15th Triennial World Congress, Barcelona, Spain

OPTIMAL ROUTING OF CONTINUOUS TIMED PETRI NETS

Bruno Gaujal [*] **Alessandro Giua** [**]

[*] *INRIA Lorraine, 54000 Nancy, France. Email: gaujal@loria.fr*
[**] *Dept. of Electrical and Electronic Engineering, University of Cagliari, Italy. Email: giua@diee.unica.it*

Abstract: In this paper, we consider deterministic timed continuous Petri nets where conflicts at places are solved by assigning stationary routing parameters. We show how to compute the stationary firing rate at all transitions as well as how to determine the optimal routing parameters to maximize the firing rates. Finally, we discuss the relations with discrete Petri nets.

Keywords: Continuous Petri nets, optimal routing, linear programming

1. INTRODUCTION

This paper deals with the problem of determining an optimal periodic mode of operation for a system described by a continuous timed Petri net.

The model we consider, is taken from a paper by Cohen *et al.* (1998). It consists of a place-transition net where the marking and the arc weight are not restricted to integer values but may take non negative real values, i.e., we have fluid quantities moving from place to place rather than discrete tokens. A release delay τ_i is associated to each place p_i, representing the time that the fluid that enters will spend in the place before it can be available to fire the output transitions. On the contrary no delay is associated to the transitions. (Note that an equivalent model could be given associating a release delay to each transition.)

Figure 1 displays a very simple continuous timed Petri net with a delay δ associated with the place p and an initial marking equal to one: $M_p(0) = 1$. The evolution of the marking given by the dashed line is arbitrary: the model just requires that the mark remains in p for at least a time interval of length δ, the marking in the place is null $M_p(\delta) = 0$. Nothing is said about the evolution of the marking before time δ. The evolution equations in this model only provide the marking at discrete times as "total equations" with no specification on how the marking actually evolves between the sampling times. Also note that the model assumes the infinite server semantics in the places where the delay is the same no matter how many tokens are actually present in the place. To model finite server semantics, one needs to add constraints in the net by allowing a maximal quantity of token in a place (for example by recycling the places with new places containing an adequate number

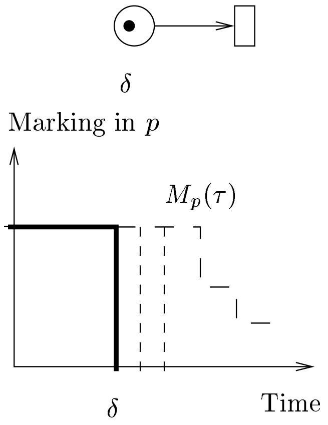

Fig. 1. Evolution of the marking in continuous Petri net

[1] This work was partially supported by the action GARP in the theme QSL of the contrat de plan état-région Lorraine

of tokens).

A rather different continuous Petri net model has been presented by Recalde and Silva (2001).

The timing of the net is given under the form of "speed parameters" in all the transitions. Once again, the infinite server semantics is assumed, which is modeled by an evolution equation of the marking under the form of a differential equation, where the decrease rate of the marking depends on its size.

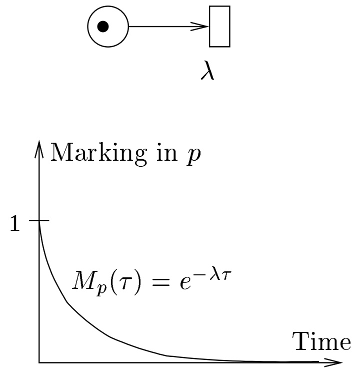

Fig. 2. Marking evolution in the model of Recalde and Silva

In the example of Figure 2, where the transition has a speed parameter λ, then the marking evolution is given by the equation $\frac{dM_p(t)}{dt} = -\lambda$ with initial condition $M_p(0) = 1$ which solution is $M_p(t) = exp(-\lambda t)$. Note that unlike in the Cohen *et al.* model, it takes an infinite amount of time to empty place p.

In both previous papers, an additional assumption (that we call STAR) was that of solving conflicts at places by assigning a stationary routing.

Finally, note that Recalde and Silva made two additional assumptions (equal conflict nets and conflicts only at immediate transitions) that are not necessary in the continuous case.

