The Task Completion Time in Degradable Systems

Andrea Bobbio
Dipartimento di Informatica
Università di Torino, 10149 Torino, Italy

Miklós Telek
Department of Telecommunications
Technical University of Budapest, 1521 Budapest, Hungary

Abstract

This paper assumes a user-oriented point of view in examining the performability of a dependable computing system. The investigated performability measure is the effective time that a task, with an assigned work requirement, takes to be executed by the system. Assuming that the system changes its performance characteristics randomly in time, the stochastic model representing the task completion time is formulated and analyzed. Applications and extensions of the basic model are discussed. Finally, the completion time model is reformulated in the language of stochastic Petri nets, and possible computational approaches are illustrated.

1 Introduction

The completion time of a task measures the time that a task takes to be executed by a computing system. If the system changes its computational power randomly in time during the execution, the task completion time is a random variable. The analytical and numerical computation of the cumulative distribution function (*Cdf*) of the task completion time is the subject of this chapter.

The adopted modeling framework consists of describing the behavior of the system configuration in time by means of a stochastic process, called the *structure-state process*, and by associating to each state of the structure-state process a non-negative real constant representing the effective working capacity or performance level of the system in that state. The real variable associated to each structure-state is called the *reward rate* [36]. The structure-state process together with the reward rates forms the *Stochastic Reward Model (SRM)* [63].

The properties of stochastic reward processes have been studied since a long time [50, 21, 41, 42, 36], however, only recently *SRM*'s have received attention as a modeling tool in performance/reliability evaluation. Indeed, the possibility of associating a reward variable to each structure state increases the descriptive power and the flexibility of the model.

Different interpretations of the structure-state process and of the associated reward structure give rise to different applications [53]. Common assignments of the reward rates are: execution rates of tasks in computing systems (the computational capacity) [5, 67], number of active processors (or processing power) [7, 34], throughput [52, 29, 35], average response time [37, 43, 47] or response time distribution [71, 62, 72]. The classical reliability theory [4] can be viewed as a particular case of *SRM* obtained by constraining the reward rates to be binary variables.

Two main different points of view have been assumed in the literature when dealing with *SRM* for degradable systems [45]. In the *system-oriented* point of view the most significant measure is the total amount of work done by the system in a finite interval. The accumulated reward is a random variable whose *Cdf* is called the *performability* [51]. Various numerical techniques for the evaluation of the performability have appeared in the literature: [38, 24, 33, 66, 25, 60, 59, 26]. In the *user-oriented* (or *task-oriented*) point of view the system is regarded as a server, and the emphasis of the analysis is on the ability of the system to provide a prescribed service in due time. Consequently, the most characterizing measure becomes the probability of accomplishing an assigned task in a given time. The task-oriented point of view is a more direct representation of the quality of service, which, in turn, is the main target of a dependable computation.

Gaver [28] analyzed the distribution of the completion time for a two-state server with different mechanisms of interruption and recovery policies. Extensions to the above model were provided in [56], while the completion time problem for fault tolerant computing systems was addressed in [14]. A unified formulation to the system-oriented and the user-oriented point of view was provided by Kulkarni et al. in [45, 46, 58]. An alternative interpretation of the completion time problem can be given in terms of the hitting time of an appropriate cumulative functional [21] against an absorbing barrier equal to the work requirement. The definition of a cumulative functional was first suggested by Kulkarni et al. [45] and then explicitly exploited in [10], where the completion time was modeled as a first hitting time against an absorbing barrier. This interpretation leads the above problem into the main stream of absorption problems in stochastic models and has proved to be useful in association with stochastic Petri nets [8] and with the extension to multi-reward models [9, 10].

In Section 2, the completion time problem is formulated as a first passage time across an absorbing barrier. The distribution of the completion time is derived in Section 3, in the Laplace transform domain and under the hypothesis that all the states pertain to the same preemption class. Section 4 illustrates some applications and extensions. Section 5 shows how to represent the formulated non-Markovian stochastic model by means of Petri nets, and compares the results obtained from two non-Markovian *PN*-based models on a simple example. Section 6 summarizes the chapter.

2 The barrier hitting problem

Given that $F(t)$ is a *Cdf*, the Laplace transform *(LT)* $F^*(s)$ and the Laplace-Stieltjes transform *(LST)* $F^{\sim}(s)$ are given by, respectively:

$$F^*(s) = \int_0^\infty e^{-st} F(t)\, dt \qquad ; \qquad F^\sim(s) = \int_0^\infty e^{-st}\, d\, F(t)$$

Let the structure-state process $(X(t),\ t \geq 0)$ be a right-continuous semi-Markov process [21, 44] defined over a discrete and finite state space $\mathcal{S}$ of cardinality n. We denote by H the time duration until the first embedded time point of the semi-Markov process starting from state i at time 0 ($X(0) = i$), and by $\mathbf{p}(0)$ the row vector of the initial probabilities. Let $\mathbf{K}(t) = [K_{ij}(t)]$ be the kernel of the semi-Markov process. The generic element

$$K_{ij}(t) = \mathbb{P}\left\{H \leq t, X(H) = j | X(0) = i\right\}$$

(with $i, j = 1, ..., n$) is the distribution of H starting in state i at time 0 supposed that a transition to state j took place. Moreover,

$$K_i(t) = \mathbb{P}\left\{H \leq t | X(0) = i\right\} = \sum_{j=1}^{n} K_{ij}(t) \qquad\qquad (i = 1, ..., n)$$

is the distribution of H starting in state i at time 0 independent of the state reached after the first embedded time point. The probability of jumping from state i to state j at time $H = t$ can be defined in terms of the kernel elements:

$$\mathbb{P}\left\{X(H) = j | H = t, X(0) = i\right\} = \frac{dK_{ij}(t)}{dK_i(t)}$$

Let $r_{X(t)}$ be a non-negative real-valued function defined as:

$$r_{X(t)} = r_i \quad \text{if} \quad X(t) = i \quad ; \quad \text{with} \quad r_i \geq 0 \quad \text{and} \quad i = 1, 2, \ldots, n. \tag{1}$$

$r_{X(t)}$ represents the instantaneous reward associated to state i. We now define a functional $Y(t)$ that represents the accumulation of reward in time. $Y(t)$ is a stochastic process that depends on $X(\tau)$ for $\tau \leq t$ [21]. During the sojourn of $X(t)$ in state i between t and $(t + dt)$, $Y(t)$ increases at the rate $r_i\, dt$. However, a transition in $X(t)$ may induce a modification in the accumulation process depending whether the transition entails a *loss of work*, or *no loss of work*. A transition which does not entail a loss of the work already accumulated by the system on the task in execution is called *preemptive resume (prs)*, and its effect on the model is that the functional $Y(t)$ resumes the previous value in the new state. A transition which entails a loss of the work done by the system on the task in execution is called *preemptive repeat (prt)*, and its effect on the model is that the functional $Y(t)$ is reset to 0 in the new state.

