A Method for Computing Global Delays of Time Petri Nets

Li Pan, Song Chen
School of Information and Communication Engineering
Hunan Institute of Science and Technology
Yueyang 414006, China
panli.hnist@gmail.com

Hong Zheng
Department of Computer Science and Engineering
East China University of Science and Technology
Shanghai 200237, China
zhenghong@ecust.edu.cn

Abstract—State class methods are a fundamental and most widely applied technique for timing analysis of Time Petri Nets (TPNs). However, the existing state class methods are not suitable for the computation of global time delay, which is an important issue for real-time systems. This paper presents a state class method for the analysis and verification of global time properties in TPNs that addresses the problem. Our method adopts a global time mode, which can efficiently evaluate global time stamps of state classes. This benefits from the introduction of a special reference transition, called zero transition, whose global firing time is always set to zero. We also decompose firing rules into several basic operations. The separation facilitates the introduction of new operations and brings us more flexibility.

Keywords-Time Petri nets; state class; zero transition; global delay.

I. INTRODUCTION

Time plays an important role in real-time systems. This has motivated the development and extension of Petri nets to support modeling and analysis of time-critical systems. Two main directions in using the time are recognized: considering the time as duration and considering intervals, such as time Petri nets (TPNs) [1], and timed Petri nets [2]. It has been shown that TPNs are more general than timed Petri nets [3].

A fundamental and most widely applied method for analyzing TPNs is reachability analysis. However, suffering from infinity of the time domain, the state space of a TPN model is usually infinite. So, contracting the infinite state space into a finite structure is essential. Berthomieu and Menasche first introduced the concept of state classes in [4]. A state class of a TPN is an aggregated pseudo-state associated with a firing sequence of transitions, which represents all states reachable from the initial state by firing all feasible firing values corresponding to the same firing sequence. This approach has been shown to produce finite state class graphs for all bounded TPN models.

Berthomieu’s classical state class approach can preserve Linear Temporal Logic (LTL) properties of the model but not necessarily its Computation Tree Logic (CTL) ones. Yoneda and Ryuba [5] introduced a geometric region graph method to address CTL properties for TPNs by using a partition refinement technique. Berthomieu and Vernadat [6] proposed an alternative construction of the Atomic State Class Graphs (ASCG), which can be then further refined to satisfy the atomicity of all classes in shorter times. In [7], Boucheneb and Hadjidj proposed a method in [10] to refine a very compact intermediate abstraction, such that time and space required for computing an atomic Concrete State Zone Graph (CSZG) are reduced significantly.

Another limitation of the classical state class approach is that it is not suitable to evaluate the global time delay in an execution trace, which is a critical issue in the design and analysis of real-time systems [9]–[11]. The classical approach uses the relative time mode for state classes, i.e., the firing time of each transition is relative to the time when the class was entered, so the absolute (global) time value cannot be obtained efficiently through a state class graph.

In [11], Vicario proposed an enumerative method for evaluating timing profiles of an execution trace. Vicario’s approach can efficiently compute the time span between any two reachable classes. However, the approach still adopts the relative time mode, so the computation of the global time needs to gather all relative domains along a fired trace. Therefore, the evaluation of global time stamps depends on all previous classes but not the latest fired class. This undoubtedly increases the difficulty in verifying global time properties and the need for memory.

In [9], a state class approach based on the global time, called clock-stamped state class approach, was proposed for TPNs by Wang, Deng and Xu. In particular, a clock-stamped state class is composed of three parts: (1) a marking $M$, (2) a global firing domain $D$, which represents the set of global firing intervals of all enabled transitions; and (3) a clock stamp $ST$ that corresponds to the global arrival time interval of the current state class. However, this approach cannot determine the firability of transitions in some cases, because timing inequalities on transition pairs are unknown, which are not able to be derived from the global firing domains.

In [10], Xu, He, and Deng presented a combined approach based on both relative and absolute time modes to the analysis of schedulability of TPNs. This approach uses the relative firing domains to determine the schedulability of transitions, and the absolute firing domains to compute the global time delay. It is to be regretted that the authors also disregard the timing inequalities on transition pairs and only retain the firing intervals of individual transitions in relative domains. As a result, the firability problem is not yet solved in the approach.