In the STAR case, the evolution equations of the model by Cohen *et al.* (assuming that each transition fires as soon as it is enabled) can be written as the coupling of a conventional linear system with a (min,×) linear system, i.e., it is very easy to implement. The evolution equations of the model of Recalde and Silva, on the contrary, can be written as the coupling of a conventional system of linear differential equations with a (min,×) linear system, and are more difficult to solve.

The two models, as we have seen from the previous examples, have a very different transient behavior. However, under the hypothesis that the net is live and bounded, both models reach the same steady state, in the sense that the average firing rate of each transition reaches the same constant value (see section 3). In this case the model by Recalde and Silva has been shown to lead to an easy computation of the steady state by means of a linear program, while Cohen *et al.* require a more complex procedure based on dynamic programming for their model.

Here, we show that the model of Cohen *et al.* can also be solved by a linear program (slightly different from the program of Recalde and Silva). Our approach can also be applied to nets that are not live or bounded but that admit one or more consistent firing vector, i.e., it can also be used to determine an optimal steady state firing vector among several. We show that an optimal steady state firing vector always corresponds to a particular STAR, but in general it is not reached from all initial markings using the same STAR during the transient. The optimal transient, i.e., the transient that leads as soon as possible to the optimal steady state can also be easily computed using our approach.

2. BACKGROUND

A *continuous Petri net* (CPN) is a structure $N = (P, T, \boldsymbol{Pre}, \boldsymbol{Post})$, where P is a set of m places; T is a set of n transitions; $\boldsymbol{Pre} : P \times T \to \mathbb{R}_0^+$ and $\boldsymbol{Post} : P \times T \to \mathbb{R}_0^+$ are the *pre*– and *post*–incidence functions that specify the weighted arcs; $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the incidence matrix. The preset and postset of a node $x \in P \cup T$ are denoted ${}^\bullet x$ and $x^\bullet$, respectively. A *marking* is a vector $\boldsymbol{M} : P \to \mathbb{R}_0^+$ that assigns to each place a non–negative fluid quantity called *mark*; we denote the marking of place p_i as M_i and the marking at time τ as $\boldsymbol{M}(\tau)$. A *CPN system* $\langle N, \boldsymbol{M}(0)\rangle$ is a net N with an initial marking $\boldsymbol{M}(0)$.

Given a CPN N with incidence matrix $\boldsymbol{C}$: a P-semiflow is a vector $\boldsymbol{x} \in \mathbb{R}^m$ such that $\boldsymbol{x} \gneq \boldsymbol{0}$ and $\boldsymbol{x}^T\boldsymbol{C} = \boldsymbol{0}$; a T-semiflow is a vector $\boldsymbol{y} \in \mathbb{R}^n$ such that $\boldsymbol{y} \gneq \boldsymbol{0}$ and $\boldsymbol{C}\boldsymbol{y} = \boldsymbol{0}$. The support of a P-semiflow $\boldsymbol{x}$ (resp., T-semiflow $\boldsymbol{y}$) is the set $||\boldsymbol{x}|| = \{p_i \in P \mid x_i > 0\}$ (resp., $||\boldsymbol{y}|| = \{t_j \in T \mid y_j > 0\}$.

A deterministic timed CPN (DTCPN) is a pair (N, δ), where N is a CPN, and $\delta : P \to \mathbb{R}_0^+$, called release delay, assigns a non–negative fixed duration to each place. The mark that enters place p_i at time τ must remain within the place for a time greater or equal to δ_i before it becomes available for firing an output transition. A place with a release delay equal to 0 is said to be immediate and we assume there are no cycles in the net whose places are all immediate. The mark $M_i(0)$ assigned by the initial marking, is assumed to enter place p_i at time $\tau = 0$, hence it will be available at time $\tau = \delta_i$.

A *stationary routing* (STAR) is a function $\rho : P \times T \longrightarrow [0, 1]$ such that $\boldsymbol{Pre}(p, t) = 0 \Longrightarrow \rho(p, t) =$

0 (i.e., $\rho(p,t) > 0$ only if $t \in p^\bullet$) and such that for all $p \in P$: $\sum_{t \in T} \rho(p,t) = 1$. If $\rho(p_i, t_j) = \alpha_{i,j} > 0$, then a fraction $\alpha_{i,j}$ of the mark entering place p_i is reserved only for the firing of its output transition t_j and cannot be "used" to enable and fire any other output transition. Thus, a STAR solves all conflicts that may arise in the net with a preassigned fixed policy.

