First Conference on
Fault-Tolerant Systems, IIT Madras, Dec 1995

Combined Preemption Policies in MRSPN *

Andrea Bobbio
Dip. Elettronica per l'Automazione
Università di Brescia
25123 Brescia, Italy
bobbio@bsing.ing.unibs.it

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

Abstract

Markov Regenerative Stochastic Petri Nets *(MRSPN)* have been recently recognized as a valuable tool to model systems with non-exponential timed activities. Some restrictive assumptions should be specified in order to arrive to analytically and numerically computable models. The usual assumption is that at most a single non-exponential transition, with associated enabling memory policy, can be enabled in each marking. More recently, new preemption policies have been studied, and closed-form solutions in the Laplace transform domain have been provided. This paper illustrates how to activate different preemption policies in *MRSPN*.

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. *MRSPN*'s are defined as *SPN*'s whose underlying marking process is a *Markov Regenerative Process (MRGP)* [4, 8]. *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. An analytical expression for the transition probability matrix of the process can be formulated [9, 11], based on the sequence of the embedded regeneration time points.

Choi et al. [7] recognized that the previous model referred to as *DSPN (Deterministic and Stochastic PN)* [2] belonged to the class of *MRSPN* and provided closed-form expressions for both the transient and the steady-state equations. 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 its memory policy is of enabling type (according to the taxonomy in [1]). Some structural extensions [8] and improved numerical techniques [13, 10] where proposed based on the same assumptions.

*This work was performed while A. Bobbio was visiting the Electrical Engineering Department of the Duke University with the support of NATO grant No. CRG.940308.

Bobbio and Telek have recently discussed how to incorporate both *preemptive resume (prs)* [5] and *preemptive repeat identical (pri)* [3] policies in a *MRSPN* and have provided Laplace transform solutions. These new preemption policies are of crucial importance for modelling fault-tolerant systems where an interrupted job must be resumed from the point of interruption or restarted with an identical requirement.

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. The transitions can be distinguished into EXP and GEN. 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 considered PN's contain GEN transitions, the underlying marking process $\mathcal{M}(t)$ is not memoryless. In order to completely specify the model at the PN level, a memory policy needs to be superimposed to the basic PN. The memory policy is specified by assigning to each GEN transition a memory variable that accounts for the time the transition has been enabled [1].

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 t_g is represented as in Figure 1b. Suppose E is the time at which t_g 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 time of a GEN transition is completely determined by the value of the memory variable at time t and the value of the barrier height γ_g. In Figure 1c, 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. If, eventually, a new token arrives in p_A, the three transitions become enabled and their clocks start counting again. In the new state, the values of the memory variables and of the threshold levels must be reassigned . The memory variables 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. 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 reported. However in [1], the barrier was implicitly assumed to be resampled each time the memory variable was reset.

A new modeling framework for accommodating *pri* policies has been devised in [3]. A *pri* policy means that the interrupted job is restarted in the new state with an identical requirement. Notice that the *pri* memory policy destroys the Markov property even if the corresponding firing time is EXP, since at any new enabling the old barrier level must be remembered. The memory

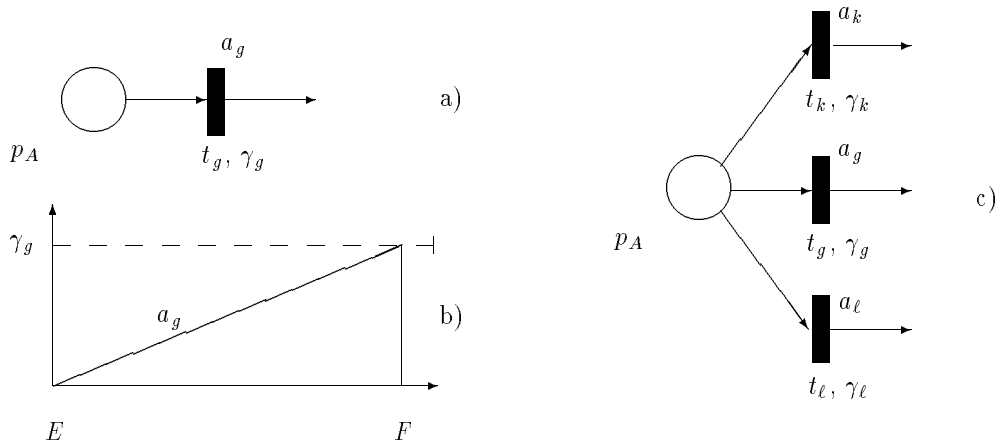

Figure 1 - Firing process of GEN transitions.

