Steady State Solution of MRSPN with Mixed Preemption Policies

Miklós Telek
Department of Telecommunications
Technical University of Budapest, 1521 Budapest, Hungary
telek@plan.hit.bme.hu

Andrea Bobbio
Dipartimento di Informatica
Università di Torino, 10149 Torino, Italy
bobbio@di.unito.it

Antonio Puliafito
Istituto di Informatica e Telecomunicazioni
Università di Catania, 95125 Catania, Italy
puliafito@iit.unict.it

Abstract

Markov Regenerative Stochastic Petri Nets (MRSPN) have been recently recognized as a valuable tool to model systems with non-exponential timed activities. The usual assumption in the implementation of such models is that at most a single non-exponential transition, with associated enabling memory policy, can be enabled in each marking. More recently, new memory policies have been studied in order to represent more complex and effective preemption mechanisms in real systems. Closed-form solutions in the Laplace transform domain have been provided also in this case. This paper concentrates on the steady-state analysis of MRSPN and provides an unified analytical approach to include mixed memory policies into a single model. A numerical example concludes the paper.

Key words: Stochastic Petri Nets, Markov regenerative processes, preemptive policies, steady-state analysis.

1 Introduction

There is an increasing interest in the implementation of tools for performance/dependability analysis of computer and communication systems which incorporate the possibility of including, to some extent, non-exponential timed activities. *MRSPN*'s are a possible candidate to provide a useful interface language between the modeler's representation and the analytical representation [15]. *MRSPN*'s are defined as *SPN*'s with generally distributed firing times whose underlying marking process is a *Markov Regenerative Process (MRGP)* [5, 9]. *MRGP*'s are characterized by an embedded sequence of *regeneration time points*, such that the future evolution of the stochastic process depends only on the state entered when a regeneration time point occurs, and not on its past history.

The analysis technique proposed for this class of models, consists in identifying the sequence of the regeneration time points. Based on this sequence, an analytical formulation for the transition probability matrix of the process is available both in transient and in steady state [10, 16].

Choi et al. [8] recognized that the previous model referred to as *Deterministic and Stochastic Petri Net (DSPN)* [3] belonged to the class of *MRSPN* and provided closed-form expressions for the transition probability matrix. The main restriction on which this model is based is that at most a single non-exponential transition can be enabled in each marking, and the memory policy is of enabling type (according to the taxonomy in [1]). Some structural extensions were reported in [9] and improved numerical techniques in [17, 18]. An alternative numerical approach for the solution of the steady state or time domain equations based on the method of supplementary variables has been discussed in [12, 11]. A comparison of the different numerical techniques has been reported in [13].

In order to improve the modeling capabilities of *MRSPN*, Bobbio and Telek have recently discussed how to incorporate new and more complex preemption policies into the model. The *preemptive resume (prs)* policy, by means of which the model can remember the time spent by a generally distributed transition in previous enabling times, has been discussed in [6] and the steday state analysis has been provided in [19]. The *preemptive repeat identical (pri)* policy, by means of which an interrupted activity can be repeated with an identical requirement, has been discussed in [4].

The present paper concentrates on the possibility of including mixed preemption policies into a single model. A unified analytical procedure is envisaged, and the steday state case is discussed. Section 2 illustrates how different memory policies can be implemented into a *SPN* with generally distributed transitions. Section 3 discusses the unified analytical approach and Section 4 illustrates a completely developed example in which mixed preemption policies are put to work.

2 Preemption mechanisms and memory policies

A marked Petri Net is a tuple $PN = (P, T, I, O, H, M_0)$, where: P is the set of places, T the set of transitions, I, O and H are the input, the output and the inhibitor functions, respectively, and M_0 is the initial marking. The reachability set $\mathcal{R}(M_0)$ is the set of all the markings that can be generated from the initial marking M_0. The marking process $\mathcal{M}(t)$ denotes the marking occupied by the *PN* at time t. Since we are interested only in the time behavior of $\mathcal{M}(t)$, we can neglect immediate transitions [2] and suppose that the tangible part of the reachability set has been generated. The transitions can be distinguished as EXP transitions and GEN transitions. EXP transitions have associated an exponentially distributed firing time, while GEN transitions have associated a generally distributed firing time. A particular class of GEN transitions is the class of the DET transitions for which the firing time is assumed to be deterministic.

Since the basic *PN* considered in this paper contains GEN transitions, the underlying marking process $\mathcal{M}(t)$ is not memoryless. In order to completely specify the behaviour of the process at the *PN* level, not only firing time distributions, but also the memory policy of the transitions must be defined [1]. The memory policy is accounted for by assigning to each GEN transition a memory variable that accounts for the time the transition has been enabled. This approach to modeling memory into a *SPN* has been independently proposed in [1], in the analytical setting, and in [14], in the simulative setting.

With reference to Figure 1a, t_g is a generally distributed transition, γ_g the associated random firing delay, and a_g the memory variable. According to the above notation, the firing process of a transition can be represented as in Figure 1b. Suppose E is the time at which the transition becomes enabled: a clock associated to the transition starts counting linearly from 0 and the memory variable is assigned a value equal to the clock count. The transition fires as soon as a_g reaches a value equal to γ_g for the first time. Therefore, γ_g acts as an absorbing barrier for the functional a_g, and the firing process can be modeled as the first passage time of a_g across an absorbing barrier of height γ_g. The firing process of a given GEN transition is completely defined if the value of the memory variable at time t and the value of the barrier height γ_g are known.