If we assign to each transition t_j a counter $Z_j(\tau)$ that represents the cumulative firing quantity of the transition from time 0 to τ, and if we assign to each place p_i a counter $H_i(\tau)$ that represents the fluid quantity that has entered the place from time 0 to τ (including the initial marking), the evolution of such a model can be written as:

$$
\begin{aligned}
Z_j(\tau) &\leq \min_{p_i \in {}^\bullet t_j} \left\{ \frac{\alpha_{i,j} H_i(\tau - \delta_i)}{Pre(p_i, t_j)} \right\} && \forall t_j \in T \\
H_i(\tau) &= M_i(0) + \sum_{t_j \in {}^\bullet p_i} Post(p_i, t_j) Z_j(\tau) && \forall p_i \in P
\end{aligned}
\tag{1}
$$

The evolution that corresponds to choosing the equality in the first equation in (1) will be called ASAP (as soon as possible) because it corresponds to firing a transition as soon as the fluid that has entered the input places becomes available for the firing.

Note that said $\boldsymbol{Z} = [Z_1 \cdots Z_n]^T$, the marking at time τ can be determined by the state equation

$$\boldsymbol{M}(\tau) = \boldsymbol{M}(0) + \boldsymbol{C}\boldsymbol{Z}(\tau).$$

It is important to recall the following result. The existence of a vector $\boldsymbol{Z}(\tau)$ satisfying the state equation is a necessary but not sufficient condition for the reachability of marking $M(\tau)$ in a discrete net. However, in the case of continuous nets this condition is necessary and sufficient provided that *all transitions may be fired from the initial marking* as shown by Recalde *et al.* (1999).

3. STEADY STATE COMPUTATION WITH A UNIQUE T-SEMIFLOW

We first note that a DTCPN (N, δ) whose conflicts are solved using a STAR ρ can be transformed into an equivalent *structurally conflict-free* net (N', δ'), i.e. a net in which each place has at most one output transition (Campos *et al.*, 1991). Each place p_i in the original net is split into h places $p_i^1, \cdots, p_i^h$ (where h is the size of $p^\bullet$). The new preset and postset are such that $\boldsymbol{Pre'}(p_j^k, t_i) = \boldsymbol{Pre}(p_j, t_i) \times \rho(p_i, t_k)$, $\boldsymbol{Post'}(p_j^k, t_k) = \boldsymbol{Post}(p_j, t_k)$ and $\boldsymbol{Post'}(p_j^k, t_i) = 0$ if $i \neq k$. An example of such a construction is shown in Figure 3.

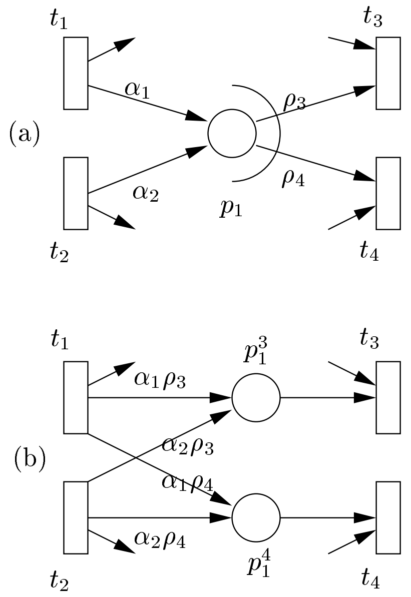

Fig. 3. (a) A place with conflict; (b) transformation to solve the conflict using a STAR.

Let us now consider a structurally conflict-free net. In the case of discrete nets, Campos *et al.* (1991) have shown that if such a net is live and bounded, then it admits a single T-semiflow whose support contains all the transitions (in the case of discrete PN a T-semiflow is a non-negative integer vector). We prove a slightly different result.

Proposition 1. A CPN N that is structurally conflict-free and strongly connected admits at most one T-semiflow (up to a multiplicative constant). Moreover, its support contains all the transitions.

