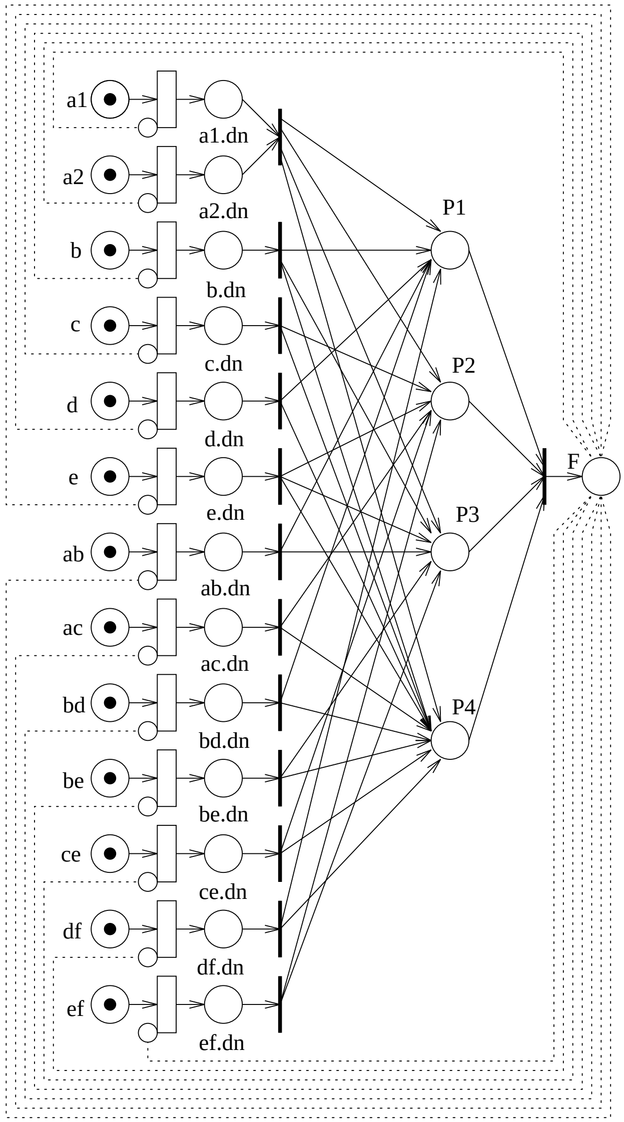


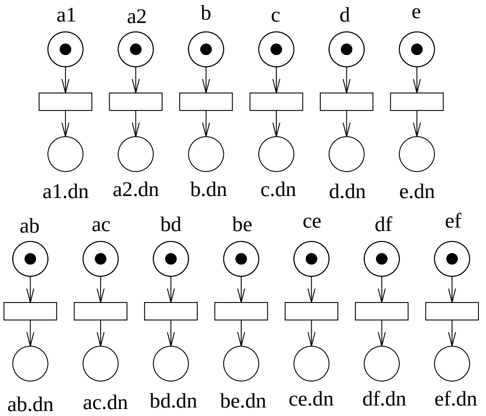
Figure 9: GSPN model without repair

column of places represent the nodes and links in our network in working condition. They all independently fail upon which a token is deposited in the corresponding place labeled with the suffix ".dn". A token in either $P1$, $P2$, $P3$ and $P4$ represents that that particular path is down because at least one node or link in that path has failed. Finally, a token in F indicates that there is no path for the client and the server to communicate and that the system has failed. Note that a token in F inhibits all the transitions in the net and hence no further change in marking can take place.

4.3 Stochastic Reward Nets

The specification of a system using GSPN's can still be tedious and troublesome. To remedy this, Ciardo et al. [9] introduced several structural extensions to GSPNs. Variable multiplicity arcs, enabling functions (also known as guards) for transitions, marking dependent arc multiplicities and timed transition priorities. The resulting net with all these extensions and the capability of assigning a real valued reward to any marking is termed as a stochastic reward net (SRN).

