PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 41, NO. 2, PP. 123–146 (1997)

TRANSIENT ANALYSIS OF A PREEMPTIVE RESUME M/D/1/2/2 THROUGH PETRI NETS[1]

Andrea BOBBIO* and Miklós TELEK**

* Dipartimento di Informatica
Università di Torino, 10149 Torino, Italy
** Department of Telecommunications
Technical University of Budapest,
H–1521 Budapest, Hungary

Received: March 1, 1996

Abstract

Stochastic Petri Nets *(SPN)* are usually designed to support exponential distributions only, with the consequence that their modelling power is restricted to Markovian systems. In recent years, some attempts have appeared in the literature aimed to define *SPN* models with generally distributed firing times. A particular subclass, called Deterministic and Stochastic Petri Nets *(DSPN)*, combines into a single model both exponential and deterministic transitions. The available *DSPN* implementations require simplifying assumptions which limit the applicability of the model to preemptive repeat different service mechanisms only. The present paper discusses a semantical generalization of the *DSPN*s by including preemptive mechanisms of resume type. This generalization is crucial in connection with fault tolerant systems, where the work performed before the interruption should not be lost. By means of this new approach, the transient analysis of a M/D/1/2/2 queue (with 2 customers, 1 server, exponential thinking and deterministic service time) is fully examined under different preemptive resume policies.

Keywords: Markov regenerative processes, Stochastic Petri Nets, transient analysis, deterministic service time, queuing systems with preemption.

1. Introduction

Petri nets have become an usual tool for modelling and analysing discrete state stochastic systems. The standard definition of Stochastic Petri Nets *(SPN)* implies that all the timed activities are exponentially distributed, and hence the modelling capability of this class of models is limited to markovian systems. In recent years, some attempts have appeared in the literature aimed to analyse *SPN*s supporting generally distributed firing times. We refer to this class of model as *Generally Distributed Transition SPN (GDT_SPN)*. The semantics of *GDT_SPN*s has been discussed in [1] where different firing policies have been examined and their potential implications in the modelling of real systems compared.

[1]This work was partly supported by OTKA T-16637

With the goal of including into a single *PN* model both exponentially distributed and constant timings, AJMONE and CHIOLA [3] have elaborated a model called Deterministic and Stochastic Petri Net *(DSPN)*. In their work, the authors have provided a method to compute the steady state solution, under two simplifying assumptions: *i)* at most one deterministic transition is enabled in a marking, and *ii)* the deterministic transitions are assigned an enabling memory policy (after the taxonomy in [1]). The solution algorithm was revisited in [17] and some structural extensions were proposed in [9]. The transient analysis of the *DSPN* model was successively developed by CHOI et al. [8]. In [7, 13, 10] the replacement of the deterministic transitions with generally distributed transitions was proposed and examined.

The main limitation of the models discussed in the mentioned references is that the deterministic (or generally distributed) transitions must be assigned a firing policy of enabling memory type [1]. The enabling memory policy means that each time the transition becomes enabled its firing time is resampled from the original distribution and the time eventually spent without firing in prior enabling periods is lost. The memory of the underlying stochastic process cannot be extended beyond a single cycle of enableing/disableing the non-exponential transition. In the language of queuing systems this assumption implies that the server should work on the job up to completion (the non-exponential transition fires), or if the job is interrupted before completion (due, for instance, to failure or preemption), the work already done is lost. In the literature, the above preemption mechanism is usually referred to as *preemptive repeat different (prd)* policy [12, 6][2].

CHOI et al. [8] have shown that the stochastic process underlying the formulated *DSPN* model is a *Markov Regenerative Process (MRP)* [11] for which general analytical solution equations for both the transient and the steady state behavior are provided. For this reason, CHOI et. al. refer to this model as *Markov Regenerative SPN (MRSPN)* [7]. A classification of *GDT_SPN*s and of the related underlying stochastic processes is in CIARDO et al. [10].

This paper proposes a semantical generalization of the *DSPN* model by including the possibility of modelling preemptive mechanisms of resume type. A *preemptive resume (prs)* service policy means that the server is able to recover an interrupted job by keeping memory of the work already performed so that, upon restart, only the residual service needs to be completed. This modelling extension is crucial in connection with fault tolerant

[2]The enabling memory assumption is relaxed in [9] where a deterministic transition can be disabled in vanishing markings only. Since vanishing markings are transversed in zero time, this assumption does not modify the behavior of the marking process versus time

and parallel computing systems, where a single task may be interrupted either during a fault/recovery cycle or for the execution of a higher priority task, but when the cause originating the interruption is ceased, the dormant task is resumed from the point it was interrupted. Even if a *prs* execution policy is the main goal of a dependable fault tolerant design, its analytical modelling was not possible in the framework of the available *DSPN* tools.

The aim of this paper is to develop a new analytical methodology to deal with a class of systems with a service policy of *prs* type. The proposed methodology is applied to revisit the case study of a closed M/D/1/2/2 queuing system, already considered in [3, 8]. The transient state probabilities are derived with two different *prs* service policies. In the first case, the customers are considered identical so that any job joining the queue preempts the job eventually under service. In the second case, the customers have different priorities so that the jobs submitted by the customer with higher priority preempts the lower priority jobs but not vice versa.

After introducing notation and the general solution equations for a *MRSPN* in Section 2, the paper presents the M/D/1/2/2 system with the two considered *prs* mechanisms, in Section 3. Section 4 is devoted to derive the closed form transient equations for the considered model, and finally, Section 5 illustrates the results.

2. Markov Regenerative Stochastic Petri Nets

A marked Petri Net (*PN*) is a tuple $PN = (P, T, I, O, H, M)$, where:

- $P = \{p_1, p_2, \dots, p_{np}\}$ is the set of places (drawn as circles);
- $T = \{t_1, t_2, \dots, t_{nt}\}$ is the set of transitions (drawn as bars);
- I, O and H are the input, the output and the inhibitor functions, respectively. The input function I provides the multiplicities of the input arcs from places to transitions; the output function O provides the multiplicities of the output arcs from transitions to places; the inhibitor function H provides the multiplicity of the inhibitor arcs from places to transitions.
- $M = \{m_1, m_2, \dots, m_{np}\}$ is the marking. The generic entry m_i is the number of tokens (drawn as black dots) in place p_i, in marking M.

Input and output arcs have an arrowhead on their destination, inhibitor arcs have a small circle. A transition is enabled in a marking if each of its ordinary input places contains at least as many tokens as the multiplicity of the input function I and each of its inhibitor input places contains fewer tokens than the multiplicity of the inhibitor function H. An enabled transition fires by removing as many tokens as the multiplicity of the input function I from each ordinary input place, and adding as many tokens as the multiplicity of the output function O to each output place. The number of tokens in an inhibitor input place is not affected.

A marking M' is said to be *immediately reachable* from M, when it is generated from M by firing a single enabled transition t_k. The reachability set $\mathcal{R}(M_0)$ is the set of all the markings that can be generated from an initial marking M_0 by repeated application of the above rules. If the set T comprises both timed and immediate transitions, $\mathcal{R}(M_0)$ is partitioned into tangible (no immediate transitions are enabled) and vanishing markings, according to [2]. Let $\mathcal{N}$ be the cardinality of the tangible subset of $\mathcal{R}(M_0)$.

Definition 1 – *A stochastic GDT_SPN is a marked SPN in which [1]:*

- *To any timed transition $t_k \in T$ is associated a random variable γ_k modelling the time needed by the activity represented by t_k to complete, when considered in isolation.*
- *Each random variable γ_k is characterized by its (possibly marking dependent) cumulative distribution function.*
- *Each timed transition t_k is attached an age variable a_k and a memory policy; the memory policy specifies the functional dependence of the age variable on the past enabling time of the transition.*
- *An initial probability is given on the reachability set.*

The age variable a_k, associated to transition t_k, is a functional that depends on the time during which t_k has been enabled. The age variables together with their memory policy specify univocally how the underlying stochastic process is conditioned upon its past history. The semantics of different memory policies has been discussed in [1] where three alternatives have been proposed and examined.

