Timed Arc Petri Nets: The Time-Element Approach

Alpaslan Yufka, Hanife Apaydın Özkan, and Aydın Aybar

Department of Electrical and Electronics Engineering
Anadolu University
Eskisehir, Turkey
{ayufka, hapaydin1, aaybar}@anadolu.edu.tr

Abstract

In this work, a novel model is proposed for Deterministic Timed Arc Petri Nets. In the proposed Timed Arc Petri Net, transition firing processes are associated with time, where time is interpreted as firing delays, hence deterministic time-delays are assigned with arcs. In Timed Petri Nets with firing delay, corresponding tokens can become temporarily invisible and re-visible during the firing process of a transition. In order to overcome this temporary invisibility, a new triangular graphical element, namely time element, is introduced. This new time element allows user to observe temporarily invisible tokens and it depicts the whole firing process graphically.

1. Introduction

Discrete-Event Systems (DESs) are event-driven systems described by the occurrence of discrete events [1]. Petri Nets (PNs) have been used to model such systems whose state evolution depends on this occurrence. PNs are the modeling paradigm such that they was first introduced without the notation of time [1, 2, 3, 4]. Untimed PNs are insufficient to define the complete model of the system since the delay of system activities are not taken into consideration [5]. Usage of time is necessary to model such time delayed systems [4]. A time extension of untimed PNs was introduced as Timed Petri Nets (TdPNs) [6, 7, 8, 9, 10] where the time can be associated with transitions, places or arcs of a PN [9]. Time can be associated with an event in three ways as deterministic time delays, stochastic time delays and time intervals [6].Besides, time delays in TdPNs are essentially interpreted as in three ways as firing, holding and enabling delays [6]. In this work, Timed Arc Petri Nets with deterministic time delays are considered, where time is interpreted as firing.

Some studies have been proposed for Timed Arc Petri Nets. They could be classified into two categories. Firstly, arcs are associated with time-intervals and tokens have their own age. Each arc with a time interval restricts the age of tokens traveling through the arc [11, 12, 13, 14, 15, 16, 17]. In these studies time delays are interpreted as enabling. Secondly, only arcs are associated with deterministic time-delays and tokens have no age [6, 9, 18]. In these studies, time delays are interpreted as enabling [6, 9, 18] and holding [6]. To the best of the authors' knowledge, for Timed Arc Petri Nets, time delays are not associated with firing process (not interpreted as firing) at any work in the literature such that authors have proposed the hourglass representation [19]. However, in the case of holding and enabling delays especially concerning to time intervals, tokens resides in places such that one may need the time left to the availability of each token [20]. One may also need to observe the firing process. In TdPNs with firing delay, corresponding tokens can become temporarily invisible and re-visible during the firing process of a transition. The approach, namely Stretched Petri Net (SPN), proposed for TdPNs with firing delays [10, 20]. SPN has been proposed by introducing new transitions with unit delays, new places and new arcs with unity weights. SPN allows user to observe the firing process by converting the TdPN to an untimed PN; however, new transitions, places and arcs have to be added into the timed net. The number of these new elements are increased as related time delay of an event rises.

In this work, a novel Deterministic Timed Arc Petri Net model is proposed, where deterministic time delays are associated with arcs and time is interpreted as firing delays. In the graphical representation, in order to overcome the temporary invisibility in TdPNs and to observe the firing process via only one element, a new triangular graphical element, namely time element, is introduced. This new time element allows user to observe temporarily invisible tokens and it depicts the whole firing process graphically. In the mathematical representation, the resulting states are represented by marking vectors which represent the number of tokens in all places, by time element vectors which represent the number of flowing tokens in time elements, and by remaining time vectors which represent the time-status of flowing tokens in time elements.