The SRN reliability model for our network example is shown in Figure 10. The net specification now only contains places and transitions which represent failure behavior for each individual component. System fault tree is "encoded" as a set of boolean functions (as opposed to a series of immediate transitions and places in GSPN) and the inhibitor arcs in the GSPN model are replaced by a simple halting condition. The structural enhancements, therefore, allow the modeler to keep a simpler representation of the system at the net level.



Name	Function
G1	(#(a1.dn) == 1) ∧ (#(a2.dn) == 1)
G2	(#(b.dn) == 1) ∨ (#(be.dn) == 1) ∨ (#(e.dn) == 1)
G3	(#(bd.dn) == 1) ∨ (#(d.dn) == 1) ∨ (#(df.dn) == 1)
G4	(#(ac.dn) == 1) ∨ (#(c.dn) == 1) ∨ (#(ce.dn) == 1)
G5	G1 ∨ (#(ab.dn) == 1) ∨ (#(b.dn) == 1) ∨ G3
G6	G1 ∨ G4 ∨ (#(e.dn) == 1) ∨ (#(ef.dn) == 1)
G7	G1 ∨ G2 ∨ (#(ab.dn) == 1) ∨ (#(ef.dn) == 1)
G8	G1 ∨ G2 ∨ G3 ∨ G4

Halting Condition
if (G5 ∧ G6 ∧ G7 ∧ G8) **then** disable all the transitions

Figure 10: SRN model without repair

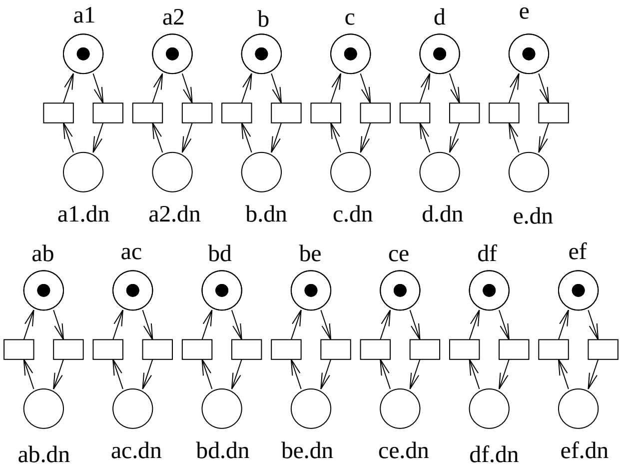


Figure 11: SRN model with independent repair

Figure 11 shows the SRN model of the network when repairs are allowed. Each component has its own repair facility and is independent of failure and repair of any other component. The SRN simply consists of the failure and repair behavior of the system. The system being up or down is still encoded by exactly the same functions as listed in Figure 10 with the exception that there is no halting condition. Instead, a reward function, which when evaluated by solving the underlying Markov reward model, gives the instantaneous and steady state availability of the system. The reward rate r for availability of this system is assigned in terms of the boolean functions defined earlier as:

```
if ($G5 \wedge G6 \wedge G7 \wedge G8$)
    return 0; (system is unavailable)
else
    return 1 (system is available);
```

We saw earlier that both reliability and availability evaluation is possible using Fault trees and reliability graphs as long as the individual component behaviors are stochastically independent. Unfortunately, in real world situations, this is not always true. Let us consider our network again but with only a single repair facility. This means that whenever a component fails, it might have to wait for the start of repairs. Figure 12 shows the SRN model. The repair policy is priority based but non-preemptive. For example, if the link *df* failed first and while it is undergoing repairs, the node *b* as well as the links *be* and *ce* failed thus causing the communication to stop. Upon completion of the current repair, the facility must choose one of the already failed nodes or links. Note that if

