Proceedings of the 8th International Workshop on Discrete Event Systems Ann Arbor, Michigan, USA, July 10-12, 2006

On performance monotonicity and basic servers semantics of continuous Petri nets

C. Mahulea, L. Recalde, M. Silva

Abstract— **Continuous Petri nets were introduced as an approximation to deal with the state explosion problem which can appear in discrete event models. When time is introduced, the flow through a fluidified transition can be defined in many ways. The most used in literature are infinite and finite servers semantics. Both can be seen as derived from stochastic Petri nets. The practical problems addressed in this contribution are: (1) a sufficient condition for the performance monotonicity, and (2) a study of the transition semantics, always related to continuous Petri nets. We prove that under some conditions, the subclass of mono-T-semiflow is monotone and also for the same class of nets we prove a property for which infinite servers semantics offers a better approximation than finite servers semantics for the discrete model.**

I. INTRODUCTION

PETRI nets (PNs) are a well-known formalism to deal with discrete event systems (DES) [1]. The state explosion problem appears frequently in such systems, making enumerative analysis methods practically inapplicable. Fluidification is an approximation technique that relaxes the description by removing the integrality constraints. Applying this idea to the discrete PNs, the firing of the transitions is not limited to natural numbers but to positive real numbers leading to continuous Petri nets (contPNs)[2][3].

As in the discrete case, contPNs can be autonomous (untimed) or can have time associated with the transitions or places, namely timed contPNs. In the literature, two servers semantics are often used in the continuous case, both closely related to the semantics used in discrete stochastic PNs. These semantics are: *finite servers semantics* (*constant speed*) and *infinite servers semantics* (*variable speed*) [2] [3].

In the first part of the paper the firing speed *monotonicity* (i.e. if $\boldsymbol{\lambda_1} \geq \boldsymbol{\lambda_2}$, with $\boldsymbol{\lambda}_i$ the vector of firing rate of the transitions, then the system throughput with $\boldsymbol{\lambda_1}$ is greater than or equal to the system throughput with $\boldsymbol{\lambda_2}$) is studied for the case of mono-T-semiflow PNs. The second part deals with the two firing semantics. The objective is to provide some guiding rule about their use: which semantics is better and in which conditions? In [4] the authors mention to have observed that in several cases *infinite servers semantics* provides a ”very accurate” approximation of discrete PNs. In this paper, we prove that infinite servers semantics is always better for mono-T-semiflow nets under some broad conditions.

The results provided here are proved for mono-T-semiflow nets but can be extended immediately to a more general class mono-T-semiflow reducible nets (as in [5]). This class (mono-T-semiflow reducible) offers a significant modeling power from a practical point of view. Focusing on live and bounded systems, the class of mono-T-semiflow reducible nets includes the class of equal conflict nets [6], which is a superset of the classes of free-choice, choice-free [7], weighted T-systems and marked graphs nets [8].

This paper is organized as follows: in Section II basic definitions of timed contPN and some ideas about fluidification are given. A characterization of performance monotonicity with respect to $\boldsymbol{\lambda}$ is proved in Section III. Section IV provides a comparison of the most used firing semantics for conPNs and is demonstrated that infinite servers semantics gives a better approximation of a discrete model. In Section V these two semantics are compared using an example. Some conclusions are given in Section VI.

II. ON THE FLUIDIFICATION OF PN: BASICS SEMANTICS

In the first part of this section we will give a definition of contPN and timed contPN. We assume that the reader is familiar with discrete PNs (for more details see [1] [2])

Definition 2.1: A contPN system is a pair $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$, where $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ is a net structure with a set of places P, a set of transitions T and the pre and post incidence matrices $\boldsymbol{Pre}$ and $\boldsymbol{Post}$, and $\boldsymbol{m_0}$ is the initial marking (distributed state).

A transition $t_j \in T$ is enabled at $\boldsymbol{m}$ iff $\forall p_i \in {}^{\bullet}t_j$, $m_i > 0$ and its enabling degree is:

$$enab(t_j, \boldsymbol{m}) = \min_{p_i \in {}^{\bullet}t_j} \left\{ \frac{m_i}{\boldsymbol{Pre}(p_i, t_j)} \right\}$$

An enabled transition t can fire in any amount $0 < \alpha < enab(t, \boldsymbol{m})$ leading to a new marking $\boldsymbol{m'} = \boldsymbol{m} + \alpha \boldsymbol{C}(:, t)$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the token-flow matrix.

A contPN is *bounded* when every place is bounded ($\forall p \in P, \exists b_p \in \mathbb{R}_{\geq 0}$ with $m(p) \leq b_p$ at every reachable marking $\boldsymbol{m}$). It is *live* when every transition is *live* (it can ultimately occur from every reachable marking). A net $\mathcal{N}$ is *structurally bounded* when $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ is bounded for every initial marking $\boldsymbol{m_0}$ and is *structurally live* when a $\boldsymbol{m_0}$ exists such that $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ is live.

Right and left non negative annullers of the token flow matrix $\boldsymbol{C}$ are called T- and P-semiflows, respectively. If non negativity is not required, the annullers are called T- and P-flows. When $\boldsymbol{y} \cdot \boldsymbol{C} = 0$, $\boldsymbol{y} > 0$ the net is said to be *conservative*, and when $\boldsymbol{C} \cdot \boldsymbol{x} = 0$, $\boldsymbol{x} > 0$ the net is said to

This work was partially supported by the project CICYT and FEDER DPI2003-06376.

The authors are with Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, María de Luna 1, 50018 Zaragoza, Spain {cmahulea, lrecalde, silva}@unizar.es

Clients	Servers	Semantics of the transition
Many(C)	Many(C)	infinite servers semantics
Many(C)	Few(D)	finite servers semantics
Few(D)	Few(D)	discrete transitions
Few(D)	Many(C)	discrete transitions

TABLE I
ON THE FLUIDIFICATION OF A TRANSITION [9]

be *consistent*. The support of a vector $\boldsymbol{v}$ is denoted by $||\boldsymbol{v}||$ and is the set of the non-zero components.