2. Timed Petri Nets

A TdPN with firing delay is defined as a tuple $G_{TdPN}(P, T, N, O, M_0, \mathfrak{D})$ [10]. Here, $P := \{p_i | \ i = 1, 2, ..., m \in \mathbb{N}\backslash\{0, \infty\}\}$ is the finite set of places, where $\mathbb{N}$ is the set of natural numbers, and $T := \{t_i | \ i = 1, 2, ..., n \in \mathbb{N}\backslash\{0, \infty\}\}$ is the finite set of transitions ($P \cap T = \emptyset$). $N : P \times T \rightarrow \mathbb{N}$ is the input matrix which specifies weights of the arcs directed from places to transitions, namely ingoing arcs, and $O : P \times T \rightarrow \mathbb{N}$ is the output matrix which specifies weights of the arcs directed from transitions to places, namely outgoing arcs. $M_0 : P \rightarrow \mathbb{N}$ is the initial marking vector at the initial time k_0. $\mathfrak{D} := \{d_{t_i} \in \mathbb{N}\backslash\{0\} | \ i = 1, 2, ..., |T|\}$ is the set of time delays, where d_t is the time delay of the transition $t \in T$ ($|*|$ indicates the number of elements in the set *.) In this work, it is assumed that the time, i.e. $k \in \mathbb{N}$, can be discretized into time slots (*ts*) by using an appropriate sampling period. An

This work has been supported by Anadolu University through Research Project BAP-1610F665, namely "*Dynamic Control Approaches for Timed Discrete Event Systems*".

example TdPN model is shown in Fig. 1.

The state of the TdPN at time k is given by $S(k) := \{M(k), Q(k)\}$, where $M(k) : P \to \mathbb{N}$ is the marking vector at time k, and $Q(k) := \{(t, k - \kappa_f(t)) | k - d_t < \kappa_f(t) < k\}$, where $\kappa_f(t)$ presents the firing time of the transition $t \in T$. For the sake of simplicity, after this $M(k)$ represents the marking vector at time k and $M(p)$ represents the number of tokens assigned to place $p \in P$ by $M(k)$.

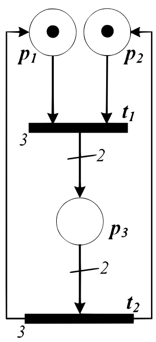

Figure 1. An example of TdPN.

A transition $t \in T$ is enabled at time k iff it is enabled at $M(k)$, that is iff $M(p) \geq N(p,t)$ for all $p \in P$. $E(G_{TdPN}, M(k)) \subseteq T$ represents the set of enabled transitions of G_{TdPN} at $M(k)$. A transition $t \in T$ can fire at time k iff $t \in E(G_{TdPN}, M(k))$.

An example of TdPN's firing process concerning to the transition $t_1 \in T$ model is shown in Fig. 2. In TdPN with firing delay, the firing process concerning to the transition $t \in T$ starts with the firing of the transition $t \in T$ at time $k = \kappa_f(t)$. An enabled transition removes the token(s) from the input place $p \in \bullet t$ immediately and does not create any token(s) in the output place during the firing duration. Here, $\bullet t := \{p \in P | N(p,t) \neq 0\}$ describes a finite set of input places concerning to $t \in T$. The transition holds the token(s) during the firing delay. These causes a temporary invisibility of token(s) in the marking vector during the firing delay (See Fig. 2.b). The token(s) is created in the output place $p \in t\bullet$ after the firing delay is elapsed (See Fig. 2.c). Here, $t\bullet := \{p \in P | O(p,t) \neq 0\}$ describes a finite set of output places concerning to $t \in T$.

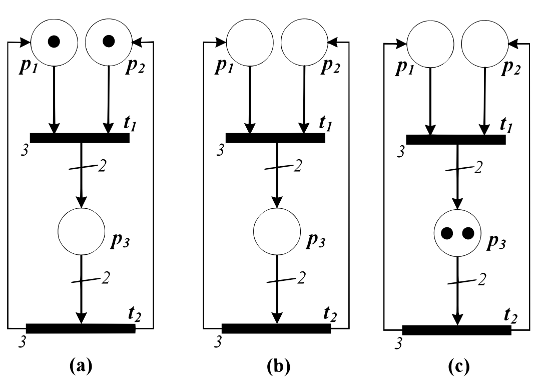

Figure 2. An example of TdPN's firing process. (a) $k < \kappa_f(t_1)$, (b) $\kappa_f(t_1) \leq k < \kappa_f(t_1) + d_{t_1}$, c) $k \geq \kappa_f(t_1) + d_{t_1}$

3. Timed Arc Petri Nets

In this section, the proposed Timed Arc Petri Net (TdAPN) is introduced to show temporarily invisible tokens transmitted through arcs. In TdAPN, arcs are associated with deterministic time delays in terms of the appropriate *ts*.