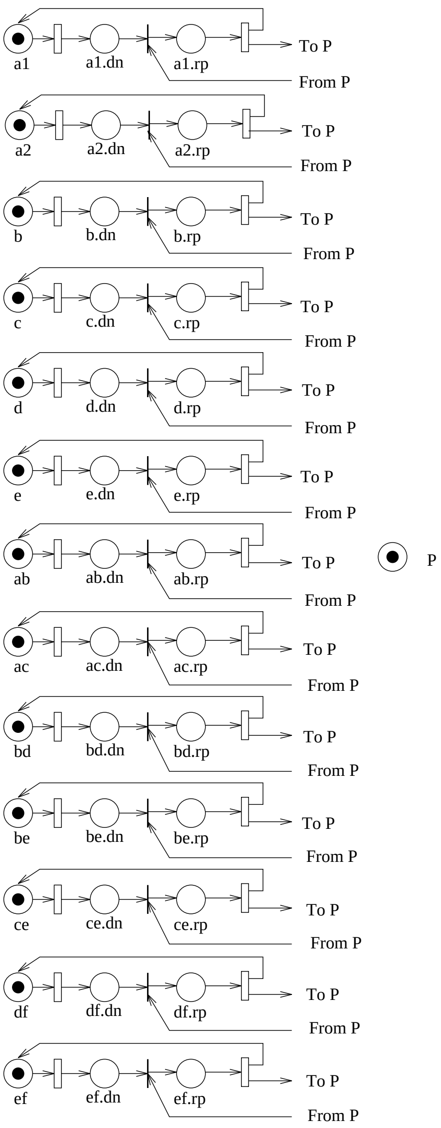


Figure 12: SRN model with dependent repair

the link *be* is repaired next, the communication will still be stopped. However, if the node *b* or the link *ce* is repaired, the communication can be resumed. In the modeled policy, the repairs are performed according to preassigned priorities. Other possible policies include FCFS or even marking dependent repair. For further examples of dependability modeling using SRN's, the reader is referred to [22, 21].

4.4 Results

The SRN models were solved using the SPNP (Stochastic Petri Net Package). SPNP [10] provides support for specifying the SRN using a "C" like language and allows for the modeler to do steady state, transient, cumulative transient and sensitivity analysis. Figure 13 shows the instantaneous availability for network with a single repair facility overlayed on

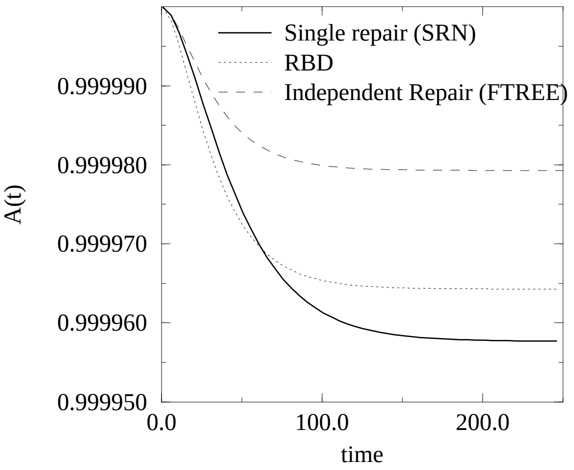


Figure 13: Instantaneous Availability of the network

the instantaneous availability plots obtained earlier in Section 3. The results obtained by solving the SRN model shown in Figure 11 match with those obtained from the FTREE model. As expected, the steady state availability for the case of a single repair facility obtained as 0.99995768 is lower than the independent repair case.

5 Advanced Modeling Techniques

The modeling framework presented so far allows the solution of stochastic problems enjoying the **Markov property**: *the probability of any particular future behavior of the process, when its current state is known exactly, is not altered by additional knowledge concerning its past behavior* [29]. If the past history of the process is completely summarized in the current state and is independent of the current time, then the process is said to be *(time-) homogeneous*. Otherwise, the exact characterization of the present state needs the associated time information, and the process is said to be *non-homogeneous*. A wide range of real problems fall in the class of Markov models (both homogeneous and non-homogeneous). However, some important aspects of system behavior in a dependability model cannot be easily captured in a Markov model. The common characteristic these problems share is that the Markov property is not valid (if valid at all) at all time instants. This category of problems is jointly referred to as *non-Markovian* models and can be analyzed using several approaches:

- supplementary variables;
- phase-type expansions; and
- Markov renewal theory.

