On Multi-enabledness in Time Petri Nets*

Hanifa Boucheneb[1], Didier Lime[2], and Olivier H. Roux[2]

[1] Department of Computer Engineering and Software Engineering,
École Polytechnique de Montréal,
P.O. Box 6079, Station Centre-ville, Montréal, Québec, Canada, H3C 3A7.
hanifa.boucheneb@polymtl.ca
[2] LUNAM Université, École Centrale de Nantes, IRCCyN
1, rue de la Noë, 44 300 Nantes, France.
{didier.lime, olivier-h.roux}@irccyn.ec-nantes.fr

Abstract. We consider time Petri nets with multiple-server semantics. We first prove that this setting is strictly more expressive, in terms of timed bisimulation, than its single-server counterpart. We then focus on two choices for the firing of multiple instances of the same transition: the more conservative safety-wise non deterministic choice, where all firable instances may fire in any order, and a simpler alternative, First Enabled First Fired (FEFF), where only the oldest instance may fire, obviously leading to a much more compact state-space. We prove that both semantics are not bisimilar but actually simulate each other with strong timed simulations, which in particular implies that they generate the same timed traces. FEFF is then very appropriate to deal with linear timed properties of time Petri nets.

Keywords: Time Petri nets, multiple/single-server semantics, firing choice policies, strong/weak timed simulations.

1 Introduction

The theory of Petri Nets provides a general framework to specify the behavior of real-time reactive systems, including their time constraints. Time constraints may be expressed in terms of stochastic delays of transitions (stochastic Petri nets), fixed values associated with places or transitions [13], or intervals labelling places, transitions or arcs [7, 10–12, 15]. Among these time extensions of Petri nets, we consider here time Petri nets [11] "à la Merlin" (threshold semantics) in both single-server and multiple-server semantics. This model associates with each transition a static firing interval constraining their firing dates. The multi-enabledness appears as soon as we consider non-safe Petri nets and is consequently both a theoretical and a practical problems. As an example, in a production line, a multi-enabled transition can either model a queue for a machine able to process one peace at a time or a conveyer belt able to move more than one object. The multiple-server semantics allows to handle, at the same time, several enabling instances of the same transition, whereas it is not allowed in the

* This work has been partially funded by project ImpRo ANR-2010-BLAN-0317.

single-server semantics. In [6], the authors discussed and showed the benefits of multiple-server semantics over single-server semantics such as scaling and conciseness power. Multi-enabledness allows very compact representations for some systems [6], where system resources are represented by tokens. In such a case, adding new resources to a system consists in simply adding new tokens in the marking, without changing the structure of the Petri net [6].

Timing information is either associated with tokens (age semantics) [8] or enabled transitions (threshold semantics) [4, 6]. For the multiple-server threshold semantics, each enabling instance of the same transition has its own timing information characterized by means of a clock (used to measure the time elapsing since its enabling) or a firing interval (indicating its firing delays). In [4], two firing choice policies have been proposed to manage the different enabling instances of the same transition: non deterministic (NDF) and First Enabled First Fired (FEFF) firing policies. For the NDF firing choice, all possible firing orders of these instances are considered, whereas, for the FEFF firing choice, only one firing order, corresponding to firing the oldest one first, is considered. The NDF firing choice includes the FEFF firing choice. Consequently, it strongly simulates the FEFF firing choice. However, as it considers all possible firing orders of instances of the same transitions, it may cause a blow-up of the state space, compared to the FEFF firing choice.

In this paper, we first give an overview of the different semantics of multi-enabledness for time Petri nets. Then, we show that the multiple-server semantics adds expressiveness relatively to the single-server semantics. We also prove that the FEFF firing choice strongly simulates the NDF firing choice. As, the NDF firing choice strongly timed simulates the FEFF firing choice, it follows that both firing choice policies (NDF and FEFF) strongly simulate each other and then generate the same timed language.

This paper is organized as follows. Section 2 defines formalisms and notions used in the paper such as timed transition systems, strong (weak) timed simulation relations and time Petri nets. Then, it discusses the different semantics of multi-enabledness of time Petri nets, proposed in the literature. Section 3 is devoted to the threshold semantics of time Petri nets, in the context of multiple-server policy and the comparison of the expressiveness relatively to the single-server semantics. Section 4 compares two firing choice policies: NDF and FEFF. The conclusion is presented in Section 5.

2 Preliminaries

Let $\mathbb{N}$, $\mathbb{Q}^+$ and $\mathbb{R}^+$ be the sets of natural, non-negative rational and non-negative real numbers, respectively.

2.1 Timed transition systems and timed (bi)simulation

As usual, we shall define the operational semantics of our time Petri nets by means of timed transition systems (TTS) combining both discrete (actions) and continuous (time elpasing) transitions [6]:

Definition 1 (Timed Transition System). *A TTS is a 4-tuple* $\mathcal{S} = < Q, q_0, \Sigma, \rightarrow >$ *where* Q *is a set of states,* $q_0 \in Q$ *is the initial state,* Σ *is the set of discrete actions (disjoint from the time domain* $\mathbb{R}^+$ *of the continuous actions), and* $\rightarrow \in Q \times (\Sigma \cup \mathbb{R}^+) \times Q$ *is the transition relation.*

A tuple $(q, a, q') \in \rightarrow$, also denoted $q \xrightarrow{a} q'$, represents the transition from state q to state q' by the discrete or continuous action a.

In the sequel, we assume that all TTSs satisfy the classical time-related conditions where $d, d' \in \mathbb{R}^+$:

- time determinism: if $q \xrightarrow{d} q'$ and $q \xrightarrow{d} q''$ then $q' = q''$;
- time additivity: if $q \xrightarrow{d} q'$ and $q' \xrightarrow{d'} q''$ then $q \xrightarrow{d+d'} q''$;
- null delay: $\forall q : q \xrightarrow{0} q$;
- time continuity: if $q \xrightarrow{d} q'$ then $\forall d' \leq d, \exists q'', q \xrightarrow{d'} q''$ and $q'' \xrightarrow{d-d'} q'$.

Definition 2 (Run in a TTS). *A run* ρ *in a TTS* $\mathcal{S} = < Q, q_0, \Sigma, \rightarrow >$ *is a (possibly infinite) sequence* $q_0 a_0 q_1 a_1 \ldots a_{n-1} q_n \ldots$ *such that* $\forall i, (q_i, a_i, q_{i+1}) \in \rightarrow$.

We assume w.l.o.g. that in a run, discrete and continuous transitions are strictly alternating. The case of a run ending in infinite delay raises no theoretical issue but we omit it for the sake of readability. Any run ρ can therefore be written as: $\rho = q_0 \xrightarrow{d_1} q_1 \xrightarrow{t_1} q_2 \xrightarrow{d_2} q_3 \xrightarrow{t_2} \cdots$, where d_i and t_i for $i > 0$ are continuous and discrete actions, respectively.

Definition 3 (Timed and Untimed Traces). *For any run* $\rho = q_0 \xrightarrow{d_1} q_1 \xrightarrow{t_1} q_2 \xrightarrow{d_2} q_3 \xrightarrow{t_2} \cdots$, *the* timed trace *(timed word) of* ρ *is the sequence* $d_1 t_1 d_2 t_2 \ldots$.

The untimed trace *(also called firing sequence) of* ρ *is the sequence* $t_1 t_2 \ldots$.

Definition 4 (Timed Language). *The* timed language *of* $\mathcal{S}$*, denoted* $L(\mathcal{S})$*, is the set of its timed traces.*

In order to compare the different semantic choices we shall introduce the notion of simulation:

Definition 5 (Strong timed (bi)simulation). *Let* $\mathcal{S}_1 = < Q_1, q_{10}, \Sigma, \rightarrow_1 >$ *and* $\mathcal{S}_2 = < Q_2, q_{20}, \Sigma, \rightarrow_2 >$ *be two timed transition systems.*

A binary relation $\prec_S \subseteq Q_1 \times Q_2$ *is a (strong)* timed simulation *iff* $\forall (q_1, q_2) \in \prec_S, \forall a \in \Sigma \cup \mathbb{R}^+$, $(\exists q_1' \in Q_1, q_1 \xrightarrow{a}_1 q_1') \Rightarrow (\exists q_2' \in Q_2, q_2 \xrightarrow{a}_2 q_2'$ *and* $(q_1', q_2') \in \prec_S)$.