3.1. The Proposed Model

A TdAPN is defined by a tuple $G_{TdAPN}(P, T, \nabla, N, O, S_0, D, \nabla^P, \nabla^T)$. Here, $\nabla := \{h_i | i = 1, 2, ..., r \in \mathbb{N} \backslash \{0, \infty\}\}$ is the finite set of time elements ($\nabla \cap P = \emptyset$ and $\nabla \cap T = \emptyset$). The complete initial state at the initial time k_0 is $S_0 := \{M_0, \nabla^{M_0}, \nabla^{R_0}\}$, where M_0 is the initial marking vector as defined in TdPN, $\nabla^{M_0} : \nabla \to \mathbb{N}$ is the initial time element vector at time k_0 and $\nabla^{R_0} : \nabla \to \mathbb{N}$ is the initial remaining time vector assigned to time elements at time k_0. It is assumed that ∇^{M_0} and ∇^{R_0} are equal to $0^{|\nabla|,1}$ in this study, thus the net is relaxed ($0^{|.|,1}$ denotes a $|.|$ x 1 sized column zeros'vector.). $D : P \times T \to \mathbb{N}$ denotes the time-delay matrix associated with outgoing arcs. The default value of the element $D(p,t)$ of D is zero *ts*. $D(p,t)$ of D is equal to the time delay of the outgoing arc if a transition $t \in T$ is connected to an output place $p \in t\bullet$. Note that $D(p,t)$ can be greater and equal than zero *ts*. It is assumed that for all arcs directed from places to transitions, time-delay is considered as one (*ts*). $\nabla^P : \nabla \times P \to \{0, 1\}$ is the relation matrix that specifies the connections directed from time elements $h \in \nabla$ to places $p \in P$. Thus $\nabla^P(h,p) = 1$ if $h \in \nabla$ maps to $p \in P$; otherwise, $\nabla^P(h,p) = 0$. $\nabla^T : \nabla \times T \to \{0, 1\}$ is the relation matrix that specifies the connections directed from transitions $t \in T$ to time elements $h \in \nabla$. The element $\nabla^T(h,t) = 1$ if $t \in T$ maps to $h \in \nabla$; otherwise, $\nabla^T(h,t) = 0$. An example TdAPN model is shown in Fig. 3.

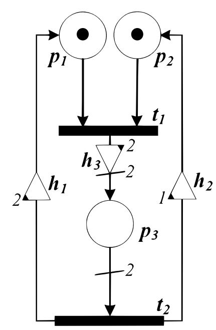

Figure 3. An example of the proposed TdAPN.

In TdPNs with firing delay, tokens in transition (flow) through arcs are not observed graphically during the firing process. These tokens are called as flowing tokens in this work. However, one needs to observe and know the state of flowing tokens, besides the state of the system. In TdAPN, $S(k) := \{M(k), \nabla^M(k), \nabla^R(k)\}$ is the complete state of the G_{TdAPN} at time k, where it can be considered in two parts as the state of the system ($M(k)$) and the state of time elements ($\nabla^M(k)$,$\nabla^R(k)$). Here, $\nabla^M(k) : \nabla \to \mathbb{N}$ represents the time element vector at time k and $\nabla^R(k) : \nabla \to \mathbb{N}$ represents the remaining time vector assigned to time elements at time k.

