



## A Mathematical Model Using Time Elements for Timed-Arc Petri Nets

Alpaslan YUFKA<sup>1,\*</sup> , Hanife Apaydin OZKAN<sup>2</sup> , Aydin AYBAR<sup>2</sup> 

<sup>1</sup>Defence Systems Technologies Division, Teknopark Ankara ASELSAN Inc., 06378, Ankara, Turkey

<sup>2</sup>Department of Electrical and Electronics Engineering, Eskisehir Technical University, 26555, Eskisehir, Turkey

### Highlights

- Modeling and analyzing Discrete Event Systems using Timed Petri Nets.
- New graphical and mathematical model for Timed-Arc Petri Nets.
- Comparison with Stretched Petri Nets.

### Article Info

Received: 03/01/2020

Accepted: 04/06/2020

### Keywords

Timed-Arc Petri Nets

Time element

Behavioral properties

Stretched Petri Nets

### Abstract

In this work, a novel graphical and mathematical model is introduced for Timed-Arc Petri Nets. In this model, operation durations related to events are associated with arcs as firing durations. Time elements are introduced on the arcs of the model for monitoring tokens in operation durations (firing durations) of firing processes (events). In the model, the status of the system is represented as states consisting of remaining time vector for indicating the status of time elements and marking vector for indicating the status of places' state information. This feature of states allows to obtain reachability set enhanced with the time information and the timed reachability graph of the system. In this work, behavioral properties of the proposed model are also defined. Moreover, via considering real world systems, behavioral properties are analyzed and performance of the model is compared with Stretched Petri Nets which is a recently proposed type of Timed Petri Nets.

## 1. INTRODUCTION

Today's large-scale and complex systems often comprise subsystems including many components. All of these components work in conjunction with each other to satisfy the current need for automation and data exchange in manufacturing technologies. These sophisticated systems are generally event-driven which are expressed by sequential events in the course of time and named as Discrete-Event Systems (DESs) [1]. Petri nets (PNs) are frequently used for modeling and analyzing DESs [1-4]. The first PN was introduced as untimed place/transition net which is not sufficient to express the dynamics of systems, since time is not associated with any event or situation [1, 4, 5]. To deal with this deficiency, time information is associated with the components of PNs, which are places, transitions, arcs and also tokens [5-11] yielding Timed PN [12].

In Timed PNs, time is associated with events and events are associated with transitions; thus, the reasonable way is to assign time delays to transitions [13]. Consequently, system states are defined by flow of tokens by the time. But these time-dependent movements, thus states, cannot be tracked graphically and calculated mathematically in the course of time throughout the events due to occurring invisibility of tokens during firing processes. Moreover, this model is insufficient to express distinct outputs, which are related to the same event and have different time delays, and it is also inadequate to express the dynamic operations such as transportation, motion, etc. These dynamics can be easily transferred into the model by associating time with arcs, yielding Timed-Arc PNs. In the literature, time is associated with arcs and tokens [7, 14-18] or with only arcs [6, 12, 19, 20] in Timed-Arc PNs. But in these works, time is not associated with events which would make the model natural and suitable for real life. In this work, a graphical and mathematical model is introduced for Timed-Arc PN model, and it is called as TdAPN. In TdAPN, places denote

\*Corresponding author, e-mail: ayufka@eskisehir.edu.tr or ayufka@aselsan.com.tr

conditions, transitions denote events. Operation durations (firing durations) of events (firing processes), are associated with arcs. First steps of graphical side of TdAPN have been taken by the authors in [21] as hourglass-representation and in [22-25] as time element representation. In the present work, time elements attached to arcs is considered. This allows monitoring of tokens in transitions throughout the firing processes providing complete picture of status of the system which is required in many practical systems avoiding a drawback of Timed PNs. The status of the system is represented as states consisting of remaining time vector for indicating the status of time elements and marking vector for indicating the status of places' state information. This feature of states allows to obtain reachability set/graph enhanced with the time information. Compared to previous works [13, 21, 22, 24, 26-30], TdAPN simplifies graphical representation of the model, more importantly reduce the size of states and computational complexity that ease the construction of the reachability set providing easier analysis. Reachability sets of TdAPN models are constructed on MATLAB according to the proposed mathematical model. In this work, conventional behavioral properties such as boundedness, safeness, liveness, deadlock, reversibility, moreover a new state property namely dynamicity, are also defined for TdAPN.

Performance of TdAPN model is observed by comparing the total time needed for construction of the net and reachability set of TdAPN with that of Stretched PNs which is a type of Timed PN and based on the methodology of transition-stretching or place-stretching by introducing pairs of place-transition or transition-place for time delays into the model, respectively. For these simulations authors' MATLAB software [25, 31] is used, because as far as authors' knowledge, any software tool which constructs the reachability set of the given Timed PN and calculates the computational time does not exist in the literature.

The paper is organized as follows: The proposed TdAPN model including the mathematical and graphical side is introduced in Section II, behavioral properties of TdAPN are defined in Section III, real-world case studies are given in Section IV and conclusions are presented in Section IV.

## 2. TIMED-ARC PETRI NETS

This section introduces graphical and mathematical model of proposed Timed-Arc PN model, namely TdAPN.

### 2.1. Mathematical Model and Definition

TdAPN is defined by a tuple  $G(P, T, N, O, D, S_0)$ . Here,  $P$  is the set of *places*, denoting conditions,  $T$  is the set of *transitions*, denoting events, ( $P \cap T = \emptyset$  and  $P \cup T \neq \emptyset$ ),  $N: P \times T \rightarrow \mathbb{N}$  is the *input matrix* that specifies the arcs directed from places to transitions, namely *ingoing arcs*,  $O: P \times T \rightarrow \mathbb{N}$  is the *output matrix* that specifies the arcs directed from transitions to places, namely *outgoing arcs*. If there exist a directed arc from a place  $p \in P$  to a transition  $t \in T$ , this place is called as *input place* of the corresponding transition. The set of input places of a transition  $t$  is represented by  $\bullet t = \{p \in P | N(p, t) \neq 0\}$ . If there exist a directed arc from a transition  $t \in T$  to a place  $p \in P$ , this place is called as *output transition* of the corresponding place. The set of output places of a transition  $t$  is represented by  $t\bullet = \{p \in P | O(p, t) \neq 0\}$ . In this work, time is discretized into *time slots*, i.e.,  $ts$ , by using an appropriate sampling period. Time delays in  $G$  are firing durations related to firing processes and represent the time passed from starting to ending time of a firing process of a transition. Through a firing process of a transition, corresponding number of tokens flow through connected arcs. These tokens are called as *flowing tokens*. In order to monitor and represent flowing tokens during firing processes, a triangular-formed component, namely *time elements*, are introduced mathematically and graphically in TdAPN. Time elements are attached to outgoing arcs for monitoring status of flowing tokens and denote ongoing operations related to events.

In TdAPN, an event occurred at the present time affects the net at the next time (after one  $ts$  elapses); therefore, 1  $ts$  delay is associated for all ingoing arcs in this paper. In  $G$ , time delays of outgoing arcs are represented by *time delay matrix*, i.e.,  $D: P \times T \rightarrow \mathbb{N}$ .  $D(p, t)$  denotes the time delay of an outgoing arc directed from the transition  $t \in T$  to the place  $p \in P$  and calculated by subtracting 1  $ts$  from real delay of event since 1  $ts$  delay is already associated with ingoing arc.

A time element on an outgoing arc from the transition  $t \in T$  to a place  $p \in t\bullet$  is denoted by  $h_p^t$  and the number of time elements introduced to a TdAPN is equal to the number of outgoing arcs with nonzero time delay. Hence, the set of time elements in a TdAPN is  $\nabla := \{h_p^t | O(p, t) \neq 0 \text{ and } D(p, t) \neq 0\}$ . In Figure 1, a time element  $h_p^t \in \nabla$  is represented. In this representation, the number of flowing tokens at time  $k$  is given inside of the time element, the right corner with filled triangular indicates the time delay of the outgoing arc, the left corner indicates the remaining time of flowing tokens at time  $k$ ; the line at the bottom corner indicates the weight of the outgoing arc.