A strong timed simulation $\simeq_S \subseteq Q_1 \times Q_2$ *is a (strong)* timed bisimulation *if* $\simeq_S^{-1} \subseteq Q_2 \times Q_1$ *is also a strong timed simulation.*

We say that transition system $\mathcal{S}_1$ is strongly simulated by $\mathcal{S}_2$ (i.e., $\mathcal{S}_2$ strongly simulates $\mathcal{S}_1$), if there exists a strong timed simulation relation $\prec_S \subseteq Q_1 \times Q_2$ s.t. $(q_{10}, q_{20}) \in \prec_S$. Note that such a simulation implies that $L(\mathcal{S}_1) \subseteq L(\mathcal{S}_2)$.

Similarly, transition systems $\mathcal{S}_1$ and $\mathcal{S}_2$ are strongly timed bisimilar if there exists a strong timed bisimulation $\simeq_S \subseteq Q_1 \times Q_2$ and $(q_{10}, q_{20}) \in \simeq_S$.

An invisible action is any action which does not belong to $\Sigma \cup \mathbb{R}^+$. We denote all invisible actions by $\epsilon \notin \Sigma \cup \mathbb{R}^+$.

Let $\sigma \in (\Sigma \cup \{\epsilon\} \cup \mathbb{R}^+)^+$ be a timed trace and $vis(\sigma)$ the timed trace obtained by eliminating invisible actions (ϵ) of σ and grouping continuous actions.

Definition 6 (Weak timed simulation). *Let* $\mathcal{S}_1 = < Q_1, q_{10}, \Sigma \cup \{\epsilon\}, \rightarrow_1 >$ *and* $\mathcal{S}_2 = < Q_2, q_{20}, \Sigma \cup \{\epsilon\}, \rightarrow_2 >$ *be two timed transition systems and* $\preceq_W \subseteq Q_1 \times Q_2$ *a binary relation. Relation* $\preceq_W$ *is a* weak timed simulation *iff* $\forall (q_1, q_2) \in \preceq_W, \forall a \in \Sigma \cup \mathbb{R}^+, (\exists q_1' \in Q_1, q_1 \xrightarrow{a}_1 q_1') \Rightarrow (\exists q_2' \in Q_2, \exists \sigma$ *s.t.* $vis(\sigma) = a, q_2 \xrightarrow{\sigma}_2 q_2'$ *and* $(q_1', q_2') \in \preceq_W)$.

We derive the notions of weak timed bisimulation and weak timed (bi)similarity of transitions systems in exactly the same way as for strong simulations.

2.2 Time Petri nets

We now introduce the formalisms considered in this article:

Definition 7 (Petri net). *A Petri net is defined by a 4-tuple:* $< P, T, \mathsf{Pre}, \mathsf{Post}, M_0 >$, *where:*

- P *and* T *are finite sets of places and transitions, respectively (s.t.* $P \cap T = \emptyset$*);*
- $\mathsf{Pre}, \mathsf{Post} \in [P \times T \longrightarrow \mathbb{N}]$ *are the backward incidence and the forward incidence functions, respectively. They indicate, for each transition, the tokens needed for its firing and those produced;*
- $M_0 \in [P \longrightarrow \mathbb{N}]$ *is the initial distribution of tokens in places, called the initial marking.*

A marking M of a Petri net is a function from P to $\mathbb{N}$. Let $M \in [P \longrightarrow \mathbb{N}]$ be a marking and t a transition. Transition t is k-enabled for $k > 0$ iff $M \geq k \times \mathsf{Pre}(., t)$ and $M \ngeq (k+1) \times \mathsf{Pre}(., t)$. In this case, k is the enabling degree of t in M. If t is k-enabled for some $k > 0$, we simply say it is enabled. When we say t is *multi-enabled* we emphasize the fact that $k > 1$. By convention, t is said to be 0-enabled in M if it is not enabled in M. Note that in case, a transition has at least an input place, the set of its enabling instances in M is finite. We suppose here that each transition has at least an input place.

If t is enabled in M, it may *fire*, leading to the marking M' s.t. $\forall p \in P, M'(p) = M(p) - \mathsf{Pre}(p, t) + \mathsf{Post}(p, t)$.

Let INT be the set of intervals of $\mathbb{R}^+$ of the form $[a, b]$ or $[a, \infty[$, where $a, b \in \mathbb{Q}^+$. For any interval $I \in INT$, the lower and the upper bounds of I are denoted $\downarrow I$ and $\uparrow I$, respectively.

Definition 8 (Time Petri Net). *A time Petri net (TPN) is defined by a 7-tuple:* $< P, T, \mathsf{Pre}, \mathsf{Post}, Is, M_0 >$ *where:*

- $< P, T, \mathsf{Pre}, \mathsf{Post}, M_0 >$ *is a Petri net;*

– $Is \in [T \to INT]$ *is a function which associates with each transition* t *a static firing interval* $Is(t)$.

Intuitively, a transition t is firable if it is maintained enabled during a time inside its static firing interval. It must be fired without any additional delay, if it is maintained enabled $\uparrow Is(t)$ time units, unless it is immediately disabled by a conflicting transition. Firing a transition takes no time but leads to a new marking.

2.3 TPN semantics

Several semantics are proposed in the literature for TPN that can be classified according to four policies (service, granularity, memory and choice) [1–3, 9] and the characterization of timing information (clock state vs interval state).

Server policy: For time Petri Nets, time is used to model a non instantaneous service represented by a timed transition. In this context, the service policy specifies whether several enabling instances of the same transition may be considered simultaneously (multiple-server semantics) or not (single-server semantics). The multiple-server semantics allows to handle, at the same time, several services per transition whereas it is not allowed in the single-server semantics. For single-server semantics, the multi-enabledness is not ambiguous since a transition can do only one thing at the same time (only one enabling instance of each transition is considered at each state), whereas different interpretations can be defined for multiple-server semantics.

Granularity policy: This policy indicates which objects timing information is associated with. Timing information is either associated with tokens (age semantics) [8] or enabled transitions (threshold semantics) [6]. In [5], the authors considered the age semantics where tokens are managed FIFO based on their ages. As tokens are handled FIFO, an enabled transition will always use the oldest tokens from each place. The difference between age and threshold semantics is highlighted in [9] by the difference between the *individual token interpretation* and the *collective token interpretation*. It is particularly significant when two or more tokens are needed, in a given place, to fire a transition. Let us consider the example of [6] depicted in Fig. 1 showing the difference between age and threshold semantics. In this example, a server answers to requests, in a delay between 2 and 3 time units for each request and we want to detect a too heavy load of the system. More precisely, we want to detect the presence of more than 40 requests during a period of 30 time units. The modeling of such a system is given in Fig. 1. Using the age semantics, the transition *Loaded* will never be fired. Indeed, each token will stay at most 3 time units in place *Running*: no multi-set of 40 tokens will exist with a token older than 3 time units. Using the threshold semantics, the transition *Loaded* will be enabled once 40 tokens will be in place *Running*, and it will fire 30 time units after, as long as at least 40 tokens are in the place, independently of their ages.

In this paper, we focus on the threshold semantics.

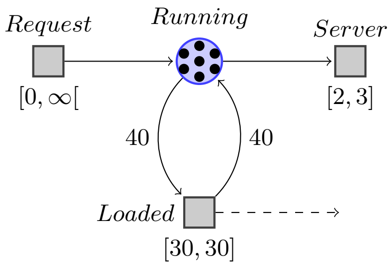

Fig. 1. A TPN illustrating the difference between age and threshold semantics [6]

Memory policy: This policy specifies when the timing information is set or reinitialized. For the age semantics, the timing information of tokens is set at their creation. In the context of threshold semantics, the memory policy relies on the notion of newly enabled transitions. In the classical semantics (also called intermediate semantics), this notion is defined using intermediate markings (markings resulting from the consumption of tokens): when a transition is fired, all transitions not enabled in the intermediate marking but enabled in the successor marking are considered as newly enabled. The firing of a transition is then not atomic w.r.t. markings. In [1, 2], the authors have discussed other semantics where the firing of transitions is considered atomic: atomic and persistent atomic semantics. In such semantics, all transitions not enabled before firing a transition t but enabled after its firing are newly enabled. Another difference between the intermediate, atomic and persistent atomic semantics lies in the particular case of the fired transition. If the fired transition enables again itself, it is considered as newly enabled in the intermediate and atomic semantics but not newly enabled in the persistent atomic semantics. For the single-server and threshold semantics, the intermediate, atomic and persistent atomic semantics are equivalent w.r.t. weak timed bisimulation if the intervals are all right-closed and persistent atomic is more expressive otherwise [2, 3, 14].

