Chapter 16
Petri Nets with Time

Béatrice Bérard, Maria Paola Cabasino, Angela Di Febbraro, Alessandro Giua, and Carla Seatzu

16.1 Introduction and Motivation

Place/Transition nets have been used in previous chapters to model Discrete Event Systems (DESs) with the aim of analyzing logical properties. However, since they do not consider the duration associated with the activities occurring in a system, they cannot be used for performance analysis of a DES, i.e., for computing the execution time of a given process, identifying bottlenecks, optimizing the use of resources, and so on. Petri nets with time are an extension of Place/Transition (P/T) nets endowed with a timing structure and can be used as performance models.

When defining Petri nets with time, three main elements should be specified: *topological structure*, *timing structure*, and *transition firing rules*. While the topological structure is generally that of a P/T net, the definition of the timing structure is a crucial problem: several timing structures have been proposed in the literature to extend P/T nets and firing rules are also based on them.

The chapter is structured as follows. Next section briefly describes the timing structures and other basic concepts related to Petri nets with time. Starting with Section 16.3 we focus on T-Timed Petri nets, the most commonly used class of Petri nets with time, and discuss different firing rules that can be used in this context. In Sections 16.4 and 16.5 we present several results related to deterministic

Béatrice Bérard
LIP6, Univ. P. et M. Curie, France
e-mail: Beatrice.Berard@lip6.fr

Maria Paola Cabasino · Alessandro Giua · Carla Seatzu
Department of Electrical and Electronic Engineering, University of Cagliari, Italy
e-mail: {cabasino,giua,seatzu}@diee.unica.it

Angela Di Febbraro
Department of Mechanical Engineering, Energetics, Production,
Transportation and Mathematical Models, University of Genova, Italy
e-mail: angela.difebbraro@unige.it

C. Seatzu et al. (Eds.): Control of Discrete-Event Systems, LNCIS 433, pp. 319–341.
springerlink.com

and stochastic T-Timed Petri nets. Section 16.6 deals with a different class of Petri nets with time, called T-Time Petri nets. Further readings are finally suggested in Section 16.7.

16.2 Timing Structure and Basic Concepts

In this section, we point out a few general issues associated with a timing structure that can be associated with Petri nets.

A P/T net is a *logical* DES model, and a possible evolution of a net is described by a sequence

$$\boldsymbol{m}_0\ [t_{j_1}\rangle\ \boldsymbol{m}_1\ [t_{j_2}\rangle\ \boldsymbol{m}_2\ [t_{j_3}\rangle\ \boldsymbol{m}_3\ \ldots\ \boldsymbol{m}_{k-1}\ [t_{j_k}\rangle\ \boldsymbol{m}_k\ \ldots$$

of markings (i.e., states) $\boldsymbol{m}_k$ (for $k = 0, 1, 2 \ldots$) and transitions (i.e., events) t_{j_k} (for $k = 1, 2, 3 \ldots$). Marking $\boldsymbol{m}_0$ is the initial marking and the firing of transition t_{j_k} changes the marking from $\boldsymbol{m}_{k-1}$ to $\boldsymbol{m}_k$.

In a PN with time the evolution of a system initialized at time τ_0 is described by a sequence

$$\boldsymbol{m}_0\ [t_{j_1}, \tau_1\rangle\ \boldsymbol{m}_1\ [t_{j_2}, \tau_2\rangle\ \boldsymbol{m}_2\ [t_{j_3}, \tau_3\rangle\ \boldsymbol{m}_3\ \ldots\ \boldsymbol{m}_{k-1}\ [t_{j_k}, \tau_k\rangle\ \boldsymbol{m}_k\ \ldots$$

where $\tau_k \geq \tau_{k-1}$ and τ_k denotes the *firing time* of transition t_{j_k} (for $k = 1, 2, 3 \ldots$) or equivalently

$$\boldsymbol{m}_0\ [t_{j_1}, \theta_1\rangle\ \boldsymbol{m}_1\ [t_{j_2}, \theta_2\rangle\ \boldsymbol{m}_2\ [t_{j_3}, \theta_3\rangle\ \boldsymbol{m}_3\ \ldots\ \boldsymbol{m}_{k-1}\ [t_{j_k}, \theta_k\rangle\ \boldsymbol{m}_k\ \ldots$$

where $\theta_k = \tau_k - \tau_{k-1}$ (for $i = 2, 3, 4 \ldots$) denotes the *delay* between the firing of transition $t_{j_{k-1}}$ and t_{j_k} and $\theta_1 = \tau_1 - \tau_0$ denotes the *delay* between the firing of transition t_{j_1} and the initial time.

A *timing structure* specifies the value that these delays may take.

16.2.1 Timed Elements

Although, in a timed evolution the delays denote the time elapsed between the firing of two transitions, from a structural point of view a delay can be associated with different elements of a net, such as places, transitions, or arcs.

As an example, consider the simple net in Fig. 16.1. Assume the systems behavior is such that the firing of transition t should occur $\theta = 3$ seconds after the initial time. This can be done associating a delay θ with one of the following elements.

- *Place* p_1: this denotes that the token in the place becomes available for transition firings only after it has been in the place for θ seconds.

- *Transition* t: this denotes that the transition will fire only after it has been enabled for θ seconds.
- *Arc* (p_1, t): this denotes that the token in the place becomes available for this arc only after the token has reached an age of θ seconds (assuming its age at the initial time was 0).

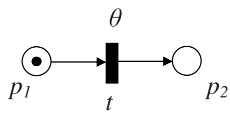

Fig. 16.1 A simple Petri net

In the rest of the chapter, we will assume that the timing structure associates delays to transitions, and we denote θ_i the delay associated with transition t_i.

16.2.2 Timed Petri Nets and Time Petri Nets

Another significant difference is among *Timed Petri nets* and *Time Petri nets*.

In *Timed Petri nets* (TdPNs) a delay is represented by a single value θ. As an example, consider a net with delays associated with transitions: if a transition becomes enabled at time τ and remains enabled henceforth, it must fire at time $\tau + \theta$.

In *Time Petri nets* (TPNs) a delay is represented by a time interval of the form $[l, u]$, where $l \in \mathbb{R}_{\geq 0}$, $u \in \mathbb{R}_{\geq 0} \cup \{+\infty\}$, and $l \leq u$. As an example, consider a net where interval $[l, u]$ is associated with a transition: if the transition becomes enabled at time τ and remains enabled henceforth, it cannot fire before time $\tau + l$ and it must fire at latest at time $\tau + u$.

When it is required to specify that delays are associated with transitions, one speaks of *T-Timed Petri nets* and *T-Time Petri nets*. On the contrary, when delays are associated with places one speaks of *P-Timed Petri nets* and *P-Time Petri nets*.

16.2.3 Deterministic and Stochastic Nets

The timing structure of a net can be: *deterministic*, when the delays are known *a priori*, or *stochastic*, when the delays are random variables.

Consider, as an example, the class of T-Timed Petri nets to which most of this chapter is dedicated. According to the nature of the associated delay, timed transitions can be classified as follows.