5.1 Supplementary Variables

This method, originally discussed in [12], allows for the solution of dependability models when the lifetime and/or repair distributions of network components are non-exponential. It is the most direct method of solving the modeling problem and is based on the inclusion of sufficient supplementary variables in the specification of the state of the system to make the whole process Markovian. In dependability models the supplementary variables are the times expended in repairs and ages of network components. The purpose of the added suplementary variables is to include all necessary information about the history of the stochastic process. The resulting Markov process is in continuous time and has a state space which is multidimensional of mixed type, partly discrete and partly continuous.

Since, after the inclusion of the supplementary variables, the stochastic process describing the system behavior is now memoryless, then it is possible to derive the Chapman-Kolmogorov equations describing the dynamic behavior for such a process. The resultant set of ordinary or partial differential equations can be defined together with boundary conditions and analyzed.

5.2 Phase Type Expansions

The use of phase type distributions dates back to the pionner work of Erlang on congestion in telephone systems at the beginning of this century [7]. His approach (named *method of stages*), although simple, was very effective in dealing with non-exponential distributions and has been considerably generalized since then. The age (repair time) of a component is assumed to consist of a combination of stages each of which is exponentially distributed. The whole process becomes Markovian provided that the description of the state of the system contains the information as to which stage of the component state duration has been reached. The division into stages is an operational device and may not necessarily have any physical significance, and any distribution with a rational Laplace transform can, in principle, be represented exactly by a phase type expansion.

The application of this technique involves the following steps [27, 19]:

- *Selection of a Stage Combination:* When the distribution has a rational Laplace transform, the stage combination can be found by examining the roots of this tranform. In other cases of known probability distributions or of directly fitting the data, a suitable guess has to be made. Several stage approximations are described in detail in [19].

- *Determination of Parameters:* When a stage model has been selected, the next step is the derivation of its parameters from those of the distribution being approximated. There are no general explicit formula for directly deriving the stage model parameters and a numerical solution is required.

The basic phase type expansion techniques approximate a non-exponential distribution by connecting dummy stages with independent and exponential sojourn time distribution in series or parallel (or combination of both). A process with sequential phases gives rise to hypoexponential or an Erlang distribution, depending upon whether or not the phases have identical distributions. Instead, if a process consists of alternate phases (parallel connection) then the overall distribution is hyperexponential. The basic instrument when sellecting one of these distributions to represent a non-exponential interval is given by the *coefficient of variation.* The coefficient of variation, C_X, of a random variable is a measure of deviation from the exponential distribution and is given by

$$C_X = \frac{\sigma_X}{E[X]},$$

where σ_X is the standard deviation of the random variable and $E[X]$ is its expectation. This coefficient varies as follows according to the sellected distribution:

C_X	**Distribution**
> 1	Hyperexponential
1	Exponential
< 1	Hypoexponential Erlang
0	Deterministic

Important generalizations of the basic stage devices are the Coxian distributions, Phase Type (PH), and Generalized Hyperexponential (GH).

5.2.1 Coxian Distributions

Cox [13] extended the concept of stages by considering the class of distributions having rational Laplace transforms. He showed that the method of stages can still be employed for this class if one is willing to tolerate stages having complex roots. The basic structure of an Coxian distribution is depicted in Figure 14. In SHARPE terminology, such distributions are called exponomials.

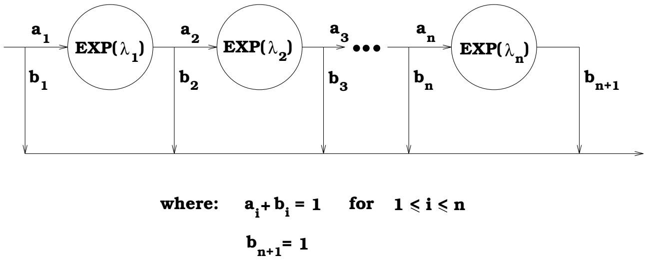


Figure 14: Coxian distribution structure.