Proof We first show that if a CPN admits two different T-semiflows $\boldsymbol{y}$ and $\boldsymbol{y'}$, with $\boldsymbol{y} \neq \beta \boldsymbol{y'}$ for all $\beta \in \mathbb{R}^+$, then it also admits a T-semiflow $\tilde{\boldsymbol{y}}$ with at least a null component. This is clearly true if either $\boldsymbol{y}$ or $\boldsymbol{y'}$ admits a null component. Assume now $\boldsymbol{y}, \boldsymbol{y'} > \boldsymbol{0}$. Then let $\beta = \min_i \{y_i / y'_i\} = y_l / y'_l$: we have that $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \beta \boldsymbol{y'}$ is a T-semiflow whose $l-$th component is null.

Assume now a CPN admits a T-semiflow $\tilde{\boldsymbol{y}}$ whose $l-$th component is null. If the net is conflict free, for each place $p \in {}^\bullet t_l$: $C(p, t_l)$ is the only negative element of $\boldsymbol{C}(p, \cdot)$. Hence $\boldsymbol{C}(p, \cdot)\tilde{\boldsymbol{y}} = 0$ only if all transitions in ${}^\bullet p$ do not belong to the support of $\tilde{\boldsymbol{y}}$. Since the net is strongly connected, ${}^\bullet p$ is not empty and we can repeat this reasoning until we find that $\tilde{\boldsymbol{y}} = \boldsymbol{0}$, clearly a contradiction. □

Based on the following result we now present a linear program (based on Little's law) to compute the average steady state transition firing rates of the model by Cohen *et al.* This firing rate of a given transition t_j is defined by the following limit, which exists as soon as the net is live and bounded, or similarly, as soon as it admits a positive T-

invariant: $y_j = \lim_{\tau \to \infty} Z_j(\tau)/\tau$. See (Cohen *et al.*, 1998) for a proof.

Theorem 2. Given a live and bounded DTCPN $\langle N, \delta, \boldsymbol{M}(0)\rangle$ with incidence matrix $\boldsymbol{C}$, we define the following linear programming problem (LPP) with constraint set $\mathcal{L}(N, \boldsymbol{M}(0))$:

$$\max \ \mathbf{1}^T \boldsymbol{y}$$
$$s.t. \ \begin{cases} \boldsymbol{C}\boldsymbol{y} = \mathbf{0} \\ M_i \geq \boldsymbol{Pre}(p_i, \cdot)\boldsymbol{y}\delta_i \ \text{ for all } p_i \in P \\ \boldsymbol{M} = \boldsymbol{M}(0) + \boldsymbol{C}\boldsymbol{z}, \end{cases} \tag{2}$$

where the variables are $\boldsymbol{y} \in (\mathbb{R}_0^+)^n$, $\boldsymbol{M} \in (\mathbb{R}_0^+)^m$, and $\boldsymbol{z} \in (\mathbb{R}_0^+)^n$.

A solution $(\boldsymbol{y}, \boldsymbol{M}, \boldsymbol{z})$ of this LP is such that $\boldsymbol{y}$ is a T-semiflow of the net, which gives the asymptotic firing rate in all the transitions and $\boldsymbol{z}$ is the transient firing vector that leads from the initial marking to a steady state.

Proof. The proof goes along the following lines. First, note that the first constraint $\boldsymbol{C}\boldsymbol{y} = \mathbf{0}$ is a necessary condition since the net is assumed to reach a stationary marking.

The constraint $M_i \geq \boldsymbol{Pre}(p_i, \cdot)\boldsymbol{y}\delta_i$ for all $p_i \in P$ is Little's law for the transition following place p_i which is true once the stationary regime is reached.

As for the last constraint, $\boldsymbol{M} = \boldsymbol{M}(0) + \boldsymbol{C}\boldsymbol{z}$, it says that the stationary marking must be reachable from the initial marking. This is a relaxation on the reachability using the ASAP firing policy, however, we will show in the remaining of the proof that the earliest firing policy is optimal for this linear program.

Note that any firing vector $\boldsymbol{z}$ can be reached using the evolution given in Equations (1) relaxing the earliest firing semantics.

Second, note that the asymptotic firing rate of a transition j is defined as

$$y_j = lim_{\tau \to \infty} Z_j(\tau)/\tau.$$

Using the fact that Equation (1) is obviously non-decreasing in $\{Z_i, \tau\}_{t_i \in T, \tau \in \mathbb{R}}$, then the limit y_j is maximal when all the firings occur as soon as possible.