Firing choice policy: This policy specifies the enabled transitions to fire first and those to disable first, in case of conflicts. In the context of TPNs, the choice of transition to fire first is non deterministic for different transitions. For the age semantics, in [5], tokens are managed First in First Out (FIFO semantics). When a transition is fired, it consumes the oldest tokens first. In the case of the threshold and multiple-server semantics, the multi-enabledness of a transition t can be considered as different transitions, which we call *enabling instances*, and which are either totally independent (non deterministic firing choice (NDF)) or managed so as to fire the oldest one first (First Enabled First Fired (FEFF) policy).

Disabling choice policy: This policy specifies which enabling instances of transitions to disable first: the most recent ones first (Last Enabled First Disabled (LEFD)) or the oldest ones first (First Enabled First Disabled (FEFD)) are possible policies. As for the firing choice policy we can also take into account all possible choices non-deterministically (NDD).

Clock vs interval states: Besides these policies, there are, in the setting of threshold semantics, two known characterizations of timing information. The first one is based on clocks. A clock is either associated with each enabled transition to measure its enabling time (time elapsed since it became enabled most recently) or associated with each token to measure its age (time elapsed since its creation). The second characterization of timing information is based on dynamically decreasing intervals associated with enabled transitions indicating the time remaining until they can fire.

TPN " à la Merlin": The classical semantics of TPN is single-server, threshold and intermediate semantics with non deterministic choice of transitions to fire first. The timing information is either characterized by means of clocks (clock states) or time intervals (interval states).

The clock state is defined as a marking and a function which associates with each enabled transition the value of its clock. The clock of a transition t is set to 0, when it is newly enabled. Afterwards, its value increases synchronously with time until it is fired or disabled by firing a conflicting transition. It is firable if its clock value reaches its static firing interval. It must be fired without any additional delay transition when its clock reaches $\uparrow Is(t)$, unless it is disabled.

The interval state is defined as a marking and a function which associates with each enabled transition the time interval in which the transition can be fired. When a transition t is newly enabled, its firing interval is set to its static firing interval. The bounds of this interval decrease synchronously with time, until t is fired or disabled by another firing. t is firable, if the lower bound of its firing interval reaches 0. It must be fired, without any additional delay, if the upper bound of its firing interval reaches 0, unless it is disabled.

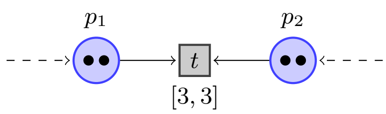

Fig. 2. A simple TPN with multiple enabledness.

Illustrative example: Let us point out, by means of the simple example of Fig. 2, some subtle differences between the semantics discussed above. Assume that initially, the marking is $\binom{p_1}{p_2} = \binom{1}{1}$. At date 1, a token arrives in p_1 leading to the marking $\binom{2}{1}$. Then, another token arrives in p_2 at date 2 leading to the marking $\binom{p_1}{p_2} = \binom{2}{2}$. The transition t is 2-enabled in the marking $\binom{2}{2}$.

For all semantics, the transition t will be fired twice. However, the firing dates of the transitions vary according to the service, granularity and memory policies.

For the single-server and threshold semantics, only one enabling instance of transition t is considered from each state. This instance is enabled since date 0. When this instance is fired, transition t is again enabled. The timing information of this enabling instance of t may be either reinitialized or not, according to the memory policy (intermediate, atomic or persistent atomic semantics). For both intermediate and atomic semantics, the transition t will be fired at dates 3 and 6. Indeed, t is enabled since date 0 and then is fired at date 3. When t is fired, it is again enabled in the resulting marking, its timing information is reinitialized. Therefore, it will be fired again 3 time units later (i.e., at date 6). For persistent atomic semantics, the transition t will be fired at dates 3 and 3, since, in this case, when t is fired for the first time, its timing information is not reinitialized.

For the age semantics, there are 2 tokens in place P_1 created at dates 0 and 1, respectively. There are also 2 tokens in place P_2 created at dates 0 and 2, respectively. In case, the tokens are managed FIFO, the dates of the first and the second firing of t are 3 and 5. The first firing of t uses the tokens of p_1 and p_2 with age 0. The second firing of t uses the remaining tokens.

For the multiple-server (multi-enabledness) semantics, the two enabling instances of transition t can be fired at dates:

- 3 and 5 for the threshold semantics, since the first and the second instances of t are enabled since dates 0 and 2, respectively.
- 3 and 5 for the age semantics with FEFF discipline. t is considered to be mutli-enabled as soon as the marking is $\binom{2}{1}$.
- 3 and 5 or 4 and 5 for the age semantics with non deterministic choice of tokens to be used. Indeed, the dates of the first and the second firing of t depend on the ages of tokens used. There are two possibilities. The first firing of t may use either the tokens of p_1 and p_2 with age 0 or the token of p_1 with age 1 and the token of p_2 with age 0. The second firing of t uses the remaining tokens. So, the dates of the first and the second firing of t are either 3 and 5 or 4 and 5.

In [6], the authors discussed and showed the benefits of multiple-server semantics over single-server semantics such as scaling and conciseness power. However, does the multiple-server semantics increase the expressive power of time Petri nets or not? So, the first aim of this paper is to investigate this question for the threshold semantics. The other aim is the comparison of two firing choice policies: NDF and FEFF.

3 Threshold semantics in the context of multiple-server policy

In the threshold and multiple-server semantics, a clock or a firing interval is associated with each enabling instance of a transition. The choice of transition to fire first is non deterministic for different transitions. The enabling instances of the same transition are either considered totally independent (non deterministic choice) or managed so as to fire the oldest one first (First Enabled First Fired (FEFF) policy). In the sequel, we consider the case of non deterministic choice of the transition to fire first.

Let M be a marking. For economy of notations, we suppose that the set of transitions is strictly ordered ($t < t'$ or $t' < t$, for any pair of transitions of T) and the enabling instances of transitions of M are managed in an ordered list, denoted en. In en, the enabling instances of the same transition are ordered from the oldest to the newest one and transitions appear in increasing order. An enabling instance of this list is referred to as t^i where $t \in T$ is its transition and i is its position in the list en.

3.1 Clock based timing information

For the clock based timing information, the TPN state is defined by the triplet $q = (M, \mathsf{en}, \nu)$, where $M \in [P \longrightarrow \mathbb{N}]$ is a marking, en is the list of enabling instances of transitions in M, where the enabling instances of the same transition are ordered from the oldest to the newest one and transitions appear in increasing order, and $\nu \in [\mathsf{en} \longrightarrow \mathbb{R}^+]$ is a clock valuation over en. For each $t^i \in \mathsf{en}, \nu(t^i)$ is the clock value of the i^{th} enabling instance. The initial state is $q_0 = (M_0, \mathsf{en}_0, \nu_0)$, where M_0 is the initial marking, en_0 is the appropriately ordered list of enabling instances of transitions in M_0 and $\forall t^i \in \mathsf{en},\ \nu_0(t^i) = 0$.

All clocks of transitions evolve uniformly with time. Let $q = (M, \mathsf{en}, \nu)$ be a state, $d \in \mathbb{R}^+$. We denote $\nu + d$ the function ν' defined by $\forall t^i \in \mathsf{en}, \nu'(t^i) = \nu(t^i) + d$. It specifies the evolution of time by d units. $(M, \mathsf{en}, \nu) \xrightarrow{d} (M, \mathsf{en}, \nu')$ iff

$$\nu' = \nu + d \text{ and } \forall t^i \in \mathsf{en}, \nu(t^i) + d \leq\uparrow Is(t)$$

Let $q = (M, \mathsf{en}, \nu)$ be a state and $t \in T$. Transition t is firable at state $q = (M, \mathsf{en}, \nu)$ iff there is at least an enabling instance t^i of t in en s.t. its clock has reached its firing interval (i.e., $\nu(t^i) \geq\downarrow Is(t)$). In case transition t^i is firable at state $q = (M, \mathsf{en}, \nu)$, its firing will consume $\mathsf{Pre}(p, t)$ tokens from each place p and produce $\mathsf{Post}(p, t)$ tokens in each place p. Consequently, it will disable transitions that are in conflict and enable new instances of transitions. We denote $\mathsf{CF}(M, \mathsf{en}, t^i)$ the set of enabling instances of transitions of en in conflict in M with t^i. The set of newly enabled instances in the marking reached from M by firing t^i is denoted $\mathsf{Nw}(M, \mathsf{en}, t^i)$.