The state of TdAPN at time  $k$  is represented by  $S(k) := \{\mathbf{M}(k), \nabla^R(k)\}$  Here,  $\mathbf{M}: P \rightarrow \mathbb{N}$  is the marking vector representing the status of places' state information, thus tokens at places at time  $k$ ;  $\nabla^R(k): \nabla \rightarrow \mathbb{N}$  is the remaining time vector representing the status of time elements, thus flowing tokens at time elements at time  $k$ . Note that, in this work,  $M(k, p)$  represents the number of tokens at place  $p \in P$  at time  $k$  while  $\nabla^R(k, h_p^t)$  represents remaining time of flowing tokens at time element  $h_p^t$  at time  $k$ . Moreover, state of  $G$  at the initial time, namely initial state, is indicated as  $S_0 := \{\mathbf{M}_0, \nabla^{R_0}\}$ .

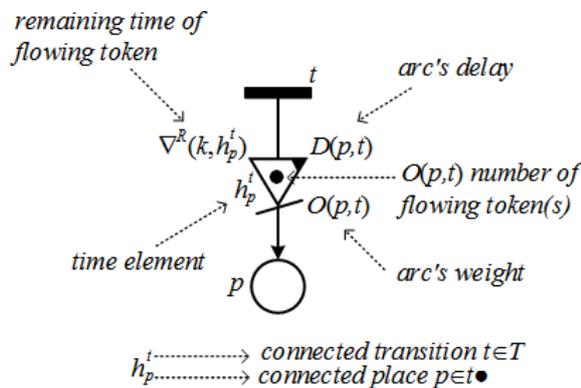


Figure 1. Representation of a time element  $h_p^t$

### 2.2. Firing Process and Enabledness

In TdAPN, a firing process started by firing transition  $t \in T$  at time  $k = \lambda \in \mathbb{N}$  is symbolized as  $t^\lambda$ . The time  $\lambda$  is also the *starting-time* for the firing process  $t^\lambda$ . TdAPN allows to associate different time delays to each outgoing arcs of a transition. That is, in a firing process  $t^\lambda$  corresponding tokens may arrive output place  $p \in t\bullet$  at different times which is calculated by  $k = \lambda + (1 + D(p, t))$  for each  $p \in t\bullet$ . The time  $\lambda + (1 + D(p, t))$  is the *ending time* for the output place  $p \in t\bullet$ . Note that, 1 *ts* comes from the time delay among ingoing arcs. The *total duration* of the firing process  $t^\lambda$ , i.e.,  $d_t \in \mathbb{N}$ , is the maximum ending time among all output places  $p \in t\bullet$  and calculated by  $d_t := \max_{p \in t\bullet} \{1 + D(p, t)\}$ . Hence,  $t^\lambda$  ends completely at time  $k = \lambda + d_t$  which is called *ending-time* for  $t^\lambda$ .

In TdAPN an active firing processes at any time  $k$  can be *previously activated firing process* or *newly activated firing process*. Previously activated firing processes ( $t^\lambda$  for  $k < \lambda$ ) at time  $k$  are the firing processes activated before time  $k$  and not finished yet, the set of which is  $F_{pre}(k) := \{t^\lambda | \lambda < k < \lambda + d_t\}$ . Newly activated processes at time  $k$  ( $t^\lambda$  for  $k = \lambda$ ) are the firing processes started at time  $k = \lambda$ , the set of which is  $F_{start}(k) := \{t^\lambda | k = \lambda\}$ . The set of all activated firing processes at time  $k$  is represented by  $F(k) := F_{pre}(k) \cup F_{start}(k) = \{t^\lambda | \lambda \leq k < \lambda + d_t\}$ . A firing processes related to a transition can be activated at a time if and only if the associated event can occur at that time, i.e., the transition is *firable*, namely *enable* at that time. In a physical system, a machine, robot etc. is mostly unavailable while an event related to it is in progress. Hence, a transition  $t$  is not enabled in TdAPN while any firing process related to it, thus a  $t^\lambda$ , continues. Under this assumption, in TdAPN a transition  $t \in T$  is *enabled* at time  $k$  if and only if any  $t^\lambda \notin F_{pre}(k)$  and the following condition is satisfied:

$$M(k, p) \geq N(p, t) \geq 1, \forall p \in \bullet t . \tag{1}$$

Here,  $N(p, t)$  denotes the weight of an outgoing arc from the transition  $t \in T$  to the place  $p \in P$ , and  $M(k, p)$  denotes the number of tokens at the place  $p \in P$  at time  $k$ . The set of enabled transitions at time  $k$  is represented by  $E(G, k)$ . More than one transition  $t \in E(G, k)$  can be simultaneously enabled at time  $k$  at the marking vector  $\mathbf{M}(k)$ . Transitions in a set  $\phi \subseteq E(G, k)$  is *simultaneously enabled* if and only if the following condition is satisfied:

$$M(k, p) \geq \sum_{t \in \phi} N(p, t), \forall p \in P . \quad (2)$$

Here,  $\phi$  is the set of simultaneously-enabled transitions and may contain more than one transition as well as a single transition. The set of sets of simultaneously-enabled transitions at time  $k$  is represented by  $\hat{E}(G, k)$ , and  $\hat{E}(G, k) \subset 2^{E(G, k)} \setminus \emptyset$  while  $\phi \in \hat{E}(G, k)$ . Here,  $2^{\{\cdot\}}$  denotes the power set of  $\{\cdot\}$ , and  $\hat{E}(G, k)$  is computed at the marking vector  $\mathbf{M}(k)$  at time  $k$ .  $\hat{E}(G, k)$  includes all possible enabled set of transitions at time  $k$  except an empty set. In this study, any transition or any set of simultaneously enabled transitions fire immediately as they become enabled.

### 2.3. Next State

In TdAPN, when enabled transitions of the set  $\phi \in \hat{E}(G, k)$  fire at a time  $k$ , firing processes of these transitions start at time  $k$  and are added into the set  $F_{start}(k)$ . The next state is calculated by considering both newly activated firing processes  $F_{start}(k)$  and previously activated firing processes  $F_{pre}(k)$ , thus considering all firing processes in the set  $F(k) := F_{pre}(k) \cup F_{start}(k)$ .

In TdAPN, for time  $k$ , the next state, that is the state at time  $k + 1$ , consists of the next marking vector  $\mathbf{M}(k + 1)$ , and the next remaining time vector  $\nabla^R(k + 1)$ , and represented by  $S(k + 1) := \{\mathbf{M}(k + 1), \nabla^R(k + 1)\}$ . The time  $k$  is a natural number and is increased by one  $ts$  to reach the next time  $k + 1$ . The components of the next marking vector at time  $k + 1$  for a place  $p \in P$ , i.e.,  $M(k + 1, p)$ , and the components of next remaining time vector at time  $k + 1$  for a time element  $h_p^t \in \nabla$ , i.e.,  $\nabla^R(k + 1, h_p^t)$ , are computed as follows:

$$M(k + 1, p) := M(k, p) + \sum_{t \in F(k)} (\delta[k - (\lambda + D(p, t))] \cdot O(p, t) - \delta[k - \lambda] \cdot N(p, t)) . \quad (3)$$

and

$$\nabla^R(k + 1, h_p^t) := \nabla^R(k, h_p^t) + \sum_{t \in F(k)} (\delta[k - \lambda] \cdot D(p, t) - \sum_{l=\lambda+1}^{\lambda+D(p, t)} \delta[k - l]) . \quad (4)$$

Here,  $\delta[k] \in \{0, 1\}$  denotes the discrete-time unit impulse function.

### 3. BEHAVIORAL PROPERTIES

In TdAPN, the status of the system is represented as states consisting of remaining time vectors and marking vectors. This feature of states allows to obtain reachability set/graph enhanced with the time information and eases the analysis of behavioral properties of the model.