Therefore, there exists an optimal solution to the LPP with a firing vector $\boldsymbol{z}$ corresponding to the earliest firing policy. □

This theorem calls for several comments.

1. Proposition 1 says that the net admits a single T-semiflow. Therefore, the vector $\boldsymbol{y}$ could be written under the form $\boldsymbol{y} = \phi \boldsymbol{v}$ where $\boldsymbol{v}$ is the unique normalized T-semiflow of the net. The linear program is modified in such a way that the vector variable $\boldsymbol{y}$ is replaced by the scalar ϕ.

$$\max \ \phi$$
$$s.t. \ \begin{cases} M_i \geq \boldsymbol{Pre}(p_i, \cdot)\boldsymbol{\phi}\boldsymbol{v}\delta_i \ \text{ for all } p_i \in P \\ \boldsymbol{M} = \boldsymbol{M}(0) + \boldsymbol{C}\boldsymbol{z}, \end{cases} \tag{3}$$

where the variables are $\phi \in (\mathbb{R}_0^+)$, $\boldsymbol{M} \in (\mathbb{R}_0^+)^m$, and $\boldsymbol{z} \in (\mathbb{R}_0^+)^n$.

2. If the program does not give any solution, this means that the net does not have a proper T-semiflow. This is the case if the net is not live or if it is not bounded. The non bounded case will be treated in Section 4.1.

3. Cohen *et al.* provide a computation of the rates by means of a policy iteration over a dynamic programming problem. While this technique is known to converge very fast to the solution, it is rather difficult to compute its worst case complexity. The linear program on the other hand may be slower on average but has a polynomial complexity in the worst case.

4. By comparing this LP with that of Recalde and Silva we can show that (surprisingly, at least for us) the steady state firing rates are the same for both models. Indeed, the solution by Recalde and Silva (2001) is a LP similar to ours where a single Little law is written for each P-semiflow (i.e., not for each place), and the LP takes the slowest one. The solution by Recalde and Silva (2001) has been proved to find the optimal solution and therefore should coincide with ours.

5. Our approach, unlike the Recalde and Silva one that compacts the single place information into P-semiflows, keeps an equation for each place. There are two advantages to this.

Firstly, our approach can also be used for nets that are not "structurally live and bounded" but become live and bounded once the timing structure is superimposed to the net structure.

Secondly, our linear characterization may be used to determine the average steady state marking (or at least some of its components). While we know the earliest firing policy provides an optimal solution, it could be that many other transient behavior (leading to different markings) also yield the same value for $\boldsymbol{y}$. Since we have no control on the solution provided by the LPP solver, we may not know whether the marking provided as a solution is the marking corresponding to the ASAP policy. We are still exploring a characterization of all those cases where we can exactly compute the average steady state marking (by restricting the $\boldsymbol{z}$ vector or minimizing it in the objective function). As an immediate consequence of the form of the LP, the case where the net does not contain any synchronization (this class includes state machines) is a case where the marking is unique (equality is reached in all the constraints over $\boldsymbol{M}$).

4. OPTIMAL ROUTING POLICY

Here is the main objective of the section: given a deterministic timed continuous Petri net, find the best routing parameters $\rho(p,t)$ which give the maximal firing rates.

We will assume that the net has a final cyclic behavior, i.e., the firing rates and the average markings reach a finite constant value. This case is characterized by the existence of a T-semiflow (not necessarily positive).

The optimization could be done by choosing initial values for the routing parameters arbitrarily and then by solving the LPP (2). On that LP, using sensitivity analysis techniques on the routing parameters allows one to evaluate how they should vary in order to increase the rates. Finally, iterating this procedure until convergence provides the optimal routing parameters. However, this technique has several drawbacks. Firstly, it may have a rather high computational complexity. Secondly, it may not convergence to the optimal solution (as it will be shown below).

Another approach is to come up with a new linear program having additional degrees of freedom on the T-semiflow that we will exploit for optimization.

$$\max\ \boldsymbol{a}^T\boldsymbol{y}$$
$$s.t.\ \begin{cases}\boldsymbol{Cy}=\boldsymbol{0}\\ M_i \geq \boldsymbol{Pre}(p_i,\cdot)\boldsymbol{y}\delta_i \text{ for all } p_i\in P\\ \boldsymbol{M}=\boldsymbol{M}(0)+\boldsymbol{Cz},\end{cases} \tag{4}$$

where the variables are $\boldsymbol{y}\in(\mathbb{R}_0^+)^n$, $\boldsymbol{M}\in(\mathbb{R}_0^+)^m$, and $\boldsymbol{z}\in(\mathbb{R}_0^+)^n$ and a is a vector chosen appropriately.