Definition 16.1. *A transition t_i of a T-Timed Petri net is called:*

- Immediate*, if it fires as soon as it is enabled, or equivalently, if its time delay is null.*
- Deterministic*, if the delay θ is chosen deterministically. Note that the deterministic delay may be a* constant *value θ_i, may be* variable *according to a sequence $\{\theta_{i,1}, \theta_{i,2}, \theta_{i,3}, \ldots\}$ of delay times known a priori, and finally may also be* marking dependent.
- Stochastic*, if the delay time θ_i is a random variable with a known probability distribution.*
  If the delay θ_i has an exponential distribution $f_i(\tau) = \lambda_i e^{-\lambda_i \tau}$ (with $\lambda_i > 0$) transition t_i is called stochastic exponential. *If the delay is a random variable with a distribution different from the exponential one the transition is called* generalized stochastic. *Finally if the parameters of the distribution depend on current marking of the net, the transition is called* stochastic marking dependent.

In this chapter only immediate, deterministic constant, and exponential stochastic are considered. Therefore, in the following, the last two types of transitions are briefly called deterministic and stochastic, respectively. In the most general case, the same Petri net may contain transitions of all three types mentioned above (immediate, deterministic, and stochastic); however, this increases the analysis complexity and very few analysis results exist for such a general net.

In Fig. 16.2 different PNs are shown. A deterministic transition t_i is represented by a black rectangle and is labeled with the value of its constant delay θ_i. A stochastic timed transition t_i is represented by a white rectangle and is labeled with the value of its parameter λ_i. An immediate transition is represented by a black bar with no label.

For a detailed comparison of the various timing mechanisms we refer to [6].

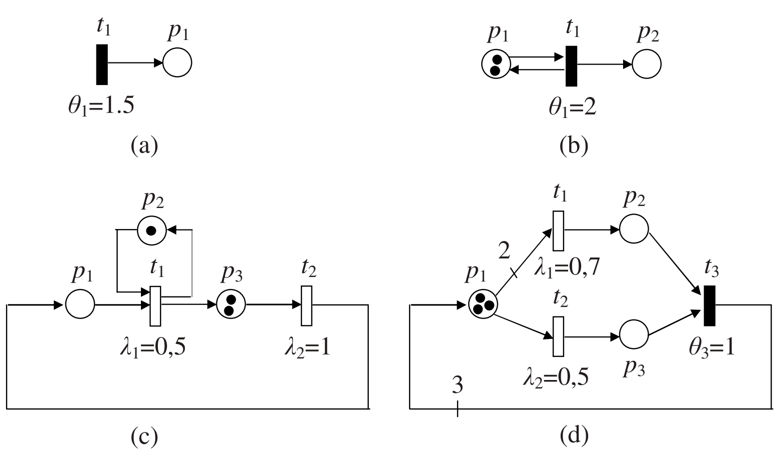

Fig. 16.2 T-Timed Petri nets

16.3 T-Timed Petri Nets and Firing Rules

In this section we focus on T-Timed Petri nets, that in the following we will simply call Timed Petri nets (TdPNs).

A series of “rules” or “conventions” should be specified in order to clarify the behavior of a given timed net. In the following subsections the most significant ones are discussed, with some comments on their expressive power.

16.3.1 Atomic vs. Non Atomic Firing

In a P/T the firing of a transition is assumed to be an *atomic* event, i.e., in the same instant the tokens that enable the transition are removed from the input places and new tokens are produced in the output places.

In the case of TdPNs this notion depends on the semantics given to the delay θ associated with a transition t.

- *The delay represents the time that must pass between the enabling and firing of a transition*. In this case, if transition t is enabled at time τ and remains enabled henceforth, it fires atomically at time $\tau+\theta$. If $\boldsymbol{m}$ is the marking before the firing, then the firing yields the new marking $\boldsymbol{m}' = \boldsymbol{m} - \boldsymbol{Pre}[\cdot,t] + \boldsymbol{Post}[\cdot,t]$.
- *The delay represents the time required to fire a transition*. In this case, assume marking $\boldsymbol{m}$ enabling transition t is reached at time τ. The transition starts its firing at τ and all tokens from the input places are removed, yielding marking $\hat{\boldsymbol{m}} = \boldsymbol{m} - \boldsymbol{Pre}[\cdot,t]$. At time $\tau+\theta$, the firing is completed producing the tokens in the output places thus yielding marking $\boldsymbol{m}' = \hat{\boldsymbol{m}} + \boldsymbol{Post}[\cdot,t]$. Such a firing policy is called *non atomic firing*.

Note that following the non atomic firing rule, the intermediate marking $\hat{\boldsymbol{m}}$ may not represent a reachable marking in the underlying P/T net and many of the analysis techniques for P/T nets, such as those based on invariants, do not apply. For this reason, we will only consider atomic firings in the rest of this chapter.

16.3.2 Enabling Semantics

Another important “rule” concerns the different strategies for the enabling of a transition.

- *Reserved marking:* as soon as a transition is enabled, the tokens of the input places of such a transition that are necessary to enable it, are reserved becoming completely invisible to all the other transitions. Moreover, in the case of actual conflict, tokens are immediately assigned to transitions, with a criterion that is in general independent of the length of their time delays.

- *Concurrent enabling:* tokens are always visible to all places and priority is given to the transition that first finishes of being enabled for a time period equal to its time delay. So, even if a transition starts being enabled later, but its time delay is small, it may happen that it fires before a transition that was enabled earlier but whose time delay is longer.

The strategy of concurrent enabling is more general than the one of reserved marking. In fact, it is always possible to transform the structure of a net following the strategy of reserved marking to an equivalent net based on concurrent enabling. This can be done using the simple scheme illustrated via the example in Fig. 16.3.

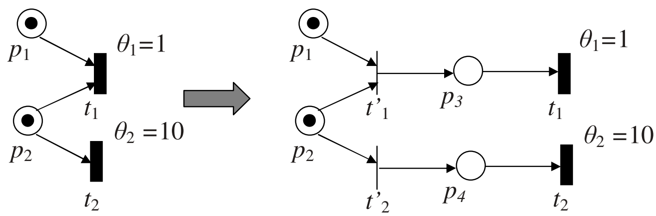

Fig. 16.3 Reserved marking strategy versus concurrent enabling strategy

16.3.3 Server Semantics

Another fundamental notion that needs to be specified when defining a TdPN is the so-called *server semantics*. Possible choices are described in the following:

- *infinite server semantics:* each transition represents an operation that can be executed by an infinite number of operation units that work in parallel; as an example, this is the case of the net in Fig. 16.4.*a*, where transition t_1 fires three times at time θ_1 because the operation units can use (process) all tokens simultaneously;
- *single server:* each transition represents an operation that can be executed by a single operation unit; an example of this is given in Fig. 16.4.*b*, where transition t_1 fires at time instants θ_1, $2\theta_1$, and $3\theta_1$ since the single operation unit can only consume (process) one token at a time;
- *multiple servers:* each transition represents an operation that can be executed by a finite number k of operation units; this is the case of the net in Fig. 16.4.*c*, where, assuming $k = 2$, transition t_1 fires twice at time θ_1 and once at time $2\theta_1$, since the two operation units can process only two tokens at a time.

In the rest of this chapter, we always assume infinite server semantics. In fact, starting from such a notion, it is possible to also represent the other two via appropriate places (as place p in Fig. 16.4.*b* and in Fig. 16.4.*c*), that limit the maximum enabling degree of the generic transition, as it will be explained in the following section.