In TdAPN, a firing process of an enabled transition $t \in T$ starting at time $k = \lambda$, $\lambda \in \mathbb{N}$, is represented by $(t)^\lambda$. Here, a transition $t \in T$ is enabled at time k iff it is enabled at $M(k)$, that is iff $M(p) \geq N(p,t) \geq 1$ for all $p \in \bullet t$, where $(t)^\lambda \notin F(k), \lambda < k$. Here, $F(k) := \{(t)^\lambda | \lambda \leq k < (\lambda+1) + max_{\forall p \in t\bullet}\{D(p,t)\}, t \in T\}$ is a set of firing processes of G_{TdAPN} which includes active firing processes $(t)^\lambda$ started previously and have not finished yet, and newly started at time $k = \lambda$. It is assumed that once a firing process $(t)^\lambda \in F(k)$ concerning to enabled $t \in T$ starts at time $k = \lambda$, another firing process of this t is not able to start again during the firing process $(t)^\lambda$. t is reconsidered for enabledness after its related firing process $(t)^\lambda$ ends at time $k = (\lambda+1) + max_{\forall p \in t\bullet}\{D(p,t)\}$. $E(G_{TdAPN}, M(k))$ represents the set of transitions of G_{TdAPN} which are enabled at $M(k)$. A transition $t \in T$ can fire at any time k iff $t \in E(G_{TdAPN}, M(k))$.

3.2. The Time-Element

In TdAPN, in order to show the state of flowing tokens graphically, a new triangular shaped graphical element such as a time element is introduced. This new element is associated with outgoing arcs in this work. Examples of the time element are shown in Fig. 4 and Fig. 5.

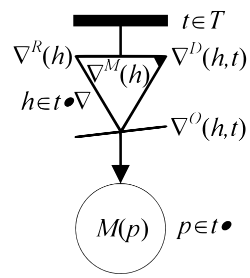

Figure 4. The representation of a time element $h \in t \bullet \nabla$ at time k.

The time element $h \in t \bullet \nabla$ is composed of four parts as follows (See Fig. 4), where $t \bullet \nabla := \{h \in \nabla | \nabla^T(h,t) = 1\}$ describes a finite post set of time elements concerning to the transition $t \in T$:

- **Right corner** indicates the outgoing arc's deterministic time-delay ($\nabla^D(h,t)$ or $D(p,t)$). Here, $\nabla^D : \nabla \times T \rightarrow \mathbb{N}$ denotes the reshaped time-delay matrix for time elements as given in (1). Moreover, in order to prevent any confusion, a filled triangular indicator at the inner corner of the time element is used to indicate the arc's time-delay.

$$\nabla^D(h,t) = \begin{cases} D(p,t) & , \nabla^P(h,p) = 1 \wedge \nabla^T(h,t) = 1 \\ 0 & , otherwise \end{cases} \tag{1}$$

- **Line at the bottom corner** denotes the outgoing arc's weight ($\nabla^O(h,t)$ or $O(p,t)$). Here, $\nabla^O : \nabla \times T \rightarrow \mathbb{N}$ denotes the reshaped weight matrix for time elements as given in (2). It is no need to indicate the middle line and weight since the weight of arc is 1.

$$\nabla^O(h,t) = \begin{cases} O(p,t) & , \nabla^P(h,p) = 1 \wedge \nabla^T(h,t) = 1 \\ 0 & , otherwise \end{cases} \tag{2}$$

- **Left corner** indicates the remaining time ($\nabla^R(h)$). Here, $\nabla^R(h)$ represents the remaining time of flowing tokens assigned to the time element $h \in t \bullet \nabla$ by $\nabla^R(k)$. The remaining time-delay next to the left corner is indicated since it is greater than 0 *ts*.
- **Inside the triangle** indicates the number of flowing tokens ($\nabla^M(h) = \nabla^O(h,t)$). Here, $\nabla^M(h)$ represents the number of tokens assigned to the time element $h \in t \bullet \nabla$ by $\nabla^M(k)$. Moreover, if $1 \leq \nabla^R(h) \leq \nabla^D(h,t)$, then it means tokens are visible in the time element $h \in t \bullet \nabla$ and are being transmitted over the time element. If $\nabla^R(h) = \nabla^D(h,t) + 1$ *ts*, then it means transition of tokens starts.

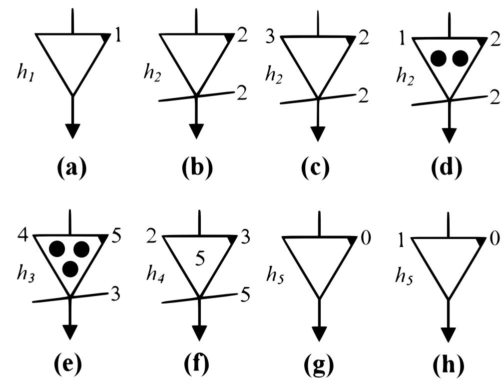

Figure 5. (a)-(f) Examples of the Time Element.