Before defining these behavioral properties for TdAPN, let we introduce the reachability set and timed reachability tree for TdAPN: In TdAPN, the state of  $G$  at time  $k$  was defined in previous section as  $S(k) = \{\mathbf{M}(k), \nabla^R(k)\}$ , and the initial state of  $G$  at time  $k_0$  is represented by  $S_0$ . The set of all reachable states from  $S_0$ , namely *reachability set*, is denoted by  $R(G, S_0) = \{S_0, S_1, \dots\}$  (e.g., see, Table 1). Here, each obtained different state  $S(k)$  from  $S_0$  is represented by  $S_j$ , where  $j = 0, 1, \dots, |R(G, S_0)| - 1$ . Though elements of  $R(G, S_0)$  are given without  $k$  notation, each state in  $R(G, S_0)$  correspond to a state obtained at a time  $k$ . For the sake of simplicity, marking vector  $\mathbf{M}(k)$  and remaining time vector  $\nabla^R(k)$  will be represented as  $\gamma_1(S(k))$  and  $\gamma_2(S(k))$ , respectively. Thus,  $\mathbf{M}(k) = \gamma_1(S(k))$  and  $\nabla^R(k) = \gamma_2(S(k))$ .

The relation between the elements of the set  $R(G, S_0)$  can be represented by a graph, namely *timed-reachability graph/tree*. Each arrow in the timed-reachability graph indicates 1 *ts* (e.g., see, Figure 3). In this work, enabled transitions in a set  $\phi \in \hat{E}(G, k)$  are fired as soon as they are enabled at  $\gamma_1(S(k))$ ; thus, it is possible to find the minimum time to reach a state from any state using the timed-reachability graph.

Conventional behavioral properties of PNs [32], i.e., boundedness, safeness, liveness, deadlock, reversibility, and a new behavioral property, namely *dynamicity*, are defined for TdAPN for the first time in the literature as follows:

**Definition 1 (Dynamicity).** Typically, states of TdAPN are divided into two types, such as the relaxed states and the dynamic states.

A state  $S(k) \in R(G, S_0)$  is called as *relaxed state* if  $\gamma_2(S(k)) = \mathbf{0}^{|\nabla| \times 1}$ . Here,  $\mathbf{0}^{|\nabla| \times 1}$  represents a  $|\nabla| \times 1$  sized zeros vector. Relaxed states preserve their status and does not yield another state until an enabled set  $\phi \in \hat{E}(G, k)$  at  $\gamma_1(S(k))$  is selected to be fired.

A state  $S(k) \in R(G, S_0)$  is called as *dynamic state* if  $\gamma_2(S(k)) \neq \mathbf{0}^{|\nabla| \times 1}$ . Dynamic states indicate that at least one firing process of a transition is active ( $F_{pre}(k) \neq \emptyset$ ); as a result, an event related to this firing process is still in progress and flowing token(s) exist in the system. The dynamic states do not preserve their status and lead the system to another state even if any enabled set  $\phi \in \hat{E}(G, k)$  is not selected to be fired.

**Definition 2 (Boundedness and Safeness).**  $G$  is said to be bounded if there exists a bound vector  $\mathbf{B}: P \rightarrow \mathbb{N}$  such that  $\gamma_1(S_j)(p) \leq B(p), \forall p \in P, \forall S_j \in R(G, S_0)$ . Here,  $\gamma_1(S_j)(p)$  is the  $p$ 'th element of the marking vector of the state  $S_j \in R(G, S_0)$  and  $B(p)$  is the  $p$ 'th element of the bound vector  $\mathbf{B}$ .

**Definition 3 (Liveness).** A transition  $t \in T$  is said to be live if, for all states  $S_j \in R(G, S_0)$ , there exists  $S(k) \in R(G, S_j)$  such that  $t \in \phi$  while  $\phi \in \hat{E}(G, k)$  at  $\gamma_1(S(k))$ . Here, the set of all reachable states from  $S_j$  is denoted by  $R(G, S_j)$ . All live transitions are considered in the set  $\tau \subseteq T$ . Therefore, TdAPN is considered to be  $\tau$ -live if  $\tau \subset T$ , and  $G$  is considered to be live if  $\tau = T$ , which is  $T$ -live.

**Definition 4 (Deadlock).** The dynamic states yield a new state due to their nonzero remaining time vectors so that the dynamic states reaches a relaxed state at the end. Thus, the deadlock property of  $G$  is examined over the relaxed states. Any relaxed state  $S(k) \in R(G, S_0)$  is considered to be a deadlock state if  $\gamma_2(S(k)) = \mathbf{0}^{|\nabla| \times 1}$  and  $\hat{E}(G, k)$  is an empty set at  $\gamma_1(S(k))$ . Note that TdAPN is not live if it has any deadlock state.

**Definition 5 (Reversibility).** If the initial state  $S_0 \in R(G, S_j)$  is reachable from all reachable states  $S_j \in R(G, S_0)$ , then TdAPN is reversible. Note that TdAPN is not reversible if it has any deadlock state.

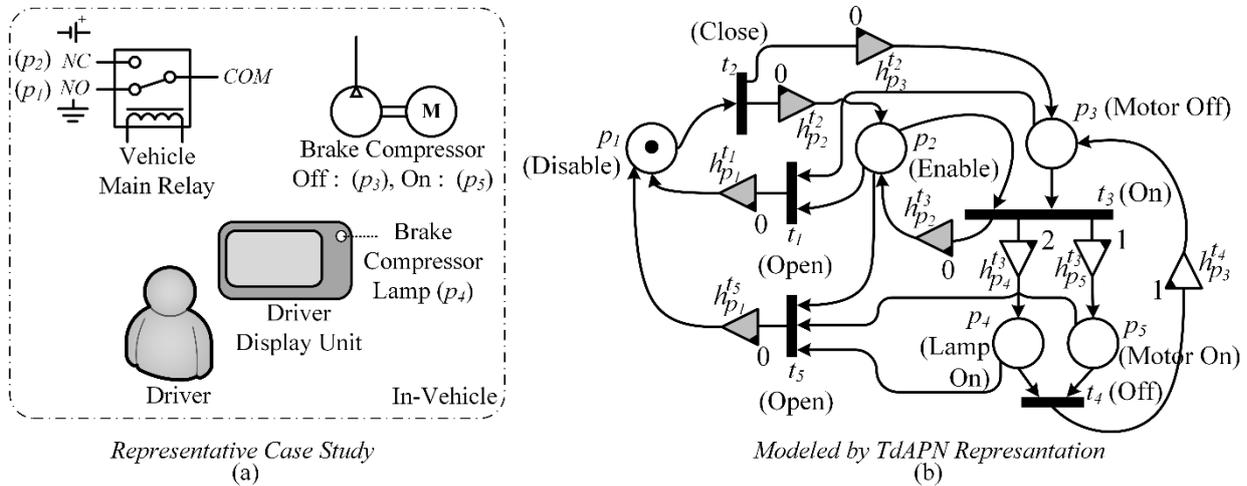
#### 4. CASE STUDIES

In this section, an air assisted brake system and a manufacturing system [13] are considered as case studies to demonstrate TdAPN methodology.

Performance of TdAPN models of these systems are compared with that of two other types of Timed PN: Transition-Stretched PNs [13, 26-30] and Place-Stretched PNs [29, 30]. In comparisons, the total computational time, which includes the construction of the net and its reachability set, is considered as the performance criterion. Reachability set of models are obtained by MATLAB software developed according to proposed mathematical model.

**4.1. Case Study I - Automotive System**

Start-stop mechanism of a compressor of air-assisted brake system is considered as a case study in this subsection. The representative simple sketch of starting the compressor is illustrated in Figure 2(a).



**Figure 2.** Representative automotive example and its model of TdAPN

Here in this figure, the vehicle is equipped with a main relay, which fully switches the electrical power of the vehicle. The vehicle is equipped with the air-assisted braking system whose air pressure is conditioned by the compressor with an electric motor. The driver can monitor the operation of the electric motor by means of a lamp via the driver display unit.