Theorem 3. The solution of the linear program (4) gives the optimal T-invariant (cyclic operation), optimal routing (coefficients to split the flow at each choice place), cycle time and finally also the transient firing vector to reach this steady state.

Proof. The proof is similar to the proof of Theorem 2 where equation $\boldsymbol{Cy}=\boldsymbol{0}$ says that $\boldsymbol{y}$ must be a T-semiflow, $M_i \geq \boldsymbol{Pre}(p_i,\cdot)\boldsymbol{y}\delta_i$ is Little's law, which must be satisfied once the stationary regime is reached and $\boldsymbol{M}=\boldsymbol{M}(0)+\boldsymbol{Cz}$ is the CNS of reachability in the underlying CPN. □

Again, this theorem calls for comments.

1. The linear program (4) looks similar to the linear program (2). However, the main difference lies in the net structure $\boldsymbol{C}$ which admits several T-semiflows. Since the T-invariant is not given, the transformation from the linear program (2) to (3) is not possible here.

2. The optimal solution $\boldsymbol{y}$ is a T-semiflow, not necessarily positive. By this we mean that some components may be zero. We are interested in a periodic mode of operation that not necessarily requires the firing of all transitions. In section 4.1 we further generalize this concept assuming that the mode of operation need only to be repetitive but not necessarily stationary (i.e., we look for vectors $\boldsymbol{y}$ such that $\boldsymbol{Cy}\geq 0$.

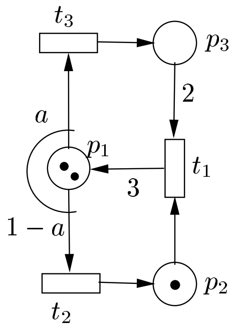

Fig. 4. A CPN where the transient regime should not be asap to get the optimal firing rate.

3. As before, some steady state marking components can also be obtained whenever the equality is reached in the Little's law equations (in particular, this is the case for all places preceding non synchronizing transitions).

4. If we want to relax the ASAP policy we put inequalities in all the equation derived from Little's law. Using this we can also prove that this optimal cyclic behavior corresponds to a stationary routing ASAP. Indeed, relaxing Little's law we get no advantage if the objective function is to maximize over $\boldsymbol{z}$.

We have just seen that once the optimal stationary regime is reached, all the transitions fire according to the earliest firing policy. However, this is not necessarily the case for the transient regime, with firing vector $\boldsymbol{z}$. Indeed, the optimal regime is not always reachable with a stationary routing using the ASAP policy. This is actually the case in the net displayed in Figure 4 where the delays in all the places are 1. In order to reach the largest firing rate for all transitions, one must fire transition t_3 twice and t_1 once before the stationary regime starts (this is not possible with the asap policy). Then, the optimal stationary routing must be $a=2/3$ and the total rate is $\boldsymbol{1}^T\boldsymbol{y}=2$. With the asap firing policy used from the start, this routing policy only leads to a total firing rate of 4/3. In this case, sensitivity analysis over LPP (2) will never provide the optimal rate of 2.

4.1 *Case of unstable nets*

We can also relax the hypothesis that a steady state marking should be reached, in the sense that we allow the possibility that the marking in some place grows unbounded. This corresponds to assume that $\boldsymbol{y}$ is not necessarily a T-invariant but a T-increase I.e., a vector such that $\boldsymbol{C} * \boldsymbol{y} \geq \boldsymbol{0}$).