If t^i is firable at $q = (M, \mathsf{en}, \nu)$, its firing leads to the state $q' = (M', \mathsf{en}', \nu')$ s.t.

- $M' = M - \mathsf{Pre}(., t) + \mathsf{Post}(., t)$,
- en' is computed from en by eliminating enabling instances of $\mathsf{CF}(M, \mathsf{en}, t^i)$ and then inserting the enabling instances of $\mathsf{Nw}(M, \mathsf{en}, t^i)$ and
- ν' is computed from ν by eliminating clock values of enabling instances of $\mathsf{CF}(M, \mathsf{en}, t^i)$ and inserting value 0 for each enabling instance of $\mathsf{Nw}(M, \mathsf{en}, t^i)$.

We write $(M, \mathsf{en}, \nu) \xrightarrow{t^i} (M', \mathsf{en}', \nu')$, for $t^i \in \mathsf{en}$ iff t^i is firable at (M, en, ν), i.e., $t^i \in \mathsf{en}$ and its firing leads to the state (M', en', ν').

Example 1. As an example, consider the time Petri net at Fig. 3. and the state $q = (M, \mathsf{en}, \nu)$, where $M(P_1) = 3$, $\mathsf{en} = \{S^1, S^2, S^3, L^1, L^2, L^3\}$, $\nu(S^1) = \nu(L^1) = 2.5$, $\nu(S^2) = \nu(L^2) = 2.1$ and $\nu(S^3) = \nu(L^3) = 1.3$.

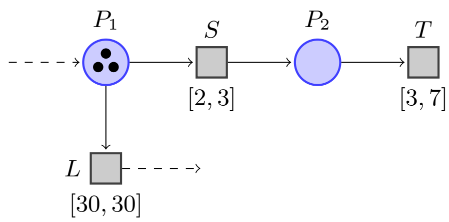

Fig. 3. A TPN illustrating CF and Nw

Let us consider the firing of the instance S^1 in the context of FEFF semantics, atomic memory policy and FEFD disabling choice policy, meaning that $\mathsf{CF}(M, \mathsf{en}, S^1) = \{S^1, L^1\}$. Thus, the firing of the transition S^1 denoted by $(M, \mathsf{en}, \nu) \xrightarrow{S^1} (M', \mathsf{en}', \nu')$ leads to the deletion of S^1 and L^1 and thus the other instances are left shifted in the list en'. Then, since $\mathsf{Nw}(M, \mathsf{en}, S^1) = \{T^1\}$, we have $\mathsf{en}' = \{S^1, S^2, L^1, L^2, T^1\}$ with $\nu'(S^1) = \nu'(L^1) = 2.1$, $\nu'(S^2) = \nu'(L^2) = 1.3$ and $\nu'(T^1) = 0$. □

Using clock based timing information, the behavior of a time Petri net is defined by means of the timed transition system $< Q, q_0, \Sigma, \longrightarrow >$, where Q is the set of clock states of the time Petri net, $q_0 = (M_0, \mathsf{en}_0, \nu_0)$ is its initial clock state, $\Sigma = T$, and $\longrightarrow$ is composed of continuous and discrete transitions defined as follows:
Let $q = (M, \mathsf{en}, \nu)$ and $q' = (M', \mathsf{en}', \nu')$ be two clock states, $d \in \mathbb{R}^+$ and $t \in T$.
$(M, \mathsf{en}, \nu) \xrightarrow{d} (M, \mathsf{en}, \nu + d)$ iff $(M, \mathsf{en}, \nu) \xrightarrow{d} (M, \mathsf{en}, \nu + d)$
$(M, \mathsf{en}, \nu) \xrightarrow{t} (M', \mathsf{en}', \nu')$ iff $\exists t^i \in \mathsf{en}, (M, \mathsf{en}, \nu) \xrightarrow{t^i} (M', \mathsf{en}', \nu')$.

3.2 Interval based timing information

For the interval based timing information, the interval TPN state is defined by the triplet (M, en, I), where M and en are defined as for the clock state and $I \in [\mathsf{en} \longrightarrow INT]$ is an interval function over en. For each $t^i \in \mathsf{en}, I(t^i)$ is the firing interval of the i^{th} enabling instance of t. The initial state is $q_0 = (M_0, \mathsf{en}_0, I_0)$, where $\forall t^i \in \mathsf{en}_0, I_0(t^i) = Is(t)$.

In this case, the behavior of TPN is defined by means of the timed transition system $< Q_I, (M_0, \mathsf{en}_0, I_0), \Sigma, \longrightarrow_I >$, where Q_I is the set of interval states of the TPN, $(M_0, \mathsf{en}_0, I_0)$ is its initial interval state, $\Sigma = T$, and $\longrightarrow_I$ is composed

of continuous and discrete transitions defined as follows:
Let (M, en, I) and (M', en', I') be two interval states, $d \in \mathbb{R}^+$ and $t \in T$.
$(M, \mathsf{en}, I) \xrightarrow{d}_I (M, \mathsf{en}, I')$ iff

$$\forall t^i \in \mathsf{en}, d \leq \uparrow I(t^i) \text{ and } I'(t^i) = [\max(0, \downarrow I(t^i) - d), \uparrow I(t^i) - d].$$

$(M, \mathsf{en}, I) \xrightarrow{t}_I (M', \mathsf{en}', I')$ iff $\exists t^i \in \mathsf{en}$ s.t.

- $\downarrow I(t^i) = 0$,
- $M' = M - \mathsf{Pre}(., t) + \mathsf{Post}(., t)$,
- en' is computed from en by eliminating enabling instances of $\mathsf{CF}(M, \mathsf{en}, t^i)$ and then inserting the enabling instances of $\mathsf{Nw}(M, \mathsf{en}, t^i)$ and
- I' is computed from I by eliminating firing intervals of enabling instances of $\mathsf{CF}(M, \mathsf{en}, t^i)$ and inserting the interval $Is(t')$ for each enabling instance t'^j of $\mathsf{Nw}(M, \mathsf{en}, t^i)$.

Theorem 1 proves that the two formulations of the semantics, clocks or intervals, are actually equivalent wrt. strong timed bisimulation.

Theorem 1 (Equivalence of clock and interval semantics). *All other semantic choices being the same, the transition systems, obtained for both clock and interval timing information, are strongly timed bisimilar.*

Proof. Let $\simeq$ be the binary relation over $\mathcal{Q}$ defined by:
$\forall (M, \mathsf{en}, \nu) \in \mathcal{Q}, \forall (M', \mathsf{en}', I) \in \mathcal{Q}_I,\ (M, \mathsf{en}, \nu) \simeq_S (M', \mathsf{en}', I)$ iff
$M = M', \mathsf{en} = \mathsf{en}',$ and $\forall t^i \in \mathsf{en}, I(t^i) = [\max(0, \downarrow Is(t) - \nu(t^i)), \uparrow Is(t) - \nu(t^i)]$.
It is easy to verify that $\simeq$ is a strong timed bisimulation. $\square$

3.3 Conflicting and newly enabled transitions

The notions of conflicting and newly enabled transitions are not dependent of the characterization of timing information. They mainly depend on the marking M, the list en and the memory (intermediate, atomic or persistent atomic semantics) and disabling choice (FEFD or LEFD) policies.

Conflicting transitions: Let M be a marking, en its list of enabling instances ordered appropriately as explained previously, t and t' two enabled transitions in M, $k > 0$ and $k' > 0$ their enabling degrees in M.

For the intermediate semantics, an enabling instance of t is in conflict with some enabling instances of t' in M iff the enabling degree of t' is decreased in the intermediate marking $M - \mathsf{Pre}(., t)$ (i.e., $M - \mathsf{Pre}(., t) \ngeq k' \times \mathsf{Pre}(., t')$). Let k'' be the enabling degree of t' in the intermediate marking $M - \mathsf{Pre}(., t)$. The firing of an enabling instance of t will disable the $k' - k''$ oldest or youngest enabling instances of t', dependently of the discipline LEFD or FEFD used to manage conflicting transitions. Moreover, the fired instance t^i is supposed to be in conflict with itself (i.e., $t^i \in \mathsf{CF}(M, \mathsf{en}, t^i)$).