TdAPN model of the system is illustrated in Figure 2(b). In this model, the set of places is  $P = \{p_1, p_2, p_3, p_4, p_5\}$ , where  $p_1$  denotes the open-status of the main relay;  $p_2$  denotes the close-status of the main relay;  $p_3$  denotes the off-status of the electric motor;  $p_4$  denotes the on-status of the lamp on the driver display unit;  $p_5$  denotes the on-status of the electric motor. On the other hand, the set of transitions is  $T = \{t_1, t_2, t_3, t_4, t_5\}$ , where  $t_1$  denotes the event to open the main relay;  $t_2$  denotes the event to close the main relay when the electric motor is off;  $t_3$  denotes the event to start the electric motor;  $t_4$  denotes the event to stop the electric motor;  $t_5$  denotes the event to close the main relay when the electric motor is on. When start request is taken, that is  $t_3$  fires, the electric motor starts after 200 ms, and the lamp lights up after 300 ms. When stop request is taken, that is  $t_4$  fires, electric motor stops after 200 ms. Appropriate sampling period is chosen according to greatest common divisor of these times ( $1 \text{ ts} = 100 \times 10^{-3}$  seconds).

The input matrix  $N$ , the output matrix  $O$ , and the time-delay matrix  $D$  in terms of  $ts$  are as follows:

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

Note that, since time delay of each ingoing arcs are fixed 1  $ts$ , one minus of the above explained delays are associated with outgoing arcs, and the matrix  $D$  is determined correspondingly. Time elements are attached to the outgoing arcs directed from transitions  $t \in T$  to  $p \in P$  if  $O(p, t) \neq 0$  and  $D(p, t) \neq 0$ . Hence, the set of time elements for this net is  $\nabla = \{h_{p_4}^{t_3}, h_{p_5}^{t_3}, h_{p_3}^{t_4}\}$ , where  $h_{p_4}^{t_3}$  represents the delay of turning on the lamp,  $h_{p_5}^{t_3}/h_{p_3}^{t_4}$  represents the delay of starting/stopping the motor after request.

The state of  $G$  at the initial time  $k_0 = 0$  is  $S_0 = \{\mathbf{M}_0, \nabla^{R_0}\}$ , where  $\mathbf{M}_0 = [1 \ 0 \ 0 \ 0 \ 0]'$  and  $\nabla^{R_0} = [0 \ 0 \ 0]'$  (Here,  $[\cdot]'$  denotes the transpose of the vector  $[\cdot]$ ). The set of sets of simultaneously-enabled transitions at time 0 is  $\hat{E}(G, 0) = \{\{t_2\}\}$ , and the set of previously activated firing processes is  $F_{pre}(0) = \emptyset$ . If the set  $\phi = \{t_2\}$  of  $\hat{E}(G, 0)$  is selected to fire at time 0, then the set of newly activated firing processes is  $F_{start}(0) = \{t_2^\lambda\}$ ; as a result,  $F(0) = \{t_2^\lambda\}$  and the next state, i.e., the state at time 1, is obtained as  $S(1) = \{[1 \ 0 \ 0 \ 0 \ 0]', [0 \ 0 \ 0]'\}$  by using Equations (3) and (4). Note that, if no transition is selected to fire at time 0, then  $F_{start}(0) = \emptyset$  and no new state is obtained. In order to obtain all reachable states, the same procedure is applied to next states for obtaining new next states.

For this system, reachability set of TdAPN has 6 states, such as  $R(G, S_0) = \{S_0, S_1, S_2, S_3, S_4, S_5\}$  as given in Table 1.

**Table 1.** Reachability set for TdAPN given in Figure 2(b)

$k$	$S_j$	$\{\gamma_1(S_j), \gamma_2(S_j)\}$	$\phi$	Next State
0	$S_0$	$\{[1 \ 0 \ 0 \ 0 \ 0]', [0 \ 0 \ 0]'\}$	-	$S_0$
			$\{t_2\}$	$S_1$
1	$S_1$	$\{[0 \ 1 \ 1 \ 0 \ 0]', [0 \ 0 \ 0]'\}$	-	$S_1$
			$\{t_1\}$	$S_0$
			$\{t_3\}$	$S_2$
2	$S_2$	$\{[0 \ 1 \ 0 \ 0 \ 0]', [2 \ 1 \ 0]'\}$	-	$S_3$
3	$S_3$	$\{[0 \ 1 \ 0 \ 0 \ 1]', [1 \ 0 \ 0]'\}$	-	$S_4$
4	$S_4$	$\{[0 \ 1 \ 0 \ 1 \ 1]', [0 \ 0 \ 0]'\}$	-	$S_4$
			$\{t_4\}$	$S_5$
			$\{t_5\}$	$S_0$
5	$S_5$	$\{[0 \ 1 \ 0 \ 0 \ 0]', [0 \ 0 \ 1]'\}$	-	$S_0$

The first column of Table 1 gives the time in terms of  $ts$  to reach the state  $S_j \in R(G, S_0)$  from  $S_0$ . The second column gives the state-label, i.e.,  $S_j = \{\mathbf{M}_j, \nabla^{R_j}\}$ . The third column gives the marking vector  $\mathbf{M}_j = \gamma_1(S_j)$  and the remaining time vector  $\nabla^{R_j} = \gamma_2(S_j)$ . The fourth column gives the selection of the set  $\phi \in \hat{E}(G, k)$  at  $\gamma_1(S_j)$ . The fifth column gives the next state from the present state  $S_j$  according to the selection of the set  $\phi$ . Tree representation of  $R(G, S_0)$  of the net is given in Figure 3. In this tree, each arrow has an elapsed 1  $ts$  time delay, and the choice of enabled transitions with regard to the enabled set  $\phi \in \hat{E}(G, k)$  at  $\gamma_1(S_j)$  are also shown next to the arrows in bold. Using this time information, it is possible to determine how many  $ts$  required for reaching from one state to another. For example, the state  $S_4$  is reachable from the initial state  $S_0$  by the path  $(S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4)$  of  $\{t_2, t_3\}$  after 4  $ts$  has elapsed. Note that, this is the minimum time duration to reach  $S_4$  from  $S_0$ .

Behavioral properties of the system can be analyzed easily through the reachability set or reachability tree. For example, states  $S_0, S_1$  and  $S_4$  are relaxed states since their remaining time vectors are zero vector, that is, they are same as the states of the equivalent untimed PN.  $S_2, S_3$  and  $S_5$  are dynamic states, since their remaining time vectors are different from zero vector. As it is clear from reachability tree in Figure 3 (where relaxed states are illustrated by octagonal boxes while dynamic states are illustrated in circular boxes), relaxed states do not change if any  $\phi$  is not selected, while dynamic states change by time resulting another state, even if any  $\phi$  is not selected. Moreover, through reachability set one can conclude that TdAPN is bounded for  $\mathbf{B} = [1 \ 1 \ 1]'$  and safe since each state in the reachability set has at most 1 token. There is no deadlock state in  $R(G, S_0)$ , and the given TdAPN is live and also reversible.

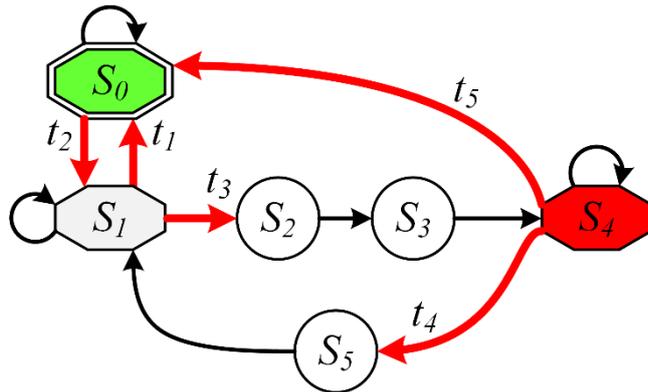


Figure 3. Timed-reachability tree for TdAPN in Figure 2(b)

In order to observe the performance of TdAPN for this system, computational times of reachability sets for Stretched PNs and TdAPN are compared. For this system construction of TdAPN model takes 1.4728 seconds, while reachability set is obtained by 0.0946 seconds (totally, 1.5674 seconds). For Transition-Stretched PN/Place-Stretched PN models of this system, construction of the model takes 1.7533/1.7076 seconds, while reachability set is obtained by 0.1119 /0.1978 seconds. That is, total time for creation of reachability tree is 1.8652 seconds for Transition-Stretched PN model and 1.9054 seconds for Place-Stretched PN model. So, for this system reachability set of TdAPN is created 18.99% shorter time than that of Transition-Stretched PN, and 21.56% shorter time than that of Place-Stretched PN.