- *Resampling policy* – The age variable a_k is reset to zero at any change of marking.
- *Enabling memory policy* – The age variable a_k accounts for the work performed by the activity corresponding to t_k from the last epoch in which t_k has been enabled. When transition t_k is disabled (even without firing) the corresponding enabling age variable is reset.
- *Age memory policy* – The age variable a_k accounts for the work performed by the activity corresponding to t_k from its last firing up to the current epoch and is reset only when t_k fires.

At the entrance in a new tangible marking, the residual firing time is computed for each enabled timed transition its age variable is given, so that the next marking is determined by the minimal residual firing time among the enabled transitions (*race policy* [1]). Because of the memoryless property, the three mentioned policies are equivalent for the exponential distribution. Hence, for an exponential transition t_k, we assume, conventionally, that the corresponding memory policy is of resampling type, so that a_k is reset at each transition.

Definition 2 – *The stochastic process underlying a GDT_SPN is called the marking process $\mathcal{M}(x)$ $(x \geq 0)$. $\mathcal{M}(x)$ is the marking of the GDT_SPN at time x.*

A single realization of the marking process $\mathcal{M}(x)$ can be written as:

$$\mathcal{R} = \{(\tau_0, M_0); (\tau_1, M_1); \dots ; (\tau_i, M_i); \dots \},$$

where M_{i+1} is a marking immediately reachable from M_i, and $\tau_{i+1} - \tau_i$ is the sojourn time in marking M_i. With the above notation, $\mathcal{M}(x) = M_i$ for $\tau_i \leq x < \tau_{i+1}$.

If at time τ_i^+ of entrance in a tangible marking M_i all the age variables a_k $(k = 1, 2, \dots, n_t)$ are equal to zero, we say that τ_i is a regeneration time point for the marking process $\mathcal{M}(x)$. Let us denote by τ_n^* the sequence of the regeneration time points embedded into a realization $\mathcal{R}$. The tangible marking $M_{(n)}$ entered at a regeneration time point τ_n^* is called a regeneration marking.

Definition 3 – *A regeneration time point τ_n^* in the marking process $\mathcal{M}(x)$ is the epoch of entrance in a tangible marking $M_{(n)}$ in which all the age variables are equal to 0.*

The embedded sequence of regeneration time points $(\tau_n^*, M_{(n)})$ is a Markov renewal sequence and the marking process $\mathcal{M}(x)$ is a Markov regenerative process [11, 7, 10].

Definition 4 – *A GDT_SPN, for which an embedded Markov renewal sequence $(\tau_n^*, M_{(n)})$ exists, is a Markov Regenerative Stochastic Petri Nets (MRSPN)* [7].

Since $(\tau_n^*, M_{(n)})$ is a Markov renewal sequence, the following equalities hold:

$$\begin{aligned}
&Pr\{M_{(n+1)} = j, (\tau_{n+1}^* - \tau_n^*) \leq x | M_{(n)} = i, \tau_n^*, M_{(n-1)}, \tau_{n-1}^*, \dots, M_{(0)}, \tau_0^*\} = \\
&Pr\{M_{(n+1)} = j, (\tau_{n+1}^* - \tau_n^*) \leq x | M_{(n)} = i\} = \qquad (1)\\
&\quad Pr\{M_{(1)} = j, \tau_1^* \leq x | M_{(0)} = i\}.
\end{aligned}$$

The first equality expresses the Markov property (i.e. the condition on the past is condensed in the present state). The second equality expresses the time homogeneity (i.e. the probability measures are independent of a translation along the time axis). According to [7, 11], we define the following matrix valued functions (of dimension $\mathcal{N} \times \mathcal{N}$):

$$
\begin{aligned}
\mathbf{V}(x) &= [V_{ij}(x)] \text{ such that } V_{ij}(x) = Pr\{\mathcal{M}(x) = j | \mathcal{M}(\tau_0^*) = i\} \\
\mathbf{K}(x) &= [K_{ij}(x)] \text{ such that } K_{ij}(x) = Pr\{M_{(1)} = j, \tau_1^* \leq x | \mathcal{M}(\tau_0^*) = i\} \\
\mathbf{E}(x) &= [E_{ij}(x)] \text{ such that } E_{ij}(x) = Pr\{\mathcal{M}(x) = j, \tau_1^* > x | \mathcal{M}(\tau_0^*) = i\}
\end{aligned}
\tag{2}
$$

Matrix $\mathbf{V}(x)$ is the transition probability matrix and provides the probability that the stochastic process $\mathcal{M}(x)$ is in marking j at time x given it was in i at $x = 0$. The matrix $\mathbf{K}(x)$ is the *global kernel* of the *MRP* and provides the cdf of the event that the next regeneration time point is τ_1^* and the next regeneration marking is $M_{(1)} = j$ given marking i at $\tau_0^* = 0$. Finally, the matrix $\mathbf{E}(x)$ is the *local kernel* since it describes the behavior of the marking process $\mathcal{M}(x)$ inside two consecutive regeneration time points. The generic element $E_{ij}(x)$ provides the probability that the process jumps in state j starting from i at $\tau_0^* = 0$ before the next regeneration time point. From the above definitions:

$$\sum_j [K_{ij}(x) + E_{ij}(x)] = 1.$$

The transient behavior of the *MRSPN* can be evaluated by solving the following generalized Markov renewal equation (in matrix form) [11, 7]:

$$\mathbf{V}(x) = \mathbf{E}(x) + \mathbf{K} * \mathbf{V}(x), \tag{3}$$

where $\mathbf{K} * \mathbf{V}(x)$ is a convolution matrix, whose (i, j)-th entry is:

$$[\mathbf{K} * \mathbf{V}(x)]_{ij} = \sum_k \int_0^x dK_{ik}(y) V_{kj}(x - y). \tag{4}$$

By denoting the Laplace Stieltjes transform *(LST)* of a function $F(x)$ by $F^\sim(s) = \int_0^\infty e^{-sx}\, dF(x)$, *Eq.* (3) becomes in the *LST* domain:

$$\mathbf{V}^\sim(s) = \mathbf{E}^\sim(s) + \mathbf{K}^\sim(s)\mathbf{V}^\sim(s) \tag{5}$$

whose solution is:

$$\mathbf{V}^\sim(s) = [\mathbf{I} - \mathbf{K}^\sim(s)]^{-1}\, \mathbf{E}^\sim(s). \tag{6}$$

As specified by (2), $\mathbf{K}(x)$ and $\mathbf{E}(x)$ depend on the evolution of the marking process between two consecutive regeneration time points. By virtue of the time homogeneity property (1), we can always define the two successive regeneration time points to be $x = \tau_0^* = 0$ and $x = \tau_1^*$. The steady state solution can be evaluated as $\lim_{s \to 0} \mathbf{V}^{\sim}(s)$.

Definition 5 – *The stochastic process subordinated to state i (denoted by $\mathcal{M}^{(i)}(x)$) is the restriction of the marking process $\mathcal{M}(x)$ for $x \leq \tau_1^*$ given $\mathcal{M}(\tau_0^*) = i$:*

$$\mathcal{M}^{(i)}(x) = [\mathcal{M}(x) : x \leq \tau_1^*, \mathcal{M}(\tau_0^*) = i].$$

According to Definition 5, $\mathcal{M}^{(i)}(x)$ describes the evolution of the *PN* starting at the regeneration time point $x = 0$ in the regeneration marking i, up to the next regeneration time point τ_1^*. Therefore, $\mathcal{M}^{(i)}(x)$ includes all the markings that can be reached from state i before the next regeneration time point. The entries of the i-th row of the matrices $\mathbf{K}(x)$ and $\mathbf{E}(x)$ are determined by $\mathcal{M}^{(i)}(x)$.