In this paper, we present a state class method for the timing analysis of TPN models. To capture the global time properties, a special reference transition, called zero-transition, is introduced into firing domains. The global firing time of the zero-transition is always set to 0. A firing domain of a state class with zero-transition consists of two parts: 1) the timing inequalities on enabled transition pairs,
which are used for the determination of the transition firability; and 2) the timing inequalities between enabled transitions and the zero transition, which are used to compute the time stamp of a state class. Therefore, the time stamp of a state class can be easily obtained by calculating the time span between the zero-transition and the latest fired transition. Moreover, firing rules are decomposed into several basic operations. The separation will bring us more flexibility.

The rest of the paper is organized as follows. In section 2, we define the syntax and semantics of a TPN. Section 3 proposes a state class approach with zero-transition to the analysis of global timing properties for TPNs. Finally, Section 4 concludes the paper.

II. TIME PETRI NETS

A. Syntax

Let \( R \) be the set of real numbers and \( R' \) the set of nonnegative real numbers. An interval is a connected subset of \( R \). Formally, \( I = \{a, b\} \) is an interval iff \( I = \{x \in R | a \leq x \leq b\} \) where \( a \in R, b \in R \) and \( a \leq b \). The lower (upper) bound of an interval \( I \) is denoted by \( l(I) \) (\( u(I) \) respectively).

The set of all (nonnegative) intervals is denoted as \( IR \) (\( IR' \)). Intervals with property \( l(I) = l(I') \) are called point intervals, any of them contains exactly one real number and can be thus formally identified with this very number.

TPNs are a time extension of Petri nets [12]. In TPNs, with each transition is associated a time interval representing the possible firing times of the transition relative to its enabling instant.

**Definition 1.** A time Petri net is a 6-tuple \( TPN = (P, T, \text{Pre}, \text{Post}, M_0, S) \) where

- \( P = \{p_1, p_2, \ldots, p_m\} \) is a finite nonempty set of places;
- \( T = \{t_1, t_2, \ldots, t_n\} \) is a finite nonempty set of transitions;
- \( \text{Pre} : P \times T \rightarrow N \) is the forward incidence matrix;
- \( \text{Post} : P \times T \rightarrow N \) is the backward incidence matrix;
- \( M_0 : P \rightarrow N \) is the initial marking;
- \( S : T \rightarrow IR' \) is a mapping called static firing interval.

For any \( t \in T, S(t) \) represents \( t \)’s static firing interval relative to the time at which transition \( t \) begins enabled.

Let \( \text{Pre}(t) \in N^p \) be the multi-set of input places of transition \( t \), corresponding to the vector of column \( t \) in forward incidence matrix. \( \text{Pre}(p, t) > 0 \) denotes the set of input places of transition \( t \) and \( p = \{t \mid \text{Pre}(p, t) > 0\} \) represents the set of output transition of place \( p \). Similarly, \( \text{Post}(t) \), \( t^p \) and \( p^t \) are defined.

B. Semantics

**Definition 2.** A transition \( t \) is enabled at marking \( M \), if and only if

\[ \forall p \in P: \text{Pre}(p, t) \leq M(p) \]

Let \( \text{En}(M) \) be the set of transitions enabled at marking \( M \). The number of simultaneous enabling of a transition \( t \) at marking \( M \) is called its enabling degree, and is denoted by

\[ \text{Degree}(M, t) = \min\{M(p) / \text{Pre}(p, t) | p \in t^p \} \]

If \( \text{Degree}(M, t) \geq 2 \), we say that transition \( t \) has multi-enabledness at marking \( M \) [13]. In order to simplify the treatment of the problem, as assumed in [3], [9] and [11], we do not consider multi-enabledness of transitions. Let \( \text{Newly}(M, t) \) denote the set of newly enabled transitions by firing \( t \) from \( M \), which is defined by

\[ \text{Newly}(M, t) = \{t \in T | t \in \text{En}(M - \text{Pre}(t) + \text{Post}(t)) \land (t \in \text{En}(M - \text{Pre}(t)) \lor (t = t_0)) \} \]

Note that a transition which is still enabled after its own firing is always considered as a newly enabled one.