5. HEURISTIC OPTIMIZATION OF DISCRETE NETS

In this section we consider classical discrete Petri nets where the marking must remain integer. The stationary routing parameters of the continuous nets are replaced by routing functions: each token entering a place p is assigned an output transition according to a routing function $r_p : \mathbb{N} \to p^\bullet$. The link with the routing parameters of the continuous net is done by assuming that the function r_p sends a proportion of $\rho(p,t)$ tokens to transition t asymptotically:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}_{\{r_p(n)=t\}} = \rho(p,t). \qquad (5)$$

This is done, for example, by choosing a *stochastic iid Bernoulli function* r_p where $Pr(r_p(n) = t) = \rho(p,t)$. Another (deterministic) possibility is to choose a *stair case routing function* r_p such that

$$\sum_{n=1}^{N} \mathbf{1}_{r_p(n)=t} = \lfloor \rho(p,t)N \rfloor .$$

The evolution equations in the discrete case become

$$Z_j(\tau) \leq \min_{p_i \in {}^\bullet t_j} \left\lfloor \frac{A_{i,j}(H_i(\tau - \delta_i))}{Pre(p_i,t_j)} \right\rfloor \qquad \forall t_j \in T$$

$$H_i(\tau) = M_i(0) + \sum_{t_j \in {}^\bullet p_i} Post(p_i,t_j) Z_j(\tau) \ \forall p_i \in P,$$

where $A_{i,j}(N) = \sum_{n=1}^{N} \mathbf{1}_{\{r_i(n)=t_j\}}$. When the net is live and bounded, the firing rate of the transitions, $\phi_j = \lim_{\tau\to\infty} Z_j(\tau)/\tau$ is shown to exist using transition blocking techniques by Gaujal *et al.* (2001). However, the computation of ϕ_j seems to be rather difficult. There are several reasons for that.

Firstly, the firing rate in the discrete case depends on the choice of the routing functions. In particular, the stair case and the Bernoulli routings may not have the same rates (there is strong evidence to claim that the stair case is always the best routing function satisfying the asymptotic condition (5)).

Secondly, the reachability condition in the discrete case is known to be difficult to check (this is the main reason why LPP 2 does not work here).

Another difficulty comes from the fact that the optimal stationary regime in the discrete case may not be reached by the earliest firing policy. The example in Figure 5 is a straightforward translation of a Tetris heap taken from (Mairesse and Vuillon, 1998) into a Petri net. 5. If the delays in the places are chosen such that $\delta_1 = 8/89, \delta_2 = 32/89, \delta_3 = 81/89, \delta_4 = 8/89, \delta_5 = 0$ then the stationary optimal asap routing is $(4333)^\infty$ with a total rate of 89/32 while the optimal routing policy is $(433)^\infty$ with a higher rate: 3.

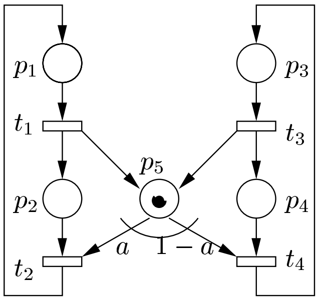

Fig. 5. A Petri net with routing

It is not difficult to prove (just looking at the evolution equations) that the firing rates in the discrete case are not larger than in the continuous case. It is also easy to construct examples where the rate in the discrete case is strictly smaller than in the continuous case.

Finally, one can note that any routing function in the discrete case can be implemented by using supervisory control (as it is done by Alpan and Gaujal (2000)).

REFERENCES

Alpan, G., B. Gaujal, "Supervisory control of Petri nets using routing functions: starvation avoidance issues", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 30, no. 5, pp. 684-695, 2000.

Campos, J., G. Chiola, M. Silva, "Ergodicity and Throughput Bounds of Petri Nets with Unique Consistent Firing Count Vector", *IEEE Trans. on Software Engineering*, vol. 17, no. 2, pp. 117-125, 1991.

Cohen, G., S. Gaubert, J.P. Quadrat, "Algebraic System Analysis of Timed Petri Nets", in *Idempotency*, J. Gunawardena (ed.), Cambridge University Press, 1998.

Gaujal, B., S. Haar, J. Mairesse, "Blocking a transition in a Petri net, and what it tells us about its asymptotic behavior", *Technical Report RR-, INRIA*, 2001.

Mairesse, J., L. Vuillon, "Optimal Sequences in a Heap Model with Two Pieces", *Technical Report 98/09, LIAFA*, University of Paris 7, 1998.

Recalde, L., E. Teruel, M. Silva, "Autonomous Continuous P/T Systems", *Application and Theory of Petri Nets 1999*, LNCS, vol. 1639, pp. 107-126, 1999.

Recalde, L., M. Silva, "Petri Net Fluidification Revisited: Semantics and Steady State", *European Journal of Automation APII-JESA*, 2001.