The subclass of *DSPN*s considered in [3, 8] is a restriction of Definition 4 obtained by adding the following specifications: *i)* in each marking, at most a single deterministic transition is enabled being all the other transitions exponential; *ii)* the memory policy associated to every deterministic transition is of enabling memory type[3]. The above restrictions imply that the regeneration markings can be partitioned in two classes:

1 *No deterministic transition is enabled* – all the transitions enabled in the regeneration marking i are exponential, so that the next regeneration time point is any epoch of entrance in a marking immediately reachable from i. The subordinated process is a single step *CTMC*.

2 *A single deterministic transition is enabled* – the next regeneration time point is the epoch at which the deterministic transition fires or is disabled (indeed, in both cases the enabling memory variable is reset). During the interval between the two consecutive regeneration time points the deterministic transition is continuously enabled and the subordinated process is a *CTMC*.

In the present paper, we derive a closed form analytical solution for an extended class of *DSPN*s referred to as *Age Memory DSPN*:

[3]In their definition of *DSPN*s, CIARDO and LINDEMANN [10] include the presence of more than one deterministic transition enabled in a marking, but still maintaining the enabling memory policy.

Definition 6 – *An Age Memory DSPN is a MRSPN (Definition 4) on which the following conditions are imposed:*

1. *In any regeneration marking i, a single deterministic transition of age memory type (say t_d with associated age memory variable a_d and deterministic duration α_d) is allowed to start its firing process.*
2. *The next regeneration time point is the epoch at which t_d fires (the corresponding age variable a_d is reset); hence, firing of t_d must lead to a regeneration marking in which all the other age variables are zero.*
3. *The related subordinated process $\mathcal{M}^{(i)}(x)$ is semimarkov.*

The above definition has two major implications. Since the subordinated process is semimarkov, multiple deterministic transitions can be simultaneously enabled inside the firing process of t_d, provided they are no more enabled as t_d fires. The second implication is that the deterministic transition t_d need not to be continuously enabled in $\mathcal{M}^{(i)}(x)$. Indeed, disabling an age memory transition does not reset its age variable. In order to track the enabling/disabling condition of t_d in $\mathcal{M}^{(i)}(x)$, we introduce a reward (indicator) variable which is equal to 1 in markings in which t_d is enabled and equal to 0 in markings in which t_d is not enabled. We group the binary reward variables into a reward vector denoted by $\underline{r}^{(i)}$. With the above definition $r_k^{(i)} = 1$ means that the deterministic transition t_d is enabled whenever $\mathcal{M}^{(i)}(x) = k$.

The behavior of the marking process between two consecutive regeneration time points is formulated in terms of the semimarkov reward model $(\underline{r}^{(i)}, \mathcal{M}^{(i)}(x))$ [14,18,4]. The age variable a_d corresponding to the deterministic transition is computed as the accumulated reward in the semimarkov reward subordinated process and the successive regeneration time point τ_1^* (the firing epoch of t_d) occurs when the age variable a_d accumulates a time equal to the deterministic duration α_d. Resorting to the computational properties of stochastic reward models [5], the cdf of τ_1^* is evaluated as the first time at which the functional a_d hits an absorbing barrier of height α_d.

The firing of t_d in the subordinated process $\mathcal{M}^{(i)}(x)$, can only occur in a state k in which $r_k^{(i)} = 1$. After the firing of t_d in state k, the successor tangible marking ℓ is determined by the branching probability matrix $\Delta^{(i)} = [\Delta_{k\ell}^{(i)}]$ [8, 10], where:

$$\Delta_{k\ell}^{(i)} = Pr\{\text{next tangible marking is } \ell \text{ |current marking is } k, t_d \text{ fires}\}$$

If marking k is not connected to vanishing markings, the kth row of $\Delta^{(i)}$ contains at most one nonzero element, equal to 1 [8].

3. The M/D/1/2/2 Preemptive Queuing System

A *PN* model for the non-preemptive M/D/1/2/2 queue has been introduced in [3], where the steady state solution was derived. The transient analysis for the same system was carried on in [8]. In the following, we examine two different mechanisms of preemptive service.

A. – Preemptive M/D/1/2/2 with identical customers

The M/D/1/2/2 queue has a preemptive service with the same kind of customers. The job in execution is preempted as soon as a new job eventually arrives at the server. The preempted job is restarted as soon as the server becomes idle again. Two different recovery policies can be considered depending whether the server is able to remember the work already performed on the job before preemption or not. In the latter case, the prior work is lost due to the interruption and the recovered job must be repeated from scratch with a service time resampled from the original cdf (*prd policy)*. In the former case, the prior work is not lost and the service time of the recovered job equals the residual service time given the work already executed before preemption (*prs policy)*. *Fig. 1a* shows a *PN* describing the M/D/1/2/2 system in which any new job preempts the job eventually under service. Place p_1 contains the customers thinking, while place p_2 contains the number of submitted jobs (including the one under service). Starting from the initial marking $s_1 = (2\ 0\ 0\ 1)$ (*Fig. 1b*), t_1 is the only enabled transition. Firing of t_1 represents the submission of the first job and leads to state $s_2 = (1\ 1\ 1\ 0)$. In s_2 transitions t_2 and t_3 are competing. t_2 represents the service of the submitted job and its firing returns the system to the initial state s_1. t_3 represents the submission of the second job and its firing disables t_2 by removing one token from p_3 (the first job becomes dormant). In $s_3 = (0\ 2\ 0\ 1)$ one job is under service and one job is dormant, and the only enabled activity is the service of the active job. Firing of t_4 leads the system again in s_2, where the dormant job is recovered. Assuming the thinking time of both customers to be exponentially distributed with parameter λ, t_1 is associated an exponential firing rate equal to (2λ) and t_3 a firing rate equal to λ. Transitions t_2 and t_4 are assigned a deterministic service time of duration α.
A.1 – t_2 and t_3 are assigned an enabling memory policy. – Each time t_2 is disabled by the arrival of the second job (t_3 fires before t_2), the corresponding enabling age variable a_2 is reset. As soon as t_2 becomes enabled again (the second job completes and t_4 fires) no memory is kept of the prior service, and the execution restarts from scratch. This behavior corresponds to a *prd* service policy, and is covered by the model definition in [8, 9].
A.2 – t_2 and t_3 are assigned an age memory policy. – Each time t_2 is disabled without firing (t_3 fires before t_2) the age variable a_2 is not reset.