There are two main approaches to the representation of states for TPNs. The first approach, called clock approach [14], [15], associates each enabled transition \( t \) with a clock that records the time elapsed since \( t \) became enabled most recently. The second one, called firing interval approach [3], [11], assigns to each enabled transition \( t \) a time interval in which \( t \) can be fired. We focus on the second state characterization in the remaining part of this paper, but the definition of states in [3] will be modified slightly, so as to capture global time properties.

**Definition 3.** A state of a TPN = \( (P, T, \text{Pre}, \text{Post}, M_0, S) \) is a pair \( s = (M, f) \), where

- \( M \) is a marking;
- \( f \) is a global firing interval function. \( \forall t \in \text{En}(M), f(t) \) represents \( t \)’s global firing interval in which each value is a possible firing time relative to the initial state.

Let \( s_0 = (M_0, f_0) \) be the initial state of the TPN, where

- \( M_0 \) is the initial marking, and
- \( \forall t \in \text{En}(M_0); f_0(t) = S(t) \).

The semantics of a TPN model can be characterized by a Labeled Transition System that is defined below.

**Definition 4.** A labeled transition system is a quadruple \( L = (S, s_0, \Sigma, \rightarrow) \) where

- \( S \) is a finite set of states;
- \( s_0 \in S \) is the initial state;
- \( \Sigma \) is a set of labels representing activities;
- \( \rightarrow \) is the transition relation.

**Definition 5.** The semantics of a TPN = \( (P, T, \text{Pre}, \text{Post}, M_0, S) \) is defined as a labeled transition system \( L = (S, s_0, \Sigma, \rightarrow) \) such that

\[ \begin{cases}
S = N^6 \times (IR')^7; \\
\Sigma = (M_0, f_0); \\
\rightarrow = N \times S \times S, \text{ and } \forall t \in T \\
\langle Pre(t), \rangle \overset{(t, d)}{\rightarrow} \langle M', f \rangle \iff \\
\forall t \in \text{En}(M), f(t) \leq d \leq f(t) \\
\langle M' = M - \text{Pre}(t) + \text{Post}(t), \rangle \\
\langle f(t) = \left[ max(\{f(t - d), \delta f(t)\}) \right] \if t \in \text{En}(M - \text{Pre}(t)) \rangle \end{cases} \]
transition; (3) describes the standard marking transformation; and (4) calculates global firing intervals of all enabled transitions at the state \( s' \). More specifically, persistently enabled transitions have their intervals at state \( s \) truncated to intervals beyond \( d \), and newly enabled transitions are assigned their static intervals plus \( d \).

A transition (firing) sequence \( \sigma \) is a finite (or infinite) string consisting of symbols in transition set \( T \). The empty sequence, denoted by \( \lambda \), is the sequence with zero occurrences of symbols. A sequence \( w \) is called a prefix of \( \sigma \), if and only if there is a sequence \( v \) such that \( \sigma = vw \). Let \( \sigma_i \) be the \( i \)-length prefix of \( \sigma \).

### III. State Class Method

#### A. State Classes with Zero-Transition

In this subsection, we define a canonical state class with zero-transition. Different from the definitions given in [3], [9] and [11], we introduce timing inequalities between individual enabled transitions and the zero transition, and remain timing inequalities on transition pairs. However, the inequalities of individual enabled transitions relative to the current state class are disregarded, because they are not necessary for the determination of the transition firability.

Let \( Var = \{ x_0, x_1, \ldots, x_n \} \) be a finite set of variables with value in \( R \cup \{ \infty \} \), where \( x_0 = 0 \). Let \( D \) be a set of time constraints \( (x_i - x_j \leq c) \) where \( c \in R \) and \( x_i, x_j \in Var \). The global time constraint \( (x_i \leq c) \) can also be expressed as the uniform form \( (x_i - x_0 \leq c) \).

The DBM (Difference Bound Matrix) representing \( D \), denoted by \( B(D) \), is an \( n \times n \) matrix [16], [17]. Its element \( b_{ij} \) can be computed in three steps:

- For each constraint \( x_i - x_j \leq c \) in \( D \), let \( b_{ij} = c \).
- If \( x_j - x_i \) is not in \( D \), let \( b_{ij} = \infty \).
- Add the implicit constraints \( x_i - x_j \leq 0 \), let \( b_{ij} = 0 \).