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 Closed form analysis of MRSPN

In this paper, we consider *MRSPN*, that 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.*

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*. The transient behavior of the *MRSPN* can be evaluated by solving the following generalized Markov renewal equation (in matrix form) [9, 7]:

$$\mathbf{V}(t) = \mathbf{E}(t) + \mathbf{K} * \mathbf{V}(t) \tag{1}$$

where $\mathbf{K} * \mathbf{V}(t)$ is a convolution matrix, and matrices $\mathbf{V}(t)$, $\mathbf{K}(t)$ and $\mathbf{E}(t)$ are defined as follows. $\mathbf{V}(t)$ 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 $t = 0$. $\mathbf{K}(t)$ is the *global kernel* and provides the cdf of the event that 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.

Equation (1) implies that 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). Equation (1) becomes in the Laplace-Stieltjes *(LST)* domain:

$$\mathbf{V}^{\sim}(s) = [\mathbf{I} - \mathbf{K}^{\sim}(s)]^{-1}\ \mathbf{E}^{\sim}(s) \tag{2}$$

The steady-state solution can be evaluated as $\lim_{s\to 0} \mathbf{V}^{\sim}(s)$. However, according to [2, 6, 14], the steady-state probabilities can be derived directly from the the local and global kernels. Let us define:

$$\alpha_{ij} = \int_{t=0}^{\infty} E_{ij}(t)dt \quad ; \qquad \alpha_i = \sum_j \alpha_{ij} \tag{3}$$

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 transition probability matrix of the $DTMC$ embedded into the regeneration time points is

$$\mathbf{\Pi} = \{\pi_{ij}\} = \lim_{t\to\infty} \mathbf{K}(t) \tag{4}$$

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

$$P = P\mathbf{\Pi} \quad ; \qquad \sum_i p_i = 1 \tag{5}$$

The steady-state probabilities of the $MRGP$ can be evaluated based on α_{ij} and p_i (or π_{ij}) as follows [2, 11]:

$$v_{ij} = \lim_{t\to\infty} Pr\{\mathcal{M}(t) = j \,|\, \mathcal{M}(0) = i\} = \frac{\sum_k p_k \alpha_{kj}}{\sum_k p_k \alpha_k} \tag{6}$$

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 [5], where non-overlapping dominant transitions have been defined. In this model, 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 firing process of the dominant transition.

In the past, three alternatives have been considered for taking into account how the memory variable and the barrier height of the dominant transition are reassigned in case of disabling or firing.

Preemptive repeat different (prd) policy - 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 is restarted from zero, and the barrier level is

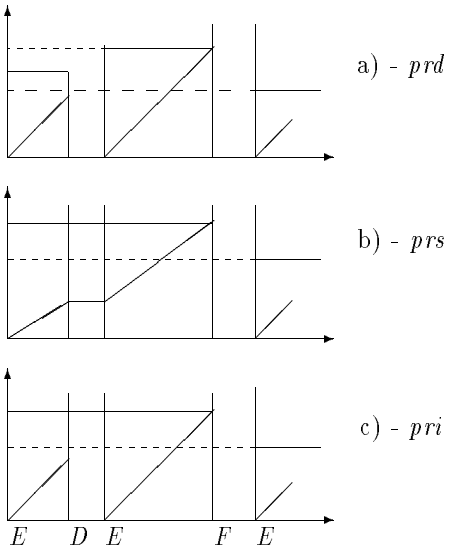

Figure 2 - Pictorial representation of different firing time sampling policies.

resampled from the same distribution assuming a different value. When the transition fires the memory variable is reset and the barrier 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 is the *DSPN* proposed in [2]. Choi et al. [6] 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 [7]. Improved numerical techniques have been presented in [13] and some structural extension in [8]. An analysis technique, based on the use of supplementary variables, has been elaborated in [10].

Preemptive resume (prs) policy - The dominant transition is assigned a *prs* memory policy. 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, so that the firing time becomes the first passage time of the total accumulated reward against the barrier level. In the context of *MRSPN* this model has been proposed for the first time in [5, 14].

Preemptive repeat identical (pri) policy - The dominant transition is assigned a *pri* memory policy. Under this policy (Figure 2c), each time a transition is disabled, the memory variable is reset, but the barrier level remains active, so that in the next enabling period an identical work requirement should be accomplished. Only when the transition fires the barrier level is resampled and the memory variable is 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 [3].