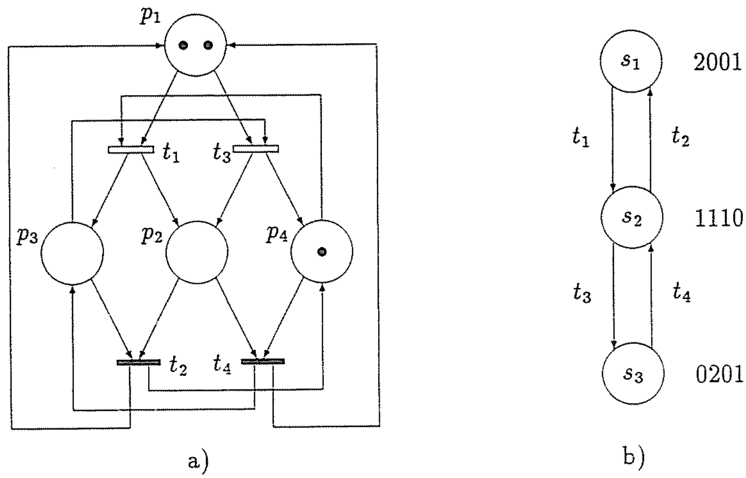

Fig. 1. Preemptive M/D/1/2/2 queue with identical customers

Hence, as the second job completes (t_4 fires), the system returns in s_2 keeping the value of a_2, so that the time to complete the interrupted job can be evaluated as the residual service time given a_2. a_2 counts the total time during which t_2 is enabled before firing, and is equal to the cumulative sojourn time in s_2. The assignment of the age memory policy to t_2 realizes a *prs* service mechanism. This behavior is not compatible with the definition of *DSPN* given in [8] and requires a new analysis methodology. The regeneration time points in the marking process $\mathcal{M}(x)$ correspond to the epochs of entrance in markings in which the age variables associated to all the transitions are equal to zero. By inspecting *Fig. 1b)*, the regeneration time points result to be the epochs of entrance in s_1 and of entrance in s_2 from s_1. The process $\mathcal{M}^{(1)}(x)$ subordinated to state s_1 is a single step *CTMC* (being the only enabled transition t_1 exponential) and includes the only immediately reachable state s_2.

The process $\mathcal{M}^{(2)}(x)$ subordinated to state s_2 includes all the states reachable from s_2 before firing of t_2: these states are s_3, s_2. Since s_2 is the only state in which t_2 is enabled, the corresponding reward rate vector is $\underline{r}^{(2)} = [0\ 1\ 0]$. Firing of t_2 can only occur from state s_2 leading to state s_1; it turns out that the only relevant nonzero entry of the branching probability matrix is $\Delta_{21}^{(2)} = 1$.

A possible realization of the subordinated marking process $\mathcal{M}^{(2)}(x)$ is shown in *Fig. 2*. Notice that $\mathcal{M}^{(2)}(x)$ is semimarkov since t_4 is deterministic. The age variable a_2 grows whenever $\mathcal{M}^{(2)}(x) = s_2$, and the firing of t_2

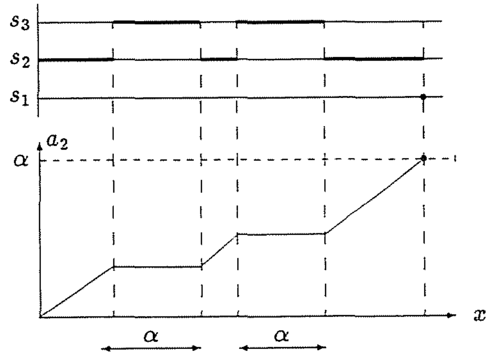

Fig. 2. A possible realization of the subordinated marking process $\mathcal{M}^{(2)}(x)$

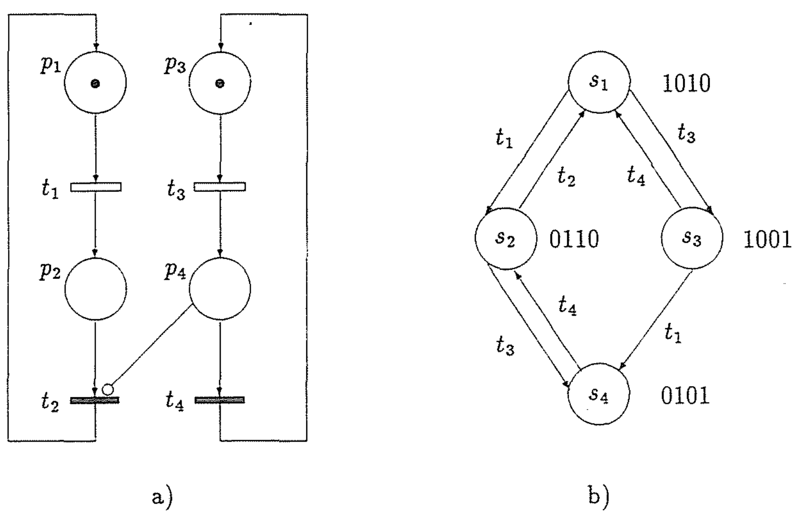

Fig. 3. Preemptive M/D/1/2/2 queue with different customers

occurs when a_2 reaches the value α (the deterministic duration assigned to t_2). Considering α as an absorbing barrier for the accumulation functional represented by the age variable a_2, the firing time of t_2 is determined by the first passage time of a_2 across the absorbing barrier α.

In the present example, s_3 can never be a regeneration marking, since a_2 is not reset at the entrance in s_3.

B. – Preemptive M/D/1/2/2 with different customers

The two customers are of different classes, and customer of class 2 preempts customer of class 1 but not viceversa. Two possible preemption policies are again possible depending whether the server is able to remember the work done before the interruption. A *PN* modelling the M/D/1/2/2 queue in which the jobs submitted by customer 2 have higher priority over the jobs submitted by customer 1 is reported in *Fig. 3a.* Place p_1 (p_3) represents customer 1 (2) thinking, while place p_2 (p_4) represents job 1 (2) under service. Transition t_1 (t_3) is the submission of a job of type 1 (2), while transition t_2 (t_4) is the completion of service of a job of type 1 (2). The inhibitor arc from p_4 to t_2 models the described preemption mechanism: as soon as a type 2 job joins the queue the type 1 job eventually under service is interrupted. The reachability graph of the *PN* of *Fig. 3a* is in *Fig. 3b.* Under a *prs* service policy, after completion of the type 2 job, the interrupted type 1 job is resumed continuing the new service period from the point reached just before the last interruption. In the *PN* of *Fig. 3a* this service policy is realized by assigning to transitions t_2 and t_4 an age memory policy. The submitting times (transitions t_1 and t_3) are exponentially distributed with parameters λ, while the service times (transitions t_2 and t_4) are deterministic with duration α.

From *Fig. 3b*, it is easily recognized that s_1, s_2 and s_3 can all be regeneration states, while s_4 can never be a regeneration state (in s_4 a type 2 job is always in execution so that its corresponding age variable a_2 is never 0). Only exponential transitions are enabled in s_1 and the next regeneration states can be either s_2 or s_3 depending whether t_1 or t_3 fires first. From s_2 the next regeneration state can be only s_1, but multiple cycles (s_2 – s_4) can occur depending whether type 2 jobs arrive to interrupt the execution of the type 1 job. From state s_3 the next regeneration marking can be either state s_1 or s_2 depending whether during the execution of the type 2 job a type 1 job does require service (but remains blocked until completion of the type 2 job) or does not.

3.1 A Fault Tolerant Example

Consider a 'work conservative' system of two machines and a single repairman. The system is work conservative if the repairman is always working when at least one machine is down. The repairman can deal at most with one machine at a time. Hence the question of choosing the machine to repair arises when both of the machines are down. The machine under service is chosen according to one of the following policies:

- the repair of the machine failed first is completed without any interruption, when the second one fails,
- the repair of the machine failed first is stopped and the repairman switches to the second one when it fails,
- with machines of different priorities the repair of the high priority machine goes when both the machines are down.

The first policy was studied in [3] and [8]. The second one results in the stochastic model referred to as *Preemptive M/D/1/2/2 with identical customers* while the third one gives a *Preemptive M/D/1/2/2 with different customers.*

4. Transient Analysis of the Subordinated Process

The entries of the i-th row of the matrices $\mathbf{K}(x)$ and $\mathbf{E}(x)$ are determined by the subordinated process $\mathcal{M}^{(i)}(x)$. In the *prs* M/D/1/2/2 system, examined in the previous section, two classes of subordinated processes have been encountered:

1. *Single step CTMCs.*
2. *Reward Semimarkov Process.*

4.1 Single Step CTMC

In the regeneration marking i only exponential transitions are enabled. The next regeneration time point is the epoch of jump into any one of the immediately reachable states. $\mathcal{M}^{(i)}(x)$ is a *CTMC* with a single transient state (state i with initial probability equal to 1) and a number of absorbing states equal to the number of immediately reachable states. Let q_{ij} be the transition rate from i to j, and $q_i = \sum_{j;j\neq i}^{\mathcal{N}} q_{ij}$.

