This is a repository copy of *Deterministic Scheduling of Time Petri Nets*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/80379/

Monograph:

Hayat, A. and Bennett, S. (1996) Deterministic Scheduling of Time Petri Nets. Research Report. ACSE Research Report 457 . Department of Automatic Control and Systems Engineering

Reuse

Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

629
.8
(S)

Deterministic Scheduling of Time Petri Nets

A. Hayat and S. Bennett

Deptt. of Automatic Control and Systems Engg.

University of Sheffield

Mappin Street, Sheffield S1 3JD.

June 05, 1996

Research Report No. 628

200361619

Deterministic Scheduling of Time Petri Nets

Abstract

Time Petri nets (TPNs) can be viewed as a natural time-extension of causal PNs for modeling of discrete event systems. In this report we define states and firing rules for TPNs, and give a reachability algorithm to enumerate states under arbitrary decisions for scheduling time of events. Deterministic Scheduling of TPNs involve defining a rule for these decisions, and the resulting model may be called a DSTPN. Beside the modeling of non-determinism, zero-testability is another important feature of TPNs, which has been the base of undecidability results about some important properties of TPNs. We show that while DSTPNs retain the modeling power of TPNs, reachability analysis similar to causal Petri nets is possible for some DSTPNs having important applications. We also point out a flaw in the proof for undecidability of TPN properties, which renders it invalid.

Index Terms : Time Petri nets, discrete event systems, scheduling, reachability analysis, decidability, boundedness, verification, transited-states, transition age vectors, zero testability.

1 Introduction

A discrete event system (DES) consists of a number of activities or conditions, each of which starts or ends by an instantaneous event. The temporal constraints on each activity may be expressed either by its duration or by delays and time-outs on its associated events. The activities, and their associated events, also have causal relations such as concurrency, conflicts, precedence and synchronization, with one another. An event is called *enabled* when all of its constraints are satisfied. The number of events in a DES, which are simultaneously enabled, and the time interval in which given event must occur are usually bounded. However, DESs are generally non-deterministic with respect to occurrence of the events because, most of the time both: (i) *the next event to occur (or selection of event)* among a number of simultaneously enabled concurrent or conflicting events, and (ii) *the time of occurrence (or scheduling time)* of the event, are undecided. This degree of non-determinism complicates the analysis of DESs with respect to properties such as safeness, liveness etc., as well as other performance measures.

Petri nets [1][2] are a mathematical tool which support modeling and analysis of the causal relations of activities and events in a DES. The nondeterminism of DESs about the selection of events (modeled as transitions) is expressible by Petri nets, but being a causal model, time is not relevant to Petri nets and therefore temporal constraints and scheduling time of events can not be considered. In order to express temporal constraints and to study their effect on the qualitative properties as well as to compute other quantitative performance measures, such as completion time or periods of cyclic activities and resource utilization, causal Petri nets need time extensions.

Several extensions to the Petri nets have been reported, with different objectives. Timed Petri nets (TdPNs) [3], Time Petri nets (TPNs) [4], and Stochastic Petri nets (SPNs) [5] are the three basically different time extensions, whose theory and applications is reviewed by Freedman [6]. In TdPNs the firing rule of transitions of causal Petri nets is changed to allow the specification of fixed duration or delay of activities and events; the resulting model is completely deterministic with respect to scheduling time of the events. Temporal constraints like time out and due dates can not be specified in TdPNs, however, the model has been useful to study cyclic processes [7][8]. SPNs and their variants, allow specification of probability distributions with the firing of transitions, making them suitable for markovian analysis to compute probability distribution of markings and other average performance measures. TPNs can be viewed as a natural extension to Petri nets in temporal space which allow realistic time bounds for the scheduling time of any event. These are expressed as (α,β) (both being real numbers, with $\alpha \leq \beta$) that is, a lower bound 'α' and an upper bound 'β'. Causal Petri nets, under the assumption of being in temporal space, can be considered as a special case of TPNs by setting (α,β) equal to $(0,\infty)$. TdPNs can also be expressed by TPN, by setting α equal to β, while the converse is not true. TPNs require an extension, rather than a change, to the firing rule of transitions, so that the non-determinism of PNs in both respects is preserved, and most of the temporal constraints, particularly time-outs, can be specified. TPNs have been used for verification of communication protocols [9][10], and safety analysis of real time systems [11].