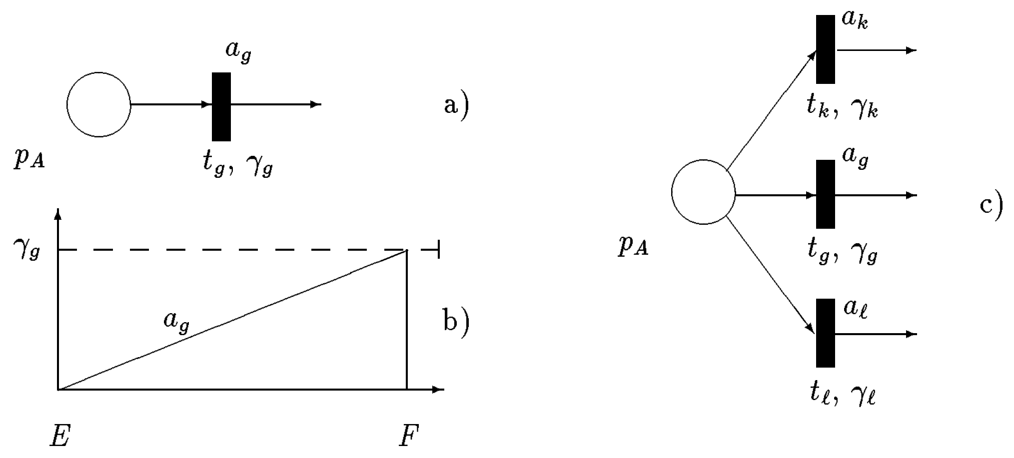


Figure 1 - Firing process of GEN transitions.

In Figure 2, several GEN transitions are output transitions to the same place p_A. Each transition has an independent firing time and an independent memory variable. When a token arrives for the first time in place p_A, all the transitions become enabled and their respective clocks start counting. When a transition fires the memory variable associated to that particular transition is reset, while the clocks associated to the other transitions are stopped. A clock can count (the memory variable increases) only when the corresponding transition is enabled. When a new token arrives in p_A, the three transitions become enabled again and their clocks restart counting.

In order to completely specify the behaviour of the *PN*, the values of the memory variables and of the threshold levels must be reassigned in the new state. The memory variable can retain the value previously reached or can restart from zero, and the barrier height can be resampled or not. The way in which these two reassignments are combined gives rise to different execution policies for the *PN*. In [1], an extensive discussion of the semantics implied by the alternative ways in which the memory variable can be reset or resumed has been discussed. However, the barrier was implicitly assumed to be resampled each time the memory variable was reset. Hence, borrowing the terminology from the queueing theory, if the memory variable is reset at each enabling the corresponding activity is assigned a *preemptive repeat different* (*prd*) execution policy, while if the memory variable is resumed the execution policy becomes *preemptive resume* (*prs*).

A new modeling framework for accommodating *preemptive repeat identical* (*pri*) policies has been devised in [4]. Under this policy, an interrupted activity is restarted from scratch in the new enabling period, but with a time requirement identical to the one before the interruption.

It is important to emphasize that a transition with exponentially distributed firing time and *pri* policy should be considered as a GEN one since it does not enjoy the memoryless property [4]. Thus, the marking process of a *PN* with only exponentially distributed firing times is not a *Continuous Time Markov Chain* (*CTMC*) if *pri* policy occur.

The memory of the global marking process is considered as the superposition of the memories of the individual transitions. In general, the underlying marking process is not analytically tractable unless some restrictions are specified.

2.1 Definition and analysis of MRSPN

MRSPN can be formally defined as follows.

Proposition 1 *A regeneration time point* τ_n^* *in the marking process* $\mathcal{M}(t)$ *of a* SPN *is the epoch of entrance in a marking* M_n *in which all the memory variables are zero and the barrier levels are resampled.*

By Proposition 1 the first regeneration time point is at time 0, i.e. $\tau_0^* = 0$

Definition 2 *A stochastic PN, for which a sequence of regeneration time points satisfying the condition of Proposition 1 exists, is a MRSPN.*

The marking process generated by a *PN* satisfying Definition 2 is, by definition, a *MRGP*, for which a closed-form solution is available [10, 16] in terms of the following matrix valued functions of dimension equal to the cardinality of the reachability set $\mathcal{R}(M_0)$:

$$
\begin{array}{rclcrcl}
\mathbf{V}(t) & = & [V_{ij}(t)] & \text{such that} & V_{ij}(t) & = & Pr\{\mathcal{M}(t) = j \mid \mathcal{M}(\tau_0^*) = i\} \\
\mathbf{K}(t) & = & [K_{ij}(t)] & " & K_{ij}(t) & = & Pr\{\mathcal{M}(\tau_1^*) = j \,,\, \tau_1^* \le t \mid \mathcal{M}(\tau_0^*) = i\} \\
\mathbf{E}(t) & = & [E_{ij}(t)] & " & E_{ij}(t) & = & Pr\{\mathcal{M}(t) = j \,,\, \tau_1^* > t \mid \mathcal{M}(\tau_0^*) = i\}
\end{array}
\quad (1)
$$

Matrix $\mathbf{V}(t)$ is the transition probability matrix and provides the probability that the stochastic process $\mathcal{M}(t)$ is in marking j at time t given it was in marking i at $t = 0$. The matrix $\mathbf{K}(t)$ is the *global kernel* of the *MRGP* 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 $\mathbf{E}(t)$ is the *local kernel* since it describes the behavior of the marking process $\mathcal{M}(t)$ between two consecutive regeneration time points. The generic element $E_{ij}(t)$ provides the probability that the process is in state j at time t starting from i at $\tau_0^* = 0$ before the next regeneration time point.
The transient behavior of the *MRSPN* can be evaluated by solving the following generalized Markov renewal equation (in matrix form) [10, 8]:

$$\mathbf{V}(t) \;=\; \mathbf{E}(t) \,+\, \mathbf{K} \,*\, \mathbf{V}(t) \tag{2}$$

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

$$[\mathbf{K} \,*\, \mathbf{V}(t)]_{ij} \;=\; \sum_k \int_0^t V_{kj}(t-y)\; dK_{ik}(y) \tag{3}$$

Equation (2) shows that the transition probability matrix depends on the knowledge of the local and global kernels. From (1), the kernel elements of the i-th row are determined by the single regeneration period starting in state i. Therefore, the analysis of the whole process can be decomposed into the analysis of the marking process between any two successive regeneration points (called the subordinated process).