The entry $K_{ij}(x)$ of the global kernel provides the probability of being absorbed in state j before time x, while the entry $E_{ij}(x)$ of the local kernel gives the probability of jumping from i to j before the next regeneration time point. Since, in this case, any firing provides a new regeneration time point, the only nonzero entry of the i-th row of matrix $\mathbf{E}(x)$ corresponds to $j = i$. It follows (for $j = 1, 2, \ldots, \mathcal{N}$) [8]:

$$K_{ij}(t) = (1-\delta_{ij})\frac{q_{ij}}{q_i}(1-e^{-q_i t}), \qquad E_{ij}(t) = \delta_{ij}e^{-q_i t}, \tag{7}$$

and in *LST* domain:

$$K_{ij}^{\sim}(s) = (1-\delta_{ij})\frac{q_{ij}}{s+q_i}, \qquad E_{ij}^{\sim}(s) = \delta_{ij}\frac{s}{s+q_i}, \tag{8}$$

where δ_{ij} is the Kronecker delta.

4.2 Subordinated Reward Semimarkov Process

At $x = \tau_0^* = 0$ a single deterministic transition t_d (with age memory variable a_d and duration α_d) starts its firing process in state i ($a_d = 0$). The successive regeneration time point τ_1^* is the epoch of firing of t_d and this event occurs as the functional a_d reaches the value α_d for the first time.

Let $\Omega(i)$ be the subset of $\mathcal{R}(M_0)$ including the states of the subordinated process $\mathcal{M}^{(i)}(x)$ (i.e. the states reachable from i before firing t_d). For notational convenience we do not renumber the states in $\Omega(i)$ so that all the subsequent matrix functions have the dimensions $(\mathcal{N} \times \mathcal{N})$ (the tangible part of $\mathcal{R}(M_0)$), but with the significant entries located in position (k, ℓ) only, with $k, \ell \in \Omega(i)$.

Let $Z^{(i)}(x)$ $(x \geq 0)$ be the semimarkov process defined over $\Omega(i)$ and $\underline{r}^{(i)}$ the corresponding binary reward vector. The subordinated process $\mathcal{M}^{(i)}(x)$ (Definition 5) coincides with $Z^{(i)}(x)$ when the initial state is state i with probability 1 ($Pr\{Z^{(i)}(0) = i\} = 1$). The age variable a_d increases at a rate $r_j^{(i)}$ (which is either equal to 0 or to 1) when $\mathcal{M}^{(i)}(x) = j$.

Let $\mathbf{Q}^{(i)}(x) = [Q_{k\ell}^{(i)}(x)]$ be the kernel of the semimarkov process $Z^{(i)}(x)$. The initial probability vector is $\underline{Q}_0^{(i)} = [0, 0, \ldots, 1_i, \ldots, 0]$ (a vector with all the entries equal to 0 but entry i equal to 1). We denote by H the time duration until the first embedded time point in the semimarkov process starting from state k at time 0 ($Z^{(i)}(0) = k$). The generic element (for $k, \ell \in \Omega(i)$)

$$Q_{k\ell}^{(i)}(x) = Pr\left\{H \leq x, Z^{(i)}(H^+) = \ell | Z^{(i)}(0) = k\right\}$$

is the distribution of H supposed that a transition from state k to state ℓ took place at the embedded time point. With nonzero diagonal elements in $\mathbf{Q}^{(i)}(x)$ the next embedded time point can be determined by a transition from state k to state k. The distribution of H is:

$$[Q_k^{(i)}(x) = \sum_{\ell \in \Omega(i)} Q_{k\ell}^{(i)}(x) \qquad (k = 1, ..., n)$$

and, finally, the probability of jumping from state k to state ℓ at time $H = x$ is:

$$\frac{dQ_{k\ell}^{(i)}(x)}{dQ_k^{(i)}(x)} = Pr\left\{Z^{(i)}(x^+) = \ell |\ H = x, Z^{(i)}(0) = k\right\}.$$

Let us introduce two matrix functions: $\mathbf{F}^{(i)}(x, \alpha_d)$ and $\mathbf{P}^{(i)}(x, \alpha_d)$ so defined:

$$F_{k\ell}^{(i)}(x, \alpha_d) = Pr\{Z^{(i)}(\tau_1^{*-}) = \ell \ , \ \tau_1^* \leq x \mid Z^{(i)}(0) = k\} \tag{9}$$

and

$$P_{k\ell}^{(i)}(x, \alpha_d) = Pr\{Z^{(i)}(x) = \ell \ , \tau_1^* > x \mid Z^{(i)}(0) = k\}. \tag{10}$$

$F_{k\ell}^{(i)}(x, \alpha_d)$ is the probability of hitting the absorbing barrier α_d in state ℓ before x starting in state k at $x = 0$. $P_{k\ell}^{(i)}(x, \alpha_d)$ is the probability of being in state ℓ at time x before absorption at the barrier α_d starting in state k at $x = 0$. From (9) and (10) follows:

$$\sum_{\ell} [F_{k\ell}^{(i)}(x, \alpha_d) + P_{k\ell}^{(i)}(x, \alpha_d)] = 1.$$

By these definitions, the i-th row vectors $\underline{K}_i(x)$ and $\underline{E}_i(x)$ of matrices $\mathbf{K}(x)$ and $\mathbf{E}(x)$, respectively, are related to $\mathbf{F}^{(i)}(x, \alpha_d)$ and $\mathbf{P}^{(i)}(x, \alpha_d)$ by the following relations:

$$\underline{K}_i(x) = \underline{Q}_0^{(i)} \mathbf{F}^{(i)}(x, \alpha_d) \Delta^{(i)} \qquad ; \qquad \underline{E}_i(x) = \underline{Q}_0^{(i)} \mathbf{P}^{(i)}(x, \alpha_d). \tag{11}$$

Due to the particular structure of the initial probability vector $\underline{Q}_0^{(i)}$, *Eq.* (11) reduces to a relation among the entries of the i-th row of the corresponding matrices:

$$K_{ij}(x) = \sum_k F_{ik}^{(i)}(x, \alpha_d) \Delta_{kj}^{(i)} \qquad ; \qquad E_{ij}(x) = P_{ij}^{(i)}(x, \alpha_d). \tag{12}$$

In the definition of matrices $F_{k\ell}^{(i)}(x, \alpha_d)$ and $P_{k\ell}^{(i)}(x, \alpha_d)$ we maintain the explicit dependence on the barrier level α_d, since this dependence will be exploited in the subsequent analytical treatment. Evaluation of $F_{k\ell}^{(i)}(x, \alpha_d)$ and $P_{k\ell}^{(i)}(x, \alpha_d)$ can be inferred from [16, 5]. We include the derivation for completeness. In order to avoid unnecessarily cumbersome notation in the following expressions, we neglect the explicit dependence on the particular subordinated process $\mathcal{M}^{(i)}(x)$, by eliminating the superscript (i). It is however tacitly intended, that all the quantities $\underline{r}$, $\mathbf{Q}(x)$, $\mathbf{F}(x, \alpha_d)$, $\mathbf{P}(x, \alpha_d)$, Δ and Ω refer to the specific process subordinated to state i.