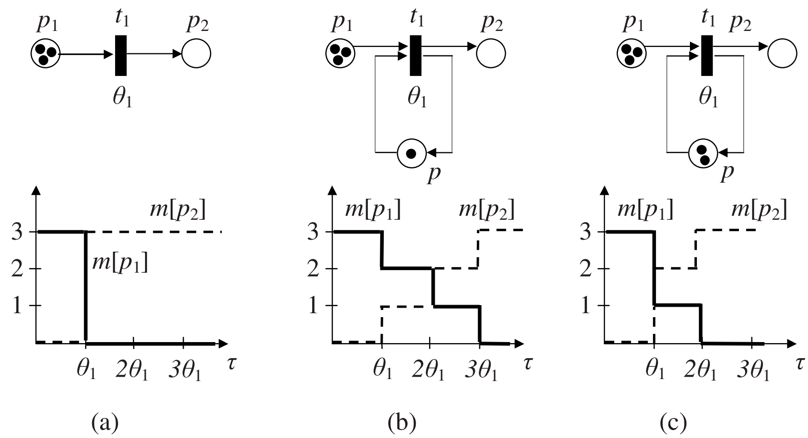

Fig. 16.4 Transitions with different server semantics

16.3.4 Memory Policy

Another notion that has to be specified concerns the *memory* associated with transitions. We have seen that a transition t_i can fire only if a time θ_i has elapsed since its enabling. Now, let us observe the net in Fig. 16.5; assuming $\theta_2 < \theta_1 < 2\theta_2$, from the initial marking (at time $\tau_0 = 0$) transitions t_1 and t_2 are enabled, thus at time $\tau_1 = \theta_2$ transition t_2 fires and yields to the marking $[0\ 0\ 1]^T$. After a delay equal to θ_3, i.e., at time $\tau_2 = \tau_1 + \theta_3$, transition t_3 fires and the net reaches again the initial marking. Two different notions of memory can be introduced.

1. *Total memory*: transition t_1 "remembers" being already enabled for a time interval equal to θ_2 and fires after a delay equal to $\theta_1 - \theta_2$, i.e., at time $\tau_3 = \tau_2 + (\theta_1 - \theta_2)$;
2. *enabling memory*: transition t_1 has only memory of the current enabling and can only fire after a delay equal to θ_1, i.e., at time $\tau_4 = \tau_2 + \theta_1$.

In the rest of this chapter, we consider as basic notion the enabling memory policy.

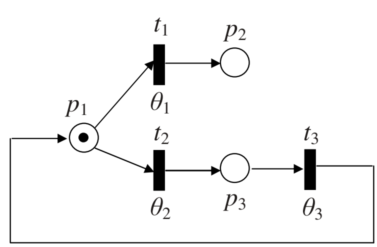

Fig. 16.5 Timed net with conflict

16.4 Deterministic Timed Petri Nets

The first extension of the P/T nets via deterministic delays has been presented in [21]. This approach uses timed transitions to address the idea of modeling the duration of activities of the represented DES, being in general the actions associated with the transitions. These nets are called *Deterministic Timed Transitions Petri Nets* or *Deterministic T-Timed Petri Nets*.

As discussed in Subsection 16.2.1, another timing structure is the one that assigns the time to places [11] that are seen as processes that require a given execution time. These nets, that are called *Deterministic Timed Places Petri Nets* or *Deterministic P-Timed Petri Nets*, represent an excellent applicative field of DES modeling approach based on *max-plus algebra* or *minimax* [2].

Finally there have been proposed also nets where the time is associated with arcs. As an example, Zhu and Denton [24] showed that such Petri nets are more general than those where the time is associated either with transitions or places.

In the rest of this section we focus on *Deterministic T-Timed Petri Nets* that are most commonly used in the literature. As a result of this, they are often called *Deterministic Timed Petri Nets* (DTdPN), without making explicit that delays are associated with transitions. Delays can be either constant or variable as clarified in the following definition.

Definition 16.2. *A deterministic timed Petri net is characterized by the algebraic structure* $N_d = (N, \Theta)$ *where:*

- $N = (P, T, \boldsymbol{Pre}, \boldsymbol{Post})$ *is a P/T net defined as in Definition 10.1 in Chapter 10;*
- $\Theta = \{\Theta_i : t_i \in T\}$*, with* $\Theta_i = \{\theta_{i,1}, \theta_{i,2}, \ldots\}$*,* $t_i \in T$*,* $\theta_{i,k} \in \mathbb{R}_+ \cup \{0\}$*,* $k \in \mathbb{N}_+$ *is a deterministic timing structure; if the time delays are constant, the generic element* $\theta_{i,k}$ *is denoted* θ_i*,* $\forall k \in \mathbb{N}_+$.

Even for timed Petri nets it is possible to define a marked Petri net. In general, the marking vector at the time instant τ_j is denoted $\boldsymbol{m}_j$.

Definition 16.3. *A deterministic timed Petri net* N_d *with a marking* $\boldsymbol{m}_0$ *at the initial time instant* τ_0 *is said a* marked deterministic timed Petri net*, and is denoted* $\langle N_d, \boldsymbol{m}_0 \rangle$.

16.4.1 Dynamical Evolution

The state of a DTdPN is determined not only by the marking, as for P/T nets, but also by the clocks associated with transitions.

Definition 16.4. *A transition* t_i *is* enabled *at a marking* $\boldsymbol{m}_j$ *if each place* $p \in P$ *of the net contains a number of tokens equal to or greater than* $\boldsymbol{Pre}[p, t_i]$*, i.e.,* $\boldsymbol{m}_j \geq \boldsymbol{Pre}[\cdot, t_i]$.

The enabling degree *of a transition t_i enabled at a marking $\boldsymbol{m}_j$ is the biggest integer number k such that $\boldsymbol{m}_j \geq k\, \boldsymbol{Pre}[\cdot, t_i]$. The enabling degree of t_i at $\boldsymbol{m}_j$ is denoted $\alpha_i(j)$.*

In the net in Fig. 16.2.*a*, transition t_1 has an infinite enabling degree; in the marked net in Fig. 16.2.*b*, transition t_1 has enabling degree equal to 2; in the marked net Fig. 16.2.*c*, transition t_1 is not enabled because p_1 is empty, while transition t_2 has enabling degree equal to 2; in the marked net in Fig. 16.2.*d*, transition t_1 has enabling degree equal to 1, because its firing needs 2 tokens and in its pre place p_1 there are only 3 tokens, transition t_2 has enabling degree equal to 3, while transition t_3 is not enabled. The case in which the enabling degree of a transition is infinite, as the case of Fig. 16.2.*a*, is a degenerate case. In the following, we assume that the enabling degree of a transition is finite (but not necessarily bounded).

At each time instant the number of clocks associated with a transition t_i is equal to its current enabling degree; this number changes with the enabling degree, thus it can change each time the net evolves from one marking to another one, namely each time that a transition fires.

The net evolution occurs in an asynchronous way on the basis of the events occurrence that is regulated by the clocks associated with the events according to the algorithm of evolution here reported.