Due to continuous nature of time, the non-determinism of scheduling time implies the existence of an infinite number of schedules for a given event. Berthomieu and Diaz [9] therefore, introduce state-classes as the infinite set of all states resulting by an infinite number of schedules from a given state. They enumerate the state-classes of a TPN to verify the correctness of the timing values specified for the system. However, termination of the enumeration process is guaranteed only for the TPNs whose underlying causal PN is bounded, which means unreliable verification of some properties for other TPNs. The state-class graph also, being a representation of an infinite number of schedules, can not be used for control and scheduling problems.

In this work, we introduce a restriction on scheduling time of transitions. We show that the TPN properties under the restriction, are decidable as for the causal Petri nets, and for the general case we invalidate the undecidability proof given by Jones et. al. [12]. Our approach makes the scheduling problem solvable for most of the quantitative and qualitative performance measures, and it may also provide a step towards solving the problem of the reliable verification of timing values.

2 Time Petri Net

A TPN at time $\tau = 0$, is a tuple $(\mathcal{T}, M_0)$, where $\mathcal{T}$ is a 6-tuple $(P, T, B, F, \alpha, \beta)$ defining structure of the TPN, as:

- P is a finite nonempty set of places p_i;
- T is finite nonempty set of transitions t_i;
- B is the backward incidence function with signature

$$B : T \times P \rightarrow N$$

- F is the forward incidence function with signature

$$F : T \times P \rightarrow N$$

- α *and* β are the minimum and maximum time functions with signatures

$$\alpha : T \rightarrow \Re$$

$$\beta : T \rightarrow \Re \qquad \text{respectively,}$$

and $\quad \forall t \in T: \ \alpha(t) \leq \beta(t)$

- M_0 is the initial marking function with signature

$$M_0 : P \rightarrow N$$

where N is the set of nonnegative integers, $\Re$ is the set of nonnegative real numbers, and all values are static. The entries of α and β, will be denoted as a pair of values (α_i, β_i) for transition t_i, where α_i and β_i represent the minimum time and maximum time, respectively, to be elapsed before firing t_i.

Definition 1

A transition t_i is called *mark enabled*, iff:

$$\forall p \in \{p \in P \mid B(t_i, p) > 0\}: \ M(p) \geq B(t_i, p) \qquad \blacklozenge$$

A transition t_i is called *k-bounded* if there exists a finite natural number k such that it can not be mark-enabled more than k times simultaneously, in any state of the TPN. That is, if $R(S_0)$ is the set of all states of a TPN;

$$\exists p \in \{p \in P \mid B(t_i, p) > 0\}; \forall M(p) \in R(S_0): \ M(p) \leq k \cdot B(t_i, p) \qquad (1)$$

A transition may be called *unbounded*, if k can not be defined and (1) does not hold. A TPN is called *t-bounded* if (1) holds for all transitions, and it is called *t-safe* if k=1 for all transitions. Correspondingly a TPN is *t-unbounded* if there is at least one unbounded transition. We will restrict our discussion in the following for *t*-safe TPNs.

2.1 States in a TPN

The definition of TPN suggests that the causal Petri nets can be handled as a special case of a TPN. Hence, to facilitate our discussion, we need to keep a distinction between the terms. We will consider '*causal Petri nets*' as the Petri nets which do not involve consideration of time in any way, and '*Petri nets*' as the special case of TPNs for which the following holds:

$$\forall t \in T: \; (\alpha(t), \beta(t)) = (0, \infty)$$

The effect of unbounded firing interval in Petri nets is that the possible firing sequences (and the resulting markings) are independent of the transition firing time. However, the firing sequences of a TPN, with one or more transitions with bounded interval, may vary by firing a transition at different times. This means that, while the state of PNs (or causal PNs) could sufficiently be described by its marking, we generally need to incorporate time based parameters as well as marking, to express the state of TPNs. The time related parameter, used by Berthomieu and Diaz [9], is called the *remaining enabling time* (RET) vector, which is conceptually the same as the *residual firing time* vector used for TdPN models [8]. In the following, we use a time parameter which leads to a natural definition of TPN states. It also allows simple rules for transition firings and a reachability algorithm similar to causal Petri nets.