5.2.2 Phase-type Distributions

Another important category of phase type expansions are the PH distributions. Neuts [23] popularized the class of PH distributions, which correspond to the time until absorption in finite dimensional Markov chains with at least one absorbing state. That is, $F(t)$ is PH if it can be written as

$$F(t) = 1 - \alpha e^{\mathbf{Q} t} \mathbf{1}$$

where $\mathbf{Q}$ is the infinitesimal generator matrix of an (n+1)-state CTMC with absorbing state (n+1). The vector $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ is the vector of initial state probabilities at $t = 0$, and the vector $\mathbf{1}$ is an n-dimensional column vector of all ones. The entries in the generator matrix $(q_{ij}, i \neq j)$ represent the instantaneous rate of the transition from state i to state j. Each component of $e^{\mathbf{Q}t}\mathbf{1}$ corresponds to a phase-type distribution that results from starting at a particular state. Therefore, the CDF $F(t)$ can be interpreted as a mixture of phase-type distributions, that is,

$$F(t) = \sum_{i=1}^{n} \alpha_i [1 - (e^{\mathbf{Q}t}\mathbf{1})_i].$$

The major advantage of using PH distributions is computational, instead of dealing with differential equations, complex variables and numerical integration, they can be handled using matrix methods [5]. A drawback of PH distributions is their non-uniqueness of representation. Many different combinations of defining parameters lead to the same CDF.

5.2.3 Generalized Hyperexponential

Generalized hyperexponential distribution functions were proposed by Botta and Harris [5] and are of the form

$$F(t) = \sum_{i=1}^{n} a_i (1 - e^{-\lambda_i t}),$$

with $\sum_{i=1}^{n} a_i = 1,\ a_i \in \mathcal{R},\ \lambda_i > 0$. They are extensions of the hyperexponential distributions, which are of the same form but without the additional requirements that the coefficients $\{a_i\}$ be positive. This added freedom makes the GH distributions extremelly versatile. The GH family has the same computational advantage over transform methods (e.g., Coxian distributions) that the PH family has, namely, avoidance of complex arithmetic.

5.2.4 Additional Information and Examples

Botta, Harris and Marchal [6] provide an excellent comparison among several commonly used classes of approximating distributions, including Erlang, Coxian and PH types. For examples on how to incorporate non-exponential distributions into stochastic Petri nets we suggest [8]. [19] discusses a complete approach to phase approximation, including choice of phase approximation class, numerical fitting of appropriate parameters, and implementation of the approximation approach in a modeling toolkit.

5.3 Markov Renewal Theory

A set of techniques that proved very powerful for the solution of non-Markovian models of computer networks is based on concepts grouped under the umbrella of Markov renewal theory [15, 11], a collective name that includes Markov renewal sequences (MRS's), and two other important classes of stochastic processes with embedded MRS's, named semi-Markov processes (SMP's) and Markov regenerative processes (MRGP's). In this subsection we review the definitions and some of the concepts of Markov renewal theory. We start with the mathematical definition of Markov renewal sequences. We then proceed with a study of processes in which an embedded MRS's can be identified. We conclude illustrating the equations that provide the solution of Markov renewal problems and suggest references for especific applications of the theory solving computer network problems.

5.3.1 Markov Renewal Sequence

Assume the system we are modeling is described by a stochastic process $\mathbf{Z} = \{Z_t; t \in \mathcal{R}_+ = [0, \infty)\}$, and we observe that at these particular times the stochastic process $\mathbf{Z}$ exhibits the Markov property. In this scenario we are dealing with a countable collection of renewal processes progressing simultaneously such that successive renewals form a discrete-time Markov chain (DTMC). The superposition of all the identified renewal processes gives the points $\{S_n; n \in \mathcal{N}\}$, known as *Markov renewal moments*, and together with the states of the DTMC defines a **Markov renewal sequence**.