For the persistent atomic semantics, an enabling instance of t is in conflict with some enabling instances t' in M iff the enabling degree of t' is decreased in the successor marking of M by t (i.e., $M - \mathsf{Pre}(.,t) + \mathsf{Post}(.,t) \not\geq k' \times \mathsf{Pre}(.,t')$). Let k'' be the enabling degree of t' in the successor marking of M by t. The firing of an enabling instance of t will disable the $k' - k''$ oldest or youngest enabling instances of t', dependently of the discipline LEFD or FEFD used to manage conflicting transitions.

For the atomic semantics, the set $\mathsf{CF}(M, \mathsf{en}, t^i)$ is computed in the same manner as for the persistent atomic semantics, except that t^i is supposed to be in conflict with itself (i.e., $t^i \in \mathsf{CF}(M, \mathsf{en}, t^i)$).

Newly enabled transitions: Let M be a marking, en its list of enabling instances ordered appropriately as explained previously, t^i an enabling instance of transition t in M and $M' = M - \mathsf{Pre}(.,t) + \mathsf{Post}(.,t)$ the successor marking of M by any enabling instance of t. The set $\mathsf{Nw}(M, \mathsf{en}, t^i)$ relies to the memory policy (intermediate, atomic or persistent atomic semantics) used.

For the intermediate semantics, there are new enabling instances of a transition $t' \in T$ in M', if its enabling degree in M' is greater than its enabling degree in the intermediate marking $M - \mathsf{Pre}(.,t)$. In other words, if a transition t' is k-enabled (for some $k \geq 0$) in $M - \mathsf{Pre}(.,t)$ but k'-enabled in M' with $k' > k$, then there are $k' - k$ new enabling instances of t' in M'.

For the atomic semantics, the firing of a transition is atomic. So, there are new enabling instances of a transition $t' \in T$ in M', if its enabling degree in M' is greater than its enabling degree in M. In other words, if a transition t' is k-enabled (for some $k \geq 0$) in M but k'-enabled in M' with $k' > k$, then there are $k' - k$ new enabling instances of t' in M'.

For the persistent atomic semantics, the set $\mathsf{Nw}(M, \mathsf{en}, t^i)$ is computed in the same manner as for the atomic semantics, except that if there are new enabling instances of the fired transition t in M', one of these enabling instances inherits the timing information of the fired transition.

For the rest of the paper, we fix a TPN $\mathcal{N}$ with multiple-server and threshold semantics. Moreover, we focus on the clock based timing information. Theorem 1 implies that the results shown here are also valid for the interval based timing information.

Property 1 follows from the definitions of CF and Nw: Note that according to these definitions, it holds that:

Property 1. Let $q = (M, \mathsf{en}, \nu)$ be a clock state of $\mathcal{N}$. For any pair (t^i, t^j) of enabling instances of the same transition t in en:

1. $\mathsf{CF}(M, \mathsf{en}, t^i) - \{t^i\} = \mathsf{CF}(M, \mathsf{en}, t^j) - \{t^j\}$
2. $\mathsf{Nw}(M, \mathsf{en}, t^i) = \mathsf{Nw}(M, \mathsf{en}, t^j)$
3. $t^i \in \mathsf{CF}(M, \mathsf{en}, t^i)$ iff $t^j \in \mathsf{CF}(M, \mathsf{en}, t^j)$

3.4 Multiple-server semantics adds expressiveness

In the context of threshold semantics, we establish in theorems 2 and 3 that the multiple-server semantics adds expressiveness relatively to the single-server policy.

Theorem 2. *Every TPN $\mathcal{N}$ can be translated into a TPN $\mathcal{N}'$ s.t. $\mathcal{N}$ in the single-server and threshold semantics is strongly or weakly timed bisimilar to $\mathcal{N}'$ in the multiple-server and threshold semantics.*

Proof. (sketch of the proof) In the single-server and threshold semantics, there is one clock (one firing interval) per transition, even if it is multi-enabled.

For the intermediate semantics, to achieve the translation, it suffices to add a place p_t for each transition t of $\mathcal{N}$ with $\mathsf{Pre}(p_t, t) = \mathsf{Post}(p_t, t) = 1$ and $M_0(p_t) = 1$. Doing so, we eliminate the multi-enabledness of t. Moreover, if t is again enabled after its firing, it is newly enabled in $\mathcal{N}'$, because t is not enabled in the intermediate marking (p_t is empty). Therefore, $\mathcal{N}'$ under the multiple-server, threshold and intermediate semantics is strongly timed bisimilar to $\mathcal{N}$ under the single-server, threshold and intermediate semantics.

This translation works also for the persistent atomic semantics, since the tokens of the added places are present before and after any firing. Therefore, if a transition t is again enabled after its firing in $\mathcal{N}$, it is also enabled again in $\mathcal{N}'$ after its firing. Under multiple-server, threshold and persistent atomic semantics, $\mathcal{N}'$ is strongly timed bisimilar to $\mathcal{N}$ under single-server, threshold and persistent atomic semantics.

For the atomic semantics, if a transition t is again enabled after its firing, it is considered as newly enabled. To deal with this case, the translation needs to add two places $p_{t_{in}}$ and $p_{t_{out}}$, and a transition t_t for each transition t of $\mathcal{N}$ with $\mathsf{Pre}(p_{t_{in}}, t) = 1, \mathsf{Post}(p_{t_{out}}, t) = 1$, $\mathsf{Pre}(p_{t_{out}}, t_t) = 1, \mathsf{Post}(p_{t_{in}}, t_t) = 1$, $M_0(p_{t_{in}}) = 1$, $M_0(p_{t_{out}}) = 0$ and $Is(t_t) = [0, 0]$. If a transition t is again enabled after its firing in $\mathcal{N}$, its firing in $\mathcal{N}'$ will empty place $p_{t_{in}}$, enable transition t_t and then disable t. As there is no delay between firings of t and t_t and the unique role of t_t is to allow the enabling of t (i.e., invisible transition), it follows that $\mathcal{N}$ under single-server, threshold and atomic semantics is weakly timed bisimilar to $\mathcal{N}'$ under multiple-server, threshold and atomic semantics. □

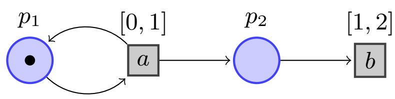

Fig. 4. Multiple-server threshold semantics with no equivalent single-server semantics

Theorem 3. *There is no TPN under single-server and threshold semantics equivalent to the TPN at Fig. 4 under multiple-server and threshold semantics (neither w.r.t. timed bisimulation nor w.r.t. timed language acceptance).*

Proof. Let $n_a(d)$ and $n_b(d)$ be the numbers of firings of transitions a and b at date d, respectively. Necessarily, $n_a(d) \geq n_b(d)$ and at date d, it remains

$n_a(d) - n_b(d)$ occurrences of b in less than 2 time units. For each of these occurrences of b, we need a clock to ensure that it occurs in the interval $[1, 2]$, relatively to the corresponding occurrence of a. Under single-server and threshold semantics, the number of clocks is finite, since we have one clock per transition. Since $n_a(d) - n_b(d)$ can grow to infinity, the translation into an equivalent TPN under single-server and threshold semantics should have an infinite number of transitions. This translation is then impossible. □

4 NDF vs FEFF firing choice policies

Consider a TPN $\mathcal{N}$ under multiple-server and threshold semantics. There are two main firing choice policies: non deterministic (NDF) and FEFF. We compare these two possibilities by supposing that the memory and disabling choice policies are the same in both cases. The results are however valid whatever these choices for memory and disabling policies are, thanks to the parametrization using CF.

For NDF firing choice, if several enabling instances of the same transition t are firable from a state $q = (M, \nu)$, then all these instances will be fired from q and the firing of one of them will not disable the others. Since NDF firing choice includes the FEFF one, we have the following obvious lemma.