A possible realization of the accumulation process $Y(t)$ is shown in Figure 1. The transition from state j to state k is of *prs* type while the transitions from state k to i and from i to j are *prt*. In order to model the completion time problem, let W be the actual work requirement of a task. W represents the time that a task would

require to be executed in isolation on a perfect system. In a degradable environment, the task completes as soon as the work accumulated by the system reaches the actual work requirement for the first time. Hence, W acts as an absorbing barrier for the functional $Y(t)$. With reference to Figure 1, the task completion time is the time at which $Y(t)$ hits the barrier W for the first time.

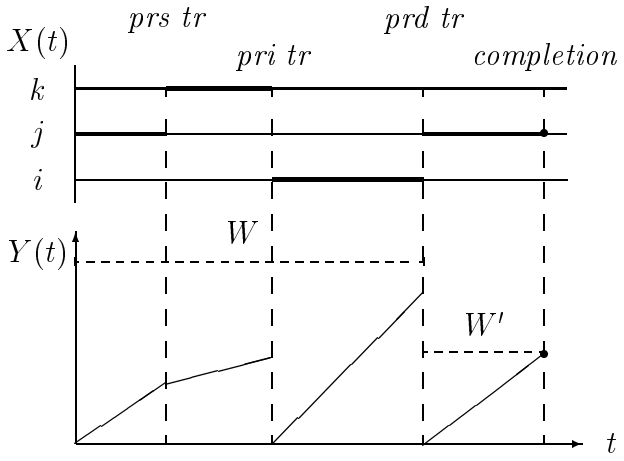

Figure 1 - The behavior of the functional $Y(t)$ versus time.

We assume, in general, that W is a random variables with distribution $G(x)$ with support on $[0,\ \infty)$. The degenerate case in which W is deterministic and the distribution $G(x)$ becomes the unit step function $U(x)$ located at $W = x$, can be considered as well. When W is not deterministic and the preemption policy is *prt*, two cases arise depending whether the repeated task has an identical work requirement as the original preempted task (*preemptive repeat identical (pri)* - policy), or a different work requirement sampled from the same distribution (*preemptive repeat different (prd)* - policy). With reference to Figure 1, it is assumed that the transition from state k to state i is *pri* and the transition from state i to state j is *prd*. According to the previous assumptions, the accumulated reward $Y(t)$ is reset and the same value W of the barrier is retained when jumping from state k to state i since the corresponding transition is *pri*. On the other hand, since transition from i to j is *prd*, the work requirement W is resampled in state j assuming a new value W' sampled from the same distribution and $Y(t)$ is reset.

For a structure-state process with only *prs* and *pri* transitions the barrier height W is constant up to the completion. In these cases, conditioned to a fixed value of the barrier height $W\ =\ x$, the completion time $T(x)$ is defined as:

$$T(x)\ =\ \min\left[t \geq 0\ :\ Y(t)\ =\ x\right]\ . \tag{2}$$

Let $F_T(t,x)$ be the conditional *Cdf* of the task completion time $T(x)$:

$$F_T(t,x)\ =\ \mathbb{P}\left\{T(x)\ \leq\ t\right\} \tag{3}$$

The unconditional completion time T is characterized by the following distribution:

$$F_T(t) \;=\; \mathbb{P}\,\{\,T \le t\} \;=\; \int_0^\infty F_T(t,x)\,dG(x) \tag{4}$$

and is the measure that can be evaluated if all the transitions are *prd*.

The distribution of the completion time, $F_T(t)$, incorporates the effect of a random variation of the execution speed consequent to a degradation and reconfiguration process, combined with the effect of the preemption and recovery policy on the execution of the task.

The following relationships between the different preemption policies can be easily established. If the work requirement W is an exponential random variable, the two policies *prs* and *prd* give rise to the same completion time (due to the memoryless property of the exponential distribution, the residual task requirement under the *prs* policy coincides with the resampled requirement under the *prd* policy). On the other hand, if W is deterministic, the two policies *pri* and *prd* are coincident (resampling a step function provides always the same constant value).

Moreover, assuming that the structure-states are all of *prs* type, so that no loss of reward occurs,

$$Y(t) = \int_0^t r_{X(\tau)}\,d\tau$$

and the distribution of the completion time is closely related to the distribution of the accumulated reward (*performability*) by means of the following relation:

$$\mathbb{P}\,\{Y(t) \le x\} \;=\; \mathbb{P}\,\{T(x) \ge t\} \tag{5}$$

Kulkarni et al. [45] derived the closed-form Laplace transform equations of $F_T(x,t)$ when $X(t)$ is a *CTMC* and all the states belong to the same preemption class. The extension to a semi-Markov process $X(t)$ whose state space is partitioned in the three preemption classes has been considered in [46]. Bobbio and Trivedi [13] studied the case where $X(t)$ is a *CTMC*, the work requirement W is a *PH* random variable [54] and the task execution policy is a probabilistic mixture of *prs* and *prd* policies. The combination of *prs* and *pri* policies has been investigated in [15] for the evaluation of the completion time of a program on a gracefully-degradable computing system.

3 The distribution of the completion time

A state whose outgoing transitions are all of *prs* type is called a *prs* state; similarly, a state whose outgoing transitions are all of *prd* (*pri*) type is called a *prd* (*pri*) state. The following closed form expressions for the *Cdf* of the completion time are derived under the hypothesis that the structure-state process is semi-Markov and all the states are of the same preemption class. A more general derivation, in which the states are allowed to belong to the three different defined preemption classes is in [46].

In order to evaluate (3), let us introduce the following vector valued functions $\mathbf{F}(t)$ and $\mathbf{F}(t,x)$ whose entries $F_i(t)$ and $F_i(t,x)$, $(i = 1,2,\ldots,n)$ are defined by:

$$F_i(t,x) \;=\; \mathbb{P}\{T(x) \le t \,|\, X(0) = i\} \;\;, \quad x \ge 0 \tag{6}$$

$$F_i(t) \;=\; \mathbb{P}\{T \;\leq\; t \,|\, X(0) \;=\; i\} \quad . \tag{7}$$

Notice that, when all the states are *prs* or *pri* the quantity to be evaluated is $F_i(t,x)$ while in the *prd* case only the function $F_i(t)$ can be derived. From the above definitions, it follows that:

$$F_T^{\sim}(s) \;=\; \mathbf{p}(0)\,\mathbf{F}^{\sim}(s) \;=\; \int_0^{\infty} \mathbf{p}(0)\,\mathbf{F}^{\sim}(s,x)\,dG(x) \quad . \tag{8}$$

Theorem 1 - Given that $X(t)$ is a semi-Markov process and all the states are of *prs* type, the double transform $\mathbf{F}^{\sim *}(s,w)$ satisfies the following equation:

$$F_i^{\sim *}(s,w) \;=\; \frac{r_i}{s \;+\; w\,r_i}\,[1 \;-\; K_i^{\sim}(s \;+\; w\,r_i)\,] \;+\; \sum_{j=1}^{n} K_{ij}^{\sim}(s \;+\; w\,r_i)\,F_j^{\sim *}(s,w) \tag{9}$$