Some examples of the time element are given in Fig. 5. Here, in Fig. 5.a, the time element h_1 indicates no active firing process with unit time delayed and unit weighted arc. In Fig. 5.b, the time element h_2 shows no active firing process with 2 time delayed and 2 weighted arc. In Fig. 5.c, it indicates a firing process started newly (at $k = \lambda$). Here, note that left corner of the time element h_2 indicates the time delay in order to make token(s) visible inside the output place connected to h_2. In Fig. 5.d, two flowing tokens are transmitted through the time element h_2 and they would become visible in the output place after 1 *ts* elapsed. In Fig. 5.e, three flowing tokens transmit through the time element h_3 and they would become visible in the output place after 4 *ts* . In Fig. 5.f, five flowing tokens transmit through the time element h_4 and they would become visible in the output place after 2 *ts* elapsed. Note that flowing tokens are transmitted as the number of the arc's weight. In Fig. 5.g, the time element h_5 indicates no active firing process with zero time delayed and unit weighted arc. In Fig. 5.h, it indicates a firing process started newly, but no token appears in the time element h_5 at the next time instant due to the zero time delayed arc. It appears in the output place connected to the time element h_5.

3.3. The Firing Process

In practice, system activities and delays such as timer operations, preparations, known delays, etc. can be considered as exact time-delays, and starting, ending, activating, controlling points, etc. can be considered as time-instants. A firing process concerning to transition $t \in T$ can be described in terms of time-delays and time-instants. In order to obtain the true model, where the system activities are dynamic and in motion, the model can be best described by interpreting time as firing durations.

In TdAPN, a firing process $(t)^\lambda$ concerning to the transition $t \in T$ is expressed in three parts in terms of time-instants as starting time-instant, ending time-instant for an output place and ending time-instant for the firing process.

3.3.1. Starting time-instant

It is a time-instant denoted by $\lambda \in \mathbb{N}$. It shows the starting time-instant for the firing process $(t)^{\lambda}$, where this process starts and the enabled transition $t \in T$ fires at time $k = \lambda$. At time $k = \lambda$, concerning tokens in all input places $p \in \bullet t$ becomes unavailable for another processes and these tokens are graphically illustrated by unfilled tokens. At the next time instant $k = (\lambda + 1)$, these tokens become temporarily invisible in input places and are transferred to concerning time elements $\forall h \in t \bullet \nabla$ as the number of $\nabla^{O}(h, t)$. Once a firing process concerning to the transition t starts, this transitions is considered as disabled until its firing process is finished completely.

3.3.2. Ending time-instant for an output place

It shows the ending time-instant for an output place $p \in t\bullet$ in the firing process $(t)^{\lambda}$, where the number of $\nabla^{O}(h, t)$ flowing tokens completes their transition and appears in this output place at time $k = (\lambda + 1) + D(p, t)$.

3.3.3. Ending time-instant for the firing process

It shows the completely ending time-instant for the firing process $(t)^{\lambda}$, where the transition $t \in T$ completes its firing process at time $k = (\lambda + 1) + max_{\forall p \in t\bullet}\{D(p, t)\}$ so that the transition t is reconsidered for enabledness. Note that $d_t = max_{\forall p \in t\bullet}\{D(p, t)\} + 1$ *ts*, where d_t indicates the complete time of $(t)^{\lambda}$ concerning to $t \in T$ and 1 *ts* comes from the maximum time-delay among ingoing arcs. Moreover, $d_t \geq 1$ *ts* since the time-delay of ingoing arcs is one *ts* and the minimum time-delay of an outgoing arc is zero *ts*. It can be considered that the value of d_t in TdAPN is equal to be the value of d_t in TdPN.