Two transitions t_1 and t_2 are said to be in *conflict relation* if ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$. They are said to be in *equal conflict* relation when $\boldsymbol{Pre}[P, t_1] = \boldsymbol{Pre}[P, t_2] \neq 0$. As in discrete nets, continuous nets can be classified according to their structure: a net is *Join Free* (JF) if $\forall t \in T$, $|{}^\bullet t| \leq 1$; a net is *Choice Free* (CF) if $\forall p \in P$, $|p^\bullet| \leq 1$; a net is *Equal Conflict* (EQ) iff $\forall t_1, t_2 \in T$ such that ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$, $\boldsymbol{Pre}[P, t_1] = \boldsymbol{Pre}[P, t_2]$.

***Definition** 2.2:* A time contPN system is a contPN system together with a vector $\boldsymbol{\lambda} : T \to \mathbb{R}_{>0}$, where λ_j is the firing rate or maximum firing speed (depending on the semantics) associated to the transition t_j.

In this way, the fundamental equation depends on time: $\boldsymbol{m}(\tau) = \boldsymbol{m_0} + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau)$. Differentiating with respect to time the following equation is obtained: $\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$. The derivative of the firing sequence is the *flow* of the timed model: $\boldsymbol{f}(\tau) = \dot{\boldsymbol{\sigma}}(\tau)$.

Notice that not every transition can be fluidified, for example in a traffic system, the ”power on” or ”power off” of a semaphore is purely discrete and in many cases can be inappropriate to fluidify it. If some transitions remain discrete and some are continuous then the model is conceptually *hybrid* [2], [9].

In timed continuous models, in order to associate a time semantics to a transition, it should be taken into account that a transition is like a station in Queuing Networks (QNs): the meeting point of servers and clients. If a transition *can be fluidified*, the more appropriate firing semantics depends on the number of servers and clients. According to the values of these variables, among other elements, the fluidification may be appropriate or not.

It is evident that if the number of clients is small (Few-Few and Few-Many in Table I), the transitions ”should” remain discrete and the fluidification may be unsuitable. Notice that, in the second case simply the place that models the servers is *implicit* [10] and can be removed.

If there are many clients and many servers (Many-Many), a continuous model with infinite servers semantics ”would” be reasonable (since there are so many servers that there is no need to make them explicit). On the other hand, in the case of many clients and a few servers (Many-Few) the relaxation is at the level of clients, and finite servers semantics can provide a good approximation (the firing speed of the transition is bounded by the product of the speed of a server and the number of servers in the station), but it may be not the case (as should be proved later for some cases).

Therefore, (according to Table I) two servers semantics are often used for continuous approximation: *finite servers semantics* (*constant speed*) and *infinite servers semantics* (*variable speed*) [2], [3]. It might seem that finite servers is more accurate than infinite servers. However, in PNs the distinction is not so clear as in Queuing Networks, since ”resources” are usually shared and this is made explicit in the model. This means that in practice infinite servers semantics is often more accurate, since the restrictions are more precise.

Under infinite servers semantics it is assumed that the flow through a synchronization uses the *min* operator expressing the number of processable tuples of clients in synchronizations (in population dynamic systems the product is used instead [3], expressing probabilities of meeting of tuples of clients in a synchronization).

Hence, under *infinite servers semantics* the flow of a transition t_j can be expressed as:

$$f_j = \lambda_j \cdot enab(t_j, \boldsymbol{m}) = \lambda_j \cdot \min_{p_i \in {}^\bullet t_j} \left\{ \frac{m_i}{Pre[p_i, t_j]} \right\} \qquad (1)$$

So, the enabling degree of the transition t_j gives the number of active servers of the transition and the flow will be its firing rate (λ_j) times the number of active servers. Notice that the number of these servers depends only on the marking of the input places and it is not bounded to any value.

Unlike the previous case, the number of active servers in a transition under *finite server semantics* is bounded to a natural number. Each transition has associated a real number called *maximal firing speed* and if the markings of the input places of a transition are strictly greater than zero, its flow will be constant, equal with this value (all servers working at full speed). Otherwise, the flow will be the minimum between its *maximal firing speed* and the total input flow of the places with zero marking.

$$f_j = \begin{cases} \lambda_j, \text{ if } \nexists p_i \in {}^\bullet t \text{ with } m_i = 0 \\ \min \left\{ \min\limits_{p_i \in {}^\bullet t | m_i = 0} \left\{ \sum\limits_{t' \in {}^\bullet p_i} \frac{f[t'] \cdot Post[t', p_i]}{Pre[p_i, t_j]} \right\}, \lambda_j \right\} \\ \text{otherwise} \end{cases} \qquad (2)$$

***Example** 2.3:* Let us consider the continuous relaxed view of the PN system in Fig. 1 with $Buf1, Buf2 > 0$. The flow of the transitions will be:

- Under infinite servers semantics
  $f_1(\tau) = \lambda_1 \cdot \boldsymbol{m}[M1]$
  $f_2(\tau) = \lambda_2 \cdot \boldsymbol{m}[p1]$
  $f_3(\tau) = \lambda_3 \cdot \min \{\boldsymbol{m}[p2], \boldsymbol{m}[Buffer1]\}$
  $\cdots$
- Under finite servers semantics
  $\boldsymbol{m}[M1] > 0 \Longrightarrow f_1(0) = \lambda_1; f_{10}(0) = \lambda_{10}$
  $\boldsymbol{m}[p1] = 0 \Longrightarrow f_2(0) = \min \{\lambda_2, f_1\} = \min \{\lambda_2, \lambda_1\}$
  $\boldsymbol{m}[p2] = 0; \ \boldsymbol{m}[Buffer1] > 0 \Longrightarrow f_3(0) = \min \{\lambda_3, f_2\} = \min \{\lambda_3, \lambda_2\}$
  $\cdots$