As an example, Fig. 1 shows a constraint set together with its corresponding DBM.

\[
D = \begin{bmatrix}
1 & 3 & 0 \\
3 & 1 & -1 \\
0 & 5 & 1 \\
-5 & 0 & 0
\end{bmatrix}
\]

\[B(D) = \begin{bmatrix}
0 & 3 & 5 \\
-1 & 0 & \infty \\
-5 & \infty & 0
\end{bmatrix}
\]

(a) (b)

Figure 1. (a) A constraint set \( D \), and (b) its DBM \( B(D) \).

To characterize global timing properties, we introduce an additional reference transition called zero-transition, denoted by \( t_0 \), which is associated with a special variable \( x_0 \).

**Definition 6.** A State Class with Zero-transition of a TPN = \((P, T, \text{Pre}, \text{Post}, M_0, S)\) is a pair \( C = (M, D) \) where:

- \( M \) is a marking;
- \( D \) is a set of inequalities called the firing domain with zero-transition. The inequalities in \( D \) are constraints of the form:
  \[
x_i - x_j \leq b_{ij} \quad \forall t_i, t_j \in \text{En}(M) \cup \{ t_0 \}
\]

where \( x_i \in Var \) is a variable that represents the global firing time of enabled transition \( t_i \) and \( b_{ij} \) is the maximum values of the difference \( x_i - x_j \). Especially, \( b_{ij} = 0 \) when \( i = j \).

Let \( S(t_0) = 0 \), the initial state class \( C_0 = (M_0, D_0) \) is defined as follows:

- \( M_0 \) is the initial marking, and
- \( D_0 = \{ x_i - x_j \leq |S(t_i) - |S(t_j) | \mid t_i, t_j \in \text{En}(M_0) \cup \{ t_0 \} \} \).

**Definition 7 (Firable condition).** A transition \( t_j \in \text{En}(M) \) is firable at state class \( C = (M, D) \), if and only if

\[
\forall t_i \in \text{En}(M): b_{ij} \geq 0.
\]

Let \( Fr(C) \) be the set of all firable transitions at state class \( C \).

Next, we introduce three basic operations on DBMs.

- **Execute.** Given a DBM \( B(D) \) and a transition \( t_f \in \text{En}(M) \), the operation \( \text{Execute}(D, t_f) \) produces a new DBM \( B(D') \), whose element \( b'_{ij} \) is computed by
  \[
b'_{ij} = \begin{cases}
  \min\{b_{ij}, b_{fi} + b_{ij}\} & \text{if } i = 0 \\
  \min\{b_{ij}, b_{fi} + b_{ij}, b_{ij}\} & \text{otherwise}
  \end{cases}
\]

We compute \( \min\{b_{ij}\} \) for all \( i \) with a complexity \( O(N^2) \).

Then, \( b_{ij} \) can be derived from \( b_{ij} \). The computation of \( \text{Execute}(D, t_f) \) runs in time \( O(N^2) \).

- **Append.** Given a DBM \( B(D) \), a transition \( t_f \in \text{En}(M) \) and a transition \( t_i \not\in \text{En}(M) \). The operation \( \text{Append}(D, t_f, t_i) \) computes a new DBM \( B(D') \), whose element \( b'_{ij} \) is defined as follows:
  \[
b'_{ij} = \begin{cases}
  0 & \text{if } i = r \land j = r \\
  \downarrow S(t_f) + b_{ij} & \text{if } i = r \land j \neq r \\
  b_{ij} + \uparrow S(t_i) & \text{if } i \neq r \land j = r \\
  b_{ij} & \text{otherwise}
  \end{cases}
\]

The Append operation is to generate a new DBM \( B(D') \) by adding constraints \( x_i - x_j \leq b_{ij} + \uparrow S(t_i) \) and \( x_j - x_i \leq b_{ij} - \downarrow S(t_f) \) to \( D \). The operation results in the complexity \( O(n) \).