4.2.1 Derivation of the Matrix Function $\mathbf{F}(x, \alpha_d)$

Conditioning on $H = h$, let us define:

$$F_{k\ell}(x, \alpha_d | H = h) = \begin{cases} \delta_{k\ell} U\left(x - \frac{\alpha_d}{r_k}\right) & \text{if: } h r_k \geq \alpha_d, \\ \sum_{u \in \Omega} \frac{dQ_{ku}(h)}{dQ_k(h)} \cdot F_{u\ell}(x - h, \alpha_d - h r_k) & \text{if: } h r_k < \alpha_d, \end{cases} \tag{13}$$

where $U(x)$ is the unit step function. In (13), two mutually exclusive events are identified. If $r_k \neq 0$ and $h\, r_k \geq \alpha_d$, a sojourn time equal to α_d is accumulated before leaving state k, so that the firing time (next regeneration time point) is $\tau_1^* = \alpha_d / r_k$. If $h\, r_k < \alpha_d$ then a transition occurs to state u with probability $dQ_{ku}(h)/dQ_k(h)$ and the residual service $(\alpha_d - h r_k)$ should be accomplished starting from state u at time $(x - h)$. Taking the LST transform of (13) with respect to x, we get:

$$F_{k\ell}^{\sim}(s, \alpha_d | H = h) = \begin{cases} \delta_{k\ell} e^{-s\alpha_d / r_k} & \text{if: } h r_k \geq \alpha_d, \\ e^{-sh} \sum_{u \in \Omega} \frac{dQ_{ku}(h)}{dQ_k(h)} F_{u\ell}^{\sim}(s, \alpha_d - h r_k) & \text{if: } h r_k < \alpha_d, \end{cases} \tag{14}$$

Unconditioning with respect to h, (14) becomes:

$$\begin{aligned} F_{k\ell}^{\sim}(s, \alpha_d) = & \delta_{k\ell} \left[1 - Q_k\left(\frac{\alpha_d}{r_k}\right)\right] e^{-s\alpha_d / r_k} + \\ & \sum_{u \in \Omega} \int_{h=0}^{\frac{\alpha_d}{r_k}} e^{-s\,h} F_{u\ell}^{\sim}(s, \alpha_d - h r_k) dQ_{ku}(h). \end{aligned} \tag{15}$$

Taking the Laplace transform (LT) with respect to α_d (denoting w the transform variable), and evaluating the integrals we obtain that the double LST-LT transform $F_{k\ell}^{\sim *}(s, w)$ satisfies the following equation:

$$F_{k\ell}^{\sim *}(s, w) = \delta_{k\ell} \frac{r_k \left[1 - Q_k^{\sim}(s + w r_k)\right]}{s + w r_k} + \sum_{u \in \Omega} Q_{ku}^{\sim}(s + w r_k) F_{u\ell}^{\sim *}(s, w). \tag{16}$$

4.2.2 Derivation of the Matrix Function $\mathbf{P}(x, \alpha_d)$

The derivation follows the same pattern as for the function $\mathbf{F}(x, \alpha_d)$. Conditioning on $H = h$, let us define:

$$P_{k\ell}(x, \alpha_d | H = h) = \begin{cases} \delta_{k\ell}\left[U(x) - U\left(x - \frac{\alpha_d}{r_k}\right)\right] \\ \qquad\qquad \text{if: } hr_k \geq \alpha_d \\ \delta_{k\ell}\left[U(x) - U(x-h)\right] + \\ \qquad + \sum_{u \in \Omega} \frac{dQ_{ku}(h)}{dQ_k(h)} P_{u\ell}(x-h, \alpha_d - hr_k) \\ \qquad\qquad \text{if: } hr_k < \alpha_d. \end{cases} \tag{17}$$

In (17), two mutually exclusive events are identified. If $r_k \neq 0$ and $\alpha_d \leq hr_k$, then the process spends all its time up to absorption in the initial state k. If $hr_k < \alpha_d$ then a transition occurs to state u with probability $dQ_{ku}(h)/dQ_k(h)$ and then the process jumps to state ℓ in the remaining time $(x - h)$ before completing the residual work $(\alpha_d - hr_k)$. Taking the LST transform of (17) with respect to x, we get:

$$P_{k\ell}^{\sim}(s, \alpha_d | H = h) = \begin{cases} \delta_{k\ell}\left[1 - e^{-s\alpha_d/r_k}\right] \\ \qquad\qquad \text{if: } hr_k \geq \alpha_d, \\ \delta_{k\ell}\left[1 - e^{-sh}\right] + \\ \qquad + e^{-sh} \sum_{u \in \Omega} \frac{dQ_{ku}(h)}{dQ_k(h)} P_{u\ell}^{\sim}(s, \alpha_d - hr_k) \\ \qquad\qquad \text{if: } hr_k < \alpha_d. \end{cases} \tag{18}$$

Unconditioning (18) with respect to h, taking the LT transform with respect to α_d (denoting w the transform variable), and finally evaluating the integrals we obtain that the double LST-LT transform $P_{k\ell}^{\sim *}(s, w)$ satisfies the following equation:

$$P_{k\ell}^{\sim *}(s, w) = \delta_{k\ell} \frac{s\left[1 - Q_k^{\sim}(s + wr_k)\right]}{w(s + wr_k)} + \sum_{u \in \Omega} Q_{ku}^{\sim}(s + wr_k) P_{u\ell}^{\sim *}(s, w). \tag{19}$$

4.2.3 The Subordinated Process is a Reward CTMC

Let us consider the particular case in which the subordinated process $Z(x)$ is a reward $CTMC$ with infinitesimal generator $\mathbf{A}$. The entries of the matrix functions $F_{k\ell}(x, \alpha_d)$ and $P_{k\ell}(x, \alpha_d)$ can be obtained from (16) and (19) by substituting the proper kernel describing the given *CTMC*:

$$Q_{k\ell}(x) = \begin{cases} \frac{a_{k\ell}}{-a_{kk}}(1 - e^{a_{kk}x}) & \text{if: } k \neq \ell, \\ 0 & \text{if: } k = \ell, \end{cases} \tag{20}$$

and in *LST* domain:

$$Q^{\sim}_{k\ell}(s) = \begin{cases} \frac{a_{k\ell}}{s - a_{kk}} & \text{if: } k \neq \ell, \\ 0 & \text{if: } k = \ell. \end{cases} \tag{21}$$

Keeping in mind that $a_{kk} = -\sum_{\ell \in \Omega} a_{k\ell}$ we have

$$(s + wr_k)F^{\sim *}_{k\ell}(s, w) = \delta_{k\ell} r_k + \sum_{u \in \Omega} a_{ku} F^{\sim *}_{u\ell}(s, w) \tag{22}$$

and

$$(s + wr_k)P^{\sim *}_{k\ell}(s, w) = \delta_{k\ell} s/w + \sum_{u \in \Omega} a_{ku}\, P^{\sim *}_{u\ell}(s, w). \tag{23}$$

Eqs. (22) and (23) can be rewritten in matrix form:

$$\mathbf{F}^{\sim *}(s, w) = (s\mathbf{I} + w\mathbf{R} - \mathbf{A})^{-1}\mathbf{R},$$

$$\mathbf{P}^{\sim *}(s, w) = \frac{s}{w}(s\mathbf{I} + w\mathbf{R} - \mathbf{A})^{-1},$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{R}$ is the diagonal matrix of the reward rates (r_k).

4.3 Steps of the Analysis Method

Finally the analysis method is composed by the following steps:

- find the regenerative instances and states of the studied model,
- classify the associated subordinated processes together with their concluding condition,
- evaluate the rows of the kernel elements in *LST* domain as it is in the previous subsections,
- then the transient behaviour is given by (6) in *LST* domain,
- the time domain transient behaviour is evaluated by a numerical inverse Laplace transformation.