***Remark** 2.4:* The constant λ_j associated to the transition t_j has a different meaning under each semantics: it is the

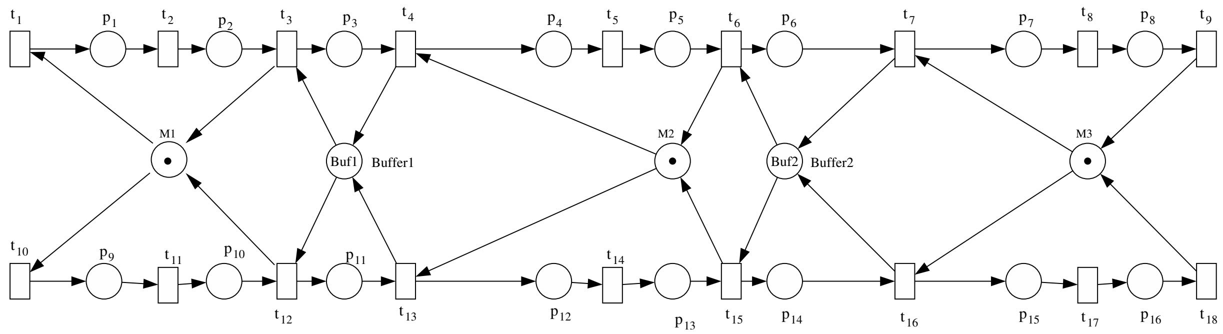


Fig. 1. Continuous mono-T-semiflow reducible net system used in Example 2.3.

firing rate of a transition in the case of infinite servers semantics and it is a maximal firing speed in the case of finite servers semantics (it is the product of the number of the servers and the firing rate of one server).

Piecewise linear behaviors are obtained, either under finite or infinite servers semantics. Under finite servers semantics the behavior is changed when a place is emptied; under infinite servers semantics the change happens when the place that gives the minimum marking into a synchronization is changed. In both situations, the switching between the linear system is given by an internal event.

Under infinite servers semantics, *nonlinearity* appears due to synchronizations ($|^\bullet t| > 1$). One linear system is defined by the set of arcs in $\boldsymbol{Pre}$ constraining the firing of the transitions.

***Definition** 2.5:* Let $\Sigma = \langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a timed contPN and $\boldsymbol{m}$ a reachable marking. It will be said that the arc (p, t) *constraints* the dynamic of t at $\boldsymbol{m}$ iff: $f[t] = \lambda[t] \cdot \frac{m[p]}{\mathrm{Pre}[p,t]}$.

***Definition** 2.6:* A *configuration* of Σ at $\boldsymbol{m}$ is a set of (p, t) arcs, one per transition, constraining the dynamic of the system.

Abusing notation, through this paper a configuration will also represent the set of places that are contained in the configuration. Each configuration corresponds to one linear system that can govern the evolution of the contPN system under infinite servers semantics.

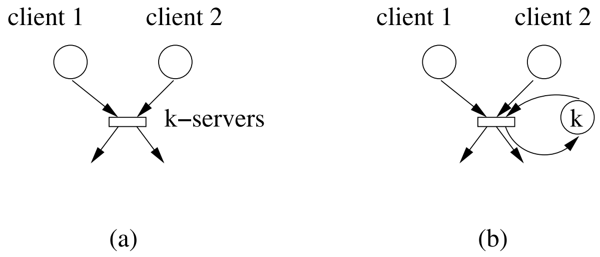
The number of configurations is given by the net structure:

$$\gamma = \prod_{t \in T} |^\bullet t|$$

We denote by *equilibrium configuration* a (possible) configuration corresponding to a (possible) steady state marking, where a steady state marking is the solution of $\dot{\boldsymbol{m}}(\tau) = 0$. If the net is consistent and all the transitions are fireable at least once then $\boldsymbol{m}$ is an equilibrium marking iff is solution of the following system [11]:

$$\begin{cases} \boldsymbol{B}^T \cdot \boldsymbol{m_0} = \boldsymbol{B}^T \cdot \boldsymbol{m} \\ \boldsymbol{C} \cdot \boldsymbol{f_{ss}} = 0 \\ \boldsymbol{f_{ss}} \geq 0 \end{cases} \tag{3}$$

where $\boldsymbol{B}^T$ is a basis of P-semiflow and $\boldsymbol{f_{ss}}$ defined as in (1).



(a)

(b)

Fig. 2. Petri net with implicit (a) and explicit (b) servers.

A crucial question is: given a PN, how to decide the better semantics to use in a given case? For example, looking at the PN system in Fig. 1, if $Buf1 = Buf2 = 5$, the reachability tree has 55125 nodes. Increasing the size of these buffers makes impractical the computation of the reachability tree. A fluidification of the model may be a solution, but which semantics approximates better the discrete PN model?

In generalized stochastic PNs a transition can model one server (*single server semantics*), ”k” servers working in parallel (*multiple servers semantics*) or an infinite number of servers (*infinite servers semantics*). To simulate ”k” servers for a transition a new place is added to the model marked with ”k” tokens, as in Fig. 2. Hence, in discrete nets, finite servers semantics can be simulated by infinite servers, just adding these places [12]. Moreover, sometimes they are even redundant, as happens for example in the net in Fig. 1. So, assuming these ”servers places” are made explicit, both semantics are equivalent. However, their continuous approximations are not equal, because finite servers semantics is hybrid at the conceptual level: considers infinite ”clients tokens” and finite number of servers. It is not immediate to decide which one is better.