#### 4.2. Case Study II - Manufacturing System

The manufacturing-example that is modeled by Timed PN and Transition-Stretched PN in [13] is considered as a case study in this subsection. The system, illustrated in Figure 4(a), comprises: a machine; a buffer whose capacity is limited to storing a single part; one main store whose storage capability is limited to two parts; two flat pallets to transfer parts; and an industrial robot. Each pallet is able to transport only one part in one go. TdAPN model of this system is illustrated in Figure 4(b).

In this model, the set of places is  $P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}\}$ , where  $p_1$  denotes the number of available pallets,  $p_2$  denotes the operation of the machine (if any token exists, it is completed),  $p_3$  denotes the availability of the machine (or the part is produced) (if any token exists, it is available),  $p_4$  denotes that the robot is assigned to unload the machine (if any token exists, it is assigned),  $p_5$  denotes the fullness of the buffer (if any token exists, it is full),  $p_6$  denotes the availability of the buffer (if any token exists, it is available),  $p_7$  denotes that the robot is assigned to transfer the produced part to an unoccupied room in the main store (if any token exists, it is assigned),  $p_8$  denotes the number of occupied rooms for storage in the main store,  $p_9$  denotes the status of the robot (if any token exists, it is free),  $p_{10}$  denotes the number of unoccupied rooms for storage in the main store. The set of transitions is  $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ , where  $t_1$  is the event to start the machine in order to produce a part,  $t_2$  is the event to make the robot unload the produced part,  $t_3$  is the event to make the robot load the produced part to the buffer,  $t_4$  is the event to make the robot unload the buffer,  $t_5$  is the event to set a pallet free and make the robot transfer the produced part from buffer to an unoccupied room of main store,  $t_6$  is the event to make the robot unload one part from the main store (or to make room for one part in the main store). The input matrix  $N$ , the output matrix  $O$ , and the time-delay matrix  $D$  are as follows:

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

The set of time elements is  $\nabla = \{h_{p_2}^{t_1}, h_{p_1}^{t_5}, h_{p_8}^{t_5}, h_{p_3}^{t_5}, h_{p_9}^{t_6}, h_{p_{10}}^{t_6}\}$ . Here,  $h_{p_2}^{t_1}$  denotes that the machine is producing a part, and this operation takes 2 ts,  $h_{p_1}^{t_5}$  denotes that the pallet is going to be free, and this operation takes 1 ts,  $h_{p_8}^{t_5}$  denotes that the robot is transferring the produced part to an unoccupied room of main store, and this operation takes 1 ts,  $h_{p_9}^{t_6}$  denotes that the robot is going to be available after it transfers the produced part, and this operation takes 1 ts,  $h_{p_{10}}^{t_6}$  denotes that the robot is going to be available after it unloads one part, and this operation takes 1 ts,  $h_{p_3}^{t_5}$  denotes that the robot is unloading one part from the main store, and this operation takes 1 ts.  $S_0 = \{\mathbf{M}_0, \nabla^{R_0}\}$  is the initial state of  $G$  at  $k_0 = 0$ , where  $\mathbf{M}_0 = [2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 2]'$  and  $\nabla^{R_0} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]'$ . Tokens in  $\mathbf{M}_0$  mean that;  $M(0, p_1) = 2$  indicates that initially two pallets are available;  $M(0, p_3) = 1$  denotes the machine is available;  $M(0, p_6) = 1$  shows the buffer is available;  $M(0, p_9) = 1$  denotes the robot is available; and  $M(0, p_{10}) = 2$  shows that there are two unoccupied rooms for storage in the main store, where its maximum storage capacity is 2 parts.  $\nabla^{R_0} = \mathbf{0}^{|\nabla| \times 1}$  indicates that there is no dynamic operation currently.

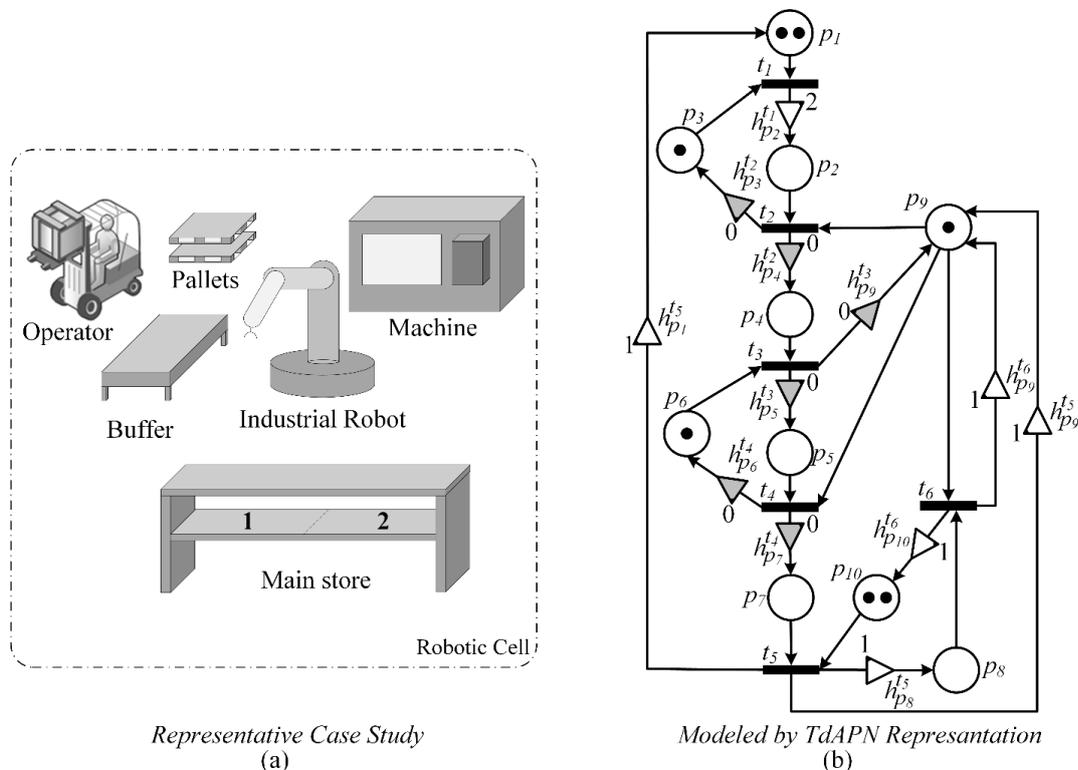


Figure 4. Representative manufacturing example and its model of TdAPN

Reachability set of TdAPN model has 75 states as given in Table 2, i.e.,  $R(G, S_0) = \{S_0, S_1, \dots, S_{74}\}$ . The corresponding timed-reachability tree is shown in Figure 5. As it is clear from the reachability tree, TdAPN has 27 relaxed states:  $\{S_0, S_3, S_4, S_6, S_{10}, S_{13}, S_{14}, S_{15}, S_{20}, S_{21}, S_{24}, S_{29}, S_{31}, S_{35}, S_{41}, S_{42}, S_{43}, S_{49}, S_{50}, S_{53}, S_{58}, S_{60}, S_{64}, S_{70}, S_{71}, S_{72}, S_{74}\}$  and the states  $S_{20}, S_{49}, S_{72}, S_{74}$  are deadlocks. Hence the net is neither

live nor reversible. As for performance observation: Construction times of TdAPN/Transition-Stretched PN/Place-Stretched PN models are 1.3337/1.9399/1.8475 seconds, while total reachability set construction take 2.8643/7.3033/20.9044 seconds. So, reachability set of TdAPN is created much shorter time than that of Transition-Stretched PN and Place-Stretched PN models.