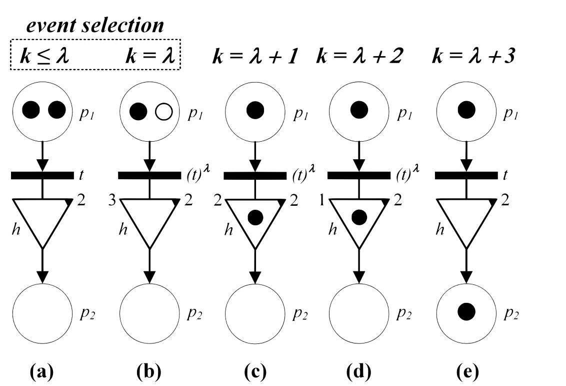

Figure 6. For TdAPN, (a)-(e) Stages of the Firing Process at each ascending time slot.

An animated example of a firing process $(t)^{\lambda}$ is given in Fig. 6. At time $k \leq \lambda$, there are two options for the net such that the $(t)^{\lambda}$ concerning to the enabled t can be started or can be postponed. $(t)^{\lambda}$ can be started at any time k (not necessarily immediately) since the transition t is enable. These occurs two distinct situations at the same time k. The choice from between these situations would be called as the event selection and it would affect only $\nabla^{R}(h)$. In the example, no firing process starts at time $k < \lambda$ ($(t)^{\lambda}$ is postponed.) even if the transition t is enable (See Fig. 6.a). At time $k = \lambda$, the firing process $(t)^{\lambda}$ starts and concerning transition t is fired. The token (as the weight of the ingoing arc) in the input place $p_1 \in \bullet t$ becomes unfilled (See Fig. 6.b). At time $k = \lambda + 1$, the flowing token resides in the time element $h \in t \bullet \nabla$ and it is being transmitted through the time $k \in [\lambda{+}1, \lambda{+}2]$ (See Fig. 6.c-d). At time $k = \lambda + 3$, where $D(p_2, t) = 2$ *ts*, the token appears in the output place p_2 (See Fig. 6.e). The firing process $(t)^{\lambda}$ is completely finished at time $k = \lambda + 3$, where $d_t = 3$ *ts*.

4. Example

In this section, we consider the TdAPN shown in Figure 3 as an example. This TdAPN is described as $G_{TdAPN}(P, T, \nabla, N, O, S_0, D, \nabla^{P}, \nabla^{T})$, where $P = \{p_1, p_2, p_3\}$, $T = \{t_1, t_2\}$, $\nabla = \{h_1, h_2, h_3\}$,

$$N = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 2 \end{bmatrix}, O = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 2 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 2 \\ 0 & 1 \\ 2 & 0 \end{bmatrix}$$

$$\nabla^{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \nabla^{T} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$S_0 = \{[1\,1\,0]^{T}, [0\,0\,0]^{T}, [0\,0\,0]^{T}\}$ .The complete time of firing processes concerning to $t \in T$ are $d_{t_1} = d_{t_2} = 3$ *ts*.

An animated example of all possible firing processes for the given TdAPN is depicted in Fig. 7. Here, 8 complete states ($S = \{M, \nabla^{M}, \nabla^{R}\}$) are found for the given TdAPN as given in Table 1. Relations among states are shown in Fig. 8, where square boxes indicate relaxed states, which are identical to untimed PN's; circular boxes indicate dynamic states, where tokens are being transmitted; arrows in dotted lines denote the event selection, whereas straight lines denote the 1 *ts* time delay elapsed; dotted boundaries denote the event selection area, where M and ∇^{M} are identical.