Through this paper we will always assume that the places that model the servers are made explicit.

III. Monotonicity and fluidification

In this section an interesting property of contPNs is studied, namely monotonicity w.r.t. firing rate $\boldsymbol{\lambda}$. This property is checked in the case of mono-T-semiflow nets.

***Definition** 3.1:* A PN is mono-T-semiflow if it is conservative, consistent and has only one T-semiflow, $\boldsymbol{X}$, normalized for t_i (i.e. $\boldsymbol{X}[t_i] = 1$).

***Definition** 3.2:* A contPN system $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is *monotone in steady-state with respect to* $\boldsymbol{\lambda}$ if $\forall \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}$ with $\boldsymbol{\lambda_1} \leq \boldsymbol{\lambda_2}$ the steady-state throughput of the system $\langle \mathcal{N}, \boldsymbol{\lambda_1}, \boldsymbol{m_0} \rangle$ is less than or equal to the steady-state throughput of the system $\langle \mathcal{N}, \boldsymbol{\lambda_2}, \boldsymbol{m_0} \rangle$.

Under infinite servers semantics, from the flow definition in (1) it is easy to observe that if the vector $\boldsymbol{\lambda}$ is multiplied by a constant $k > 1$ then the flow will be also multiplied by k. An increase of the flow by k is obtained also when the initial marking of the net is multiplied by k. But what occurs if only some components of $\boldsymbol{\lambda}$ are increased? Monotonicity w.r.t. $\boldsymbol{\lambda}$ does not always hold, but the following can be stated:

***Theorem** 3.3:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a mono-T-semiflow contPN system under infinite servers semantics. If every possible *equilibrium configuration* contains the support of a P-semiflow then $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is monotone in steady state with respect to $\boldsymbol{\lambda}$.

Proof: Let $\boldsymbol{\lambda_1} \leq \boldsymbol{\lambda_2}$, $\boldsymbol{m_0} \geq 0$, $\boldsymbol{f_1}$ the steady-state flow for the net system $\langle \mathcal{N}, \boldsymbol{\lambda_1}, \boldsymbol{m_0} \rangle$ and $\boldsymbol{m_1}$ the corresponding steady-state marking. Let $\boldsymbol{y_1}$ be the P-semiflow included. Let $\boldsymbol{f_2}$ be the steady-state flow for the net system $\langle \mathcal{N}, \boldsymbol{\lambda_2}, \boldsymbol{m_0} \rangle$, $\boldsymbol{m_2}$ the steady-state marking and $\boldsymbol{y_2}$ the P-semiflow included.

Let us focus on $\boldsymbol{f_2}$ and $\boldsymbol{y_2}$. Every place $p_{j2} \in ||\boldsymbol{y_2}||$ restricts the flow of one output transition, denoted by t_{j2} and the flow can be written as:

$$\boldsymbol{f_2}[t_{j2}] = \boldsymbol{\lambda_2}[t_{j2}] \cdot \frac{\boldsymbol{m_2}[p_{j2}]}{\boldsymbol{Pre}[p_{j2}, t_{j2}]}$$

Using the P-semiflow $\boldsymbol{y_2}$, we can write the token conservation law for every marking taking $\boldsymbol{m_2}$ from the previous equation:

$$\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j2}] \cdot \boldsymbol{f_2}[t_{j2}]}{\boldsymbol{\lambda_2}[t_{j2}]} = \sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \boldsymbol{m_1}[p_{j2}]$$

Now, $\boldsymbol{m_1}[p_{j2}]$ is not necessarily the marking that limits the flow of the transition t_{j2} in the case of $\boldsymbol{f_1}$, but obviously it is greater than or equal to the marking that limits the transition:

$$\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \boldsymbol{m_1}[p_{j2}] \geq \sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j2}] \cdot \boldsymbol{f_1}[t_{j2}]}{\boldsymbol{\lambda_1}[t_{j2}]} \geq$$

$$\geq \sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j2}] \cdot \boldsymbol{f_1}[t_{j2}]}{\boldsymbol{\lambda_2}[t_{j2}]}$$

The net is mono-T-semiflow, so $\boldsymbol{f_1} = k_1 \cdot \boldsymbol{X}$, $\boldsymbol{f_2} = k_2 \cdot \boldsymbol{X}$ where $\boldsymbol{X} > 0$ is a T-semiflow and we obtain:

$$\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j2}] \cdot \boldsymbol{X}[t_{j2}]}{\boldsymbol{\lambda_2}[t_{j2}]} \cdot (k_2 - k_1) \geq 0 \Longrightarrow k_2 \geq k_1$$

□

Unfortunately, in general cases, the monotonicity depends on the initial marking $\boldsymbol{m_0}$.

***Example** 3.4:* Let us consider the contPN in Fig. 3. This net is mono-T-semiflow, with $\boldsymbol{X} = [1, 1, 1]$. The net has 4 configurations: $\mathcal{C}_1 = \{p_1, p_2, p_3\}$, $\mathcal{C}_2 = \{p_1, p_3, p_4\}$, $\mathcal{C}_3 = \{p_2, p_3, p_4\}$ and $\mathcal{C}_4 = \{p_3, p_4\}$. The P-semiflows are: $\boldsymbol{y_1} = p_1 + p_2 + p_3$ and $\boldsymbol{y_2} = p_1 + 4 \cdot p_3 + p_4$, therefore we have two configurations that contain a P-semiflow ($\mathcal{C}_1$ and $\mathcal{C}_2$), one configuration that contains a P-flow ($\mathcal{C}_3$) (but no P-semiflow) $\boldsymbol{y_3} = \boldsymbol{y_1} - \boldsymbol{y_2}$ and one configuration that does not contain any P-flow ($\mathcal{C}_4$).