Lemma 1. *The NDF firing choice strongly timed simulates the FEFF firing choice.*

Proof. Let $\mathcal{S}_1 =< Q_1, q_0, \Sigma, \rightarrow_1>$ and $\mathcal{S}_2 =< Q_2, q_0, \Sigma, \rightarrow_2>$ be the transition systems of $\mathcal{N}$ under multiple-server and threshold semantics for NDF and FEFF firing choice policies, respectively. Since NDF firing choice includes the FEFF one, it follows that $\mathcal{Q}_2 \subseteq \mathcal{Q}_1$ and $\rightarrow_2 \subseteq \rightarrow_1$. Therefore, the NDF firing choice strongly simulates the FEFF firing choice. □

However, for the NDF firing choice policy, all enabling instances of the same transition will be fired in different orders, which may cause a blow-up of the state space. It would be interesting if we can consider only one firing order without loosing properties of the model. In this sense, we show in the following that under FEFF or NDF firing choice semantics, $\mathcal{N}$ has the same timed language. The proof of this claim is based on the strong timed simulation relation over states $\mathcal{Q}$ of $\mathcal{N}$ defined in subsection 4.1.

4.1 FEFF simulates NDF firing choice policy

We first consider a TPN $\mathcal{N}$ under multiple-server and threshold semantics with NDF firing choice policy.

Let $\preceq$ be the relation over states $\mathcal{Q}$ of the TPN $\mathcal{N}$ defined by:
$\forall (M, \mathsf{en}, \nu), (M', \mathsf{en}', \nu') \in \mathcal{Q}, (M, \mathsf{en}, \nu) \preceq (M', \mathsf{en}', \nu')$ iff $M = M'$, $\mathsf{en} = \mathsf{en}'$ and

$$\forall t^i \in \mathsf{en}, \nu(t^i) = \nu'(t^i) \quad \text{or} \quad \downarrow Is(t) \leq \nu'(t^i) < \nu(t^i).$$

Lemma 2. *The relation $\preceq$ is a strong timed simulation.*

Proof. It suffices to show that:
$\forall (M, \mathsf{en}, \nu) \in \mathcal{Q}, \forall (M', \mathsf{en}', \nu') \in \mathcal{Q}$ s.t. $(M, \mathsf{en}, \nu) \preceq (M', \mathsf{en}', \nu')$,
$\forall d \in \mathbb{R}^+, \forall t^f \in \mathsf{en}$,
(i) $(M, \mathsf{en}, \nu) \xrightarrow{d} (M, \mathsf{en}, \nu + d) \Rightarrow$

$$((M', \mathsf{en}', \nu') \xrightarrow{d} (M', \mathsf{en}', \nu' + d) \text{ and } (M, \mathsf{en}, \nu + d) \preceq (M', \mathsf{en}', \nu' + d))$$

(ii) $\exists (M_1, \mathsf{en}_1, \nu_1) \in \mathcal{Q}, (M, \mathsf{en}, \nu) \xrightarrow{t^f} (M_1, \mathsf{en}_1, \nu_1) \Rightarrow$

$$(\exists (M'_1, \mathsf{en}'_1, \nu'_1) \in \mathcal{Q}, (M', \mathsf{en}', \nu') \xrightarrow{t^f} (M'_1, \mathsf{en}'_1, \nu'_1) \text{ and}$$
$$(M_1, \mathsf{en}_1, \nu_1) \preceq (M'_1, \mathsf{en}'_1, \nu'_1))$$

Proof of (i): $(M, \mathsf{en}, \nu) \xrightarrow{d} (M, \mathsf{en}, \nu + d)$ iff $\forall t^i \in \mathsf{en}, \nu(t^i) + d \leq \uparrow Is(t)$. By assumption, $(M, \mathsf{en}, \nu) \preceq (M', \mathsf{en}', \nu')$, which means that:
(1) $M = M'$, $\mathsf{en} = \mathsf{en}'$ and

$$\forall t^i \in \mathsf{en}, \nu'(t^i) = \nu(t^i) \ \text{ or } \downarrow Is(t) \leq \nu'(t^i) < \nu(t^i).$$

Therefore, $\forall t^i \in \mathsf{en}, \nu'(t^i) \leq \nu(t^i)$. It follows that $M = M'$, $\mathsf{en} = \mathsf{en}'$ and $\forall t^i \in \mathsf{en}, \nu'(t^i) + d \leq \nu(t^i) + d \leq \uparrow Is(t)$ and then

$$(M', \mathsf{en}', \nu') \xrightarrow{d} (M', \mathsf{en}', \nu' + d).$$

Moreover, (1) implies that $M = M'$, $\mathsf{en} = \mathsf{en}'$,

$$\forall t^i \in \mathsf{en}, \nu'(t^i) + d = \nu(t^i) + d \ \text{ or } \downarrow Is(t) \leq \downarrow Is(t) + d \leq \nu'(t^i) + d < \nu(t^i) + d.$$

and then $(M, \mathsf{en}, \nu + d) \preceq (M', \mathsf{en}', \nu' + d)$.

Proof of (ii): $(M, \mathsf{en}, \nu) \xrightarrow{t^f} (M_1, \mathsf{en}_1, \nu_1)$ and $(M, \mathsf{en}, \nu) \preceq (M', \mathsf{en}', \nu')$ imply that $M = M'$, $\mathsf{en} = \mathsf{en}'$ and $t^f \in \mathsf{en}, \nu(t^f) \geq \downarrow Is(t)$ and $(\nu'(t^f) = \nu(t^f)$ or $\downarrow Is(t) \leq \nu'(t^f) < \nu(t^f))$. Therefore, $t^f \in \mathsf{en}', \nu'(t^f) \geq \downarrow Is(t)$ and then $\exists (M'_1, \mathsf{en}'_1, \nu'_1) \in \mathcal{Q}, (M', \mathsf{en}', \nu') \xrightarrow{t^f} (M'_1, \mathsf{en}'_1, \nu'_1)$.
Moreover, it holds that $M_1 = M'_1$, $\mathsf{en}_1 = \mathsf{en}'_1$ and $\forall t'^i \in \mathsf{en}_1$,
if $t'^i \in \mathsf{Nw}(M, \mathsf{en}, t^f)$, $\nu_1(t'^i) = 0, \nu'_1(t'^i) = 0$.
Otherwise, $\nu_1(t'^i) = \nu(t'^{i_o}), \nu'_1(t'^i) = \nu'(t'^{i_o})$, t'^{i_o} being the reference in en to t'^i.
Therefore, $\forall t'^i \in \mathsf{en}_1$,

$$\nu_1(t'^i) = \nu'_1(t'^i) = 0 \text{ or } \downarrow Is(t) \leq \nu'_1(t'^i) = \nu'(t'^{i_o}) < \nu(t'^{i_o}) = \nu_1(t'^i).$$

The relation $\preceq$ is then a strong timed simulation over states of the model, whatever the intermediate/atomic/persistent atomic semantics. □

Let us now show that from the same state (M, en, ν), the states reached by firing two enabling instances of the same transition are s.t. the one reached by the older enabling instance strongly simulates the other one. It means that applying FEFF firing choice will preserve the timed traces of (M, en, ν) obtained by NDF firing choice, whatever the intermediate/atomic/persistent atomic semantics.

Lemma 3. *Let* (M, en, ν), $(M_f, \mathsf{en}_f, \nu_f)$ *and* $(M_g, \mathsf{en}_g, \nu_g)$ *be three states,* t^f *and* t^g *two distinct enabling instances of the same transition* t *in* M *s.t.*

$$(M, \mathsf{en}, \nu) \xrightarrow{t^f} (M_f, \mathsf{en}_f, \nu_f),\ (M, \mathsf{en}, \nu) \xrightarrow{t^g} (M_g, \mathsf{en}_g, \nu_g) \quad \textit{and} \quad \nu(t^f) \geq \nu(t^g).$$

Then $(M_f, \mathsf{en}_f, \nu_f) \preceq (M_g, \mathsf{en}_g, \nu_g)$.