- **Eliminate.** Given a DBM \( B(D) \) and a transition \( t_k \in \text{En}(M) \), the operation \( \text{Eliminate}(D, t_k) \) generates a new DBM \( B(D') \), which is defined by
  \[
b'_{ij} = \begin{cases}
  \infty & \text{if } i = k \lor j = k \\
  b_{ij} & \text{otherwise}
  \end{cases}
\]

The operation \( \text{Eliminate} \) is to compute a new DBM \( B(D') \) where all information about \( x_k \) are lost. The operation requires time \( O(n) \).

Let \( W \) be a set of transitions, we generalize the definition of Append as follows:

- \( D = \text{Append}(D, t_0, \emptyset) \), and
- \( \text{Append}(D, t_f, W) = \text{Append}(\text{Append}(D, t_f, U), t_f, t_i) \), where \( W = U \cup \{ t_i \} \).

In the same way, we define \( \text{Eliminate}(D, W) \) by

- \( D = \text{Eliminate}(D, \emptyset, \emptyset) \), and
- \( \text{Eliminate}(D, W) = \text{Eliminate}(\text{Eliminate}(D, U), t_k) \), where \( W = U \cup \{ t_k \} \).

**Definition 8 (Firing rules).** State class \( C_{k+1} = (M_{k+1}, D_{k+1}) \) reached from state class \( C_k = (M_k, D_k) \) by firing transition \( t_f \in Fr(C_k) \) can be computed as follows:

- Marking \( M_{k+1} \):
$M_{k+1} = M_k - \text{Pre}(t_j) + \text{Post}(t_j)$

- Firing domain $D_{k+1}$:
  (i) $B(D'_i) = \text{Execute}(D_i, t_i)$
  (ii) $B(D''_i) = \text{Append}(D'_i, t_i, \text{Newly}(M_i, t_i))$
  (iii) $B(D'''_i) = \text{Eliminate}(D''_i, \text{En}(\text{Pre}(t_i)))$

The computation of its successive state class from the current state class requires a total complexity $O(|T|^2)$, where $|T|$ is the larger value of $\text{En}(M_k)$ and $\text{En}(M_{k+1})$.

**B. Global Time Stamps**

Let $ST_i$ denote the global time stamp of class $C_i$, which represents the global arrival time interval of $C_i$. We assume that $ST_0 = 0$.

**Theorem 1.** Let $C_n = (M_n, D_n)$ be a reachable state class from the initial state class $C_0 = (M_0, D_0)$ by firing a transition sequence $\sigma = t_{n-1}^{-1} \cdots t_n^{-1}$ where $n \geq 1$. Then

$$ST_n = \{ -b(D'_1)_{0,0}, b(D'_{n-1})_{0,0} \}.$$  

**Proof.** Because firing a transition takes no time, the global arrival time of state class $C_n$ is just the firing time of $t_n$. From the definitions of the $\text{Execute}$ operation and Firing rules, it follows that $x_{n_0} - x_0 \leq b(D'_{n-1})_{0,0}$ and $x_0 - x_{n_0} \leq b(D'_n)_{0,0}$. And then we obtain that $-b(D'_{n-1})_{0,0} \leq x_{n_0} - x_0 \leq b(D'_n)_{0,0}$. Thus, $ST_n = \{ -b(D'_1)_{0,0}, b(D'_{n-1})_{0,0} \}$. \qed

From Theorem 1, it is easy to see that $ST_n$ only depends on $D_{n-1}$, i.e., the latest fired state class.

In the sequel, we will take the example in Fig. 2 to illustrate the evolution of state classes and the evaluation of time stamps.

![Figure 2. A simple TPN.](image)