**Table 2.** Reachability set for TdAPN given in Figure 4(b)

$k$	$S_j$	$\{M, \nabla^R\}$	$\phi$	Next State
0	$S_0$	$\{[2010010012]', [00000]'\}$	-	$S_0$
			$\{t_1\}$	$S_1$
1	$S_1$	$\{[1000010012]', [20000]'\}$	-	$S_2$
2	$S_2$	$\{[1000010012]', [10000]'\}$	-	$S_3$
3	$S_3$	$\{[1100010012]', [00000]'\}$	-	$S_3$
			$\{t_2\}$	$S_4$
4	$S_4$	$\{[1011010002]', [00000]'\}$	-	$S_4$
			$\{t_1\}$	$S_5$
			$\{t_3\}$	$S_6$
			$\{t_1, t_3\}$	$S_7$
5	$S_5$	$\{[0001010002]', [20000]'\}$	-	$S_8$
			$\{t_3\}$	$S_9$
5	$S_6$	$\{[1010100012]', [00000]'\}$	-	$S_6$
			$\{t_1\}$	$S_7$
			$\{t_4\}$	$S_{10}$
			$\{t_1, t_4\}$	$S_{11}$
5	$S_7$	$\{[0000100012]', [20000]'\}$	-	$S_9$
			$\{t_4\}$	$S_{12}$
6	$S_8$	$\{[0001010002]', [10000]'\}$	-	$S_{13}$
			$\{t_3\}$	$S_{14}$
6	$S_9$	$\{[0000100012]', [10000]'\}$	-	$S_{14}$
			$\{t_4\}$	$S_{15}$
6	$S_{10}$	$\{[1010011002]', [00000]'\}$	-	$S_{10}$
			$\{t_1\}$	$S_{11}$
			$\{t_5\}$	$S_{16}$
			$\{t_1, t_5\}$	$S_{17}$
6	$S_{11}$	$\{[0000011002]', [20000]'\}$	-	$S_{12}$
			$\{t_5\}$	$S_{18}$
6	$S_{12}$	$\{[0000011002]', [10000]'\}$	-	$S_{15}$
			$\{t_5\}$	$S_{19}$
7	$S_{13}$	$\{[0101010002]', [00000]'\}$	-	$S_{13}$
			$\{t_3\}$	$S_{14}$
7	$S_{14}$	$\{[0100100012]', [00000]'\}$	-	$S_{14}$
			$\{t_2\}$	$S_{20}$
			$\{t_4\}$	$S_{15}$
7	$S_{15}$	$\{[0100011002]', [00000]'\}$	-	$S_{15}$
			$\{t_5\}$	$S_{19}$
7	$S_{16}$	$\{[2010010001]', [01100]'\}$	-	$S_{21}$
			$\{t_1\}$	$S_{22}$
7	$S_{17}$	$\{[1000010001]', [21100]'\}$	-	$S_{23}$
7	$S_{18}$	$\{[1000010001]', [11100]'\}$	-	$S_{24}$
7	$S_{19}$	$\{[1100010001]', [01100]'\}$	-	$S_{24}$
8	$S_{20}$	$\{[0011100002]', [00000]'\}$	-	$S_{20}$
8	$S_{21}$	$\{[2010010111]', [00000]'\}$	-	$S_{21}$
			$\{t_1\}$	$S_{22}$
			$\{t_6\}$	$S_{25}$
			$\{t_1, t_6\}$	$S_{26}$
8	$S_{22}$	$\{[1000010111]', [20000]'\}$	-	$S_{23}$
			$\{t_6\}$	$S_{27}$
8	$S_{23}$	$\{[1000010111]', [10000]'\}$	-	$S_{24}$
			$\{t_6\}$	$S_{28}$
8	$S_{24}$	$\{[1100010111]', [00000]'\}$	-	$S_{24}$
			$\{t_2\}$	$S_{29}$
			$\{t_6\}$	$S_{28}$
9	$S_{25}$	$\{[2010010001]', [00011]'\}$	-	$S_0$
			$\{t_1\}$	$S_{11}$
9	$S_{26}$	$\{[1000010001]', [20011]'\}$	-	$S_2$
9	$S_{27}$	$\{[1000010001]', [10011]'\}$	-	$S_3$
9	$S_{28}$	$\{[1100010001]', [00011]'\}$	-	$S_3$
9	$S_{29}$	$\{[1011010101]', [00000]'\}$	-	$S_{29}$
			$\{t_1\}$	$S_{30}$
			$\{t_3\}$	$S_{31}$
			$\{t_1, t_3\}$	$S_{32}$
10	$S_{30}$	$\{[0001010101]', [20000]'\}$	-	$S_{33}$
			$\{t_3\}$	$S_{34}$

**Table 2 (Continue).** Reachability set for TdAPN given in Figure 4(b)

$k$	$S_j$	$\{M, \nabla^R\}$	$\phi$	Next State
10	$S_{31}$	$\{[1010100111]', [00000]'\}$	-	$S_{31}$
			$\{t_1\}$	$S_{32}$
			$\{t_4\}$	$S_{35}$
			$\{t_6\}$	$S_{36}$
			$\{t_1, t_4\}$	$S_{37}$
10	$S_{32}$	$\{[0000100111]', [20000]'\}$	-	$S_{34}$
			$\{t_4\}$	$S_{39}$
			$\{t_6\}$	$S_{40}$
11	$S_{33}$	$\{[0001010101]', [10000]'\}$	-	$S_{41}$
11	$S_{34}$	$\{[0000100111]', [10000]'\}$	-	$S_{42}$
			$\{t_4\}$	$S_{43}$
11	$S_{35}$	$\{[1010011101]', [00000]'\}$	-	$S_{35}$
			$\{t_1\}$	$S_{37}$
			$\{t_5\}$	$S_{45}$
			$\{t_1, t_5\}$	$S_{46}$
11	$S_{36}$	$\{[1010100001]', [00011]'\}$	-	$S_6$
			$\{t_1\}$	$S_7$
11	$S_{37}$	$\{[0000011101]', [20000]'\}$	-	$S_{39}$
11	$S_{38}$	$\{[0000100001]', [20011]'\}$	-	$S_9$
			$\{t_5\}$	$S_{47}$
11	$S_{39}$	$\{[0000011101]', [10000]'\}$	-	$S_{43}$
			$\{t_5\}$	$S_{48}$
11	$S_{40}$	$\{[0000100001]', [10011]'\}$	-	$S_{14}$
12	$S_{41}$	$\{[0101010101]', [00000]'\}$	-	$S_{41}$
			$\{t_3\}$	$S_{42}$
12	$S_{42}$	$\{[0100100111]', [00000]'\}$	-	$S_{42}$
			$\{t_2\}$	$S_{49}$
			$\{t_4\}$	$S_{43}$
			$\{t_6\}$	$S_{44}$
			$\{t_1, t_6\}$	$S_{44}$
12	$S_{43}$	$\{[0100011101]', [00000]'\}$	-	$S_{43}$
			$\{t_5\}$	$S_{48}$
12	$S_{44}$	$\{[0100100001]', [00011]'\}$	-	$S_{14}$
12	$S_{45}$	$\{[2010010100]', [01100]'\}$	-	$S_{50}$
			$\{t_1\}$	$S_{51}$
12	$S_{46}$	$\{[1000010100]', [21100]'\}$	-	$S_{52}$
12	$S_{47}$	$\{[1000010100]', [11100]'\}$	-	$S_{53}$
12	$S_{48}$	$\{[1100010100]', [01100]'\}$	-	$S_{53}$