Proof - Conditioning on the time until the first embedded time point in the initial state $H = h$, let us define:

$$F_i^{\sim}(s,x \,|\, H=h) \;=\; \mathbb{E}\,[exp(-s\,T)|W \;=\; x, X(0)=i, H=h] \;=$$

$$\begin{cases} exp(-sx/r_i) & \text{if } \; h\,r_i \geq x \\ exp(-sh)\,\displaystyle\sum_{j=1}^{n} \frac{dK_{ij}(h)}{dK_i(h)} F_j^{\sim}(s, x - hr_i) & \text{if } \; x \;>\; h\,r_i \end{cases} \tag{10}$$

In (10), two mutually exclusive events are identified: if $h\,r_i \geq x$, then $T = x/r_i$ or if $h\,r_i \;<\; x$ then a transition occurs to state j. Taking the LT with respect to x, we obtain:

$$\begin{aligned} F_i^{\sim *}(s,w \,|\, H=h) \;\;=\;\; & \int_{x=0}^{hr_i} exp(-\,w\,x\,)\,exp(-\,s\,x/r_i)\,dx \;+ \\ & exp[-(s \;+\; wr_i)\,h]\,\sum_{j=1}^{n} \frac{dK_{ij}(h)}{dK_i(h)}\,F_j^{\sim *}(s,\,w) \end{aligned} \tag{11}$$

Unconditioning with respect to H, (11) becomes:

$$\begin{aligned} F_i^{\sim *}(s,w) \;\;=\;\; & \int_{h=0}^{\infty}\int_{x=0}^{hr_i} exp[-(\,s \;+\; w\,r_i)x/r_i]dx\;dK_i(h) \;+ \\ & \int_{h=0}^{\infty} exp[-(s + w\,r_i)h]\;\sum_{j=1}^{n} F_j^{\sim *}(s,w)\;dK_{ij}(h) \end{aligned} \tag{12}$$

Finally, Equation (9) is obtained from (12) by evaluating the integrals $\Box$.

Corollary 2 - Under the assumptions of Theorem 1, given that $X(t)$ is a *CTMC* with infinitesimal generator $\mathbf{Q}$, the double transform $\mathbf{F}^{\sim *}(s,w)$ satisfies the following matrix equation [45]:

$$\mathbf{F}^{\sim *}(s,w) \;=\; [s\mathbf{I} \,+\, w\mathbf{R} \,-\, \mathbf{Q}]^{-1}\,\mathbf{r} \tag{13}$$

where

$$\mathbf{r} \;=\; [r_1,\, r_2, \ldots, r_n\,]^T \quad ; \qquad \mathbf{R} \;=\; diag\,[r_1,\, r_2, \ldots, r_n\,] \tag{14}$$

are a vector and a matrix of reward rates.

Proof - Equation (13) is obtained by substituting the following Markovian kernel in (9):

$$K_{ij}^{\sim}(s) \;=\; \begin{cases} \dfrac{q_{ij}}{s \,+\, q_i} & \text{if}: i \,\neq\, j \\ \\ 0 & \text{if}: i \,=\, j \end{cases}$$

where $q_i = \sum_{j=1;j\neq i}^{n} q_{ij}$ $\;\square$.

In general, the kernel $\mathbf{K}(t)$ of a semi-Markov process can have non-zero positive entries on the main diagonal. Therefore, from a given state i either a "virtual" transition into state i itself can take place or a real transition to a different state $j \neq i$. In the previously considered *prs* case, the accumulation process resumes the value reached by the total reward in the previous state and there is no need to distinguish between a jump into the same state or into a different one.

The situation is different in the *prt* case (either *prd* or *pri*). Indeed, a jump into a new state reset the accumulated reward while a jump into the same state should retain the same reward level. However, it has been shown in [70] that a semi-Markov kernel can be transformed into a canonical form in which the entries on the main diagonal are zero while preserving the same transition probabilities for all the transitions from i to j with $i \neq j$. The canonical representation of the semi-Markov kernel $\mathbf{K}^u(t)$ is given by [70]:

$$K_{ij}^{u\sim}(s) \;=\; \begin{cases} \dfrac{K_{ij}^{\sim}(s)}{1 \,-\, K_{ii}^{\sim}(s)} & \text{if}: i \,\neq\, j \\ \\ 0 & \text{if}: i \,=\, j \end{cases} \tag{15}$$

With a kernel in a canonical form, the problem of distinguishing between transitions into the same state or into a different state is avoided. Therefore, in the following we implicitly assume that the kernel is, or has been transformed, in a canonical form with zero entries on the main diagonal.

Theorem 3 - Given that $X(t)$ is a semi-Markov process and all the states are of *prd* type, the *LST* $F_i^{\sim}(s)$ satisfies the following equation:

$$\begin{aligned} F_i^{\sim}(s) \;\;=\;\; & \int_0^{\infty} exp(-s\,x/r_i)\,[1 \,-\, K_i(x/r_i)]\; dG(x) \;+ \\ & \sum_{j=1}^{n} F_j^{\sim}(s) \int_0^{\infty} exp(-sh)\;[1 \,-\, G(hr_i)]\;\; dK_{ij}(h) \end{aligned} \tag{16}$$

Proof - Conditioning on the time until the first embedded time point in the initial state $H = h$, let us define:

$$F_i^{\sim}(s \mid W = x, H = h) \;=\; \mathbb{E}\,[exp(-s\,T)|W \;=\; x, X(0) = i, H = h] =$$

$$\begin{cases} exp(-sx/r_i) & \text{if } \; hr_i \geq x \\ \\ exp(-sh) \sum_{j=1}^{n} \dfrac{dK_{ij}(h)}{dK_i(h)}\, F_j^{\sim}(s) & \text{if } \; x \;>\; h\,r_i \end{cases} \tag{17}$$

In (17), two mutually exclusive events are identified: if $h\,r_i \geq x$, then $T = x/r_i$ or if $h\,r_i < x$ then a transition occurs to state j ($j \neq i$) and a different task independent and with the same distribution is restarted. By unconditioning Equation (17) with respect to W and then with respect to H, we obtain:

$$\begin{aligned} F_i^{\sim}(s) \;\; = \;\; & \int_{h=0}^{\infty} \int_{x=0}^{h/r_i} exp(-s\,x/r_i)\; d\,G(x)\; d\,K_i(h) \\ & + \int_{h=0}^{\infty} \int_{x=h/r_i}^{\infty} \sum_{j=1}^{n} F_j(s)\; exp(-sh)\; d\,G(x)\; d\,K_{ij}(h) \end{aligned} \tag{18}$$

Solving the integrals in (18), the theorem is proved $\square$.