The solution of Equation (2) becomes in the Laplace-Stieltjes (*LST*) domain:

$$\mathbf{V}^{\sim}(s) \;=[\mathbf{I} \,-\, \mathbf{K}^{\sim}(s)]^{-1}\; \mathbf{E}^{\sim}(s) \tag{4}$$

The steady-state solution can be evaluated as $\lim_{s\to 0} \mathbf{V}^{\sim}(s)$. However, according to [3, 7, 19], the steady-state probabilities can be derived directly from the local and global kernels. Let us define:

$$\alpha_{ij} = \int_{t=0}^{\infty} E_{ij}(t)dt = \lim_{s\to 0} \frac{1}{s}\; E_{ij}^{\sim}(s) \quad ; \qquad \alpha_i = \sum_j \alpha_{ij} \tag{5}$$

α_{ij} is the expected time a subordinated process starting from state i spends in state j. α_i is the expected duration of the subordinated process starting from state i before the next regeneration time point. The state transition probability matrix of the *DTMC* embedded into the regeneration time points is

$$\pi = \{\pi_{ij}\} = \lim_{t\to\infty} \mathbf{K}(t) = \lim_{s\to 0} \mathbf{K}^{\sim}(s) \tag{6}$$

Let $P = \{p_i\}$ (row vector) be the unique solution of the set of equations:

$$P = P\pi \quad ; \qquad \sum_i p_i = 1 \tag{7}$$

The steady-state probabilities of the *MRGP* can be evaluated based on α_{ij} and p_i (or π_{ij} by applying (7)) as follows [3, 16]:

$$v_{ij} = \lim_{t\to\infty} Pr\{\mathcal{M}(t) = j \mid \mathcal{M}(0) = i\} = \frac{\sum_k p_k \alpha_{kj}}{\sum_k p_k \alpha_k} \tag{8}$$

2.2 Preemption policies in MRSPN

The evaluation of the entries of the local and global kernels depends on the model structure and specification. In the present paper, we assume the modeling environment proposed in [6], where non-overlapping dominant transitions have been defined. In this environment, any two successive regeneration time points correspond to the first enabling and to the firing (or disabling) of a single GEN transition called the dominant transition. The regeneration periods dominated by different transitions cannot overlap. The entries of the i-th row of the kernel matrices $\mathbf{K}(t)$ and $\mathbf{E}(t)$ can be evaluated by analysing in isolation the subordinated process starting from state i and depend on the memory policy associated to the dominant transition.

The dominant transition is prd - Each time a *prd* dominant transition is disabled or fires, its memory variable is reset and its barrier level is resampled from the same distribution. With reference to Figure 2a, let E, D, and F be enabling, disabling or firing time instants, respectively. The transition is enabled for the first time at $t = 0$, and its memory variable starts increasing linearly. At point D, the transition is disabled and the memory is reset. At the next enabling time instant E the memory variable restarts from zero, and the barrier level is resampled from the same distribution assuming a different value. When the transition fires the memory variable is reset and the barrier

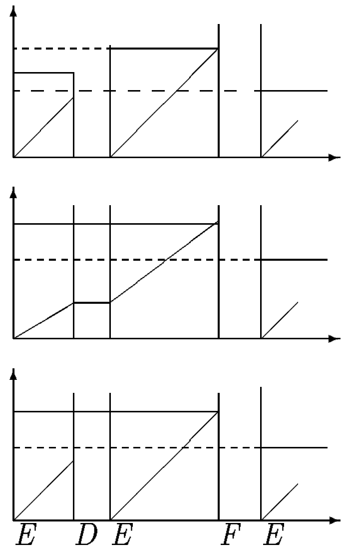


Figure 2 - Pictorial representation of different firing time sampling policies.

resampled. According to Proposition 1, both D and F are regeneration time points for the marking process. The memory of the process is confined to the period of time in which the dominant GEN transition is continuously enabled.

The first model constructed on these assumptions was the *DSPN* model proposed in [3]. Choi et al. [7] have recognized that the marking process underlying a *DSPN* is a *MRGP* and have extended the model by allowing a single GEN transition to be enabled in each marking [8]. Improved numerical techniques have been presented in [18] and some structural extension in [9]. An analysis technique, based on the use of supplementary variables, has been elaborated in [11].

The dominant transition is prs - With reference to Figure 2b, when the dominant transition is disabled (in point D), its associated clock is stopped but not reset; when the transition is enabled again, its memory variable restarts from the previously retained value. When the transition fires, the memory variable is reset and the barrier height resampled.

The regeneration period coincides with the firing cycle of the dominant transition. The states in which the dominant transition is enabled or disabled can be assigned a reward 1 or 0, respectively. In this way, the firing time of the dominant transition (corresponding to the duration of the regeneration period) can be calculated as the first passage time of the total accumulated reward against an absorbing barrier equal to firing time requirement of the dominant transition. In the context of *MRSPN*, this model has been proposed for the first time in [6, 19].