Algorithm 16.1. (Temporal evolution of a DTdPN). *Assume that the DTdPN at the time instant τ_j is in the marking $\boldsymbol{m}_j$ and that the minimum values of the clocks associated with the transitions, $o_i = min\{o_{i,1}, \ldots, o_{i,\alpha_i(j)}\}$, $\forall t_i \in T$, are known; the marking evolution of the DTdPN follows the repetition of these steps:*

1. *Let o^**

$$o^* = \min_{i:t_i \in T} \{o_i\} \tag{16.1}$$

   be the minimum among the values of the clocks o_i associated with the transitions enabled at marking $\boldsymbol{m}_j$; if o^* is not unique more than one transition could fire at the same time according to a sequence that should be specified a priori.
2. *At the time instant $\tau_{j+1} = \tau_j + o^*$ transition t^* fires yielding the system from marking $\boldsymbol{m}_j$ to the marking $\boldsymbol{m}_{j+1} = \boldsymbol{m}_j + C[\cdot, t]$.*
3. *Once marking $\boldsymbol{m}_{j+1}$ is reached, the clock associated with t^* is discarded. Clocks associated with each transition $t_i \in T$ are updated as follows:*
   - *if the enabling degree $\alpha_i(j+1)$ at marking $\boldsymbol{m}_{j+1}$ is less than the enabling degree $\alpha_i(j)$ that transition t_i had at previous marking $\boldsymbol{m}_j$, then $[\alpha_i(j) - \alpha_i(j+1)]$ clocks associated with t_i have to be discarded: clocks that are discarded from set $\{o_{i,1}, \ldots, o_{i,\alpha_i(j)}\}$ are those having the higher values;*
   - *if $\alpha_i(j+1) > \alpha_i(j)$, $[\alpha_i(j+1) - \alpha_i(j)]$ new clocks are associated with t_i and initialized to the values specified by the timing structure Θ;*
   - *if $\alpha_i(j+1) = \alpha_i(j)$, do nothing;*
   - *reduce to an amount equal to o^* the values of all old clocks.*
4. *Repeat from step 1, setting $j+1 \rightarrow j$.*

Note that if transition t_i is not enabled at a marking, it has no clocks associated with it, i.e., it has no *active* clocks. If at a marking $\boldsymbol{m}_j$ the minimum value of the clock o_i of a transition t_i corresponds to more than one clock, as an example k, in the set $\{o_{i,1},\dots,o_{i,\alpha_i(j)}\}$, this means that if the transition will be the next one to fire, it will fire k times at the same time.

The algorithm is based on the assumption of *enabling memory* and *infinite server semantics*. If *total memory* is used, step 2 of the algorithm should be modified. Note that the memory chosen depends on the kind of study one wants to do on the system while does not depend on the system itself. For the server semantics we have chosen the most general: in fact one can always lead the system to single or multiple servers adding self loops to transitions, as shown in Fig. 16.4.*b* and 16.4.*c*. Obviously, the algorithm is simplified if all transitions have single server semantics, because in such a case each transition has associated one single clock.

Finally, at step 1 it is said that a politic for the *resolution of conflicts* has to be applied if more than one transition can fire at the same time, to decide the sequence in which these transitions will fire. This is needful only when the firing of one transition can disable other transitions, namely when the net is not *persistent*. Either a firing priority or a firing probability can be associated with transitions.

Example 16.2. Let us consider the net in Fig. 16.2.*b* whose temporal evolution is shown in Fig. 16.6.

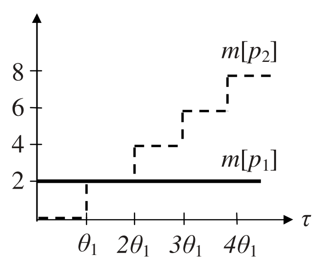

Fig. 16.6 Evolution of the net in Fig. 16.2.*b*

Transition t_1 (with time delay $\theta_1 = 2$) has enabling degree $\alpha_1(0)$ equal to 2 at the initial time instant $\tau_0 = 0$; it has initially two active clocks $o_{1,1}$, and $o_{1,2}$. After two time instants, i.e., at $\tau_1 = \tau_0 + \theta_1$, both clocks are deleted, transitions t_1 fires twice and the two clocks are again set to $\theta_1 = 2$. At the reached marking $\boldsymbol{m}_1 = [2\ 2]^T$, transition t_1 still has an enabling degree equal to 2 and the two clocks $o_{1,1}$, and $o_{1,2}$ are again active. The net continues to evolve following Algorithm 16.1. ■

Example 16.3. A production line is composed by two machines M_1 and M_2, two robotics arms R_1 and R_2, and two conveyor belts. Each machine uses one robotic arm that loads and unloads parts that the machine has to process. One of the conveyor belt can carry only two parts, while the other one carries empty pallets. Pallets in the system are three.

Each part is processed by machine M_1 first and machine M_2 later, with process times respectively equal to 10 and 20 time units. The loading and unloading processes require 1 time unit, while the time spent in the conveyor belts is assumed negligible.

The DTdPN modeling this production system is shown in Fig. 16.7, while Table 16.1 contains the meaning of places and transitions and the transitions delay.

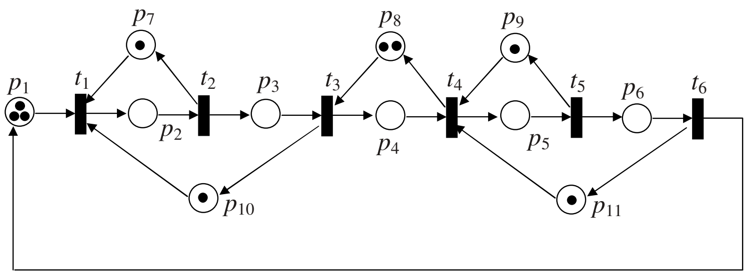

Fig. 16.7 DTdPN modeling a production line formed by two machines

At the initial marking $\boldsymbol{m}_0 = [3\ 0\ 0\ 0\ 0\ 0\ 1\ 2\ 1\ 1\ 1]^T$ transition t_1 is enabled and after one time unit fires yielding to the marking $\boldsymbol{m}_1 = [2\ 1\ 0\ 0\ 0\ 0\ 0\ 2\ 1\ 0\ 1]^T$, where transition t_2 is enabled. After 10 time units t_2 fires and yields the net to the new marking $\boldsymbol{m}_2 = [2\ 0\ 1\ 0\ 0\ 0\ 1\ 21\ 0\ 1]^T$. The net continues to evolve following the procedure indicated above. ■

16.4.2 Timed Marked Graphs

A *Timed Marked Graph* (TdMG) is a DTdPN where each place has only one input transition and one output transition and all arcs have unitary weight. A more restricted class of such nets are the *strongly connected timed marked graphs* whose importance is due to the fact that there exist some criteria to analyze the performance of the system in an easy way.

Definition 16.5. *A* (deterministic) Strongly Connected Timed Marked Graph *(SCTdMG) is a DTdPN N_d satisfying the following properties:*

- *the net structure N_d is a timed marked graph;*
- *the net is strongly connected, namely there exists an oriented path from each node to any node: this implies that each place and each transition of the net belongs to an oriented cycle; the set of the oriented elementary cycles of N_d is denoted $\Gamma = \{\gamma_1, \ldots, \gamma_r\}$;*
- *the timing structure Θ associated with transitions is deterministic and has constant time delays.*

Table 16.1 Description of places and transitions in Fig. 16.7

Place	Description
p_1	availability of parts and pallets
p_2	M_1 is working
p_3	part ready to be unloaded by M_1
p_4	part ready to be processed by M_2
p_5	m_2 is working
p_6	part ready to be unloaded by M_2
p_7	M_1 is available
p_8	availability on the conveyor belts
p_9	M_2 is available
p_{10}	R_1 is available
p_{11}	R_2 is available

Transition	Description	Delay
t_1	R_1 loads a part on M_1	1
t_2	M_1 ends the processing	10
t_3	R_1 ends the processing M_1 to the conveyor belt	1
t_4	R_2 loads a part on M_2	1
t_5	M_2 ends the processing	20
t_6	R_2 removes from M_2 a processed part	1

Although, these nets could seem too much restrictive, they can model important classes of discrete event systems. As an example two important classes of production systems, such as *job-shop* systems and systems based on the *Kanban* philosophy, can be modeled using SCTdMGs [14].