In mathematical terms, the bivariate stochastic process $(\mathbf{X}, \mathbf{S}) = \{X_n, S_n; n \in \mathcal{N}\}$ is a **Markov renewal sequence** if it satisfies

$$Pr\{X_{n+1} = j, S_{n+1} - S_n \leq t \mid X_0, ..., X_n; S_0, ..., S_n\} = Pr\{X_{n+1} = j, S_{n+1} - S_n \leq t \mid X_n\}$$

for all $n \in \mathcal{N}$, $j \in \mathcal{E}$, $t \in \mathcal{R}_+$, and such that $S_0 \leq S_1 \leq S_2 \leq ...$, assuming $S_0 \doteq 0$. In practical problems usually the MRS is assumed *time-homogeneous*; that is, the conditional transition probabilities $K_{i,j}(t)$, where

$$K_{i,j}(t) \doteq Pr\{X_{n+1} = j, S_{n+1} - S_n \leq t \mid X_n = i\}$$

are independent of n for any $i, j \in \mathcal{E}$, $t \in \mathcal{R}_+$. Therefore, we can always write

$$K_{i,j}(t) = Pr\{X_1 = j, S_1 \leq t \mid X_0 = i\},$$

The matrix of transition probabilities $\mathbf{K}(t) = \{K_{i,j}(t) : i, j \in \mathcal{E}, t \in \mathcal{R}_+\}$ is called the **kernel** of the MRS.

5.3.2 Semi-Markov Processes

Given an MRS $(\mathbf{X}, \mathbf{S})$ with state space $\mathcal{E}$ and kernel $\mathbf{K}(t)$, we can introduce the counting process

$$\mathbf{N}(t) \doteq sup\{n : S_n \leq t\}, \qquad \forall t \in \mathcal{R}_+$$

to count the number of Markov renewal moments up to time t, but not considering the one at zero. Using the counting process just defined, we introduce the process $\mathbf{Y} = \{Y_t; t \in \mathcal{R}_+\}$ defined by

$$\begin{aligned} Y_t &\doteq X_{\mathbf{N}(t)} \\ &= X_n, \qquad if\ S_n \leq t < S_{n+1} \end{aligned}$$

called **semi-Markov process**. An SMP is a stochastic process which moves from one state to another within a countable number of states with the successive states visited forming a discrete-time Markov chain, and that the time required for each successive move is a random variable whose distribution function may depend on the two states between which the move is being made. From the SMP definition it should be observed that the process only changes state at the Markov renewal moments S_n.

5.3.3 Markov Regenerative Processes

A stochastic process $\mathbf{Z} = \{Z_t; t \in \mathcal{R}_+\}$ is called *regenerative* if there exist time points at which the process probabilistically restarts itself. Such random times when the future of $\mathbf{Z}$ becomes a probabilistic replica of itself are named *times of regeneration* for $\mathbf{Z}$. This concept may be weakened by letting the future after a time of regeneration depend also on the state of an MRS at that time. We then say that $\mathbf{Z}$ is a Markov regenerative process.

MRGP's are stochastic processes $\{Z_t; t \in \mathcal{R}_+\}$ that exhibit embedded MRS's **(X,S)** with the additional property that all conditional finite distributions of $\{Z_{t+S_n}; t \in \mathcal{R}_+\}$ given $\{Z_u; 0 \leq u \leq S_n, X_n = i\}$ are the same as those of $\{Z_t, t \in \mathcal{R}_+\}$ given $X_0 = i$. [21] As a special case, the definition implies that

$$Pr\{Z_{t+S_n} = j \mid Z_u, 0 \leq u \leq S_n, X_n = i\} =$$
$$Pr\{Z_t = j \mid X_0 = i\}$$

In contrast to SMP's, state changes are allowed between two consecutive Markov renewal moments in MRGP's.