3 Analysis of a subordinated CTMC

Let us suppose that a regeneration period starts at time $t = 0$ from marking i and is dominated by a GEN transition t_g with memory variable a_g and random firing time γ_g. Let us suppose that the process subordinated to the dominant transition is a *CTMC* with infinitesimal generator $\mathbf{A^i}$. The following analysis provides a mean to evaluate the i-th row of the global and local kernels $\mathbf{K}(t)$ and $\mathbf{E}(t)$. The same analysis must be repeated for any state $i \in \mathcal{R}(M_0)$ that can be a regeneration state. Let the state space $\mathcal{R}(M_0)$ be partitioned into three subsets:

- $\mathcal{E}^i$: groups the state reachable from i in which t_g is enabled. For any $k \in \mathcal{E}^i$ the reward rate is equal to 1 and the memory variable is strictly increasing. Hence, the firing of the dominant transition can only occur from a state $k \in \mathcal{E}^i$, and after firing the process can jump to any state ℓ according to a switching probability matrix Δ^i.
- $\mathcal{D}^i$: groups the states reachable from i in which t_g is not enabled. For any $k \in \mathcal{D}^i$ the reward rate is equal to 0.
- $\mathcal{F}^i$: is the complementary set $\mathcal{F}^i = \mathcal{R}(M_0) - \mathcal{E}^i - \mathcal{D}^i$.

The different preemption policies, introduced in the previous section, are characterized by the following behaviour:

- *prd*: any transition out of $\mathcal{E}^i$ provides the next regeneration time point. Subsets $\mathcal{D}^i$ and $\mathcal{F}^i$ can be made absorbing.
- *prs*: states in $\mathcal{E}^i$ and $\mathcal{D}^i$ can be merged together: states in $\mathcal{E}^i$ have an associated reward rate equal to 1 and states in $\mathcal{D}^i$ have an associated reward rate equal to 0. The firing of the dominant transition occurs from a state $k \in \mathcal{E}^i$ [5].
- *pri*: only transitions out of $\mathcal{E}^i$ provide the next regeneration time point. The semantics of this policy requires the evaluation of the first passage time matrix out of $\mathcal{E}^i$ into $\mathcal{D}^i$ and out of $\mathcal{D}^i$ into $\mathcal{E}^i$ [3].

3.1 pri dominant transition

By a suitable reordering of the states, the infinitesimal generator of the subordinated *CTMC* starting from state i is partitioned in the following way:

$$\mathbf{A^i} = \begin{array}{|c|c|} \hline B_{\mathcal{E}} & B_{\mathcal{E}\mathcal{D}} \\ \hline B_{\mathcal{D}\mathcal{E}} & B_{\mathcal{D}} \\ \hline \end{array} \tag{7}$$

The values of α_{ij} and π_{ij} are computed in the *pri* case as a limit of the transient solution derived in [3].

$$\begin{aligned} \alpha^i_{ij}(w) &= [I_1 + L(w)B_{\mathcal{E}\mathcal{D}}B_{\mathcal{D}}^{-1}B_{\mathcal{D}\mathcal{E}}]^{-1}[L(w) \mid -L(w)B_{\mathcal{E}\mathcal{D}}B_{\mathcal{D}}^{-1}] \\ \pi^i_{ij}(w) &= [I_1 + L(w)B_{\mathcal{E}\mathcal{D}}B_{\mathcal{D}}^{-1}B_{\mathcal{D}\mathcal{E}}]^{-1}[e^{B_{\mathcal{E}}w} \mid 0] \end{aligned} \tag{8}$$

where $L(w) = \int_0^w e^{B_{\mathcal{E}}w}dw$.

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 and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In *LNCS*, volume 266, pages 132–145. Springer Verlag, 1987.

[3] 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 Int Conf PNPM95*, pages 113–122, 1995.

[4] 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), LNCS*, volume 852, pages 131–148, 1994.

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

[6] H. Choi, V.G. Kulkarni, and K. Trivedi. Transient analysis of deterministic and stochastic Petri nets. In *Proc of the 14-th Int Conf Application and Theory of Petri Nets*, Chicago, 1993.

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

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

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

[10] R. German. New results for the analysis of deterministic and stochastic Petri nets. In *IPDS95*, pages 114–123. IEEE CS Press, 1995.

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

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

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

[14] M. Telek, A. Bobbio, L. Jereb, A. Puliafito, and K. Trivedi. Steady state analysis of Markov regenerative SPN with age memory policy. *8-th Int Conf on Modeling Techniques and Tools for Computer Performance Evaluation, LNCS*, 977, pp 165–179, 1995.