Corollary 4 - Under the assumptions of Theorem 3, given that $X(t)$ is a *CTMC* with infinitesimal generator $\mathbf{Q}$, the *LST* $F_i^{\sim}(s)$ satisfies the following Equation [45]:

$$F_i^{\sim}(s) \;=\; G^{\sim}\left(\frac{s \;+\; q_i}{r_i}\right) \;+\; \sum_{j=1;j\neq i}^{n} \frac{q_{ij}}{(s \;+\; q_i)} \left[\, 1 \;-\; G^{\sim}\left(\frac{s \;+\; q_i}{r_i}\right)\right] F_j^{\sim}(s) \tag{19}$$

Theorem 5 - Given that $X(t)$ is a semi-Markov process and all the states are of *pri* type, the *LST* $F_i^{\sim}(s,x)$ satisfies the following equation:

$$F_i^{\sim}(s,x) \;=\; exp(-s\,x/r_i)\,[1 \;-\; K_i(x/r_i)] \;+\; \sum_{j=1}^{n} F_j^{\sim}(s,x) \int_0^{x/r_i} exp(-sh)\; d\,K_{ij}(h) \tag{20}$$

Proof - Conditioning on the sojourn time in the initial state $H = h$, let us define:

$$F_i^{\sim}(s,x \mid H = h) \;=\; \mathbb{E}\,[exp(-s\,T) \mid X \;=\; x, X(0) = i, H = h] \;=$$

$$\begin{cases} exp(-sx/r_i) & \text{if } \; h\,r_i \geq x \\ \\ exp(-sh) \sum_{j=1}^{n} \dfrac{dK_{ij}(h)}{dK_i(h)}\, F_j^{\sim}(s,x) & \text{if } \; x \;>\; h\,r_i \end{cases} \tag{21}$$

Unconditioning with respect to H yields Equation (20) $\Box$.

Corollary 6 - Under the assumptions of Theorem 5, given that $X(t)$ is a $CTMC$ with infinitesimal generator $\mathbf{Q}$, the LST $F_i^{\sim}(s,x)$ satisfies the following Equation [45]:

$$F_i^{\sim}(s,x) \;=\; exp[-(s+q_i)\,x/r_i] + \sum_{j=1;j\neq i}^{n} \frac{q_{ij}}{s\;+\;q_i}(\,1 - exp[-(s+q_i)\,x/r_i])\;F_j^{\sim}(s,x) \quad (22)$$

4 Applications and Extensions of the Basic Model

In this section we present some applications and extensions of the basic model, considered so far in the literature.

Binary reward variables - When the reward rates are constrained to be binary variables, a binary partition of the state space is induced. Classical reliability-availability models fall in this class. The conceptual framework, formulated in the previous sections, offers an unified view to subtle reliability problems in which the system catastrophic failure depends on the duration of the downtime. The problem has a long history in the reliability literature [27, 64, 61, 65] and can be formulated in terms of the completion time of a "virtual task" whose work requirement is equal to the assigned downtime threshold [57]. If the down state is either of *pri* or *prd* type, a fatal failure occurs as soon as a single downtime greater than the threshold is encountered, while if the down state is of *prs* type, the fatal failure occurs when the threshold level is exceeded by the total accumulated down time. Nicola et al. [57] have calculated the completion time under fairly more general conditions, and have derived several related measures from the knowledge of the completion time distribution.

State space partition in preemption classes - The expressions in Section 3 are derived under the simplifying hypothesis that all the structure states of $X(t)$ belong to the same preemption class. A natural and useful extension is to consider a partition of the state space into different preemption classes. The accumulation of the reward is thus resumed or reset according to the characteristics of the state just abandoned at the transition. In [46], closed-form Laplace transform solutions are provided when all the three types of preemption policies are eventually present in the system, with a semi-Markov structure-state process. More general task execution processes can be modeled and estimated, and different kinds of failures can be taken into account. The same authors [58] have further extended their analysis, by considering a stream of jobs arriving at the server according to a Poisson process.

The completion time of programs - A specialized application of the above theory has been devoted to study the execution time of programs on computing systems. In their pioneering work Castillo and Siewiorek [14] have considered the time required to correctly execute a program, taking into account hardware reliability, software (operating system)

reliability, the workload of the system while the program is executing, and the type and amount of resources required to execute the program. The hardware and software reliabilities and the workload and resource characteristics, contribute to the definition of the random environment in which the task is performed, and are represented by $X(t)$.

The evaluation of the completion time of programs executed on degradable systems with different types of checkpointing mechanisms is the subject of [15]. By a combination of *prs* and *pri* kinds of interruptions, and the consideration of block structured programs, the authors are able to compare different recovery mechanisms at different levels of nesting in the program structure.

Multiple reward models - The simultaneous execution of parallel tasks with different work requirements on a computing system has been considered in [9, 10]. Each task α ($\alpha = 1, \ldots, \nu$) is served in each state i of $X(t)$ ($i = 1, \ldots, n$) at a different reward rate $r_{i\alpha}$. The reward rates are therefore grouped into a reward matrix, whose generic row $\mathbf{r}_i$ is the ν-dimensional vector representing how the total computational capacity of the system in state i is shared among the ν parallel tasks running in state i. On the other hand, the generic column $\mathbf{r}_\alpha$ contains the service rates at which task α is executed in the different structure-states in which the system operates. The minimal completion time has been derived in [10] under various combinations of preemption policies and being $X(t)$ semi-Markovian.

5 Completion Time and Petri Nets

The functional $Y(t)$, which allowed us to define the completion time as the hitting time against an absorbing barrier (Equation 2), is a complex stochastic process even if the structure-state process $X(t)$ is a *CTMC* (Corollaries 2, 4 and 6). Stochastic Petri nets (*SPN*) are usually restricted to be Markovian and therefore cannot be invoked to model and analyze the stochastic problem formulated in the previous sections. Recently, some attempts have been presented in the literature aimed at generalizing the concept of stochastic Petri nets by allowing the firing times to be generally distributed [1, 40, 18, 11]. The inclusion of non-exponential firing times poses intriguing problems about the interpretation of the evolution of the net versus time. A detailed discussion of the semantics of a *SPN* with generally distributed transition times can be found in [1]. We refer to this model as *GDT_SPN (Generally Distributed Transitions_SPN)*. The marking process underlying a *GDT_SPN* does not have, in general, a tractable analytical formulation. Therefore, various restrictions have been proposed in the literature. A particular case of non-Markovian *SPN*, is the class of *DSPN (Deterministic and SPN)*. *DSPN's* were introduced in [3] as *generalized stochastic Petri nets (GSPN)* [2] where in each marking a single transition is allowed to have associated a deterministic firing time, being all the other timed transitions exponential. *DSPN* becomes of potential concern in the completion time analysis, when the structure state process is a *CTMC* and the work requirement is a constant.

Several extensions of the original *DSPN* model have been recently appeared in the

literature [18, 11], aimed at including into the model non-deterministic distributions, and at accommodating more complex preemption policies for the general distributed transitions.

In the following three subsections, we enumerate the features and the properties of the mentioned *SPN*-based models that are relevant in the context of the problems discussed in the present chapter. Subsection 5.4 provides a general framework for modeling completion time problems in terms of *GDT_SPN*, and a numerical example is presented in Subsection 5.5.

5.1 Generally Distributed Transitions_SPN

Definition 1: According to [1], a *GDT_SPN* is defined as a marked *PN* in which:

1. The set of transitions is partitioned into a subset of immediate transitions (thin bars) and a subset of timed transitions (thick bars). Immediate transitions fire in zero time and have higher priority over timed transitions [2].

2. To each timed transition t_k is assigned a generally distributed random firing time γ_k, with *Cdf* $G_k(t)$, modeling the time occurring to complete the activity associated to t_k.

3. An *execution policy* is defined, which specifies the way in which a transition is selected to fire (among those enabled in a given marking), and the way in which the *GDT_SPN* keeps track of the past history.