Proof. Since t^f and t^g are two enabling instances of the same transition t, it follows that (see Property 1): $M_f = M_g$, $\mathsf{CF}(M, \mathsf{en}, t^f) - \{t^f\} = \mathsf{CF}(M, \mathsf{en}, t^g) - \{t^g\}$, $\mathsf{Nw}(M, \mathsf{en}, t^f) = \mathsf{Nw}(M, \mathsf{en}, t^g)$, and $t^f \in \mathsf{CF}(M, \mathsf{en}, t^f) \Leftrightarrow t^g \in \mathsf{CF}(M, \mathsf{en}, t^g)$.
The list en_f is obtained from en by eliminating enabling instances of $\mathsf{CF}(M, \mathsf{en}, t^f)$ and adding enabling instances of $\mathsf{Nw}(M, \mathsf{en}, t^f)$. Similarly, the list en_g is obtained from en by eliminating instances of $\mathsf{CF}(M, \mathsf{en}, t^g)$ and adding instances of $\mathsf{Nw}(M, \mathsf{en}, t^g)$. Let us consider two cases: $t^f \notin \mathsf{CF}(M, \mathsf{en}, t^f)$ (which may hold for the persistent atomic semantics) and $t^f \in \mathsf{CF}(M, \mathsf{en}, t^f)$ (which always holds for the intermediate and atomic semantics).
1) If $t^f \notin \mathsf{CF}(M, \mathsf{en}, t^f)$ (i.e., $t^g \notin \mathsf{CF}(M, \mathsf{en}, t^g)$) then $\mathsf{CF}(M, \mathsf{en}, t^f) = \mathsf{CF}(M, \mathsf{en}, t^g)$ and $\mathsf{Nw}(M, \mathsf{en}, t^f) = \mathsf{Nw}(M, \mathsf{en}, t^g)$. It follows that $en_f = en_g$ and $\nu_f = \nu_g$.
2) If $t^f \in \mathsf{CF}(M, \mathsf{en}, t^f)$ (i.e., $t^g \in \mathsf{CF}(M, \mathsf{en}, t^g)$) then $\mathsf{CF}(M, \mathsf{en}, t^f) - \{t^f\} = \mathsf{CF}(M, \mathsf{en}, t^g) - \{t^g\}$ and $\mathsf{Nw}(M, \mathsf{en}, t^f) = \mathsf{Nw}(M, \mathsf{en}, t^g)$. It follows that the lists en_f and en_g are equal. In addition, both ν_f and ν_g are obtained by eliminating the same set of clocks values, except those of the fired instances. In ν_f, the clock value of the fired instance t^f is eliminated but the one of t^g is kept. Similarly, in ν_g, the clock value of the fired instance t^g is eliminated but the one of t^f is kept. In lists en_f and en_g, transitions appears in increasing order and the enabling instances of the same transition are ordered from the oldest to the newest one. Let g' and f', with $f' < g'$, be the positions in en_f and en_g of the enabling instances t^g and t^f of en, respectively. The enabling instances between positions f' and g' are all firable instances of t. Then:

$$\forall t'^i \in \mathsf{en}_f, \nu_f(t'^i) = \begin{cases} \nu_g(t'^i) & \text{if } i < f' \\ \nu_g(t'^{i+1}) \leq \nu_g(t'^i) & \text{if } f' \leq i < g' \\ \nu_g(t'^i) & \text{otherwise} \end{cases}.$$

As t^f and t^g are firable from (M, en, ν), it follows that:
$\forall i \in [f', g'[, \downarrow Is(t) \leq \nu_f(t^i) \leq \nu_g(t^i)$.
Consequently, $\forall t'^i \in \mathsf{en}_f, \nu_f(t'^i) = \nu_g(t'^i) \vee \downarrow Is(t') \leq \nu_f(t'^i) \leq \nu_g(t'^i)$.
Then: $(M_f, \mathsf{en}_f, \nu_f) \preceq (M_g, \mathsf{en}_g, \nu_g)$. □

We can now state the main result of this section:

Theorem 4. *Let* $\mathcal{N}$ *be a TPN under multiple-server and threshold semantics. The FEFF firing choice policy of* $\mathcal{N}$ *strongly timed simulates the NDF firing choice policy.*

Proof. Let $\mathcal{S}_1 =< Q_1, q_0, \Sigma, \rightarrow_1>$ and $\mathcal{S}_2 =< Q_2, q_0, \Sigma, \rightarrow_2>$ be the transition systems of $\mathcal{N}$ under multiple-server and threshold semantics for NDF and FEFF firing choice policies, respectively. $\mathcal{S}_1$ and $\mathcal{S}_2$ have the same initial state $q_0 = (M_0, \mathsf{en}_0, , \nu_0)$. The FEFF firing choice means that the oldest enabling instance of the same transition is fired first. From the initial state $q_0 = (M_0, \mathsf{en}_0, \nu_0)$, we have the following relationships between $\mathcal{S}_1$ and $\mathcal{S}_2$:

1) $\forall d \geq 0, (M_0, \mathsf{en}_0, \nu_0) \xrightarrow{d}_1 (M_0, \mathsf{en}_0, \nu_0{+}d) \Leftrightarrow (M_0, \mathsf{en}_0, \nu_0) \xrightarrow{d}_2 (M_0, \mathsf{en}_0, \nu_0{+}d)$.

2) $\forall t^g \in \mathsf{en}_0, \quad (M_0, \mathsf{en}_0, \nu_0) \xrightarrow{t^g}_1 (M_g, \mathsf{en}_g, \nu_g) \ \Rightarrow$

$$\exists t^f \in \mathsf{en}_0 \text{ s.t. } \nu(t^f) \geq \nu(t^g), (M_0, \mathsf{en}_0, \nu_0) \xrightarrow{t^f}_2 (M_f, \mathsf{en}_f, \nu_f).$$

Using Lemma 3, we can state that $(M_f, \mathsf{en}_f, \nu_f) \preceq (M_g, \mathsf{en}_g, \nu_g)$.

Inductively and thanks to Lemma 2 and Lemma 3, from any states $(M, \mathsf{en}, \nu_1) \in Q_1$ and $(M, \mathsf{en}, \nu_2) \in Q_2$ such that $(M, \mathsf{en}, \nu_1) \preceq (M, \mathsf{en}, \nu_2)$, we have:

1) $\forall d \geq 0, \quad (M, \mathsf{en}, \nu_1) \xrightarrow{d}_1 (M, \mathsf{en}, \nu_1 + d) \ \Rightarrow \ (M, \mathsf{en}, \nu_2) \xrightarrow{d}_2 (M, \mathsf{en}, \nu_2 + d)$.

2) $\forall t^g \in \mathsf{en}, \quad (M, \mathsf{en}, \nu_1) \xrightarrow{t^g}_1 (M_g, \mathsf{en}, \nu_g) \ \Rightarrow \ \exists t^f \in \mathsf{en}$ s.t.

$\nu(t^f) \geq \nu(t^g), (M, \mathsf{en}, \nu_2) \xrightarrow{t^f}_2 (M_f, \mathsf{en}_f, \nu_f)$ and $(M_f, \mathsf{en}_g, \nu_f) \preceq (M_g, \mathsf{en}_g, \nu_g)$.

Therefore, $\mathcal{S}_2$ strongly timed simulates $\mathcal{S}_1$. □

4.2 FEFF vs NDF wrt timed language

We can now extend the previous results to timed language acceptance consideration.

Theorem 5. *Let $\mathcal{N}$ be a TPN under multiple-server and threshold semantics. The timed language of $\mathcal{N}$ is the same for both NDF and FEFF firing choice policies.*

Proof. Let $\mathcal{S}_1 = < Q_1, q_0, \Sigma, \rightarrow_1 >$ and $\mathcal{S}_2 = < Q_2, q_0, \Sigma, \rightarrow_2 >$ be the transition systems of $\mathcal{N}$ under multiple-server and threshold semantics for NDF and FEFF choice policies, respectively. Lemma 1 states that $\mathcal{S}_1$ strongly timed simulates $\mathcal{S}_2$. Theorem 4 states that $\mathcal{S}_2$ strongly timed simulates $\mathcal{S}_1$. Therefore, $\mathcal{S}_1$ and $\mathcal{S}_2$ have the same timed language. □

4.3 FEFF vs NDF wrt timed bisimulation

The previous subsections prove that for the multiple-server and threshold semantics, there is a strong timed co-simulation between the NDF and FEFF firing choice policies. We will now show that this strong timed co-simulation is not a timed bisimulation.

Let us consider the TPN $\mathcal{N}$ of Fig. 5. Let $\mathcal{S}_1 = < Q_1, q_0, \Sigma, \rightarrow_1 >$ and $\mathcal{S}_2 = < Q_2, q_0, \Sigma, \rightarrow_2 >$ be the transition systems of $\mathcal{N}$ under multiple-server and threshold semantics for NDF and FEFF choice policies, respectively.