The dominant transition is pri - Under this policy (Figure 2c), each time the dominant transition is disabled, its memory variable is reset, but the barrier level remains active, so that in the next enabling period an identical work requirement should be accomplished. In Figure 2c, the same barrier level is maintained over different enabling periods up to the firing of the dominant transition. Only when the transition fires the barrier level is resampled and the memory variable reset. Hence, also in this case, the next regeneration time point can occur only upon firing of the dominant transition. In the context of *MRSPN*, this policy has been introduced in [4].

3 Steady-state analysis with mixed policies

In the following derivation, we assume that each regeneration period is dominated by a single GEN transition whose memory policy can be either *prs* or *prd* or *pri*. We provide a unified formalism for deriving the entries of the the global and local kernels $\mathbf{K}(t)$ and $\mathbf{E}(t)$ row by row or directly the elements of the α and π matrices when only the steady state values are of interest. In particular, we derive closed form expressions when the subordinated process is a *CTMC*. Once the kernels are known, application of formulas in Section (2.1) provides the transition probability matrix of the process.

Let us consider a single regeneration period starting at time $t = 0$ from marking i and let us suppose that the considered regeneration period is dominated by a GEN transition t_g with memory variable a_g and random firing time γ_g with Cdf $G_g(x)$. The process subordinated to the dominant transition is denoted by $Z^i(t)$. We restrict the following analysis to the case in which $Z^i(t)$ is a *CTMC*, i.e. only EXP transitions can be enebled during the regeneration period.

Once the entries of the i-th row have been determined, the same analysis must be repeated for any state $i \in \mathcal{R}(M_0)$ that can be a regeneration state. Given that the condition for non-overlapping dominant transitions are met over the whole state space, different preemption policies can be associated to different GEN transitions in the same model.

Let $R^i \subset \mathcal{R}(M_0)$ be the subset of states that can be reached during the subordinated process starting from state i, and let $\overline{R^i} = \mathcal{R}(M_0) - R^i$ be the complementary subset.

Any transition from $k \in R^i$ to $\ell \in \overline{R^i}$ concludes the regeneration period. Therefore, from the point of view of the analysis of the considered regeneration period, the states in $\overline{R^i}$ can be made absorbing.

The following analysis is developed in the case in which the firing time associated to the dominant transition is deterministic. If, however, γ_g is not deterministic but is GEN, the analysis proceeds in two steps:

1. Fix a value for the firing random time $\gamma_g = w$ and perform the analysis as in the deterministic case.

2. Uncondition the obtained results with respect to the distribution $G_g(w)$ of γ_g.

The state space R^i of the subordinated process can be partitioned into two subsets based on the enabling/disabling condition of the dominant transition.

- $\mathcal{E}^i \subset R^i$: groups the states reachable from i in which t_g is enabled. A reward rate equal to 1 is assigned to any $k \in \mathcal{E}^i$ so that the memory variable is strictly increasing.

- $\mathcal{D}^i$: groups the states reachable from i in which t_g is not enabled. A reward rate equal to 0 is assigned to any $k \in \mathcal{D}^i$ so that the memory variable is not increasing (the associated clock is not counting).

During the analysis of $Z^i(t)$ we renumber the states in $\mathcal{R}(M_0)$ for notational convenience so that the states numbered $1, 2, \ldots, h$ belong to the subset $\mathcal{E}^i$, in which the dominant transition t_g is enabled and the states numbered $h+1, h+2, \ldots, m+h$ belong to $\mathcal{D}^i$ in which t_g is disabled. The infinitesimal generator of the subordinated *CTMC* defined on R^i is partitioned in the following way (we drop the superscript i in the following notation where no ambiguity arises):

$$\mathbf{A^i} = \begin{array}{|c|c|} \hline \mathbf{B}_{\mathcal{E}} & \mathbf{B}_{\mathcal{E}\mathcal{D}} \\ \hline \mathbf{B}_{\mathcal{D}\mathcal{E}} & \mathbf{B}_{\mathcal{D}} \\ \hline \end{array} \qquad (9)$$

Let us introduce the following matrices of probability measures associated with the partitioned state space [4]:

- $\mathbf{P1}(t, w)$ is the state transition probability inside $\mathcal{E}$ at time t before absorption at the barrier w or before a passage to $\mathcal{D}$.

- $\mathbf{F1}(t,w)$ is the probability that t_g fires from a state in $\mathcal{E}$ before t, suppose that the subordinated process never left $\mathcal{E}$ between 0 and t.

- $\mathbf{P12}(t,w)$ is the distribution of the first passage time from $\mathcal{E}$ to a state of $\mathcal{D}$ before absorption at the barrier w.

- $\mathbf{P2}(t)$ is the state transition probability inside $\mathcal{D}$ at time t before a passage to $\mathcal{E}$.

- $\mathbf{P21}(t)$ is the distribution of the first passage time from $\mathcal{D}$ to a state of $\mathcal{E}$.

- $\boldsymbol{\Delta}$ is the branching probability matrix and represents the successor tangible marking $\ell \in \mathcal{R}(M_0)$ that is reached by firing t_g in a state $k \in \mathcal{E}$ [7].