5. Numerical Results

The closed form *LST* expressions of $\mathbf{K}(x)$ and $\mathbf{E}(x)$ for the two *prs* M/D/1/2/2 queuing systems are derived in detail, applying the technique developed in the previous section. The time domain values are obtained by performing an analytical inversion with respect to the transform variable w, and a numerical inversion with respect to the transform variable s.

A. – prs preemptive M/D/1/2/2 with identical customers – Let us build up the $\mathbf{K}^{\sim}(s)$ and $\mathbf{E}^{\sim}(s)$ matrices row by row by considering separately all the states that can be regeneration states and can originate a subordinated process. Since s_3 can never be a regeneration state the third row of the above matrices is irrelevant. The fact that s_3 is not a regeneration marking, means that the process can stay in s_3 only between two successive regeneration time points (*Fig. 2*).

A.1) – The starting regeneration state is s_1 - No deterministic transition is enabled and the next regeneration state can only be state s_2. Applying (8) we obtain:

$$K_{11}^{\sim}(s) = 0, \qquad K_{12}^{\sim}(s) = \tfrac{2\lambda}{s+2\lambda}, \quad K_{13}^{\sim}(s) = 0,$$

and (24)

$$E_{11}^{\sim}(s) = \tfrac{s}{s+2\lambda}, \quad E_{12}^{\sim}(s) = 0, \qquad E_{13}^{\sim}(s) = 0.$$

A.2) – The starting regeneration state is s_2 - Transition t_2 is deterministic so that the next regeneration time point is the epoch of firing of t_2. The subordinated process $\mathcal{M}^{(2)}(x)$ (*Fig. 2*) comprises states s_2 and s_3 and is a semimarkov process (since t_4 is deterministic) whose kernel is:

$$Q^{\sim}(s) = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{\lambda}{s+\lambda} \\ 0 & e^{-\alpha s} & 0 \end{vmatrix}$$

From Section 3, the reward vector is $\underline{r}^{(2)} = [0,\ 1,\ 0]$, and the only nonzero entry of the branching probability matrix is $\Delta_{21}^{(2)} = 1$. Applying *Eqs.* (16) and (19) we obtain the following results for the nonzero entries:

$$F_{22}^{\sim *}(s,w) = \frac{1}{s + w + \lambda - \lambda e^{-s\alpha}}$$

$$P_{22}^{\sim *}(s,w) = \frac{s/w}{s + w + \lambda - \lambda e^{-s\alpha}}$$

$$P_{23}^{\sim *}(s,w) = \frac{\lambda(1 - e^{-s\alpha})/w}{s + w + \lambda - \lambda e^{-s\alpha}}$$

Applying (12), and after inverting the LT transform with respect to w, the LST matrix functions $\mathbf{K}^{\sim}(s)$ and $\mathbf{E}^{\sim}(s)$ become:

$$K^{\sim}(s) = \begin{vmatrix} 0 & \frac{2\lambda}{s+2\lambda} & 0 \\ e^{-\alpha(s+\lambda-\lambda e^{-\alpha s})} & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \tag{25}$$

and

$$E^{\sim}(s) = \begin{vmatrix} \frac{s}{s+2\lambda} & 0 & 0 \\ 0 & \frac{s[1-e^{-\alpha(s+\lambda-\lambda e^{-\alpha s})}]}{s+\lambda-\lambda e^{-\alpha s}} & \frac{\lambda(1-e^{-\alpha s})[1-e^{-\alpha(s+\lambda-\lambda e^{-\alpha s})}]}{s+\lambda-\lambda e^{-\alpha s}} \\ 0 & 0 & 0 \end{vmatrix} \tag{26}$$

The LST of the state probabilities are obtained by solving (6). The time domain probabilities are calculated by numerically inverting (6) by resorting to the Jagerman method [15] revisited by Chimento and Trivedi in [6]. The plot of the state probabilities versus time for states s_1 and s_3 is depicted in *Fig. 4*, for $\alpha = 1$ and for two different values of the submitting rate $\lambda = 0.5$ and $\lambda = 2$.

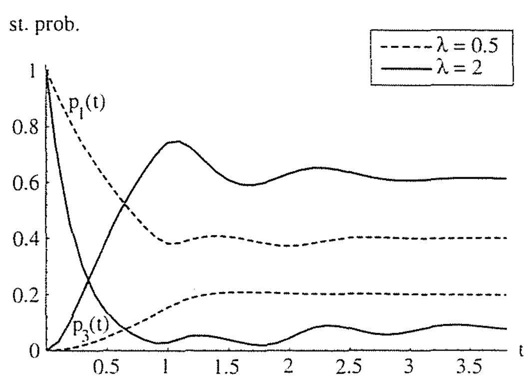

Fig. 4. Transient behavior of the state probabilities for the preemptive M/D/1/2/2 system with identical customers

B. – prs preemptive M/D/1/2/2 with different customers – The reachability graph in *Fig. 3b* comprises *4* states. Let us build up the $\mathbf{K}^{\sim}(s)$ and $\mathbf{E}^{\sim}(s)$ matrices row by row, taking into consideration that state s_4 can never be a regeneration marking since a type 2 job with nonzero age memory is always active.

B.1) – The starting regeneration state is s_1 – No deterministic transitions are enabled: the state is markovian and the next regeneration state can be either state s_2 or s_3. The nonzero elements of the 1-st row of matrices $\mathbf{K}^{\sim}(s)$ and $\mathbf{E}^{\sim}(s)$ are from (8):

$$K^{\sim}_{12}(s) = \frac{\lambda}{s+2\lambda} \quad ; \quad K^{\sim}_{13}(s) = \frac{\lambda}{s+2\lambda} \quad ; \quad E^{\sim}_{11}(s) = \frac{s}{s+2\lambda}$$

B.2) – The starting regeneration state is s_2 – The subordinated process coincides, in this case, with the subordinated process $\mathcal{M}^{(2)}(x)$ of the previous example (see *Fig. 2*), but with state s_4 in *Fig. 3b*, playing the role of state s_3 in *Fig. 1b*. Thus, with an obvious permutation of pedices, we can derive the nonzero entries $K^{\sim}_{21}(s)$, $E^{\sim}_{22}(s)$ and $E^{\sim}_{24}(s)$ from the 2-nd row in (25 and (26), respectively.

B.3) – The starting regeneration state is s_3 – The subordinated process is a *CTMC*, hence the results of Section 4.2.3 apply. The infinitesimal generator of the *CTMC* is:

$$\mathbf{A} = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\lambda & \lambda \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

and the reward vector is $\underline{r}^{(3)} = [0,\ 0,\ 1,\ 1]$. The branching probabilities arising from the firing of t_4 are $\Delta^{(3)}_{31} = 1$ and $\Delta^{(3)}_{42} = 1$. Applying (22) and (23) and solving the sets of equations, the nonzero entries take the form:

$$F^{\sim *}_{33}(s,w) = \frac{1}{s+\lambda+w} \quad ; \quad F^{\sim *}_{34}(s,w) = \frac{\lambda}{(s+w)(s+\lambda+w)}$$

$$P^{\sim *}_{33}(s,w) = \frac{s}{w(s+\lambda+w)}; \quad P^{\sim *}_{34}(s,w) = \frac{\lambda s}{w(s+w)(s+\lambda+w)}$$

Inverting the above equations with respect to w, taking into account the branching probabilities, yields:

$$K^{\sim}_{31}(s) = e^{-\alpha(s+\lambda)} \quad ; \quad K^{\sim}_{32}(s) = e^{-\alpha s}(1-e^{-\alpha\lambda})$$

$$E^{\sim}_{33}(s) = \frac{s}{s+\lambda}(1-e^{-\alpha(s+\lambda)}); \quad E^{\sim}_{34}(s) = \frac{\lambda}{s+\lambda} - (1-\frac{s}{s+\lambda}e^{-\alpha\lambda})e^{-\alpha s}$$