We consider the following run. At date 1, we fire the transition a and then we let 1 more time unit elapse. For both firing choice policies, this run is possible and leads to the same state q_3, where there are two tokens in place P_2 with different

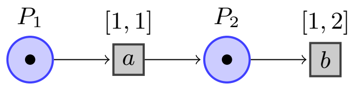

Fig. 5. A TPN $\mathcal{N}$

ages 1 and 2: $q_0 \xrightarrow{1} q_1 \xrightarrow{a} q_2 \xrightarrow{1} q_3$ with $\rightarrow \in \{\rightarrow_1, \rightarrow_2\}$. From this state, with the NDF firing choice policy, both enabling instances of b are fired. The firing of the instance of b with the youngest token of P_2 leads to the state q_4 ($q_3 \xrightarrow{b}_1 q_4$). To mime this behavior, from q_3, the FEFF choice policy has to fire the transition b with the oldest token of P_2 leading to the state q_5 ($q_3 \xrightarrow{b}_2 q_5$). Now from state q_5, the FEFF firing choice policy allows to wait 1 time unit ($q_5 \xrightarrow{1}_2 q_6$), whereas the transition b must fire immediately with the NDF firing choice policy from q_4 ($q_4 \xrightarrow{b}_1 q_7$). Then, q_4 and q_5 are not strongly timed bisimilar.

4.4 Discussion

The NDF firing choice policy is the more conservative safety-wise choice and corresponds in general to the semantics we want to use for model checking or control purposes. However, since all firable instances may fire in any order, the number of runs may lead to a state space explosion. To avoid to compute all runs, the simpler alternative, First Enabled First Fired (FEFF), where only the oldest instance may fire, leads to a much more compact state-space. We show with the following example of Fig. 6 how FEFF generates fewer runs than NDF but preserves all the timed traces of NDF.

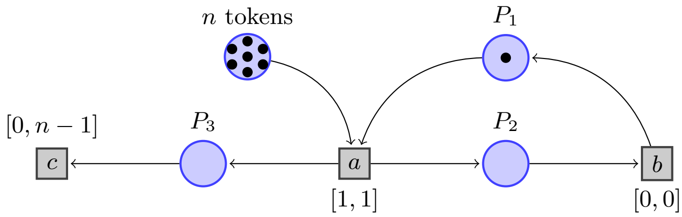

Fig. 6. A TPN illustrating the interest of the FEFF choice policy

We consider only the runs producing the trace $(ab)^n\omega$ of duration n. Since the prefix $(ab)^n$ takes also n time units, the duration of ω is 0. In all runs, the only state reachable after this prefix is q_1 with a marking with one token in P_1, n tokens in P_3 and no token in other places. As a shorthand and since the markings will now change only in place P_3, we can denote the state by the age of each n tokens enabling c. For q_1, the oldest token is $c^1 = n-1$ and the newest is $c^n = 0$ and $q_1 = \{c^1 = n-1, c^2 = n-2, c^3 = n-3 \ldots, c^n = 0\}$. All the instances of c are firable from q_1 since the timed interval of c is $[0, n-1]$. For FEFF, we obtain one only run by firing c^1 then c^2 then c^3 ... then c^n whereas for NDF we obtain all the $n!$ combinations of all the instances leading to the same state q_f. We illustrate these runs with $n = 3$ in Figure 7. All these $n!$ runs of NDF produce the same trace $(ab)^n(c)^n$ where $\omega = (c)^n$ is in null time, which is also the trace produced by the only run of FEFF.

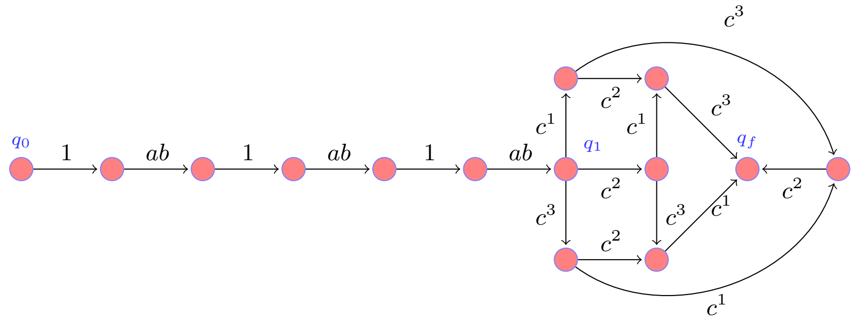

Fig. 7. Runs with NDF

5 Conclusion

In this paper, we first presented and discussed different semantics of multi-enabledness for time Petri nets. Then, we considered the threshold semantics in both contexts single-server and multiple-server semantics and investigated some questions relative to the expressiveness. We showed that multiple-server semantics adds expressiveness, in terms of timed bisimulation, relatively to single-server semantics.

For the multiple-server semantics, different firing choice policies may be used to manage the multiple instances of the same transition such as Non-Deterministic (NDF) choice and First Enabled First Fired (FEFF) choice. In the NDF firing choice, the firable instances of the same transition are fired in all possible orders, which may cause a blow-up of the state space. In the FEFF choice, these instances are fired in only one order: from the oldest ones to the more recent ones. We proved that both semantics are not bisimilar but actually simulate each other with strong timed simulations, which in particular implies that they generate the same timed traces. Consequently, a TPN under NDF or FEFF policies has the same linear (timed) properties but FEFF leads to more compact state space. FEFF is then very appropriate to deal with linear timed properties of time Petri nets.

As immediate perspective, we will investigate the extension of the results established in this paper, to the case of age-semantics.

References

1. T. Aura and J. Lilius. A causal semantics for time petri nets. *Theoretical Computer Science*, 243(2):409–447, 2000.
2. B. Bérard, F. Cassez, S. Haddad, D. Lime, and O. H. Roux. Comparison of different semantics for time Petri nets. In *Automated Technology for Verification and Analysis (ATVA'05)*, volume 3707 of *Lecture Notes in Computer Science*, Taiwan, Oct. 2005. Springer.
3. B. Bérard, F. Cassez, S. Haddad, D. Lime, and O. H. Roux. The expressive power of time Petri nets. *Theoretical Computer Science*, 474:1–20, 2013.
4. B. Berthomieu. La méthode des classes d'états pour l'analyse des réseaux temporels - mise en oeuvre, extension à la multi-sensibilisation. In *Modélisation des Systèmes Réactifs, (MSR'01)*, pages 275–290, Toulouse, France, October 2001. Hermes.
5. H. Boucheneb, A. Bullich, and O. H. Roux. FIFO Time Petri Nets for conflicts handling. In *11th International Workshop on Discrete Event Systems (WODES'12)*, Guadalajara, Mexico, Oct. 2012. IFAC.
6. M. Boyer and M. Diaz. Multiple enabledness of transitions in time Petri nets. In *Proc. of the 9th IEEE International Workshop on Petri Nets and Performance Models*, Aachen, Germany, September 11–14 2001. IEEE Computer Society.
7. M. Boyer and O. H. Roux. On the compared expressiveness of arc, place and transition time Petri nets. *Fundamenta Informaticae*, 88(3):225–249, 2008.
8. A. Cerone and A. Maggiolo-Schettini. Timed based expressivity of time Petri nets for system specification. *Theoretical Computer Science*, 216:1–53, 1999.
9. R. Glabbeek and G. Plotkin. Configuration structures. In *Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science, (LICS'95)*, pages 199–209, San Diego, USA, 1995. IEEE Computer Society Press.
10. W. Khansa, J.-P. Denat, and S. Collart-Dutilleul. P-time Petri nets for manufacturing systems. In *International Workshop on Discrete Event Systems, WODES'96*, pages 94–102, Edinburgh (U.K.), august 1996.
11. P. M. Merlin. *A study of the recoverability of computing systems*. PhD thesis, Department of Information and Computer Science, University of California, Irvine, CA, 1974.
12. C. J. Myers, T. G. Rokicki, and T. H.-Y. Meng. Poset timing and its application to the timed cirucits. *IEEE Transactions on CAD*, 18(6), 1999.
13. C. Ramchandani. *Analysis of asynchronous concurrent systems by timed Petri nets*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1974. Project MAC Report MAC-TR-120.
14. P.-A. Reynier and A. Sangnier. Weak time petri nets strike back! In M. Bravetti and G. Zavattaro, editors, *20th International Conference on Concurrency Theory (CONCUR'09)*, volume 5710 of *Lecture Notes in Computer Science*, pages 557–571, Bologna, Italy, Sept. 2009. Springer.
15. B. Walter. Timed net for modeling and analysing protocols with time. In *Proceedings of the IFIP Conference on Protocol Specification Testing and Verification*, North Holland, 1983.