According to these definitions, the following equalities hold for $\forall k, \ell \in R^i$:

$$P1_{k\ell}(t,w) + P12_{k\ell}(t,w) + F1_{k\ell}(t,w) = 1$$

$$P2_{k\ell}(t) + P21_{k\ell}(t) = 1$$

The above matrices can be evaluated in the Laplace transform domain based on the partitioned infinitesimal generator of the subordinated *CTMC*. Being s the transform variable of the time t and v the transform variable of the barrier height w the following expressions can be derived [6, 4]:

$$\mathbf{P1}^{\sim *}(s,v) \; = \; \frac{s}{v}\left((s+v)\mathbf{I} \; - \; \mathbf{B}_{\mathcal{E}}\right)^{-1} \tag{10}$$

$$\mathbf{P12}^{\sim *}(s,v) \; = \; \frac{1}{v}\left((s+v)\mathbf{I} \; - \; \mathbf{B}_{\mathcal{E}}\right)^{-1}\mathbf{B}_{\mathcal{E}\mathcal{D}} \tag{11}$$

$$\mathbf{P21}^{\sim}(s) \; = \; \left(s\mathbf{I} \; - \; \mathbf{B}_{\mathcal{D}}\right)^{-1}\mathbf{B}_{\mathcal{D}\mathcal{E}} \tag{12}$$

$$\mathbf{P2}^{\sim}(s) \; = \; s\,\left(s\mathbf{I} \; - \; \mathbf{B}_{\mathcal{D}}\right)^{-1} \tag{13}$$

$$\mathbf{F1}^{\sim *}(s,v) \; = \; \left((s+v)\mathbf{I} \; - \; \mathbf{B}_{\mathcal{E}}\right)^{-1} \tag{14}$$

after a symbolical inverse Laplace transformation according to the variable v we obtain:

$$\mathbf{F1}^{\sim}(s,w) \;=\; e^{(-s\mathbf{I}+\mathbf{B}_{\mathcal{E}})w} \tag{15}$$

$$\mathbf{P1}^{\sim}(s,w) \;=\; s\int_0^w e^{(-s\mathbf{I}+\mathbf{B}_{\mathcal{E}})w'}dw' \tag{16}$$

$$\mathbf{P12}^{\sim}(s,w) \;=\; \int_0^w e^{(-s\mathbf{I}+\mathbf{B}_{\mathcal{E}})w'}dw'\;\mathbf{B}_{\mathcal{ED}} \tag{17}$$

Let us particularize the general analysis with respect to the specific preemption policies.

3.1 The dominant transition is prd

Any transition out of $\mathcal{E}$ provides the next regeneration time point. Thus, $R^i = \mathcal{E}^i$ and subset $\mathcal{D}^i \subset \overline{R^i}$ can be made absorbing (i.e. $\mathbf{B}_{\mathcal{DE}} = \mathbf{0}$ and $\mathbf{B}_{\mathcal{D}} = \mathbf{0}$). The kernel entries of the i-row can be expressed as (compare with [6]):

$$\mathbf{E_i}^{\sim}(s|\gamma_g = w) \;=\; \mathbf{P1}^{\sim}(s,w) \tag{18}$$

$$\mathbf{K_i}^{\sim}(s|\gamma_g = w) \;=\; \mathbf{F1}^{\sim}(s,w)\,\mathbf{\Delta} \;+\; \mathbf{P12}^{\sim}(s,w) \tag{19}$$

These and the following equalities hold only for the i-row of the matrices.

For a fixed firing time w the steady state solution becomes [19]:

$$\alpha_i = LT^{-1}_{v\to w}[\;\lim_{s\to 0} 1/s\;\mathbf{P1}^{\sim *}(s,v)\;] = \mathbf{L}(w) \tag{20}$$

where $\mathbf{L}(w) = \int_{w'=0}^{w} e^{w'\;\mathbf{B}_{\mathcal{E}}}\;dw'$,

$$\begin{aligned}\pi_i &= LT^{-1}_{v\to w}[\;\mathbf{F1}^{\sim *}(0,v)\;]\;\mathbf{\Delta} + [\;\mathbf{0}\mid LT^{-1}_{v\to w}[\;\mathbf{P12}^{\sim *}(0,v)\;]\;]\\ &= e^{w\;\mathbf{B}_{\mathcal{E}}}\;\mathbf{\Delta} + [\;\mathbf{0}\mid \mathbf{L}(w)\;\mathbf{B}_{\mathcal{ED}}]\end{aligned} \tag{21}$$

3.2 The dominant transition is prs

The process alternates between subsets $\mathcal{E}$ and $\mathcal{D}$ until the total reward accumulated reaches the threshold w and the transition fires from a state in $\mathcal{E}$. The kernel entries of the i-row can be expressed as (compare with [6]):

$$\mathbf{E_i}^{\sim\sim}(s,v|\gamma_g = w) \;=\; [\mathbf{I} \;-\; \mathbf{P12}^{\sim\sim}(s,v)\,\mathbf{P21}^{\sim}(s)]^{-1}\,[\mathbf{P1}^{\sim\sim}(s,v) \mid \mathbf{P12}^{\sim\sim}(s,v)\,\mathbf{P2}^{\sim}(s)] \tag{22}$$

$$\mathbf{K_i}^{\sim\sim}(s,v|\gamma_g = w) \;=\; [\mathbf{I} \;-\; \mathbf{P12}^{\sim\sim}(s,v)\,\mathbf{P21}^{\sim}(s)]^{-1}\,\mathbf{F1}^{\sim\sim}(s,v)\,\mathbf{\Delta} \tag{23}$$