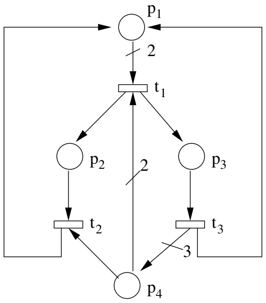


Fig. 3. Mono-T-semiflow net used in Example 3.4.

1) Let $\boldsymbol{m_0} = [1, 1, 0, 15]$. For this marking the configuration $\mathcal{C}_4$ cannot be an equilibrium one. If it were, p_4 should limit the flow of t_1 and t_2. Since the net has one T-semiflow ([1,1,1]) in steady state $f_1 = f_2$, and taking into account the weights ($f_1 = \frac{m_4}{2} = f_2 = m_4$), $m_4 = 0$. Steady state flow satisfies $f_1 = f_2 = f_3$ that implies $m_3 = 0$. Considering the second P-semiflow $m_1 = 16$, which is impossible because using the first P-semiflow m_2 and/or m_3 should be negative.
   Assume now that $\mathcal{C}_3$ is the equilibrium configuration. From the first P-semiflow: $m_1 + m_2 + m_3 = 2 \Longrightarrow m_1 \leq 2; m_2 \leq 2; m_3 \leq 2$ but $m_1 \geq m_4$ (the flow of t_1 is restricted by p_4) then $m_4 \leq 2$. Using the second P-semiflow we obtain $m_1 \geq 6$ which is impossible. Thus for $\boldsymbol{m_0}$ the only possible equilibrium configurations are $\mathcal{C}_1$ and $\mathcal{C}_2$. Both contain P-semiflows and using Theorem 3.3 the contPN system with this initial marking is monotone with respect to $\boldsymbol{\lambda}$.
2) Let $\boldsymbol{m_0} = [15, 1, 1, 0]$. $\mathcal{C}_4$ cannot be an equilibrium configuration because in that case $m_3 = m_4 = 0$, and then $m_1 = 19$ (from the second P-semiflow). From the first P-semiflow m_2 should be negative ($m_2 = \boldsymbol{y_1} - m_1 - m_3 = 17 - 19 - 0 = -2$).
   The configuration containing a P-flow ($\mathcal{C}_3$) can be equilibrium configuration, but non-existing a P-semiflow gives that the net system is not be monotone. For example, if $\boldsymbol{\lambda_1} = [1, 1, 1]^T$, the steady state marking is: $\boldsymbol{m_1} = [16, 0.5, 0.5, 1]$ and the flow: $\boldsymbol{f_1} = [0.5, 0.5, 0.5]$. Putting $\boldsymbol{\lambda_2} = [1, 2, 1]^T \geq \boldsymbol{\lambda_1}$, the steady-state marking is: $\boldsymbol{m_2} = [16.33, 0.22, 0.44, 0.88]$ and the flow: $\boldsymbol{f_2} = [0.44, 0.44, 0.44]$. Thus the net system is not monotone w.r.t. $\boldsymbol{\lambda}$.
3) Let $\boldsymbol{m_0} = [1, 15, 1, 0]$. For $\boldsymbol{\lambda_1} = [2, 1, 1]^T$, $\mathcal{C}_2$ is the equilibrium configuration: $\boldsymbol{m_1} = [1.25, 15, 0.75, 0.75]$ and $\boldsymbol{f_1} = [0.75, 0.75, 0.75]$. Making $\boldsymbol{\lambda_2} = [2, 2, 1]^T$, $\mathcal{C}_4$ is the corresponding equilibrium configuration with $\boldsymbol{m_2} = [5, 12, 0, 0]$ and $\boldsymbol{f_2} = [0, 0, 0]$. So, the contPN system is not monotone w.r.t. $\boldsymbol{\lambda}$. In this case a deadlock is obtained.

According to Theorem 3.3, independently of the initial marking, if all configurations contain a P-semiflow, then the underlying net system is monotone w.r.t. $\boldsymbol{\lambda}$ in steady state $\forall \boldsymbol{m_0}$. Moreover, we will prove that this P-semiflow condition can be relaxed to P-flow (Corollary 3.7). Let us first consider the following Lemma and Proposition.

***Lemma** 3.5:* Let $\mathcal{N}$ be a consistent JF net. For every P-flow $\boldsymbol{y}$ of $\mathcal{N}$ there exist some P-semiflow $\boldsymbol{y'}$ such that $||\boldsymbol{y}'|| \subseteq ||\boldsymbol{y}||$.

Proof: Dual of Theorem 9 of [7] (T-flows of CF nets). □

***Theorem** 3.6:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a mono-T-semiflow contPN system under infinite servers semantics. If $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is not monotone in steady state w.r.t. $\boldsymbol{\lambda}$, then there exists a configuration that does not contain any P-flow.

Proof: If $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is not monotone, applying Theorem 3.3, an equilibrium configuration exists that does not contain a P-semiflow. Assume that this equilibrium configuration contains one P-flow (or more) and let us consider the subnet $\mathcal{N}'$ formed by the set of transitions together with the places that limit their flow in steady state. Let us call $\boldsymbol{C'}$ the token flow matrix of this subnet. Since the original net is mono-T-semiflow, then $\mathcal{N}'$ is consistent being a configuration that is obtained by removing the places that do not restrict the flow of any transition.

Using the previous Lemma, $\mathcal{N}'$ has a join. Let us call t_k this transition. To simplify, let us assume p_i and p_j are the input places of t_k that belong to the configuration. Obviously, only one place restricts the flow of t_k. Let us assume p_i to be this one; thus p_j restricts the flow of another transition.