The *execution policy* is needed to univocally determine a stochastic process associated to the *PN*. The *execution policy* comprises two specifications: a criterion to choose the next transition to fire (the *selection policy*), and a criterion to keep memory of the past history of the process (the *memory policy*). A natural choice to select the next transition to fire is according to a *race policy*: if more than one transition (of the same highest priority level) is enabled in a given marking, the transition fires whose associated random delay is statistically the minimum. The memory policy specifies how to recalculate the firing time distribution of a transition which has been disabled without firing when it is enabled again. In the exponential case, the same problem is hidden by the memoryless property. Two alternative memory policies are considered:

- *age memory*: an age variable a_k, associated with transition t_k, counts the time since the last firing epoch of t_k; when t_k is enabled, its firing distribution is calculated as the residual *Cdf* of the associated random variable γ_k, conditioned to a_k.

- *enabling memory*: the age variable a_k counts the time since the last epoch in which t_k has been enabled. When t_k is disabled (even without firing), a_k is reset.

Under the *age memory* policy the time spent in a *PN* transition accumulates whenever the transition is enabled and can be utilized to realize a *prs* preemption policy. Under the *enabling memory* policy the time spent in the transition is reset as soon as the transition is disabled and therefore can realize a *prd* preemption policy.

A numerically tractable realization of the *GDT_SPN* defined in *Definition 1*, is obtained by restricting the firing time random variables γ_k to be *PH*-distributed [11]. The

non-Markovian process generated by the *GDT_SPN* is converted into a *CTMC* defined over an expanded state space. The cardinality of the expanded state space is of the order of the cross product of the timed reachability set of the basic *PN* times the state spaces of the *PH* distributions of the γ_k random variables.

The program package *ESP* [23] realizes the *GDT_SPN* model with *PH* distributions. According to *Definition 1*, the program allows the user to assign a specific memory policy to each *PN* transition so that the different execution policies can be put to work. The important point about the *ESP* package is that the expanded *CTMC* is generated automatically from the model specifications. The generation of the expanded state space is driven by the different execution policies assigned by the user at the specification level.

The applicability of the *GDT_SPN* model with *PH* distributions to the completion time problem is legitimated by the following theorem proved in [13] and rederived by Neuts in [55]:

> *The class of PH distributions is closed with respect to the completion time problem in Markov Reward Models under any probabilistic mixing of* prs *and* prd *transitions.*

Hence, when the work requirement W is a *PH* random variable, or is approximated by a *PH* random variable, we are inside the area covered by the modeling power of the *ESP* package.

5.2 Deterministic SPN

Definition 2: According to [3], a *DSPN* is defined as a marked *PN* in which:

1. The set of transitions is partitioned into a subset of immediate transitions, a subset of exponential transitions and a subset of deterministic transitions.
2. At most, a single deterministic transition is allowed to be enabled in each marking and the firing time of a deterministic transition is marking independent.
3. The time elapsed in a deterministic transition cannot be remembered when the transition becomes disabled; the only allowed execution policy is the *race* policy with *enabling memory.*

In [3], the steady-state probability distribution is the only addressed solution. An improved algorithm for the evaluation of the steady state probabilities has been successively proposed in [48, 49], and some structural extensions, with respect to the specifications of Definition 2, have been presented in [19]. However, the computation of the distribution of the completion time requires the transient analysis.

The *DSPN* model has been revisited by Choi et al. [16]. In [16], the stochastic process associated with the *DSPN* model is proved to be a Markov regenerative process and an analytical method for the derivation of both the transient and the steady-state solution is provided. The analytical solution is derived in the Laplace transform domain, whose inversion necessitates a numerical technique. The paper proposes to use the Jagerman's method [39], as adapted by Chimento and Trivedi [15].

An alternative numerical solution technique, based on the use of supplementary variables [22], was originally proposed in [32] for the steady state analysis and then extended in [30] to the transient analysis of particularly structured *DSPN*'s.

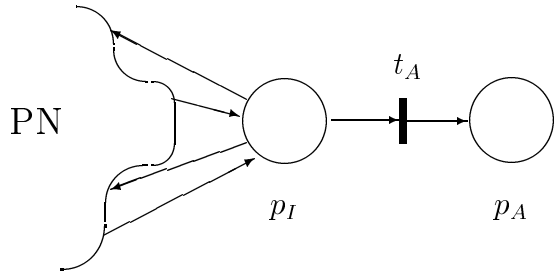

Figure 2 - A completion time problem represented as a first marking time in place p_A.

5.3 Markov Regenerative SPN

A further extension, called *Markov Regenerative SPN** (*MRSPN**) model, has been developed in [17], where the structural restrictions implied in *Definition 2* are retained, while replacing the deterministic transitions with generally distributed transitions. In particular, only the enabling memory policy can be assigned to the generally distributed transitions. This extension makes it possible to evaluate completion time problems in which the structure-state process is a *CTMC*, the work requirement is any random variable and the preemption policy is *prd*.

The supplementary variable approach to the same model has been discussed in [30] and a tool has been built based on this technique [31].

In order to relax the restriction on the enabling memory policy Bobbio and Telek [12] have defined a new class of *MRSPN* based on the concept of non-overlapping dominant transitions. In this model, any two successive regeneration time points of the marking process correspond to the first enabling and to the firing (or disabling) of a single generally distributed transition called the dominant transition. The enabling cycles of the dominant transitions cannot overlap. This definition includes the possibility that the structure-state process is semi-Markov, and allows the accommodation of different preemption policies. The *prs* case has been introduced in [12, 68], and a specific algorithm for the steady-state analysis has been elaborated in [69]. Finally, the inclusion of the *pri* policy has been discussed in [6].

5.4 Modeling Completion Time by Petri Nets

The completion time problem can be pictorially represented, at the *SPN* level, as a *first marking* problem. To this end, let us suppose that the reward rate is equal to one ($r = 1$) in all the states producing useful work. By this we physically mean that the task completes as soon as the total accumulated time spent in the markings producing useful work reaches the work requirement W. We introduce in the basic *PN* two additional places: an indicator place p_I and an absorbing place p_A (Figure 2).

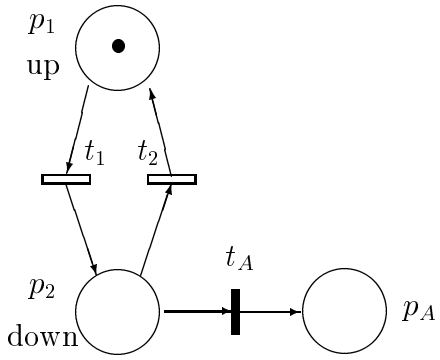

Figure 3 - PN modeling the attainment of a catastrophic failure when the down time exceeds a critical threshold.

The indicator place p_I is connected to the original *SPN* in such a way that it remains marked as long as the system is producing useful work; on the other hand, the absorbing place p_A is inserted to stop the execution of the net as soon as it becomes marked for the first time. The two places p_I and p_A are connected with each other by a single timed transition t_A. The random variable associated to the timed transition t_A is coincident with the work requirement W.