For a fixed firing time w the steady state solution becomes [19]:

$$\begin{aligned}
\alpha_i &= LST^{-1}_{v\to w}[\, \lim_{s\to 0}\; 1/s\; \mathbf{P}^{\sim\sim}(s,v)\,] \\
&= LST^{-1}_{v\to w}[\, \mathbf{I} + (v\mathbf{I} - \mathbf{B}_{\mathcal{E}})^{-1}\; \mathbf{B}_{\mathcal{E}\mathcal{D}}\; \mathbf{B}_{\mathcal{D}}^{-1}\; \mathbf{B}_{\mathcal{D}\mathcal{E}}\,]^{-1} \\
&\qquad\qquad [\,(v\mathbf{I} - \mathbf{B}_{\mathcal{E}})^{-1} \mid -(v\mathbf{I} - \mathbf{B}_{\mathcal{E}})^{-1}\; \mathbf{B}_{\mathcal{E}\mathcal{D}}\; \mathbf{B}_{\mathcal{D}}^{-1}\,] \\
&= LST^{-1}_{v\to w}[\,(v\mathbf{I} - \beta)^{-1} \mid -(v\mathbf{I} - \beta)^{-1}\; \mathbf{B}_{\mathcal{E}\mathcal{D}}\; \mathbf{B}_{\mathcal{D}}^{-1}\,] \\
&= \Big[\; \mathbf{L}_\beta(w) \;\Big|\; -\mathbf{L}_\beta(w)\; \mathbf{B}_{\mathcal{E}\mathcal{D}}\; \mathbf{B}_{\mathcal{D}}^{-1} \Big]
\end{aligned} \tag{24}$$

where

$$\beta = \mathbf{B}_{\mathcal{E}} - \mathbf{B}_{\mathcal{E}\mathcal{D}}\; \mathbf{B}_{\mathcal{D}}^{-1}\; \mathbf{B}_{\mathcal{E}\mathcal{D}} \quad \text{and} \quad \mathrm{L}_\beta(w) = \int_0^w e^{\beta w'}\, dw'$$

$$\begin{aligned}
\pi_i &= LST^{-1}_{v\to w}[\, \lim_{s\to 0}\; 1/s\; \mathbf{F}^{\sim\sim}(s,v)\,]\; \mathbf{\Delta} \\
&= LST^{-1}_{v\to w}[\, \mathbf{I} + (v\mathbf{I} - \mathbf{B}_{\mathcal{E}})^{-1}\; \mathbf{B}_{\mathcal{E}\mathcal{D}}\; \mathbf{B}_{\mathcal{D}}^{-1}\; \mathbf{B}_{\mathcal{D}\mathcal{E}}\,]^{-1}\;\; [\, v\; (v\mathbf{I} - \mathbf{B}_{\mathcal{E}})^{-1}\,]\;\; \mathbf{\Delta} \\
&= LST^{-1}_{v\to w}[\, v\; (v\mathbf{I} - \beta)^{-1}\,]\;\; \mathbf{\Delta} \\
&= [\, e^{w\beta}\,]\;\; \mathbf{\Delta}
\end{aligned} \tag{25}$$

3.3 The dominant transition is pri

The process alternates between subsets $\mathcal{E}$ and $\mathcal{D}$ but each time a transition out of $\mathcal{E}$ occurs the accumulated reward is reset. The transition fires as soon as the accumulated reward reaches the threshold w during a single stay in $\mathcal{E}$ [4].

$$\mathbf{E_i}^{\sim}(s|\gamma_g = w) \;=\; [\mathbf{I} \;-\; \mathbf{P12}^{\sim}(s,w)\,\mathbf{P21}^{\sim}(s)]^{-1}\,[\mathbf{P1}^{\sim}(s,w) \mid \mathbf{P12}^{\sim}(s,w)\,\mathbf{P2}^{\sim}(s)] \tag{26}$$

$$\mathbf{K_i}^{\sim}(s|\gamma_g = w) \;=\; [\mathbf{I} \;-\; \mathbf{P12}^{\sim}(s,w)\,\mathbf{P21}^{\sim}(s)]^{-1}\,\mathbf{F1}^{\sim}(s,w)\,\mathbf{\Delta} \tag{27}$$

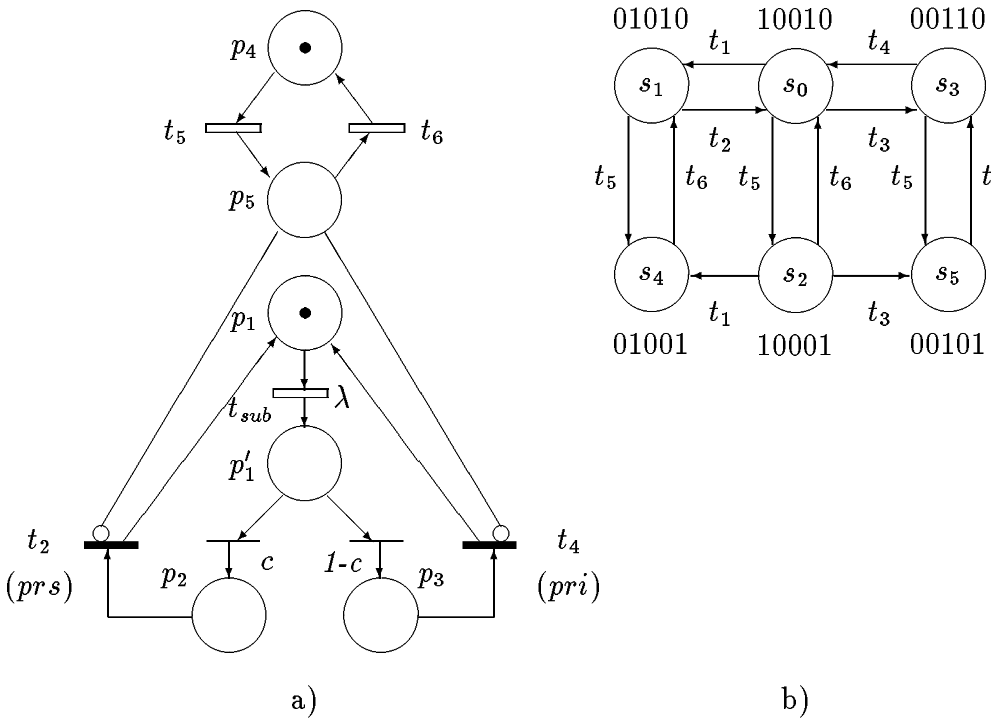


Figure 3 - The Petri net and the reduced reachability graph of the two processor system