5.4 Solution of Problems

Let $\mathbf{Z} = \{Z_t; t \in \mathcal{R}_+\}$ be an MRGP with state space $\mathcal{F}$, whose embedded MRS is $(\mathbf{X}, \mathbf{S}) = \{X_n, S_n; n \in \mathcal{N}\}$ with kernel matrix $\mathbf{K}(t)$. For such a process we can define a matrix of conditional transition probabilities as:

$$V_{i,j}(t) \doteq Pr\{Z_t = j \mid Z_0 = i\}, \ \forall i \in \mathcal{E}, \forall j \in \mathcal{F}, \forall t \in \mathcal{R}_+$$

In many practical problems involving Markov renewal processes, our primary concern is finding ways to effectively compute $V_{i,j}(t)$ since several measures of interest (e.g., reliability and availability) are related to the conditional transition probabilities of the stochastic process.

At any instant t, the conditional transition probabilities $V_{i,j}(t)$ of $\mathbf{Z}$ can be computed as:

$$\begin{aligned} V_{i,j}(t) &= Pr\{Z_t = j, S_1 > t \mid Z_0 = i\} + \\ & \sum_{k \in \mathcal{E}} \int_0^t dK_{i,k}(u) V_{k,j}(t-u) \end{aligned}$$

for all $i \in \mathcal{E}$, $j \in \mathcal{F}$, and $t \in \mathcal{R}_+$. If we define matrix $\mathbf{E}(t)$ by

$$E_{i,j}(t) \doteq Pr\{Z_t = j, S_1 > t \mid Z_0 = i\}.$$

Then, the set of integral equations $V_{i,j}(t)$ defines a **Markov renewal equation**, and can be expressed in matrix form as

$$\mathbf{V}(t) = \mathbf{E}(t) + \int_0^t d\mathbf{K}(u)\mathbf{V}(t-u)$$

where the Lebesgue-Stieltjes integral is taken term by term. The Markov renewal equation represents a set of coupled *Volterra integral equations of the second kind* and can be solved in time-domain or in Laplace-Stieltjes domain. For a discussion of approaches to solve these equations see [14, 28]. References for the application of Markov renewal theory in the solution of performance and realiability/availability models of computers networks are [17, 16, 18]

6 Conclusion

In this paper, we reviewed reliability and availability modeling techniques by applying them to a communication network example. Two categories of models viz., the non-state space based models which include reliability block diagrams, fault trees and reliability graphs and fault trees and the state space based models which include stochastic Petri nets and its variants were discussed. We showed powers and limitations of various models by solving for reliability and availability of the network example. In all of the above, the failure times and the repair times are assumed to be exponentially distributed which may not hold. Therefore, finally, we also described three methods of modeling a system which has nonexponentially distributed failure or repair times.

References

[1] Ajmone-Marsan A., Balbo G. and Conte. G., A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Trans. Comp. Systems*, Vol 2. No. 2, May 1984, pp. 93-122

[2] ATM Forum, *ATM User-Network Interface Specification, Version 3.0*, Prentice Hall (ISBN 0-13-225863-3), September, 1993.

[3] ATM Forum, *ATM User-Network Interface Specification, Version 3.1*, July 21, 1994.

[4] Balakrishnan, Meera and Trivedi, K. S., Componentwise Decomposition for an Efficient Reliability Computation of Systems with Repairable Components, *Proc. Twenty-fifth International Symposium on Fault-Tolerant Computing,* Pasadena, CA, July 1995.

[5] Botta, R.F. and Harris, C.M., Approximation with Generalized Hyperexponential Distributions: Weak Convergence Results, *Queueing Systems - Theory and Applications*, 1(2):169–190, 1986.

[6] Botta, R.F., Harris, C.M., and Marchal, W.G., Characterizations of Generalized Hyperexponential Distribution Functions, *Communications in Statistics - Stochastic Models*, 3(1):115–148, 1987.

[7] Brokemeyer, F., Halstron, H. S. and Jensen, A., The Life and Works of A.K. Erlang, *Transactions of the Danish Academy of Technical Sciences*, 2, 1948.

[8] Chen, P., Bruel, S.C., and Balbo, G., Alternative Methods for Incorporating Non-Exponential Distributions into Stochastic Timed Petri Nets, *PNPM'89*, Kyoto, Japan, pp. 187–197, December 11-13, 1989.