16.4.2.1 Performance Analysis

Let us now present some results that allow to perform the analysis, in steady conditions, in the case of TdMGs and SCTdMGs.

Theorem 16.1. *In a TdMG, the number of tokens in a cycle remains constant for any firing sequence.*

The proof of this theorem is based on the structural characteristics of a TdMG, where each place has one single input and output transition. In fact, each time a transition in a cycle fires, it removes a token from the input place that belongs to the cycle and put a token in the output place that belongs to the same cycle, thus the number of tokens in the cycle remains unchanged.

Let us now introduce the notion of *cycle time* that can be a performance index in a system modeled as a SCTdMG.

Definition 16.6. *The* cycle time C_i *of a transition* t_i *of a SCTdMG is defined on the basis of its generic kth time firing* $\tau_{i,k}$

$$C(t_i) = \lim_{k\to\infty} \frac{\tau_{i,k}}{k} \tag{16.2}$$

where $\tau_{i,k}$ *is the time instant at which transition* t_i *fires for the kth time, starting from the initial time instant* τ_0.

The above definition allows one to give two important results.

Theorem 16.2. [7, 12, 23] *In a SCTdMG, all transitions belonging to a cycle* $\gamma_j \in \Gamma$ *have the same cycle time* C_{γ_j}*, defined as the ratio between the sum of the delay times of transitions that form* γ_j *and the number of tokens circulating in it, i.e.,*

$$C_{\gamma_j} = \frac{\sum_{t_i \in \gamma_j} \theta_i}{\sum_{p_k \in \gamma_j} m[p_k]} \tag{16.3}$$

Theorem 16.3. [7, 10] *In a SCTdMG in steady conditions, all transitions in a cycle have the same cycle time C, equivalent to:*

$$C = \max_{\gamma_j \in \Gamma} C_{\gamma_j} = \max_{\gamma_j \in \Gamma} \left\{ \frac{\sum_{t_i \in \gamma_j} \theta_i}{\sum_{p_k \in \gamma_j} m[p_k]} \right\} \tag{16.4}$$

that identifies the maximum among the cycle times of all elementary cycles of a SCTdMG. This means that in steady conditions all transitions have the same firing frequency equal to $\lambda_r = 1/C$.

The result presented above is intuitive, in fact due to the structural characteristics of a SCTdMG, in steady conditions all cycles are synchronized on the "slower" cycle.

Example 16.4. Let us consider again the DTdPN shown in Fig. 16.7 already introduced in Example 16.3. The elementary cycles that form the set Γ are 6: γ_1: $p_7t_1p_2t_2p_7$; γ_2: $p_2t_2p_3t_3p_{10}t_1p_2$; γ_3: $p_8t_3p_4t_4p_8$; γ_4: $p_9t_4p_5t_5p_9$; γ_5: $p_5t_5p_6t_6p_{11}t_4p_5$; γ_6: $p_1t_1p_2t_2p_3t_3p_4t_4p_5t_5p_6t_6p_1$. By Theorem 16.2 the time cycles are respectively: $C_{\gamma_1} = 11$; $C_{\gamma_2} = 12$; $C_{\gamma_3} = 1$; $C_{\gamma_4} = 21$; $C_{\gamma_5} = 22$; $C_{\gamma_6} = 11,3$. Thus, by Theorem 16.3, the firing frequency of all transitions of this DTdPN in steady condition is $\lambda_r = 1/C = 1/\max_{\gamma_j \in \Gamma} C_{\gamma_j} = 1/22 = 0,045$. ■

16.5 Stochastic Timed Petri Nets

In this section, we present *Stochastic Timed Petri Nets* (STdPNs), i.e., P/T nets where the delays associated with transitions are random variables. As a result of this randomness STdPN can be considered stochastic processes.

We consider STdPNs with atomic firing and time delay described by a random variable with negative exponential distribution function. At each stochastic transition t_i is associated a parameter λ_i that characterizes its distribution, called *firing rate* or *firing frequency* of the transition. Note that if we denote $\bar{\theta}_i$ the average firing delay of transition t_i, it holds $\bar{\theta}_i = 1/\lambda_i$.

Definition 16.7. *A stochastic timed Petri net (STdPN) is a triple $N_s = (N, \Psi, \boldsymbol{\lambda})$ where:*

- *$N = (P, T, \boldsymbol{Pre}, \boldsymbol{Post})$ is a Petri net defined as in Definition 10.1 in Chapter 10;*
- *$\Psi = \{\Psi_i : t_i \in T\}$ is a timing stochastic structure; Ψ_i is a probability distribution function defined in $\mathbb{R}_+ \cup \{0\}$, from which are extracted the values of the random variables that form the delay firing $\theta_{i,k}$ of transition t_i, $t_i \in T$, $k \in \mathbb{N}_+$; in particular in this section we consider all Ψ_i as negative exponential distribution functions;*
- *$\boldsymbol{\lambda} = [\lambda_1\ \lambda_2\ \cdots]$ is the vector of the firing frequencies of transitions; elements λ_i can depend on the marking, namely it can be $\lambda_i = \lambda_i(\boldsymbol{m}_k)$, $k \in \mathbb{N}_+$.*

For a stochastic Petri net the firing of a transition follows the same rule of a P/T net, except for the fact that the choice of the next transition to fire is made on the basis of the firing probabilities of single transitions. The probability that transition t_i, enabled at marking $\boldsymbol{m}_k$, fires is equal to

$$Pr\{t_i \mid \boldsymbol{m}_k\} = \frac{\lambda_i(\boldsymbol{m}_k)}{\sum\limits_{t_j \in \mathscr{A}(\boldsymbol{m}_k)} \lambda_j(\boldsymbol{m}_k)} \tag{16.5}$$

where $\mathscr{A}(\boldsymbol{m}_k)$ is the set of transitions enabled at marking $\boldsymbol{m}_k$.

To describe the behavior of a STdPN, as for DTdPNs, clocks are associated with transitions. For simplicity, it is assumed that each transition is associated with a single clock, which is initialized to the value of the delay when the transition is enabled for the first time after a firing. Clocks operate as in the case of DTdPNs. In more detail, each time a new marking is reached, each enabled transition t_i resamples a new instance θ_i from the probability density function associated with its delay.

Many researchers formally demonstrated the potentiality of STdPNs as a tool for performance analysis of real systems, particularly showing that, from the point of view of the dynamic behavior, a STdPN is equivalent to a *Continuous Time Markov Chain* (CTMC). This connection has been proved by the following results that can be found in many classical books, such as [23].

Theorem 16.4. *In a STdPN, times spent in each marking are exponentially distributed.*

Theorem 16.5. *The time evolution of a STdPN can be described by a CTMC where each state corresponds to a different marking reachable by the STdPN.*

As a result of the above two theorems and of the STdPN evolution rules, it is possible to compute the distribution of times spent in a given marking $\boldsymbol{m}_j$ since the delay of all the enabled transitions is a random variable with exponential distribution. Therefore, the time spent in a marking is the smallest of the delays before next transition firing. As a consequence, the parameter that characterizes its exponential distribution is $\alpha_j = \sum_{t_i \in \mathscr{A}(\boldsymbol{m}_j)} \lambda_i(\boldsymbol{m}_j)$; this element also identifies the negative component $-q_{jj}$ along the diagonal frequency matrix $\boldsymbol{Q}$ of the CTMC equivalent to the considered STdPN.