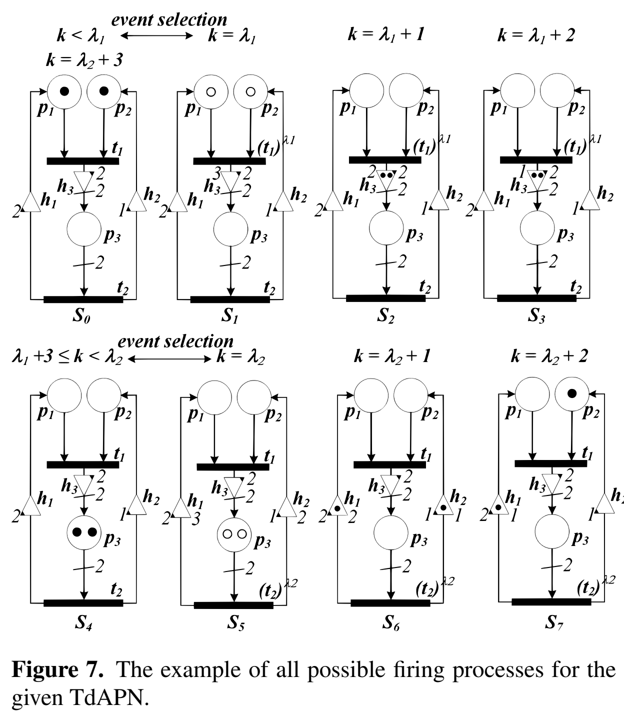

Figure 7. The example of all possible firing processes for the given TdAPN.

5. Conclusion

The proposed TdAPN represents a new notion for Deterministic Timed Arc Petri Nets and provides corresponding nota-

Table 1. The complete states for the given TdAPN.

S	E	F	M	∇^M	∇^R
S_0	$\{t_1\}$	$\{\emptyset\}$	$[1\,1\,0]^T$	$[0\,0\,0]^T$	$[0\,0\,0]^T$
S_1	$\{t_1\}$	$\{(t_1)^\lambda\}$	$[1\,1\,0]^T$	$[0\,0\,0]^T$	$[0\,0\,3]^T$
S_2	$\{\emptyset\}$	$\{(t_1)^\lambda\}$	$[0\,0\,0]^T$	$[0\,0\,2]^T$	$[0\,0\,2]^T$
S_3	$\{\emptyset\}$	$\{(t_1)^\lambda\}$	$[0\,0\,0]^T$	$[0\,0\,2]^T$	$[0\,0\,1]^T$
S_4	$\{t_2\}$	$\{\emptyset\}$	$[0\,0\,2]^T$	$[0\,0\,0]^T$	$[0\,0\,0]^T$
S_5	$\{t_2\}$	$\{(t_2)^\lambda\}$	$[0\,0\,2]^T$	$[0\,0\,0]^T$	$[3\,2\,0]^T$
S_6	$\{\emptyset\}$	$\{(t_2)^\lambda\}$	$[0\,0\,0]^T$	$[1\,1\,0]^T$	$[2\,1\,0]^T$
S_7	$\{\emptyset\}$	$\{(t_2)^\lambda\}$	$[0\,1\,0]^T$	$[1\,0\,0]^T$	$[1\,0\,0]^T$

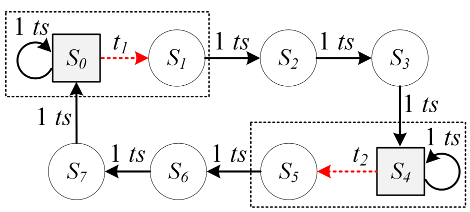

Figure 8. Relations among the complete states of the given TdAPN.

tion, where deterministic time delays assigned to arcs are interpreted as firing. Proposed time element approach allows user to observe temporarily invisible tokens whereas these are not observed in TdPNs graphically. It also gives information about the related arc, the whole firing process and corresponding flowing token(s). The concept is clear and concise. TdAPN is used with unit time in terms of appropriate time slots which is readily handled with computers in practical applications. Thus, we expect the model constructed by TdAPN could be easily implemented in certain time-delayed systems and related algorithms. Time durations such that exact time labels attached to outgoing arcs have no time intervals so that using deterministic time values instead of time intervals may provide less complexity and may decrease the computational time. It may also provide plainness for implementing algorithms. Furthermore, redundant transitions and places from the untimed PN model may be removed by using TdAPN.

In this work, the time element and the firing process of TdAPN are introduced. Using the proposed approach, TdAPN is able to give a complete model for timed delay systems. Thus, one may able to see the complete picture of the system with deterministic time delays. Next future direction will include construct mathematical relations in order to compute the complete state of TdAPN at the next time instant.

6. References

[1] Cassandras, C.G. and Lafortune, S., ”Introduction to Discrete Event Systems, Second Edition”, *Springer US,* New York, 2008.

[2] Y. Ho (Ed.), ”Discrete Event Dynamic Systems: Analyzing Complexity and Performance in The Modern World”, IEEE Press, A Selected Reprint Volume, New York. 1992.