The initial class is $C_0 = (M_0, D_0)$ where $M_0 = (1 1 0 0)$,

$$B(D_0) = x_1 \begin{pmatrix} 0 & 3 & 5 \end{pmatrix},$$

$$ST_0 = 0.$$  

Since $b_{12} = 4 \geq 0$ and $b_{21} = -2 < 0$, by the firable condition, we have $\text{Firing}(C_0) = \{ t_1 \}$. Firing $t_1$ at $C_0$ will result in $C_1$. Then it follows from the firing rules that

$$M_1 = (0 1 1 1),$$

$$B(D'_1) = \text{Execute}(D_0, t_1) = B(D_0),$$

(because $b_{21} < 0$)

$$B(D''_1) = \text{Append}(D'_1, t_1, \text{Newly}(M_0, t_1))$$

$$x_0 \begin{pmatrix} 0 & 3 & 5 & 8 \end{pmatrix},$$

$$x_1 \begin{pmatrix} -1 & 0 & 4 \end{pmatrix},$$

$$x_2 \begin{pmatrix} -5 & -2 \end{pmatrix},$$

$$x_3 \begin{pmatrix} -2 \end{pmatrix},$$

$$x_4 \begin{pmatrix} -4 & -3 & 1 & -1 \end{pmatrix}.$$  

$$B(D_1) = \text{Eliminate}(D''_1, \{ t_1 \})$$

$$x_0 \begin{pmatrix} 0 & 5 & 8 \end{pmatrix},$$

$$x_1 \begin{pmatrix} 2 & 0 & 4 \end{pmatrix},$$

$$x_4 \begin{pmatrix} -4 & 1 \end{pmatrix}.$$  

$$ST_1 = \{ -b(D'_1)_{10}, b(D''_1)_{01} \} = [1, 3].$$

Similarly, according to the firable condition, we have that $\text{Firing}(C_1) = \{ t_2, t_3 \}$. At $C_1$, firing $t_2$ will produce $C_2 = (M_2, D_2)$, where

$$M_2 = (0 0 1 1),$$

$$B(D'_2) = \text{Execute}(D_1, t_2) = B(D_1),$$

$$(b_{31} < 0)$$

$$B(D''_2) = \text{Append}(D'_2, t_2, \text{Newly}(M_1, t_2))$$

$$x_0 \begin{pmatrix} 0 & 5 & 8 \end{pmatrix},$$

$$x_2 \begin{pmatrix} -5 & 0 \end{pmatrix},$$

$$x_3 \begin{pmatrix} -5 & 0 \end{pmatrix},$$

$$x_4 \begin{pmatrix} -6 & -1 \end{pmatrix}.$$  

$$B(D_2) = \text{Eliminate}(D''_2, \{ t_2 \}) = \{ \emptyset \}$$

$$x_0 \begin{pmatrix} 0 & 5 & 8 \end{pmatrix},$$

$$x_3 \begin{pmatrix} -5 \end{pmatrix},$$

$$x_4 \begin{pmatrix} -6 \end{pmatrix}.$$  

$$ST_2 = \{ -b(D'_2)_{20}, b(D''_2)_{01} \} = [5, 5].$$

Following this way, we can generate the state class tree of the TPN model, which is depicted in Fig. 3. Its root is the initial class $C_0$, and there is an arc labeled with $t_i$ from class $C_j$ to class $C_i$ if $t_i$ is firable from class $C_j$ and if its firing leads to class $C_j$. In this class tree, each firable transition of a state class has only one successor.

![Figure 3. Reachability class tree of the TPN in Fig. 2.](image)
From the reachability class tree, we easily obtain the global delay of any state class. For example, the global delay of the state class \( C_7 \) is \([6,8]\), that is to say, the execution delay of the schedule \( t_1t_2t_3t_4 \) is \([6,8]\). If the constraint that the time of finishing all tasks is not more than 6 time units is require, only the schedule \( t_1t_3t_4t_2 \) satisfies the requirement.

IV. **Conclusion**

We have presented a state class method for the analysis and verification of timing properties. In our method, timing inequalities on transition pairs are used to determine the firability of transitions, and the zero transition is introduced to facilitate the computation of the global time. Moreover, the separation of firing rules into basic operations can bring us conveniences. For example, we may define an operation \( \text{reset} \), which is easily added to firing rules during runtime to reset the timing point of a net. We may also introduce a new operation \( \text{suspend} \) to support the suspension of a net during runtime for a period of time.

Currently, we are working on extending the proposed method for the schedulability analysis of a new TPN model, where two time semantics, i.e., strong semantics and weak one, are mixed on choices.

**ACKNOWLEDGMENT**

This work was supported by the key project of Guangdong and Hong Kong under Grant 2007A090601001.

**REFERENCES**