Example 16.5. Consider the STdPN in Fig. 16.8 whose reachability graph is reported in the same figure, where $\boldsymbol{m}_0 = [1\ 0\ 0\ 0\ 0]^T$, $\boldsymbol{m}_1 = [0\ 1\ 1\ 0\ 0]^T$, $\boldsymbol{m}_2 = [0\ 0\ 1\ 1\ 0]^T$, $\boldsymbol{m}_3 = [0\ 1\ 0\ 0\ 1]^T$, e $\boldsymbol{m}_4 = [0\ 0\ 0\ 1\ 1]^T$.

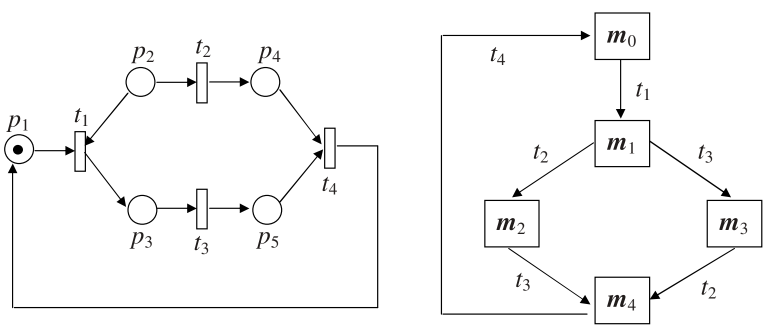

Fig. 16.8 A stochastic timed Petri net and its reachability graph

Let $\boldsymbol{\pi}$ be the row vector with as many components as the number of reachable markings, where each component represents the steady state probability associated with the corresponding marking. Since the graph is finite and strongly connected, we can compute vector $\boldsymbol{\pi}$ solving the linear system

$$\begin{cases} \boldsymbol{\pi Q} = \boldsymbol{0} \\ \sum_{x_j \in X} \pi_j = 1 \end{cases}$$

that in this example is equal to

$$\begin{cases} \boldsymbol{\pi} \begin{bmatrix} -\lambda_1 & \lambda_1 & 0 & 0 & 0 \\ 0 & -(\lambda_2+\lambda_3) & \lambda_2 & \lambda_3 & 0 \\ 0 & 0 & -\lambda_3 & 0 & \lambda_3 \\ 0 & 0 & 0 & -\lambda_2 & \lambda_2 \\ \lambda_4 & 0 & 0 & 0 & -\lambda_4 \end{bmatrix} = \begin{bmatrix} 0\,0\,0\,0\,0 \end{bmatrix} \\ \sum_{j=0}^{4} \pi_j = 1 \end{cases}$$

Supposing that $\lambda_j = 1$, $j = 0, \ldots, 4$, the system solution is $\pi_0 = \pi_4 = 2/7$, $\pi_1 = \pi_2 = \pi_3 = 1/7$. ■

The following section formalizes the rules of construction of the CTMC equivalent to a given STdPN.

16.5.1 Construction of the Markov Chain Equivalent to the STdPN

The CTMC equivalent to a given STdPN can be easily generated using the following algorithm:

Algorithm 16.6. (CTMC equivalent to a STdPN).

1. *Create a bijective correspondence between states X of the Markov chain and the reachability set $R(N_s, \boldsymbol{m}_0)$, such that to each marking $\boldsymbol{m}_k$ corresponds the state $x_k \in X$.*
2. *Let $\pi_0(0) = 1$ be the initial state probability vector, i.e., associate the maximum probability with state x_0 corresponding to $\boldsymbol{m}_0$.*
3. *Let the transition frequencies of the Markov chain, namely the elements of matrix $\boldsymbol{Q}$, equal to*

$$-q_{kk} = \sum_{t_i \in \mathscr{A}(\boldsymbol{m}_k)} \lambda_i(\boldsymbol{m}_k) \tag{16.6}$$

$$q_{kj} = \sum_{t_i \in \mathscr{A}_j(\boldsymbol{m}_k)} \lambda_i(\boldsymbol{m}_k) \tag{16.7}$$

where $\mathscr{A}_j(\boldsymbol{m}_k)$ is the subset of $\mathscr{A}(\boldsymbol{m}_k)$ that includes transitions whose firing yields to $\boldsymbol{m}_j$, i.e., $\mathscr{A}_j(\boldsymbol{m}_k) = \{t_i \in \mathscr{A}(\boldsymbol{m}_k) \mid \boldsymbol{m}_k[t_i\rangle \boldsymbol{m}_j\}$; in general there exists only one transition whose firing yields the state from marking $\boldsymbol{m}_k$ to marking $\boldsymbol{m}_j$.

Example 16.7. Consider the behavior of a machine: when it is available, it can load one part and starts its processing. The end of the process makes the machine available to process another part. The machine, however, may fail while working and therefore needs to be repaired. After the repair, the machine is ready to work again. The behavior of this machine is modeled by the STdPN in Fig. 16.9. The meaning of places and transitions is described in Table 16.2, where are also described the characteristic parameters of exponential distributions that determine the firing time of transitions.

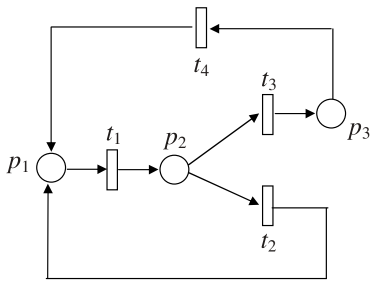

Fig. 16.9 STdPN of a machine that may fail

Table 16.2 Description of places and transitions in Fig. 16.9

Place	Description
p_1	the machine is available
p_2	the machine is working
p_3	the machine is being repaired

Transition	Description	Firing frequency
t_1	beginning of a process	α
t_2	end of a process	β
t_3	failure of the machine	μ
t_4	completed repairing	λ

States in which the machine can be are: x_0 = machine available, corresponding to marking $\boldsymbol{m}_0 = [1\ 0\ 0]^T$; x_1 = machine working, corresponding to marking $\boldsymbol{m}_1 = [0\ 1\ 0]^T$; x_2 = faulty machine, corresponding to marking $\boldsymbol{m}_2 = [0\ 0\ 1]^T$. The Markov chain equivalent to the STdPN is characterized by the frequency transition matrix $\boldsymbol{Q}$

$$Q = \begin{bmatrix} -\alpha & \alpha & 0 \\ \beta & -(\beta+\mu) & \mu \\ \lambda & 0 & -\lambda \end{bmatrix}$$

and the corresponding transitions diagram is shown in Fig. 16.10.

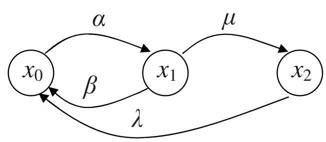

Fig. 16.10 State transition frequency diagram of the CTMC equivalent to the STdPN in Fig. 16.9

Note that such a diagram can be obtained by the reachability graph of the STdPN, represented in Fig. 16.11, substituting to each marking the corresponding state of the equivalent Markov chain and to each transition of the STdPN the parameter that characterizes the exponential distribution of the firing time. ■

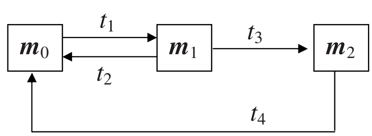

Fig. 16.11 Reachability graph of the STdPN in Fig. 16.9

16.5.2 Performance Analysis

An homogeneous CTMC that is finite and irreducible is always ergodic[1] [9]. This implies that a bounded STdPN whose reachability graph is strongly connected always corresponds to an ergodic CTMC. The reachability graph of a STdPN is equivalent to the one of a P/T net obtained removing time delays. This happens because time delays associated with transitions have probability density defined in $\mathbb{R}_+$. As a result, the criteria and methodologies introduced in Chapter 11 for the structural analysis of P/T nets also apply to STdPNs.