$k$	$S_j$	$\{M, \nabla^R\}$	$\phi$	Next State
13	$S_{49}$	$\{[0011100101]', [00000]'\}$	-	$S_{49}$
13	$S_{50}$	$\{[2010010210]', [00000]'\}$	-	$S_{50}$
			$\{t_1\}$	$S_{51}$
			$\{t_6\}$	$S_{54}$
			$\{t_1, t_6\}$	$S_{55}$
13	$S_{51}$	$\{[1000010210]', [20000]'\}$	-	$S_{52}$
			$\{t_6\}$	$S_{56}$
13	$S_{52}$	$\{[1000010210]', [10000]'\}$	-	$S_{53}$
13	$S_{53}$	$\{[1100010210]', [00000]'\}$	-	$S_{53}$
			$\{t_2\}$	$S_{58}$
			$\{t_6\}$	$S_{57}$
14	$S_{54}$	$\{[2010010100]', [00011]'\}$	-	$S_{21}$
			$\{t_1\}$	$S_{22}$
14	$S_{55}$	$\{[1000010100]', [20011]'\}$	-	$S_{23}$
14	$S_{56}$	$\{[1000010100]', [10011]'\}$	-	$S_{24}$
14	$S_{57}$	$\{[1100010100]', [00011]'\}$	-	$S_{24}$
14	$S_{58}$	$\{[1011010200]', [00000]'\}$	-	$S_{58}$
			$\{t_1\}$	$S_{59}$
			$\{t_3\}$	$S_{60}$
15	$S_{59}$	$\{[0001010200]', [20000]'\}$	-	$S_{62}$
			$\{t_3\}$	$S_{63}$
15	$S_{60}$	$\{[1010100210]', [00000]'\}$	-	$S_{60}$
			$\{t_1\}$	$S_{61}$
			$\{t_4\}$	$S_{64}$
			$\{t_6\}$	$S_{65}$
			$\{t_1, t_4\}$	$S_{66}$
			$\{t_1, t_6\}$	$S_{67}$
15	$S_{61}$	$\{[0000100210]', [20000]'\}$	-	$S_{63}$
			$\{t_4\}$	$S_{68}$
			$\{t_6\}$	$S_{69}$
16	$S_{62}$	$\{[0001010200]', [10000]'\}$	-	$S_{70}$
			$\{t_3\}$	$S_{71}$
16	$S_{63}$	$\{[0000100210]', [10000]'\}$	-	$S_{71}$
			$\{t_4\}$	$S_{72}$
			$\{t_6\}$	$S_{73}$
16	$S_{64}$	$\{[1010011200]', [00000]'\}$	-	$S_{64}$
			$\{t_1\}$	$S_{66}$
16	$S_{65}$	$\{[1010100100]', [00011]'\}$	-	$S_{31}$
			$\{t_1\}$	$S_{32}$
16	$S_{66}$	$\{[0000011200]', [20000]'\}$	-	$S_{68}$

**Table 2 (Continue).** Reachability set for TdAPN given in Figure 4(b)

$k$	$S_j$	$\{M, \nabla^R\}$	$\phi$	Next State
16	$S_{67}$	$\{[0000100100]', [20011]'\}$	-	$S_{34}$
16	$S_{68}$	$\{[0000011200]', [10000]'\}$	-	$S_{72}$
16	$S_{69}$	$\{[0000100100]', [10011]'\}$	-	$S_{42}$
17	$S_{70}$	$\{[0101010200]', [00000]'\}$	-	$S_{70}$
			$\{t_3\}$	$S_{71}$
17	$S_{71}$	$\{[0100100210]', [00000]'\}$	-	$S_{71}$
			$\{t_2\}$	$S_{74}$
			$\{t_4\}$	$S_{72}$
			$\{t_6\}$	$S_{73}$
17	$S_{72}$	$\{[0100011200]', [00000]'\}$	-	$S_{72}$
17	$S_{73}$	$\{[0100100100]', [00011]'\}$	-	$S_{42}$
18	$S_{74}$	$\{[0011100200]', [00000]'\}$	-	$S_{74}$

## 5. CONCLUSION

In this work, a graphical and mathematical model, namely TdAPN, is introduced for Timed-Arc PN model. In TdAPN time elements are attached to arcs, which allows monitoring of tokens in transitions throughout the firing processes providing complete picture of the system through evolution avoiding a drawback of Timed PNs. The status of the system is represented as states consisting of remaining time vector for indicating the status of time elements and marking vector for indicating the status of places' state information. This feature of states allows to obtain reachability set/graph enhanced with the time information. Compared to previous works [13, 21, 22, 24, 26-30], TdAPN simplifies graphical representation of the model, more importantly reduce the size of states that ease the construction of the reachability set providing easier analysis. Reachability sets of TdAPN models are constructed on MATLAB according to the proposed mathematical model. In this work, conventional behavioral properties such as boundedness, safeness, liveness, deadlock, reversibility, moreover a new state property namely dynamicity, are also defined for TdAPN.

Performance of TdAPN model is observed by comparing the total time needed for construction of the net and reachability set of TdAPN with that of Stretched PNs. Results have shown that TdAPN give admissible results. For these simulations authors' MATLAB software [25, 31] is used, because as far as authors' knowledge, any software tool which constructs the reachability set of the given Timed PN and calculates the computational time does not exist in the literature.

The concept of TdAPN is clear and concise. This approach is used with unit time, in terms of appropriate time slots, which are readily handled by computers in practical applications. Thus, any model constructed using TdAPNs could be easily implemented in specific time-delayed systems and related algorithms. Time durations are used such that exact time labels attached to outgoing arcs have no time intervals; the use of deterministic time values instead of time intervals can therefore provide less complexity and serve to decrease the computational time. It can also present fewer difficulties when implementing algorithms. Moreover, in the model, a sole firing process or multiple firing processes can start at any time.

The TdAPN provides an overall model for large-scale and complex systems. Thus, it is possible to obtain the reachability set and corresponding reachability tree so that the complete dynamic schema of the deterministically timed system can be depicted. This allows to point out problematic states in timed-delay dynamic systems, such as deadlocks. The future direction of this work will include pseudo algorithms to obtain the reachability set and a forbidden-state controller (for example, [33, 34]), such as a deadlock avoidance control, for TdAPN.