Interpreting the model as a *GDT_SPN*, the transition firing occurs according to the semantics of the race policy: t_A fires when its associated firing time W is the minimum among the activities enabled in p_I. Hence, the epoch at which p_A becomes marked for the first time is the epoch at which the time elapsed in p_I exceeds W for the first time, thus, by construction, is the completion time. In standard *PN* models, stopping the net usually requires additional elements, like immediate transitions or inhibitor arcs. Using higher level nets, like nets with enabling functions [20], the indicator, absorption and stopping property can be obtained by means of simpler and natural specifications.

The semantics of the memory policies of a *GDT_SPN* [1] is suited to model different preemptive disciplines in the task completion time problem. This feature differentiates the *GDT_SPN* model from the *DSPN* where the enabling memory policy is the only available one. If transition t_A (Figure 2) follows a race policy with age memory, it fires as the total marking time accumulated in p_I exceeds W (independently of the number of times place p_I has become marked); from the point of view of the completion time problem, a *prs* policy is realized. If t_A follows a race policy with enabling memory, it fires the first time a continuous marked interval in place p_I (without interruptions) exceeds W. The *GDT_SPN* models a completion time problem with *prd* policy.

In the *DSPN* model, t_A is a deterministic transition. The semantics of the *DSPN* model of Subsection 5.2 enforces a preemptive repeat policy: each time place p_I is enabled again, a new task is started. Since in this case, the task requirement is deterministic, the *pri* and *prd* policies are coincident.

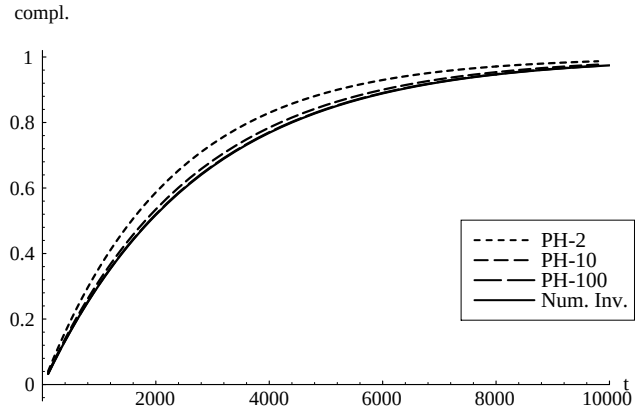

Figure 4 - *Cdf* of the lifetime of a 2-states system subject to a bounded catastrophic breakdown.

5.5 Numerical Examples

We compare the results obtained from the *GDT_SPN* with *PH* distributions and the *DSPN* models, on two simple examples. The comparison is particularly significant to explore the flexibility and the accuracy of the *PH* approximation in a limiting case, since a deterministic variable is known to be typically non-*PH*.

Case 1 - Bounded catastrophic breakdown - A system alternates between an up state (place p_1) and a down state (place p_2). Transition t_1 represents system failure (with failure rate λ) and transition t_2 system repair (with repair rate μ). A catastrophic (unsafe) condition is reached if and only if the time elapsed in the failed state exceeds a tolerance threshold W. This problem is represented in Figure 3, where place p_2 acts as indicator place and place p_A is the absorbing place representing the catastrophic condition.

Transition t_A is assigned the tolerance threshold W so that the catastrophic condition (token in p_A) is reached when the total down time exceeds W according to the assigned memory policy. If t_A is assigned an age memory policy, a *prs* strategy is realized since the time in p_A accumulates independently of the number of passages. On the other hand, if t_A is assigned an enabling memory policy, a *prd* strategy is realized. The distribution of the system lifetime can be interpreted as the distribution of the completion time of a "virtual task" of duration W, executed in place p_2. Since in the following we use the results obtained from [16] in the framework of *DSPN* models, only the *prd* policy can be considered.

Figure 4 shows the lifetime *Cdf* with $\lambda = 0.001\ h^{-1}$, $\mu = 0.1\ h^{-1}$ and with *prd* policy. The solid line represents the deterministic case with $W = 10\,h$ and is computed by numerically inverting the Laplace transform obtained from [16]. Dashed lines represent the cases in which W is Erlang with expected value $\mathbb{E}(W) = 10\,h$ and increasing number of stages (2, 10 and 100, respectively). Since the resulting *Cdf* is rather smooth also in the limiting deterministic case, the *PH* approximation becomes already close to the

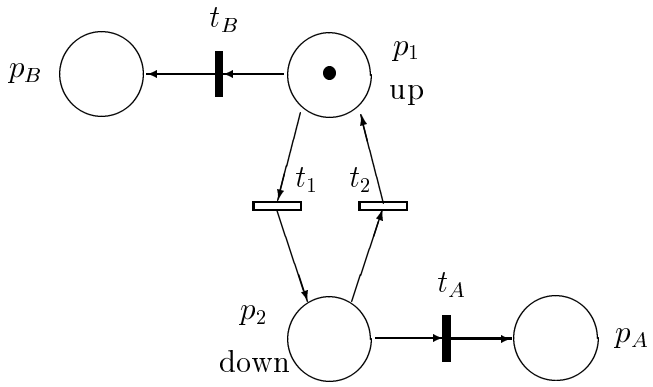

Figure 5 - *PN* modeling the completion time of a task on a 2-states server subject to a bounded catastrophic breakdown.

deterministic case with only a small number of stages.

Case 2 - Completion time with bounded catastrophic breakdown - A system, that can reach a catastrophic condition, as in the previous case, executes a task of length Z. The *PN* modeling the system is shown in Figure 5 [57], where t_B is assigned a firing time equal to the work requirement Z, and a token in p_B stops the net as soon as the task execution is completed. The *PN* models a competing multiple completion time example, and the analysis is aimed at evaluating the defective *Cdf* of completing the task before reaching the catastrophic state (a token arrives in p_B before one arrives in p_A). Computations are performed supposing that the up state is *prd* and the down state is *prs*. The deterministic case is solved by numerically inverting the closed-form Laplace transform given in [57].

Figure 6 compares the case in which both barrier levels W and Z are deterministic with the case in which both are Erlang of the same increasing order (2, 10 and 100 stages, respectively). Failure and repair rates are as in Case 1; the expected value of W is $\mathbb{E}(W) = 10\,h$, and the expected value of Z is $\mathbb{E}(Z) = 1000\,h$. As it can be observed, abrupt changes in the *Cdf* shape require *PH* variables of very high order.

6 Conclusion

The distribution of the task completion time is a performability measure that characterizes the quality of the service in a dependable computing system.