For a fixed firing time w the steay state expressions become:

$$\alpha_i = [I + \mathbf{L}(w)\mathbf{B}_{\mathcal{E}\mathcal{D}}\ \mathbf{B}_{\mathcal{D}}^{-1}\ \mathbf{B}_{\mathcal{E}\mathcal{D}}]^{-1}[\mathbf{L}(w) \mid -\mathbf{L}(w)\ \mathbf{B}_{\mathcal{E}\mathcal{D}}\ \mathbf{B}_{\mathcal{D}}^{-1}] \tag{28}$$

$$\pi_i = [I + \mathbf{L}(w)\ \mathbf{B}_{\mathcal{E}\mathcal{D}}\ \mathbf{B}_{\mathcal{D}}^{-1}\ \mathbf{B}_{\mathcal{E}\mathcal{D}}]^{-1}[e^{\mathbf{B}_{\mathcal{E}} w}] \tag{29}$$

4 Combined preemption policies: an example

A two processor system runs two types of jobs according to the following scheduling policy. Jobs of class 1 require both processors and have preemptive priority over jobs of class 2. Jobs of class 2 have lower priority and are scheduled to run on a single processor that is chosen according to a predefined switching probability.

A PN modeling the system operation according to the described scheduling policy is represented in the Figure 3a). Place p_1 is the customer of class 2 thinking. He submits jobs at a rate λ (transition t_{sub}). The jobs of class 2 are sent to a single free processor according to a random choice represented by the switching probability c and $1-c$. Place

p_2 (p_3) is processor 1 (processor 2) servicing customer 2 with service time distribution modeled by transition t_2 (t_4). Place p_4 is customer 1 thinking, while place p_5 represents job 1 running on both processors while preempting customer 2 under service. (Inhibitors arcs from p_5 to both t_2 and t_4). Transition t_5 (t_6) has an exponentially distributed firing time with parameter λ_5 (λ_6).

We assume that the service time of customer 1 is GEN and therefore we associate a GEN random firing time to both t_2 and t_4 with distribution $G_2(w)$ and $G_4(w)$. We further assume that processor 1 has fault-tolerant capabilities so that the execution of jobs is *prs*. Processor 2, instead, does not have fault-tolerant capabilities so that a recovery of an interrupted job occurs according to a *pri* policy. To this end, we associate to transition t_2 a *prs* policy, and to transition t_4 a *pri* policy. All the other transitions are assumed to be EXP.

Elimination of vanishing markings leads to the tangible reachability graph of Figure 3b). Inspection of the reachability graph leads to the following assertions:

- State s_0 and s_2 are always regeneration states with outgoing EXP transitions;
- States s_1 and s_3 are regeneration states as well. According to the described system characteristics the subordinated process starting from state s_1 (s_3) is dominated by t_2 (t_4) with *prs* (*pri*) policy and concluded by the firing of t_2 (t_4).
- States s_4 and s_5 are regeneration states when entered from s_2, while are not regeneration states in the other cases since the memory variable associated to transitions t_2 and t_4 is never zero, when entered from s_1 and s_3, respectively. The outgoing transitions of s_4 and s_5 are EXP transitions.

From state s_0 the next regeneration markings can be either s_1, s_2 or s_3. From s_2, the next regeneration markings can be s_0, s_4 or s_5. The subordinated process starting form state s_1 (s_3) is a *CTMC* involving states s_1, s_4 (s_3, s_5), the preemption mechanism is *prs* (*pri*) and the next regeneration marking can only be s_0 (s_0).

Based on the above considerations

$$\pi = \begin{bmatrix} 0 & \frac{c\,\lambda}{\lambda+\lambda_5} & \frac{\lambda_5}{\lambda+\lambda_5} & \frac{(1-c)\,\lambda}{\lambda+\lambda_5} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{\lambda_6}{\lambda+\lambda_6} & 0 & 0 & 0 & \frac{c\,\lambda}{\lambda+\lambda_6} & \frac{(1-c)\,\lambda}{\lambda+\lambda_6} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\alpha = \begin{bmatrix} \frac{1}{\lambda+\lambda_5} & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_{s_1 s_1} & 0 & 0 & \alpha_{s_1 s_4} & 0 \\ 0 & 0 & \frac{1}{\lambda+\lambda_6} & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{s_3 s_3} & 0 & \alpha_{s_3 s_5} \\ 0 & 0 & 0 & 0 & \frac{1}{\lambda_6} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\lambda_6} \end{bmatrix}$$

The subordinated process starting from marking s_1 and s_3 are the same two-state CTMC with intensity λ_5 and λ_6. The difference between the relevant rows of the α matrices comes from the different preemption policies. For the evaluation of $\alpha_{s_1 s_1}$ and $\alpha_{s_1 s_4}$ Equation (24) is applied, while for the evaluation of $\alpha_{s_3 s_3}$ and $\alpha_{s_3 s_5}$ (28) is considered.