[9] Ciardo G., Blakemore A., Chimento P. F., Muppala J. K. and Trivedi K. S., Automated generation and analysis of Markov reward models using stochastic reward nets. In *Linear Algebra, Markov Chains, and Queueing Models*, Carl Meyer and R.J. Plemmons (eds.), IMA Volumes in Mathematics & its Applications, Vol. 48, pp 145-191, Springer Verlag, Heidelberg, 1993.

[10] Ciardo, G., Muppala J. and Trivedi, K. S., SPNP: Stochastic Petri Net Package, *Proc. Third Int. Workshop on Petri Nets and Performance Models (PNPM89)*, Kyoto, pp. 142 - 151, 1989.

[11] Çinlar, E., *Introduction to Stochastic Processes*, Prentice-Hall, Englewood Cliffs, 1975.

[12] Cox, D.R., The Analysis of Non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables, *Proc. Camb. Philos. Soc.*, 51(3): 433–441, 1955.

[13] Cox, D.R., Use of Complex Probabilities in the Theory of Stochastic Processes, *Proc. Camb. Philos. Soc.*, 51:313–318, 1955.

[14] German, R., Logothetis, D., and Trivedi, K., Transient Analysis of Markov Regenerative Stochastic Petri Nets: a Comparison of Approaches, in *Petri Net Performance Models, PNPM'95*, 1995.

[15] Kulkarni, V.G., *Modeling and Analysis of Stochastic Systems*, Chapman & Hall, London, 1995.

[16] Logothetis, D. and Trivedi, K., Reliability Analysis of Various Station Attachment Schemes in a FDDI Token Ring, *Proceedings of the IEEE INFOCOM 93*, San Francisco, CA, March 1993.

[17] Logothetis, D. and Trivedi, K., The Effect of Detection and Restoration Times for Error Recovery in Communication Networks, *MILCOM*, 1996.

[18] Logothetis, D. and Trivedi, K., Time–Dependent Behavior of Redundant Systems with Deterministic Repair, in *Computations with Markov Chains*, edited by W.J. Stewart, Kluwer Academic Publishers, Norwell, pp. 135–150, 1995.

[19] Malhotra, M. and Reibman, A., Selecting and Implementing Phase Approximations for Semi-Markov Models, *Stochastic Models* , 9(4), 1993.

[20] Malhotra, M. and Trivedi, K. S., Power-Hierarchy of Dependability Model Types, *IEEE Transactions on Reliability,* Vol. 43, No. 2, pp. 493-502, Sept. 1994.

[21] Malhotra, M. and Trivedi, K. S., Dependability Modeling Using Petri-Nets, *IEEE Transactions on Reliability,* Vol. 44, No. 3, pp. 428-440, Sept., 1995.

[22] Muppala, J., Ciardo, G. and Trivedi, K. S., Stochastic Reward Nets for Reliability Prediction, *Communications in RMS*, July 1994.

[23] Neuts, M.F., Renewal Process of Phase Type, *Naval Research Logistics Quartely*, 25(3):445–454, 1978.

[24] Onvural, Raif O., *Asynchonous Transfer Mode Networks: Performance Issues*, Artech House, 1995.

[25] Peterson, J. L., *Petri net theory and the modeling of systems*, Prentice Hall, 1981.

[26] Sahner, Robin A., Trivedi, Kishor S., and Puliafito, Antonio, *Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package*, Kluwer Academic Publishers, 1996.

[27] Singh, C., Billington, R., and Lee, S.Y., The Method of Stages for Non-Markovian Models," *IEEE Transactions on Reliability*, 26(2):135–137, June 1977.

[28] Telek, M., Bobbio, A., Jereb, L., and Trivedi, K., Steady state analysis of Markov Regenerative SPN with age memory policy, in *Performance Tools and MMB '95,* Heidelberg (Germany), 1995.

[29] Trivedi, Kishor S., *Probability & Statistics with Reliability, Queuing, and Computer Science Applications*, Prentice-Hall, 1982.