[3] Zhou, M. and DiCesare, F., ”Petri Net Synthesis for Discrete Event Control of Manufacturing Systems”, *The Springer International Series in Engineering and Computer Science,*, vol. 204. 1993.

[4] Proth, J.M. and Xie, X., ”Petri Nets a Tool for Design and Management of Manufacturing Systems”, *Wiley,*. 1996.

[5] Zuberek, W.M., ”Timed Petri nets - definitions, properties, and applications”, *Microelectronics and Reliability,*, vol. 31, no. 4, pp: 627-644. 1991.

[6] Bowden, F.D.J., ”A brief survey and synthesis of the roles of time in Petri nets”, *Mathematical and Computer Modelling,*, vol. 31, pp: 55-68. 2000.

[7] Freedman, P., ”Time, Petri nets, and robotics”, *IEEE Transactions on Robotics and Automation,*, vol. 7, no. 4, pp: 417-433. 1991.

[8] Zuberek, W.M., ”Timed Petri nets in modeling and analysis of cluster tools”, *IEEE Transactions on Robotics and Automation,*, vol. 17, no. 5, pp: 562-575. 2000.

[9] Wang, J., ”Time Petri Nets, Theory and Application”, *Kluwer*. 1998.

[10] Aybar, A. and Iftar, A., ”Supervisory controller design to enforce some basic properties in timed-transition Petri nets using stretching”, *Nonlinear Analysis: Hybrid Systems,*, vol. 6, pp. 712-729. 2012.

[11] Walter, W.B., ”Timed Petri-Nets for Modelling and Analyzing Protocols with Real-Time Characteristics”, in *the 3rd Int.Work. on Protocol Specification, Testing, and Verification*, Switzerland, 1983, pp. 149-159.

[12] Bolognesi, T. and Lucidi, F. and Trigila, S., ”From Timed Petri Nets to Timed LOTOS”, in *the 10th Int.Symp. on Protocol Specification, Testing and Verification*, Canada, 1990, pp. 395-408.

[13] Hanisch, H.M., ”Analysis of Place/Transition Nets with Timed-Arcs and its Application to Batch Process Control”, in *the 14th Int.Conf. on Application and Theory of Petri Nets*, USA, 1993, vol. 691, pp. 282-299.

[14] Abdulla, P.A. and Nylen, A., ”Timed Petri nets and BQOs”, in *the 22nd Int.Conf. on Application and Theory of Petri Nets*, UK, 2001, vol. 2075, pp. 53-70.

[15] Nielsen, M. and Sassone, V. and Srba, J., ”Properties of Distributed Timed-Arc Petri Nets”, in *the 21st Int.Conf. on Foundations of Software Technology and Theoretical Computer Science*, India, 2001, vol. 2245, pp. 280-291.

[16] Jensen, P.G. and Larsen, K.G. and Srba, J., ”Real-Time Strategy Synthesis for Timed-Arc Petri Net Games via Discretization”, in *the 23rd Int.Symp. on Model Checking Software*, The Netherlands, 2016, vol. 9641, pp. 129-146.

[17] Sieverding, S. and Ellen, C. and Battram, P., ”Sequence Diagram Test Case Specification and Virtual Integration Analysis using Timed-Arc Petri Nets”, in *the 10th Int.Work. on Formal Eng. Approaches to Soft. Components and Arch.*, Italy, 2013, vol. 108, pp. 17-31.

[18] Zhu, J. and Denton, R., ”Timed Petri Nets and their Application to Communication Protocol Specification”, in *the 21st Int.Conf. on Military Communications*, USA, 1988, pp. 195-199.

[19] Yufka, A., Özkan, H.A. and Aybar, A., ”A Formal Method and Novel Graphical Representation for Deterministic Timed-Arc Petri Nets”, in *the National Conf. on Otomatik Kontrol Ulusal Toplantisi*, Turkey, 2016, pp. 209-213.

[20] Aybar, A. and Iftar, A., ”Deadlock Avoidance Controller Design for Timed Petri Nets Using Stretching”, *IEEE Systems Journal,*, vol. 2, no. 2, pp. 178-188. 2008.