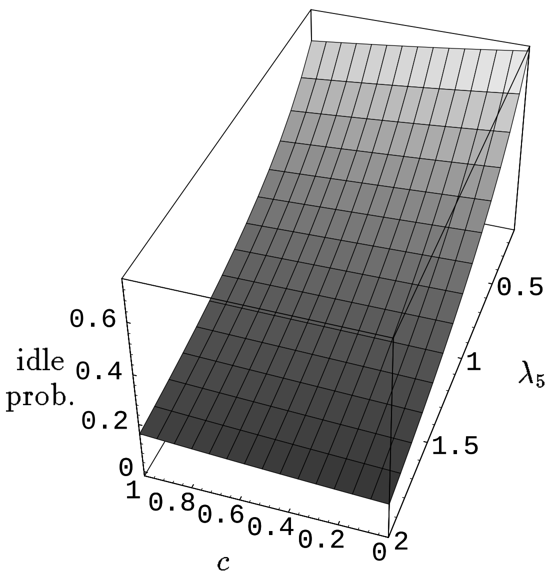
$$\alpha_{s_1 s_1} = \int_0^\infty w \; dG_2(w) \quad ; \quad \alpha_{s_1 s_4} = \int_0^\infty \frac{\lambda_5}{\lambda_6} \; w \; dG_2(w)$$

$$\alpha_{s_3 s_3} = \int_0^\infty \frac{1}{\lambda_5} \; (e^{\lambda_5 w} - 1) \; dG_4(w) \quad ; \quad \alpha_{s_3 s_5} = \int_0^\infty \frac{1}{\lambda_6} \; (e^{\lambda_5 w} - 1) \; dG_4(w)$$

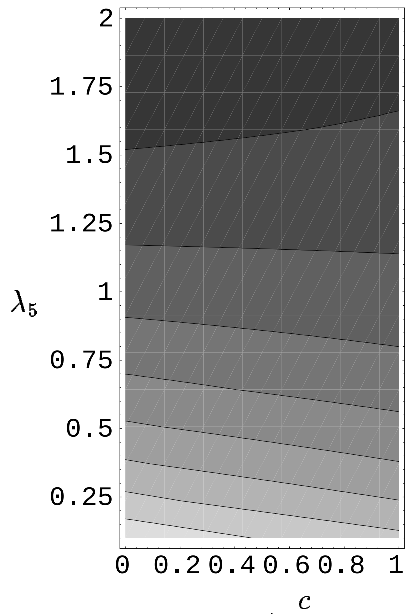
Suppose that the two processors are of different class, and processor 1 is slower than processor 2. The considered design problem consists in optimizing the switching probability c in order to optimize the performance characteristics of the system.

The steady state probabilities $(v_{ij}(t))$ can be evaluated based on $\alpha = \{\alpha_{ij}\}$ and π by Equation 7. With a given traffic pattern the dependence of the system performance on c can be measured by the steady state probability of the idle state (s_0). The better the system performance is the higher the steady state probability of s_0.

On Figure 4 the steady state probability of s_0 is depicted as a function of the switching probability and the submission rate of customer 1. Figure 4a contains the three dimensional view of the function. In order to emphasize the dependence on c Figure 4b shows the surface plot of it. The parameters of the model are set as follows:



a)



b)

Figure 4 - The dependence of the system performance on λ_5 and c

- the submission rate of customer 1 (λ_5) is varying from 0.1 to 2,
- the service rate of customer 1 (λ_6) is 1,
- the submission rate of customer 2 (λ) is 0.2,
- the service time of customer 2 is assumed to be deterministic. On the slower processor (processor 1) the service time is set equal to 2 (firing time of t_2), while on the faster processor (processor 2) is set equal to 1 (firing time of t_4),
- the switching probability (c) is varying between 0 and 1.

As can be seen on the figure the optimal value of c depends on λ_5 as well. For the case when $\lambda_5 > 1.25$ (frequent preemption of the low priority job) $c = 1$ (*prs* policy) results in the best preformance, and while $\lambda_5 < 1.25$ (rare preemption of the low priority job) $c = 0$ (*pri* policy) results in the best. No probabilistic mixture of the two preemtion policies ($0 < c < 1$) results in a better performance than the two limiting cases.

5 Conclusion

The paper has provided a common approach to include a mixture of preemption policies into a single *MRSPN*. This approach extends the modeling capabilities of previous formulations, where a single policy was considered at the time. In particular, complete equations have been provided for the steady state case, when all the subordinated processes are restricted to be *CTMC*.

6 Acknowledgements

A. Bobbio was partially supported by NATO grant No. CRG.940308. and CNR grant No.95.00431.CT12. M. Telek was partially supported by the Hungarian 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] A. Bobbio, V.G. Kulkarni, A. Puliafito, M. Telek, and K. Trivedi. Preemptive repeat identical transitions in Markov Regenerative Stochastic Petri Nets. In *6-th International Conference on Petri Nets and Performance Models - PNPM95*, pages 113–122, 1995.

[5] 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.

[6] 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.

[7] Hoon 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.

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

[9] 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.

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

[11] 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.

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

[13] R. German, D. Logothetis, and K. Trivedi. Transient analysis of Markov Regenerative Stochastic Petri Nets: a comparison of approaches. In *6-th International Conference on Petri Nets and Performance Models - PNPM95*, pages 103–112, 1995.

[14] P.J. Haas and G.S. Shedler. Regenerative stochastic Petri nets. *Performance Evaluation*, 6:189–204, 1986.

[15] B.R. Haverkort and K. Trivedi. Specification techniques for Markov Reward Models. *Discrete Event Dynamic Systems: Theory and Applications*, 3:219–247, 1993.

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

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

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

[19] 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 Modeling Techniques and Tools for Computer Performance Evaluation, Lecture Notes in Computer Science*, volume 977, pages 165–179, 1995.