If we consider now that t_k is restricted by p_j and the other transitions of the subnet by the same places, we obtain another configuration (possible a non-equilibrium one) in which place p_i was removed. This reasoning can be repeated with all the P-flows. Clearly, this new configuration does not contain any P-semiflow (from hypothesis) or P-flow. □

Therefore:

***Corollary** 3.7:* Let $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ be a continuous mono-T-semiflow net under infinite servers semantics. If all configurations contain a P-flow, then $\forall \boldsymbol{m_0}$ the underlying net system is monotone w.r.t. $\boldsymbol{\lambda}$ in steady state.

An algorithm to compute all configurations that do not contain a P-semiflow can be stated. We will give here the schema, and a more efficient algorithm will be provided in a future work. The idea here is to use boolean equations in order to find configurations (covers of the transitions by arcs connected to its input places) that do not contain a P-semiflow. Any P-semiflow will provide a boolean equation. Moreover, if a transition is not a join ($|{}^\bullet t_i| = 1$) its input place must belong to all configurations (is an essential cover). Therefore, for every synchronization we should have another boolean equation ensuring that at least one input place is taken.

Let γ_i be a boolean variable: $\gamma_i = 1$ iff p_i belongs to one solution. Let us consider the net in Fig. 3. It has two P-semiflow: $\boldsymbol{y_1} = p_1 + p_2 + p_3$ and $\boldsymbol{y_2} = p_1 + 4 \cdot p_3 + p_4$, so two boolean equations negating them in the set of solutions are obtained: $\gamma_1 \cdot \gamma_2 \cdot \gamma_3 = 0$ and $\gamma_1 \cdot \gamma_3 \cdot \gamma_4 = 0$. In order to cover transitions we need additional equations. For t_1: $\gamma_1 + \gamma_4 = 1$ and the same for t_2: $\gamma_2 + \gamma_4 = 1$. Clearly, p_3 being the only input place in t_3 is an essential cover thus $\gamma_3 = 1$. The following system of boolean equations is obtained:

$$\begin{cases} \gamma_1 \cdot \gamma_2 \cdot \gamma_3 = 0 & (a) \\ \gamma_1 \cdot \gamma_3 \cdot \gamma_4 = 0 & (b) \\ \gamma_1 + \gamma_4 = 1 & (c) \\ \gamma_2 + \gamma_4 = 1 & (d) \\ \gamma_3 = 1 & (e) \end{cases} \tag{4}$$

From (4.e), taking into account (4.a) and (4.b), $\gamma_1 \cdot (\gamma_2 + \gamma_4) = 0$. Using (4.d): $\gamma_1 = 0$, thus $\gamma_2 = \oslash$ and $\gamma_4 = 1$. In summary: $\gamma_1 = 0$, $\gamma_2 = \oslash$, $\gamma_3 = 1$ and $\gamma_4 = 1$ tell us that the net has two configurations ($\mathcal{C}_1 = \{p_3, p_4\}$, $\mathcal{C}_2 = \{p_2, p_3, p_4\}$) that do not contain the support of a P-semiflow. Depending on the initial marking these configurations can or cannot be equilibrium configurations and the monotonicity may be lost (see Example 3.4).

IV. Comparison of servers semantics for contPN

In this section we will compare the throughput of a continuous net system under infinite and finite servers semantics for the subclass of mono-T-semiflow nets. As we said, the servers should be made explicit because otherwise the comparison is inappropriate.

In [5] a branch and bound algorithm is provided to compute upper and lower bounds of the steady-state throughput. Nevertheless, let us consider a more relaxed computation:

$$\gamma = \max \{ \boldsymbol{y} \cdot \boldsymbol{PD} | \boldsymbol{y} \cdot \boldsymbol{C} = 0, \boldsymbol{y} \cdot \boldsymbol{m_0} = 1, \boldsymbol{y} \geq 0 \} \tag{5}$$

where $\boldsymbol{PD}(p) = \max_{t \in p^\bullet} \frac{\boldsymbol{Pre}(p,t) \cdot \boldsymbol{X}(t)}{\boldsymbol{\lambda}(t)}$ and $\boldsymbol{X}$ the T-semiflow normalized for a transition t_i.

Let $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ be a mono-T-semiflow net and γ the solution of LPP (5). According to [5] and [13] the throughput in steady state verifies $\boldsymbol{f}_{ss} \leq \frac{1}{\gamma} \boldsymbol{X}$ for continuous infinite servers semantics and discrete system, respectively. Moreover, this bound is reached in the continuous system iff the steady-state configuration contains the support of a P-semiflow [5]. Therefore:

***Proposition** 4.1:* [5] Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a mono-T-semiflow net system. Discrete throughput is less than or equal to the continuous infinite servers semantics throughput if all steady-state configurations contains the support of a P-semiflow.

Hence, if the steady state configuration contains the support of a P-semiflow, the upper bound of the contPN system under infinite servers semantics is equal to the throughput of the system and is superior to the throughput of the discrete net. Moreover, if the net system is live, the throughput of the contPN system under finite servers semantics is greater than or equal to the throughput under infinite servers semantics. This permits to order these values.

***Proposition** 4.2:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a live mono-T-semiflow contPN system. The throughput in steady-state under finite servers semantics is greater than or equal to the throughput under infinite servers semantics.

Proof: The net is mono-T-semiflow so the throughput in steady-state should be $\alpha \cdot \boldsymbol{X}_i$, where $\boldsymbol{X}_i$ is the T-semiflow normalized for t_i. Under finite servers semantics at least one transition should be strongly enabled in steady state (otherwise, the net is not live) and the throughput will be its maximal firing speed. Therefore, this α is maximal and under infinite servers semantics the throughput will be less than or equal to this value. □

The liveness analysis of autonomous and timed continuous systems is studied in [14] for the subclass of mono-T-semiflow nets. According to Proposition 4.1 and Proposition 4.2, the following can be stated:

***Theorem** 4.3:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a live mono-T-semiflow Petri net system with all equilibrium configuration containing a P-semiflow. The continuous model under infinite servers semantics provides a better approximation of any T-timed interpreted discrete model than the continuous model under finite servers semantics.