Finally, the complete $\mathbf{K}^{\sim}(s)$ and $\mathbf{E}^{\sim}(s)$ matrices become:

$$\mathbf{K}^{\sim}(s) = \begin{vmatrix} 0 & \frac{\lambda}{s+2\lambda} & \frac{\lambda}{s+2\lambda} & 0 \\ e^{-\alpha(s+\lambda-\lambda e^{-\alpha s})} & 0 & 0 & 0 \\ e^{-\alpha(s+\lambda)} & e^{-\alpha s}(1-e^{-\alpha\lambda}) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} \qquad (27)$$

and

$$\mathbf{E}^{\sim}(s) = \left| \begin{array}{cccc} \frac{s}{s+2\lambda} & 0 & 0 & 0 \\ 0 & \frac{s[1-e^{-\alpha(s+\lambda-\lambda e^{-\alpha s})}]}{s+\lambda-\lambda e^{-\alpha s}} & 0 & \frac{\lambda-\lambda e^{-\alpha s}[1-e^{-\alpha(s+\lambda-\lambda e^{-\alpha s})}]}{s+\lambda-\lambda e^{-\alpha s}} \\ 0 & 0 & \frac{s(1-e^{-\alpha(s+\lambda)})}{s+\lambda} & \frac{\lambda}{s+\lambda} - (\frac{s+\lambda-se^{-\alpha\lambda}}{s+\lambda})e^{-\alpha s} \\ 0 & 0 & 0 & 0 \end{array} \right| \tag{28}$$

As in the previous example, the time domain probabilities are calculated by numerically inverting (6). The plot of the state probabilities versus time for states s_1 and s_4 is reported in *Fig. 5*, for $\alpha = 1$ and for two different values of the submitting rate $\lambda = 0.5$ and $\lambda = 2$.

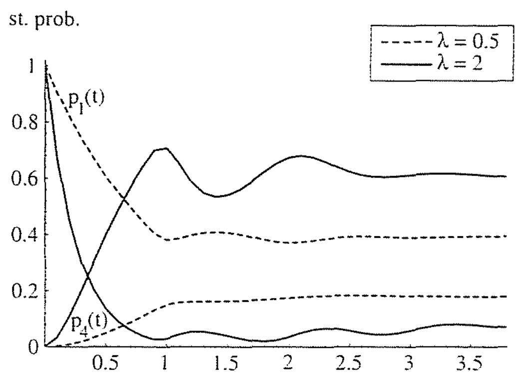

Fig. 5. Transient behavior of the state probabilities for the preemptive M/D/1/2/2 system with different customers

6. Conclusion

We have defined a new class of *DSPN*s called *Age Memory DSPN*s, which allow the inclusion of deterministic transitions with associated age memory policy. This extension was motivated by the need of modelling systems in which the execution of tasks may follow a preemptive resume policy.

We have shown that the marking process underlying an *Age Memory DSPN* is a Markov regenerative process, and hence the proposed class belongs to the class of *MRSPN*s introduced in [7]. A natural definition for the regeneration time points embedded into the Markov regenerative process is related to the simultaneous reset of all the age variables associated to the non-exponential transitions. The marking process between two consecutive regeneration time points can be, in general, a reward semimarkov process. A binary reward variable is introduced to distinguish between states in which the execution of the service is interrupted and states in which the execution is resumed with no loss of prior work. The age memory variable associated to a deterministic transition accumulates the time only in states in which the reward rate is equal to one. The firing time of a deterministic transition can be interpreted as the time at which the accumulated age memory variable reaches the value of the deterministic duration associated to the corresponding transition for the first time.

The transient analysis of a reward semimarkov process has been derived in detail, in order to show how to obtain a double *LT-LST* closed form expression for the transient state probabilities of the general process.

A M/D/1/2/2 queuing system, considered as a case study example in previous literature [3, 8], has been reexamined by introducing service policies of preemptive resume type, for the first time. Work is in progress to combine into a single *DSPN* preemptive service policies of both *prs* and *prd* type.

References

1. AJMONE MARSAN, M. – BALBO, G. – BOBBIO, A. – CHIOLA, G. – CONTE, G. – CUMANI, A.: The Effect of Execution Policies on the Semantics and Analysis of Stochastic Petri Nets. *IEEE Transactions on Software Engineering*, Vol. SE-15, pp. 832–846, 1989.
2. AJMONE MARSAN, M. – BALBO, G. – CONTE, G.: A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM Transactions on Computer Systems*, Vol. 2, pp. 93–122, 1984.
3. AJMONE MARSA, M. – CHIOLA, G.: On Petri Nets with Determinisitic and Exponentially Distributed Firing Times. In *Lecture Notes in Computer Science*, Vol. 266, pp. 132–145, Springer Verlag, 1987.
4. BOBBIO, A.: Stochastic Reward Models in Performance/Reliability Analysis, *Journal on Communications*, Vol. XLIII pp. 27–35, January 1992.
5. BOBBIO, A. – TELEK, M.: Task Completion Time. In *Proocedings 2nd International Workshop on Performability Modelling of Computer and Communication Systems (PMCC2)*, 1993.
6. CHIMENTO, P. F. – TRIVERDI, K. S.: The Completion Time of Programs on Processors Subject to Failure and Repair. *IEEE Transactions on Computers*, Vol. 42, pp. 1184–1194, 1993.
7. HOON CHOI, – KULKARNI, V. G. – TRIVERDI, K.: Markov Regenerative Stochastic Petri Nets. In G. Iazeolla and S.S. Lavenberg, editors, *Proocedings International Conference PERFORMANCE'93* pp. 339–356, 1993.
8. HOON CHOI, – KULKARNI, V. G. – TRIVERDI, K.: Transient Analysis of Deterministic and Sotchastic Petri Nets. In *Proceedings of the 14-th International Conference on Application and Theory of Petri Nets*, Chicago, June 1993.

9. CIARDO, G. – LINDEMANN, C.: Analysis of Deterministic and Stochastic Petri Nets. In *Proocedings International Workshop on Petri Nets and Performance Models – PNPM93*, pp. 160–169. IEEE Computer Society, 1993.
10. CIARDO, G. – TRIVERDI, K. S.: A Decomposition Approach for Stochastic Reward Net Models. *Performance Evaluation*, Vol. 18, pp. 37–59, 1993.
11. CINLAR, E.: *Introduction to Stochastic Processes.* Prentice-Hall, Englewood Cliffs, 1975.
12. GAVER, D. P.: A Waiting Line with Interrupted Service, Including Priorities. *Journal of the Royal Statistical Society*, B24, pp. 73–90, 1962.
13. GERMA, R. – LINDEMANN, C.: Analysis of Stochastic Petri Nets by the Method of Supplementary Variables. *Performance Evaluation*, Vol. 20, pp. 317–335, 1994.
14. HOWARD, R. A.: Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes. John Wiley and Sons, New York, 1971.
15. JAGERMAN, D. L.: Az Inversion Technique for the Laplace Transform. *The Bell System Technical Journal*, Vol. 61, pp. 1995–2002, October 1982.
16. KULKARNI, V. G. – NICOLA, V. F. – TRIVERDI, K.: On Modelling the Performance and Reliability of Multi-mode Computer Systems. *The Journal of Systems and Software*, Vol. 6, pp. 175–183, 1986.
17. LINDEMANN, C.: An Improved Numerical Algorithm for Calculating Steadystate Solutions of Deterministic and Stochastic Petri Nets Models. *Performance Evaluation*, Vol. 18, pp. 75–95, 1993.
18. REIBMAN A. – SMITH, R. – TRIVERDI, K. S.: Markov and Markov Reward Model Transient Analysis: an Overview of Numerical Approaches. *European Journal of Operational Research*, Vol. 40, pp. 257–267, 1989.