[1] A Markov chain is *ergodic* if and only if, for any initial probability distribution, there exists a limit probability distribution, i.e., there exists $\lim_{t\to\infty} \pi(t)$, and such a distribution is independent of the initial marking. Note that for ergodicity it is not necessary that the graph is irreducible. It is sufficient that there exists a unique ergodic component, i.e., a strongly connected component with no output arcs.

The analysis of a STdPN is usually targeted to measure aggregate performance indices that are more significant than the steady state probabilities $\boldsymbol{\pi}$ of the markings. In the following items are reported the most common performance indices [1].

- The probability of event e defined as a function of the marking (e.g. no token in a given set of places or at least one token in a place when another one is empty, etc.) can be computed summing up the probabilities of all markings that satisfy the condition expressed by the event; thus, the steady probability of event e is

$$Pr\{e\} = \sum_{\boldsymbol{m}_k \in \mathscr{M}_e} \pi_k$$

  where $\mathscr{M}_e$ is the set of markings satisfying the condition expressed by e; note that we can sum up the probabilities of the single markings because they are mutually exclusive.
- The probability of having a certain number of tokens in a place p_i can be computed as a special event; then, if $e_{i,j}$ denotes the event of having j tokens in place p_i, the average number of tokens in p_i can be computed as

$$\bar{n}_i = \sum_j jPr\{e_{i,j}\} \tag{16.8}$$

- The firing frequency f_j of a transition t_j, i.e., the average number of times that the transition fires in the time unit, under steady conditions can be computed as the weighted sum of the firing rates of transitions enabled at each marking $\boldsymbol{m}_i$

$$f_j = \sum_{i:t_j \in \mathscr{A}(\boldsymbol{m}_i)} \lambda_j(\boldsymbol{m}_i)\pi_i \tag{16.9}$$

- The average time $\bar{\theta}$ that a token spend to pass through a subnet in steady conditions can be computed applying the Little's law [9], that can be written for such a case as

$$\bar{\theta} = \frac{\bar{n}}{\lambda} \tag{16.10}$$

  where $\bar{n}$ is the average number of tokens passing through the subnet and λ is the average speed of the tokens that are entering in the subnet.

Finally, note that the main problem of the evaluation of the performance indices for a STdPN is the necessity of working with the equilibrium equations based on the reachability graph. In fact, the dimension of the reachability graph grows exponentially both with the number of tokens of the initial marking $\boldsymbol{m}_0$ and with the number of places; thus, except for some particular classes of nets, the dimension of the reachability graph and the computational complexity of the procedure do not allow to have exact analytical solutions.

16.6 Time Petri Nets

In this section, we focus on *T-Time Petri Nets*, i.e., P/T nets where a timing interval is associated with each transition. As in the case of T-Timed Petri nets, "T" (for transition) is often assumed as implicit and the model is more briefly called *Time Petri nets* (TPNs). This interval-based variant was first proposed in [20] and applied later to other timed models (see [8, 16, 17, 19]). The basic principle is the following. When an interval $[l_i, u_i]$ from the time domain is associated with a transition t_i in a P/T net, the bounds of the interval represent respectively the minimal and maximal delay for firing the transition. In this case, an implicit clock can be associated with the transition and the transition can be fired only if the clock value belongs to the interval.

We give a formal definition for this model and its semantics, described by timed transition systems.

We denote here by $\mathscr{I}$ the set of closed intervals with a lower bound in $\mathbb{Q}_+$ and an upper bound in $\mathbb{Q}_+ \cup \{\infty\}$, associated with transitions. In particular, $I(t_i) = [l_i, u_i]$ denotes the interval associated with transition t_i.

For an interval I, the backward closure of I is defined by: $I^{\downarrow} = \{x \mid \exists y \in I,\ x \leq y\}$.

Definition 16.8. *A* Time Petri Net *is characterized by the algebraic structure* $N_T = (N, I)$ *where:*

- *N is a P/T net defined as in Definition 10.1 in Chapter 10;*
- *$I : T \to \mathscr{I}$ associates with each transition a* firing interval.

Fig. 16.12 depicts a Time Petri net. Each transition is equipped with its firing interval. For instance, transition t_1 has interval $I(t_1) = [1, \infty[$. The initial marking has two tokens in place p_1 and one token in place p_2.

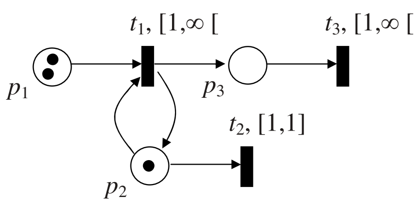

Fig. 16.12 A time Petri net, with time intervals on transitions

We now explain the timing conditions, with $\mathbb{R}_+$ as dense time domain. A transition t can be fired if the time elapsed since the last update belongs to its interval $I(t)$. Moreover, for all enabled transitions, time cannot progress when one of the upper bounds is reached, thus enforcing urgency.

A configuration of N_T is a pair $(\boldsymbol{m}, \nu)$, for a marking $\boldsymbol{m}$ and a valuation $\nu \in (\mathbb{R}_+ \cup \{\bot\})^T$. Relevant values of ν are those for which t belongs to $\mathscr{A}(\boldsymbol{m})$, and $\nu(t)$

contains the time elapsed since the last update in this case. We write $\nu(t) = \perp$ otherwise. For a real number d, the valuation $\nu + d$ is defined by $(\nu + d)(t) = \nu(t) + d$ for any t, with adequate conventions for $\perp$ values. A configuration is *admissible* if for all enabled transitions, $\nu(t) \in I(t)^{\downarrow}$. We denote by $ADM(N_T)$ the set of admissible configurations of N_T.

The important point that remains to be defined is the update of timing information upon transition firing. In other words, we should precise when the implicit clock associated with the transition is reset: transition t' is said to be *newly enabled* after firing t from marking $\boldsymbol{m}$ if the predicate $\uparrow enabled(t', \boldsymbol{m}, t)$ defined by:

$$\uparrow enabled(t', \boldsymbol{m}, t) = (t' \in \mathscr{A}(\boldsymbol{m} - \boldsymbol{Pre}[\cdot, t] + \boldsymbol{Post}[\cdot, t])) \wedge (t' \notin \mathscr{A}(\boldsymbol{m}))$$

evaluates to *true*.

Thus, t' is newly enabled if it was not enabled before firing t but becomes enabled by this firing. This corresponds to the so-called *persistent atomic* semantics, which is not the most frequently used but is easier to explain and equivalent to the other ones for safe time Petri nets. Discussions and comparisons with *atomic* and *intermediate* semantics can be found in [3, 22].