In general, the P-semiflow included in the configurations are difficult to study since the number of configurations may be very large. However, there are net subclasses for which it is immediate. For example, for structurally live and bounded EQ nets (thus conservative, consistent and the rank of the token flow matrix is equal to the number of conflicts minus one [6]). In fact, they contain a P-semiflow in every configuration [6] and can be transformed into equivalent CF nets [5]. Therefore the conditions of Theorem 4.3 are satisfied.

***Corollary** 4.4:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a structurally live and structurally bounded EQ net system. Infinite servers semantics provides a better continuous approximation of the discrete system throughput.

More general net classes exist for which this result holds too. For example, it holds for the non mono-T-semiflow net in Fig. 1. Nevertheless, there exist cases in which the fluidified model under finite servers semantics presents a smaller deviation from the discrete throughput. Even more, there are net systems such that the continuous throughput is not an upper bound of the discrete model.

***Example** 4.5:* Let us return to the net in Fig. 3 with $\boldsymbol{m_0} = [15, 1, 1, 0]$. We saw before that for the net system $\mathcal{C}_3 = \{p_2, p_3, p_4\}$ is an equilibrium configuration that does not contain any P-semiflow. Consider this model as finite servers semantics with 2 servers in each transition and maximal firing speed $\boldsymbol{\lambda} = [4, 2, 2]$. The steady state flow will be equal to 2, which is also the (upper) bound [15]. Under infinite servers semantics, $\boldsymbol{\lambda} = [2, 1, 1]$ and places modeling the servers should be added. The steady-state flow obtained by simulation is 0.75 and the bounds (lower and upper) are 0.67 and 2. As discrete, the steady-state throughput is 1.55 and clearly finite servers semantics is better in this case.

In some cases (mono-T-semiflow reducible nets [5]), timed continuous nets can be transformed into mono-T-semiflow. For T-semiflow reducible nets, the visit ratio does not depend on the initial marking, but is defined by the net structure and the rates associated to transitions. Therefore, the results presented here hold for the more general class of mono-T-semiflow reducible nets.

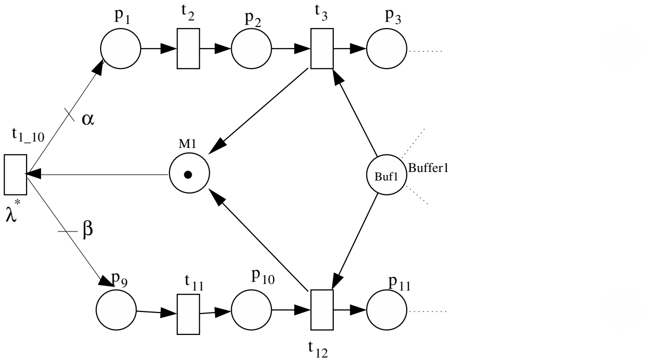


Fig. 4. Equivalent mono-T-semiflow net of the contPN in Fig. 1 ($\lambda^* = \lambda_1 + \lambda_{10}, \alpha = \frac{\lambda_1}{\lambda_1+\lambda_{10}}, \beta = \frac{\lambda_{10}}{\lambda_1+\lambda_{10}}$).

V. Case study: a manufacturing system

We have seen that in general it is not possible to give an answer regarding the continuous servers semantics which approximates better the discrete behavior. Performance of a flexible manufacturing system is compared here.

Let us consider the production system presented in Fig. 1 that has the following P-semiflows: $\boldsymbol{y_1} = p_1 + p_2 + M_1 + p_9 + p_{10}$ (corresponds to the state of machine M_1), $\boldsymbol{y_2} = p_3 + Buffer_1 + p_{11}$ (Buffer 1), $\boldsymbol{y_3} = p_4 + p_5 + M_2 + p_{12} + p_{13}$ (machine M_2), $\boldsymbol{y_4} = p_6 + Buffer_2 + p_{14}$ (Buffer 2) and $\boldsymbol{y_5} = p_7 + p_8 + M_3 + p_{15} + p_{16}$ (machine M_3). We have computed simulations for the continuous model under infinite and finite (single-server) servers semantics using $\boldsymbol{\lambda} = \boldsymbol{1}$, $Buf1 = Buf2 = 10$ and we compare it with the results obtained in the case of discrete net system. This model is not mono-T-semiflow but can be transformed into an equivalent mono-T-semiflow net (see Fig. 4) without changing the loading/scheduling policy for the underlying manufacturing system. In fact it has two T-semiflows that correspond to two production lines and for the particular value of $\lambda_1 = \lambda_{10}$, the "global" T-semiflow of net in Fig. 1 will be $\boldsymbol{X} = \boldsymbol{1}$. Therefore, in steady state all the transitions will run at the same speed.

Measuring the flow of transition T_7 we have obtained the following results: $Th(T_7) = 0.186$ for the discrete model, $Th(T_7) = 0.2$ for the continuous model under infinite servers semantics and $Th(T_7) = 1$ for the continuous model under finite servers semantics. The results are showing clearly that continuous infinite servers semantics fits much better with the discrete results.

This net is mono-T-semiflow reducible and (as it will be seen afterwards) the equilibrium configuration contains a P-semiflow. Therefore, the throughput of the system will be given by the slowest P-semiflow. Moreover the net system is live and in that case the exact throughput of the continuous system under infinite server semantics can be computed in polynomial time (is equal with the LPP bound (Eq. (5))) [5]. Hence, infinite servers semantics is a better approximation than finite servers semantics which is too optimistic in this case.