Interpreting the task completion time as the hitting time of a suitable cumulative functional against an absorbing barrier, provides a flexible and useful representation of the problem in many applications. In fact, various kinds of policies can be accommodated for modeling the interruption and the subsequent recovery of the execution of a task. Three different preemption policies have been extensively discussed and closed-form expression for the *Cdf* of the completion time have been derived in the Laplace

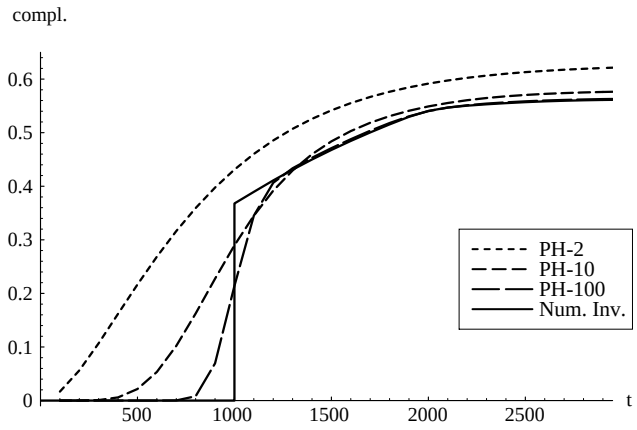

Figure 6 - *Cdf* of the completion time of a 2-states server with *prd* up state and *prs* down state.

transform domain.

Non-exponential stochastic Petri net can provide a graphical descriptive tool for the completion time analysis. In particular, recently proposed *SPN* model are reviewed. These models allow to some extent the inclusion of generally distributed firing times. A numerical example compares the results obtained from two specific models. In the first one, the firing time distributions are allowed to be of *PH* type, while in the second one a single transition in each marking is allowed to have a deterministic firing time, with all the other firing times being exponential.

Acknowledgments

The work of Andrea Bobbio was partially supported by CNR grant No. 96.01939.CT12. Miklós Telek was partially supported by OTKA grant No. T-16637.

References

[1] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Transactions on Software Engineering*, SE-15:832–846, 1989.

[2] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2:93–122, 1984.

[3] M. Ajmone Marsan and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In *Lecture Notes in Computer Science*, volume 266, pages 132–145. Springer Verlag, 1987.

[4] R.E. Barlow and F. Proschan. *Statistical Theory of Reliability and Life Testing.* Holt, Rinehart and Winston, New York, 1975.

[5] M.D. Beaudry. Performance-related reliability measures for computing systems. *IEEE Transactions on Computers*, C-27:540–547, 1978.

[6] A. Bobbio, , V.G. Kulkarni, A. Puliafito, M. Telek, and K.S. Trivedi. Preemptive repeat identical transitions in Markov Regenerative Stochastic Petri Nets. In *Proceedings 6-th International Conference on Petri Nets and Performance Models - PNPM95*, pages 113–122. IEEE Computer Society, 1995.

[7] A. Bobbio. The effect of an imperfect coverage on the optimum degree of redundancy of a degradable multiprocessor system. In *Proceedings RELIABILITY'87*, Paper 5B/3, Birmingham, 1987.

[8] A. Bobbio. Petri nets generating Markov reward models for performance/reliability analysis of degradable systems. In R. Puigjaner and D. Poitier, editors, *Modeling Techniques and Tools for Computer Performance Evaluation*, pages 353–365. Plenum Press, 1989.

[9] A. Bobbio. A multi-reward stochastic model for the completion time of parallel tasks. In A. Jensen and V.B. Iversen, editors, *Teletraffic and Datatraffic (Proceedings 13-th International Teletraffic Congress, ITC-13)*, pages 577–582. Elsevier Science Publisher, 1991.

[10] A. Bobbio and L. Roberti. Distribution of the minimal completion time of parallel tasks in multi-reward semi-Markov models. *Performance Evaluation*, 14:239–256, 1992.

[11] A. Bobbio and M. Telek. Computational restrictions for SPN with generally distributed transition times. In D. Hammer K. Echtle and D. Powell, editors, *First European Dependable Computing Conference (EDCC-1), Lecture Notes in Computer Science*, volume 852, pages 131–148, 1994.

[12] A. Bobbio and M. Telek. Markov regenerative SPN with non-overlapping activity cycles. In *International Computer Performance and Dependability Symposium - IPDS95*, pages 124–133. IEEE CS Press, 1995.

[13] A. Bobbio and K.S. Trivedi. Computation of the distribution of the completion time when the work requirement is a PH random variable. *Stochastic Models*, 6:133–149, 1990.

[14] X. Castillo and D.P. Siewiorek. A performance reliability model for computing systems. In *Proceedings 10-th International Symposium on Fault-Tolerant Computing*, pages 187–192, 1980.

[15] P.F. Chimento and K.S. Trivedi. The completion time of programs on processors subject to failure and repair. *IEEE Transactions on Computers*, 42:1184–1194, 1993.

[16] H. Choi, V.G. Kulkarni, and K. Trivedi. Transient analysis of deterministic and stochastic Petri nets. In *Proceedings of the 14-th International Conference on Application and Theory of Petri Nets*, Chicago, June 1993.

[17] H. Choi, V.G. Kulkarni, and K. Trivedi. Markov regenerative stochastic Petri nets. *Performance Evaluation*, 20:337–357, 1994.

[18] G. Ciardo, R. German, and C. Lindemann. A characterization of the stochastic process underlying a stochastic Petri net. *IEEE Transactions on Software Engineering*, 20:506–515, 1994.

[19] G. Ciardo and C. Lindemann. Analysis of deterministic and stochastic Petri nets. In *Proceedings International Workshop on Petri Nets and Performance Models - PNPM93*, pages 160–169. IEEE Computer Society, 1993.

[20] G. Ciardo, J. Muppala, and K.S. Trivedi. On the solution of GSPN reward models. *Performance Evaluation*, 12:237–253, 1991.

[21] E. Cinlar. Markov renewal theory. *Advances in Applied Probability*, 1:123–187, 1969.

[22] D.R. Cox. The analysis of non-markovian stochastic processes by the inclusion of supplementary variables. *Proceedings of the Cambridge Phylosophical Society*, 51:433–440, 1955.

[23] A. Cumani. Esp - A package for the evaluation of stochastic Petri nets with phase-type distributed transition times. In *Proceedings International Workshop Timed Petri Nets*, pages 144–151, Torino (Italy), 1985. IEEE Computer Society Press no. 674.

[24] L. Donatiello and B.R. Iyer. Analysis of a composite performance reliability measure for fault tolerant systems. *Journal of the ACM*, 34:179–199, 1987.

[25] E. De Souza e Silva and H.R. Gail. Calculating availability and performability measures of repairable computer systems using randomization. *Journal of the ACM*, 36:171–193, 1989.

[26] E. De Souza e Silva, H.R. Gail, and R. Vallejos Campos. Calculating transient distributions of cumulative reward. In *Proceedings ACM/SIGMETRICS Conference*, Ottawa, 1995.

[27] A. Von Ellenrieder and A. Levine. The probability of an excessive non-functioning interval. *Operations Research*, 14:835–840, 1966.

[28] D.P. Gaver. A waiting line with interrupted service, including priorities. *Journal of the Royal Statistical Society*, B24:73–90, 1962.

[29] F.A. Gay and M.L. Ketelsen. Performance evaluation for gracefully degrading systems. In *Proceedings 9-th International Symposium on Fault-Tolerant Computing*, pages 51–58, 1979.

[30] R. German. New results for the analysis of deterministic and stochastic Petri nets. In *International Computer Performance and Dependability Symposium - IPDS95*, pages 114–123. IEEE CS Press, 1995.

[31] R. German, C. Kelling, A. Zimmermann, and G. Hommel. *TimeNET - A toolkit for evaluating non-markovian stochastic Petri nets.* Report No. 19 - Technische Universität Berlin, 1994.