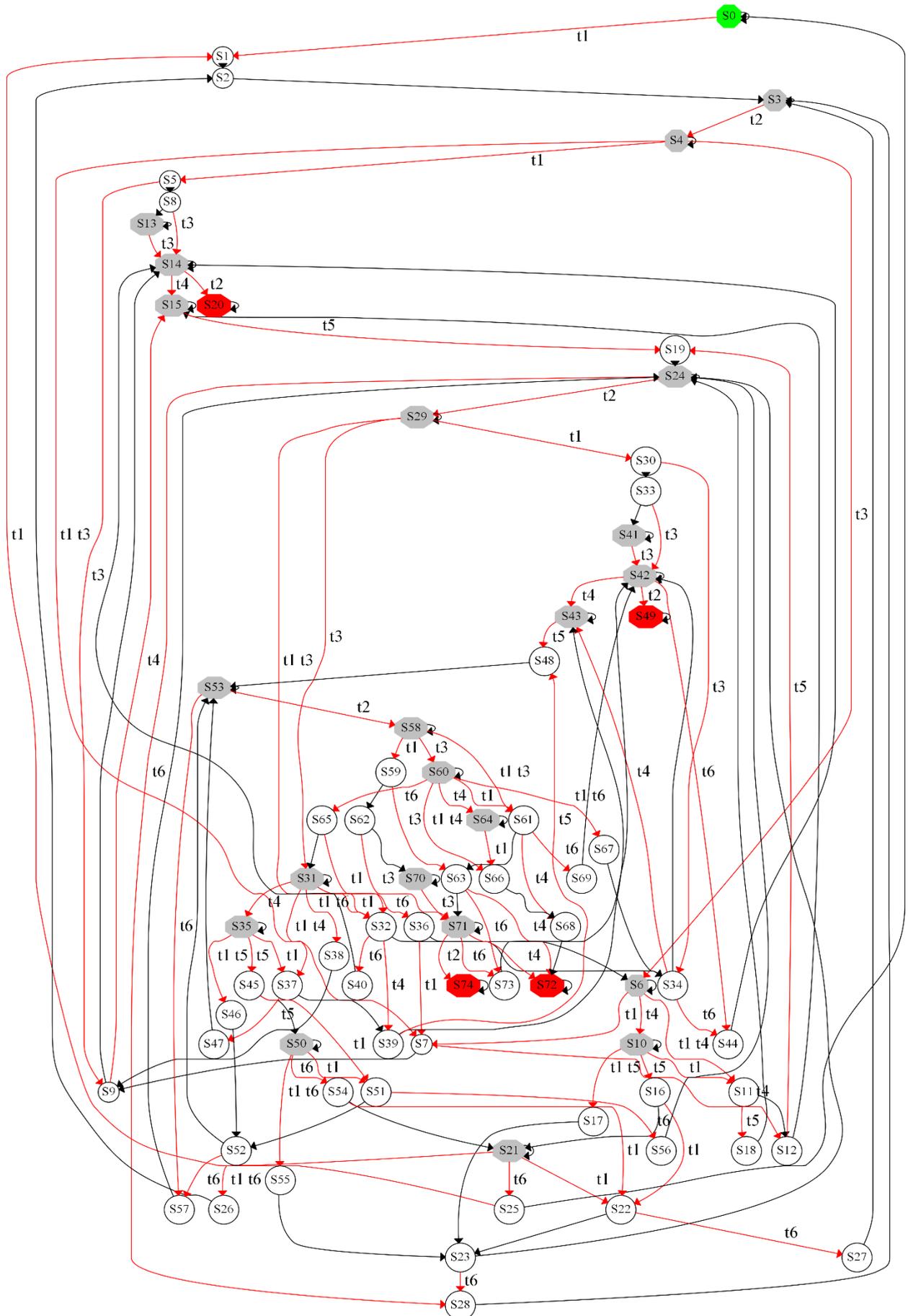


Figure 5. Timed-reachability tree for TdAPN in Figure 4(b)

## ACKNOWLEDGEMENT

This research project, which is BAP-1610F665, namely "*Dynamic Control Approaches for Timed Discrete-Event Systems*", was supported by Anadolu University/Eskisehir Technical University.

## CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

## REFERENCES

- [1] Cassandras, C., Lafortune, S., "Introduction to discrete event systems (second edition)", Springer US, (2008).
- [2] Zhou, M.C., DiCesare, F., "Petri net synthesis for discrete event control of manufacturing systems", Springer US, (1993).
- [3] Proth, J.M., Xie, X., "Petri nets a tool for design and management of manufacturing systems", Wiley, (1996).
- [4] Ho, Y. (Editor), "Discrete event dynamic systems: analyzing complexity and performance in the modern world", IEEE Press, (1992).
- [5] Zuberek, W.M., "Timed Petri nets - definitions, properties, and applications", *Microelectronics and Reliability*, 31(4): 627-644, (1991).
- [6] Bowden, F., "A brief survey and synthesis of the roles of time in Petri nets", *Mathematical and Computer Modelling*, 31: 55-68, (2000).
- [7] Nielsen, M., Sassone, V., Srba, J., "Properties of distributed timed-arc Petri nets", *Proceedings of the 21th International Conference on Foundations of Software Technology and Theoretical Computer Science*, 2245: 280–291, (2001).
- [8] Akshay, S., Helouet, L., Phawade, R., "Combining free choice and time in Petri nets", *Journal of Logical and Algebraic Methods in Programming*, 110: 1-29, (2020).
- [9] Capkovic, F., "Modeling and control of discrete-event systems with partial non-determinism using Petri nets", *Acta Polytechnica Hungarica*, 17(4): 47-66, (2020).
- [10] Lefebvre, D., Daoui, C., "Control design for bounded partially controlled TPNs using timed extended reachability graphs and MDP", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(6): 2273-2283, (2020).
- [11] Li, L., Basile, F., Li, Z.W., "A reduced computation of state space to enforce GMECs and deadlock-freeness on TPN systems", *Proceedings of the 15th IFAC International Workshop on Discrete Event System*, (2020).
- [12] Wang, J., "Time Petri nets, theory and application", Kluwer, (1998).
- [13] Aybar, A., Iftar, A., "Deadlock avoidance controller design for timed Petri nets using stretching", *IEEE Systems Journal*, 2(2): 178-188, (2008).
- [14] Abdulla, P., Nylén, A., "Timed petri nets and BQOs", *Proceedings of the 22th International Conference on Application and Theory of Petri Nets*, 53-70, (2001).

- [15] Bolognesi, T., Lucidi, F., Trigila, S., “From timed Petri nets to timed lotos”, Proceedings of the 10th International Symposium on Protocol Specification, Testing and Verification, 395–408, (1990).
- [16] Hanisch, H., “Analysis of place/transition nets with timed-arcs and its application to batch process control”, Proceedings of the 14th International Conference on Application and Theory of Petri Nets, 282-299, (1993).
- [17] Jacobsen, L., Jacobsen, M., Moller, M.H., Srba, J., “Verification of timed-arc Petri nets”, Proceedings of the 37th International Conference on Current Trends in Theory and Practice of Computer Science, 6543: 46-72, (2011).
- [18] Walter, W., “Timed petri-nets for modelling and analyzing protocols with real-time characteristics”, Proceedings of the 3rd IFIP International Workshop on Protocol Specification, Testing and Verification, 149-159, (1983).
- [19] Bowden, F., “The modelling and analysis of command and control decision processes using extended time Petri nets: Chapter 6 Super-Class (Ph.D. thesis)”, University of Adelaide, (2001).
- [20] Zhu, J., Denton, R., “Timed Petri nets and their application to communication protocol specification”, Proceedings of the 21th International Conference on Century Military Communications - What's Possible? Military Communications, 195-199, (1988).
- [21] Yufka, A., Ozkan, H.A., Aybar, A., “A formal method and novel graphical representation for deterministic timed-arc Petri nets”, Proceedings of the National Conference on Otomatik Kontrol Ulusal Toplantisi, 209-213, (2016).
- [22] Yufka, A., Ozkan, H.A., Aybar, A., “Timed arc Petri nets: the time-element approach”, Proceedings of the 10th International Conference on Electrical and Electronics Engineering, 794-798, (2017).
- [23] Yufka, A., Ozkan, H.A., Aybar, A., “Modeling basic components of railway systems using timed arc Petri nets”, Proceedings of the 5th International Conference on Control, Decision and Information Technologies, 427-432, (2018).
- [24] Yufka, A., Ozkan, H.A., Aybar, A., “Timed arc Petri nets: the impulsive approach”, Proceedings of the 5th International Conference on Control, Decision and Information Technologies, 409-414, (2018).
- [25] Yufka, A., Ozkan, H.A., Aybar, A., “Reachability set algorithms for timed-arc Petri nets”, International Journal of Industrial Electronics and Electrical Engineering, 7(5): 36-45, (2019).
- [26] Aybar, A., Iftar, A., “Supervisory controller design to enforce some basic properties in timed-transition Petri nets using stretching”, Nonlinear Analysis: Hybrid Systems, 6: 712-729, (2012).
- [27] Aybar, A., Iftar, A., “Supervisory controller design for timed Petri nets”, Proceedings of the 1st International Conference on System of Systems Engineering, 59-64, (2006).
- [28] Aybar, A., Iftar, A., “Supervisory controller design to enforce some basic properties in timed Petri nets”, Proceedings of the 13th IFAC International Symposium on Information Control Problems in Manufacturing, 940-945, (2009).
- [29] Aybar, A., Iftar, A., “Representation of the state of timed-place Petri nets using stretching”, Proceedings of the 4th IFAC International Workshop on Discrete-Event System Design, 42(21): 72-77, (2009).

- [30] Aybar, A., Iftar, A., “Supervisory controller design to enforce basic properties in timed-place Petri nets”, Proceedings of the 6th IFAC International Conference on Management and Control of Production and Logistics, 46(24): 486-492, (2013).
- [31] Yufka, A., “Timed-arc Petri nets modeling and forbidden state control approach: chapter 7 algorithms for timed-arc Petri nets (Ph.D. thesis)”, Eskisehir Technical University, (2019).
- [32] Murata, T., “Petri nets - properties, analysis and applications”, Proceedings of the IEEE, 77(4): 541-580, (1989).
- [33] Aybar, A., Iftar, A., “Decentralized supervisory controller design to avoid deadlock in Petri nets”, International Journal of Control, 76(13): 1285-1295, (2003).
- [34] Aybar, A., Iftar, A., Ozkan, H.A., “Centralized and decentralized supervisory controller design to enforce boundedness, liveness, and reversibility in Petri nets”, International Journal of Control, 78(8): 537-553, (2005).