Definition 16.9. *The semantics of a time Petri net N_T is the timed transition system $\mathscr{T} = (S, s_0, E)$ where:*

- $S = ADM(N_T)$*;*
- $s_0 = (\boldsymbol{m}_0, \mathbf{0})$*, where* $\mathbf{0}$ *denotes the valuation with null values for all transitions enabled in* $\boldsymbol{m}_0$ *and* $\perp$ *otherwise;*
- $E \subseteq S \times (T \cup \mathbb{R}_+) \times S$ *contains the two following types of transitions, from an admissible configuration* $(\boldsymbol{m}, \nu)$*:*
  - *For each transition t enabled in $\boldsymbol{m}$ such that $\nu(t) \in I(t)$, a discrete transition $(\boldsymbol{m}, \nu) \xrightarrow{t} (\boldsymbol{m} - \boldsymbol{Pre}[\cdot, t] + \boldsymbol{Post}[\cdot, t]), \nu')$ such that for all $t' \in \mathscr{A}(\boldsymbol{m} - \boldsymbol{Pre}[\cdot, t] + \boldsymbol{Post}[\cdot, t])$,*
    $$\nu'(t') = \begin{cases} 0 & \text{if } \uparrow enabled(t', \boldsymbol{m}, t), \\ \nu(t') & \text{otherwise.} \end{cases}$$
  - *For each $d \in \mathbb{R}_+$, such that for each t in $\mathscr{A}(M), \nu(t) + d \in I(t)^{\downarrow}$, a delay transition $(\boldsymbol{m}, \nu) \xrightarrow{d} (\boldsymbol{m}, \nu + d)$.*

For instance, a possible run of the net in Fig. 16.12 is the following:

$$(\boldsymbol{m}_0, [0,0,\perp]) \xrightarrow{1} (\boldsymbol{m}_0, [1,1,\perp]) \xrightarrow{t_1} (\boldsymbol{m}_1, [1,1,0]) \xrightarrow{t_1} (\boldsymbol{m}_2, [\perp,1,0])$$
$$\xrightarrow{t_2} (\boldsymbol{m}_3, [\perp,\perp,0]) \xrightarrow{1.5} (\boldsymbol{m}_3, [\perp,\perp,1.5]) \xrightarrow{t_3} (\boldsymbol{m}_4, [\perp,\perp,1.5]) \cdots$$

with markings $\boldsymbol{m}_0 = [2\ 1\ 0]^T$, $\boldsymbol{m}_1 = [1\ 1\ 1]^T$, $\boldsymbol{m}_2 = [0\ 1\ 2]^T$, $\boldsymbol{m}_3 = [0\ 0\ 2]^T$, and $\boldsymbol{m}_4 = [0\ 0\ 1]^T$.

This definition corresponds to what is called *strong* semantics, and implies "*urgency*" for transition firing, because time delays cannot disable transitions. This can be used to enforce priorities between transitions, which can lead to time locks, i.e., deadlocks due to conflicting timing constraints. In contrast, in the definition of *weak* semantics [22], time elapsing is permitted beyond the upper bound of the interval of a transition, by removing the condition: for each t in $\mathscr{A}(M), \nu(t) + d \in I(t)^{\downarrow}$ for a transition with delay d. When this occurs, the transition is disabled, in a mechanism similar to what happens in ordinary P/T nets.

The class of time Petri nets with strong intermediate semantics has been largely studied (see for instance [4, 18]), and tools like ROMÉO [15] and TINA [5] have been developed for the analysis of bounded nets in this class.

16.7 Further Reading

This chapter is based on the Italian texbook on discrete event systems by Di Febbraro and Giua [13].

Many references are already cited along the chapter. Among them, particular attention should be devoted to [1, 23] dealing with Timed nets, and to [4, 6] devoted to Time nets. Interesting comparisons among different semantics for Time Petri nets are given in [3, 22].

References

1. Ajmone Marsan, M.: Stochastic Petri Nets: an Elementary Introduction. In: Rozenberg, G. (ed.) APN 1989. LNCS, vol. 424. Springer, Heidelberg (1990)
2. Baccelli, F., Cohen, G., Olsder, G.J., Quadrat, J.P.: Synchronization and linearity. Wiley (1992)
3. Bérard, B., Cassez, F., Haddad, S., Lime, D., Roux, O.H.: Comparison of Different Semantics for Time Petri Nets. In: Peled, D.A., Tsay, Y.-K. (eds.) ATVA 2005. LNCS, vol. 3707, pp. 293–307. Springer, Heidelberg (2005)
4. Berthomieu, B., Diaz, M.: Modeling and verification of time dependent systems using time Petri nets. IEEE Transactions on Software Engineering 17(3), 259–273 (1991)
5. Berthomieu, B., Vernadat, F.: Time Petri nets analysis with TINA. In: Proc. 3rd Int. Conf. on the Quantitative Evaluation of Systems, Riverside, California, USA (2006)
6. Boyer, M., Roux, O.H.: On the compared expressiveness of arc, place and transition time Petri nets. Fundamenta Informaticae 88(3), 225–249 (2008)
7. Campos, J., Chiola, G., Colom, J.M., Silva, M.: Properties and performance bounds for timed marked graphs. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 39(5), 386–401 (1992)
8. Campos, S., Clarke, E.M.: Theories and experiences for real-time system development. AMAST Series in Computing, vol. 2, pp. 129–145. World Scientific (1995)
9. Cassandras, C.G., Lafortune, S.: Introduction to Discrete Event Systems, 2nd edn. Springer (2008)
10. Chretienne, P.: Les reseaux de Petri Temporises. Master Thesis. Univ. of Paris VI, Paris (1983)
11. Coolahan, J., Roussopoulos, N.: Timing requirements for time-driven systems using augmented Petri nets. IEEE Transactions on Software Engineering 9(5), 603–616 (1983)

12. Desrochers, A.A., Al-Jaar, R.Y.: Applications of Petri nets in manufacturing systems. IEEE Press (1995)
13. Di Febbraro, A., Giua, A.: Sistemi ad eventi discreti. McGraw Hill (2002) (in Italian)
14. Di Cesare, F., Harhalakis, G., Proth, J.M., Silva, M., Vernadat, F.B.: Practice of Petri Nets in Manufacturing. Chapman and Hall (1993)
15. Gardey, G., Lime, D., Magnin, M., Roux, O.H.: Romeo: A Tool for Analyzing Time Petri Nets. In: Etessami, K., Rajamani, S.K. (eds.) CAV 2005. LNCS, vol. 3576, pp. 418–423. Springer, Heidelberg (2005)
16. Henzinger, T.A., Manna, Z., Pnueli, A.: What Good are Digital Clocks? In: Kuich, W. (ed.) ICALP 1992. LNCS, vol. 623, pp. 545–558. Springer, Heidelberg (1992)
17. Henzinger, T.A., Manna, Z., Pnueli, A.: Temporal proof methodologies for timed transition systems. Information and Computation 112(2), 273–337 (1994)
18. Jones, N.D., Landweber, L.H., Lien, Y.E.: Complexity of some problems in Petri nets. Theoretical Computer Scienc 4, 277–299 (1977)
19. Laroussinie, F., Markey, N., Schnoebelen, P.: On Model Checking Durational Kripke Structures. In: Nielsen, M., Engberg, U. (eds.) FOSSACS 2002. LNCS, vol. 2303, pp. 264–279. Springer, Heidelberg (2002)
20. Merlin, P.M.: A Study of the Recoverability of Computing Systems. UCI, Univ. California, Irvine (1974)
21. Ramchandani, C.: Analysis of asynchronous concurrent systems by Petri nets. Project MAC, TR-120. M.I.T, Cambridge (1974)
22. Reynier, P.-A., Sangnier, A.: Weak Time Petri Nets Strike Back! In: Bravetti, M., Zavattaro, G. (eds.) CONCUR 2009. LNCS, vol. 5710, pp. 557–571. Springer, Heidelberg (2009)
23. Wang, J.: Timed Petri Nets: Theory and Applications. CRC Press (1998)
24. Zhu, J.J., Denton, R.T.: Timed Petri nets and their application to communication protocol specification. In: Proc. Military Communications Conference, San Diego, CA (1988)