[32] R. German and C. Lindemann. Analysis of stochastic Petri nets by the method of supplementary variables. *Performance Evaluation*, 20:317–335, 1994.

[33] A. Goyal and A.N. Tantawi. Evaluation of performability for degradable computer systems. *IEEE Transactions on Computers*, C-36:738–744, 1987.

[34] V. Grassi, L. Donatiello, and G. Iazeolla. Performability evaluation of multicomponent fault-tolerant systems. *IEEE Transactions on Reliability*, R-37:216–222, 1988.

[35] B.R. Haverkort. *Performability modelling, tools, evaluation techniques and applications.* Phd thesis, Computer Science Dept, University of Twente, Enschede (NL), 1990.

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

[37] R. Huslende. A combined evaluation of performance and reliability for degradable systems. In *Proceedings ACM/SIGMETRICS Conference*, pages 157–164, 1981.

[38] B.R. Iyer, L. Donatiello, and P. Heidelberger. Analysis of performability for stochastic models of fault-tolerant systems. *IEEE Transactions on Computers*, C-35:902–907, 1986.

[39] D.L. Jagerman. An inversion technique for the Laplace transform. *The Bell System Technical Journal*, 61:1995–2002, October 1982.

[40] G. Juanole and Y. Atamna. Dealing with arbitrary time distributions with the stochastic timed Petri net models - Application to queueing systems. In *Proceedings International Workshop on Petri Nets and Performance Models - PNPM91*, pages 32–41. IEEE Computer Society, 1991.

[41] J. Keilson and S.S. Rao. A process with chain dependent growth rate. *Journal of Applied Probability*, 7:699–711, 1970.

[42] J. Keilson and S.S. Rao. A process with chain dependent growth rate. Part II: the ruin and ergodic problems. *Advances in Applied Probability*, 3:315–338, 1971.

[43] H.M. Khelalfa and A.K. von Mayrhauser. Models to evaluate trade-offs between performance and reliability. In *Proceedings CMG XV*, pages 24–29, San Francisco, 1984.

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

[45] V.G. Kulkarni, V.F. Nicola, and K. Trivedi. On modeling the performance and reliability of multi-mode computer systems. *The Journal of Systems and Software*, 6:175–183, 1986.

[46] V.G. Kulkarni, V.F. Nicola, and K. Trivedi. The completion time of a job on a multi-mode system. *Advances in Applied Probability*, 19:932–954, 1987.

[47] Y. Levy and P.E. Wirth. A unifying approach to performance and reliability objectives. In *Proceedings 12-th International Teletraffic Congress, ITC-12*, pages 4.2B2.1–4.2B2.7, Torino, 1988.

[48] C. Lindemann. An improved numerical algorithm for calculating steady-state solutions of deterministic and stochastic Petri net models. *Performance Evaluation*, 18:75–95, 1993.

[49] C. Lindemann. DSPNexpress: a software package for the efficient solution of deterministic and stochastic Petri nets. *Performance Evaluation*, 22:3–21, 1995.

[50] R.A. McLean and M.F. Neuts. The integral of a step function defined on a Semi-Markov process. *SIAM Journal on Applied Mathematics*, 15:726–737, 1967.

[51] J.F. Meyer. On evaluating the performability of degradable systems. *IEEE Transactions on Computers*, C-29:720–731, 1980.

[52] J.F. Meyer. Closed form solution of performability. *IEEE Transactions on Computers*, C-31:648–657, 1982.

[53] J.F. Meyer. Performability: a retrospective and some pointers to the future. *Performance Evaluation*, 14:139–156, 1992.

[54] M.F. Neuts. *Matrix Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore, 1981.

[55] M.F. Neuts. Two further closure properties of PH-Distributions. *Asia-Pacific Journal of Operational Research*, 9:459–477, 1992.

[56] V.F. Nicola. A single server queue with mixed types of interruptions. *Acta Informatica*, 23:465–486, 1986.

[57] V.F. Nicola, A. Bobbio, and K. Trivedi. A unified performance reliability analysis of a system with a cumulative down time constraint. *Microelectronics and Reliability*, 32:49–65, 1992.

[58] V.F. Nicola, V.G. Kulkarni, and K. Trivedi. Queueing analysis of fault-tolerant computer systems. *IEEE Transactions on Software Engineering*, SE-13:363–375, 1987.

[59] K.R. Pattipati, Y. Li, and H.A. Blom. A unified framework for the performability evaluation fault-tolerant computer systems. *IEEE Transactions on Computers*, 42:312–326, 1993.

[60] K.R. Pattipati and S.A. Shah. On the computational aspects of performability models of fault-tolerant computer systems. *IEEE Transactions on Computers*, 39:832–836, 1990.

[61] K.G. Ramamurthy and N.K. Jaiswal. A two-dissimilar-unit cold standby system with allowed downtime. *Microelectronics and Reliability*, 22:689–691, 1982.

[62] A. Reibman. Modeling the effect of reliability on performance. *IEEE Transactions on Reliability*, R-39:314–320, 1990.

[63] A. Reibman, R. Smith, and K.S. Trivedi. Markov and Markov reward model transient analysis: an overview of numerical approaches. *European Journal of Operational Research*, 40:257–267, 1989.

[64] S.M. Ross and J. Schechtman. On the first time a separately maintained parallel system has been down for a fixed time. *Naval Research Logistic Quarterly*, 26:285–290, 1979.

[65] G. Shanthikumar. First failure time of dependent parallel systems with safety period. *Microelectronics and Reliability*, 26:955–972, 1986.

[66] R. Smith, K. Trivedi, and A.V. Ramesh. Performability analysis: Measures, an algorithm and a case study. *IEEE Transactions on Computers*, C-37:406–417, 1988.

[67] U. Sumita, J.G. Shanthikumar, and Y. Masuda. Analysis of fault tolerant computer systems. *Microelectronics and Reliability*, 27:65–78, 1987.

[68] M. Telek and A. Bobbio. Markov regenerative stochastic Petri nets with age type general transitions. In G. De Michelis and M. Diaz, editors, *Application and Theory of Petri Nets (16-th International Conference), Lecture Notes in Computer Science*, volume 935, pages 471–489. Springer Verlag, 1995.

[69] M. Telek, A. Bobbio, L. Jereb, A. Puliafito, and K. Trivedi. Steady state analysis of Markov regenerative SPN with age memory policy. In H. Beilner and F. Bause, editors, *8-th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Lecture Notes in Computer Science*, volume 977, pages 165–179. Springer Verlag, 1995.

[70] Miklós Telek. *Some advanced reliability modelling techniques*. Phd Thesis, Hungarian Academy of Science, 1994.

[71] K. Trivedi, A.S. Sathaye, O.C. Ibe, and R.C. Howe. Should I add a processor ? In *Proceedings 23-rd Annual Hawaii International Conference on System Sciences HICSS-23*, pages 214–221, 1990.

[72] K.S. Trivedi, J.K. Muppala, S.P. Woolet, and B.R. Haverkort. Composite performance and dependability analysis. *Performance Evaluation*, 14:197–215, 1992.