In this example, every configuration contains a P-semiflow. This can be checked writing the boolean system of equations as explained before for the net in Fig 3 and verifying the absence of solution. Let us try to find a configuration that does not contain the support of a P-semiflow. Starting with the P-semiflow corresponding to M_3 (i.e. $\boldsymbol{y_5} = p_7 + p_8 + M_3 + p_{15} + p_{16}$), the places p_7, p_8, p_{15}, p_{16} are essential covers (because have only one output transition) so should belong to the configuration. In order to not include this P-semiflow, the place M_3 should be not taken, forcing us to include p_6 and p_{14} in order to restrict the flow of t_7 and t_{16}. Now, taking into account the P-semiflow $\boldsymbol{y_4} = p_6 + Buffer_2 + p_{14}$, $Buffer_2$ cannot be taken, so p_5 and p_{13} are needed to limit the flow of t_6 and t_{15} respectively. p_4 and p_{12} are essential covers and will be taken. Observing $\boldsymbol{y_3} = p_4 + p_5 + M_2 + p_{12} + p_{13}$, M_2 cannot be taken, and so p_3 and p_{11} have to be in the configuration. Watching to $\boldsymbol{y_2} = p_3 + Buffer_1 + p_{11}$, place $Buffer_1$ will not belong to the configuration. But then, $p_2, p_1, M_1, p_{10}, p_9$ have to be in the configuration. However, all these places are the support of a P-semiflow (the one corresponding to the state of the first machine), $\boldsymbol{y_1} = p_1 + p_2 + M_1 + p_9 + p_{10}$. This means that no configuration without P-semiflow exists.

VI. Conclusions

Mono-T-semiflow contPN systems are under discussion in this paper. First, the monotonicity with respect to firing rate $\boldsymbol{\lambda}$ is studied. We prove that if all possible equilibrium configurations contain the support of a P-semiflow then the net exhibits this monotonicity. In the second part, a comparison between the two most used firing semantics for timed contPN system is provided trying to see which one is better. A ”good” firing semantics for timed continuous Petri nets should provide a time evolution ”similar” to the discrete model. Being a relaxation, an identical result is practically impossible to obtain. In this paper most used semantics in the literature, namely *infinite* and *finite servers semantics* are compared. For mono T-semiflow reducible contPN systems with all equilibrium configuration containing the support of a P-semiflow we have shown in Section III that infinite servers semantics is always a better approximation. Nevertheless, in the actual level of knowledge, in general it is difficult to answer to this question. In any case, studying many systems it appears in practice that infinite servers semantics is usually superior to finite servers semantics. To illustrate this, a real system modeled with PNs is presented. An interpretation of this result for these particular cases has been done. Finally, a preliminary method for computing all configurations not containing any P-semiflow based on boolean equations have been presented. Even so, this method will be generalized and improved from a computational point of view.

Acknowledgement

We want to thank the referees of this paper for their useful suggestions.

References

[1] F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva, and F. B. Vernadat, *Practice of Petri Nets in Manufacturing*. Chapman & Hall, 1993.

[2] R. David and H. Alla, *Discrete, Continuous and Hybrid Petri Nets*. Springer-Verlag, 2004.

[3] M. Silva and L. Recalde, “Petri nets and integrality relaxations: A view of continuous Petri nets,” *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 32, no. 4, pp. 314–327, 2002.

[4] H. Alla and R. David, “Continuous and hybrid Petri nets,” *Journal of Circuits, Systems, and Computers*, vol. 8, no. 1, pp. 159–188, 1998.

[5] J. Júlvez, L. Recalde, and M. Silva, “Steady-state performance evaluation of continuous mono-T-semiflow Petri nets,” *Automatica*, vol. 41, no. 4, pp. 605–616, 2005.

[6] E. Teruel and M. Silva, “Structure theory of equal conflict systems,” *Theoretical Computer Science*, vol. 153, no. 1-2, pp. 271–300, 1996.

[7] E. Teruel, J. M. Colom, and M. Silva, “Choice-free Petri nets: A model for deterministic concurrent systems with bulk services and arrivals,” *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 27, no. 1, pp. 73–83, 1997.

[8] F. Commoner, A. W. Holt, S. Even, and A. Pnueli, “Marked directed graphs,” *Journal on Computer Systems Science*, vol. 5, pp. 72–79, 1971.

[9] M. Silva and L. Recalde, “On fluidification of Petri net models: from discrete to hybrid and continuous models,” *Annual Reviews in Control*, vol. 28, no. 2, pp. 253–266, 2004.

[10] M. Silva, E. Teruel, and J. Colom, “Linear algebraic and linear programming techniques for the analysis of P/T net systems,” *LCNS*, vol. 1, no. 1941, pp. 309–373, 1998.

[11] C. Mahulea, A. Ramírez, L. Recalde, and M. Silva, “Steady state control, zero valued poles and token conservation laws in continuous net systems,” in *Workshop on Control of Hybrid and Discrete Event Systems*. Miami, USA: J.M. Colom, S. Sreenivas and T. Ushio, eds., June 2005.

[12] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1995.

[13] J. Campos and M. Silva, “Structural techniques and performance bounds of stochastic Petri net models,” in *Advances in Petri Nets 1992*, ser. Lecture Notes in Computer Science, G. Rozenberg, Ed. Springer, 1992, vol. 609, pp. 352–391.

[14] J. Júlvez, L. Recalde, and M. Silva, “On deadlock-freeness analysis of autonomous and timed continuous mono-T-semiflow nets,” in *Proceedings of the* 41^{st} *IEEE Conference on Decission and Control (CDC 2002)*, Las Vegas, USA, December 2002, pp. 781–786.

[15] L. Recalde and M. Silva, “Petri Nets fluidification revisited: Semantics and steady state,” *APII-JESA*, vol. 35, no. 4, pp. 435–449, 2001.