Definition 2

The state S of a TPN at time τ, can be defined as a pair $S = (M, U)$:

- M is the marking vector,
- U is the age vector, which gives duration of a transition for which it has been mark-enabled continuously;

$$U : T \rightarrow \mathfrak{R}$$ ♦

An entry of U may be denoted by u_i for brevity, and we call it the *age* of the transition. Unlike the marking vector (where zero value of $M(p_i)$ means absence of any token at p_i), for the age vector U, when t_i is not mark-enabled, u_i is non-existent and is denoted by 'ϕ '.

The Definition 2 implies that, being a continuous time model, every state of a TPN is an instantaneous state, and there may be infinite number of states between any time interval. At a given time τ, if no transition fires then the TPN is in one state only, denoted as S_τ. The firing of one or more transitions at time τ, will result in a number of valid states at the same instant, which may be denoted as $S_{i(\tau)}$, with $i = 1,2,3, \cdots$. The maximum number of states at given time, by firing of n transitions, is $(1+ n)$. The actual number may be less due to possible simultaneous firing of transitions.

Definition 3

A transition t_i is called *time enabled* or *fireable*, iff:

$$\alpha_i \leq u_i \leq \beta_i \qquad \blacklozenge$$

The times α_i and β_i for transition t_i are relative to the moment at which transition t_i is mark-enabled. Assuming that transition t_i is mark-enabled at an absolute time τ, then t_i can not fire, while being continuously enabled, before time $(\tau + \alpha_i)$ and must fire before or at the latest time $(\tau + \beta_i)$. If transition t_i does not remain mark-enabled, this means that another transition t_m has been fired; firing t_m may lead to a new marking, at a different absolute time τ'.

In a TPN model, the firing of a transition is instantaneous. This means that, the duration of the event to be represented by a transition, must be zero or negligible with respect to all other time values specified in the model, otherwise it should appear as a time label of the transition. Furthermore, the pair (α_i , β_i) may not be specified for a transition t_i, if the modeled event has an unbounded time interval of $(0,\infty)$.

As an example, consider the TPN given in Figure 1, representing a producer-consumer system. Process 'A', described by transitions t_1 and t_2, sends a message, that is consumed by process 'B', in a single action represented by t_3. Each transition has its associated firing interval. The place p_4, is to implement the restriction on simultaneous multiple enableness of t_3. The initial marking M_0 = [1, 0, 0, 1], and age vector at τ = 0.0, is U = [0, ϕ , ϕ], hence initial state is $S_{(0)}$ = (M_0, U), where only t_1 is mark-enabled, and will be fireable between τ = 1.0 to 6.0.

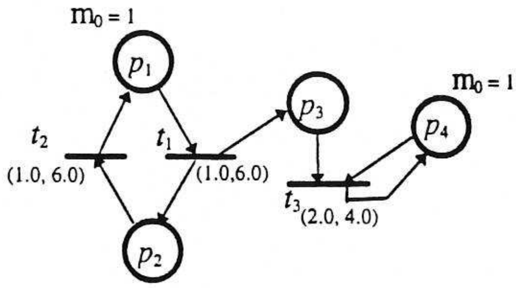


Figure 1

2.2 Firing Rule Between States

Like causal Petri nets, there may be more than one mark-enabled or fireable transitions in a TPN, simultaneously. The choice as to which should fire may be made by a transition firing control algorithm. However, the change in the state, simulating the dynamic behavior of the system, occurs according to the following *transition (firing) rule*:

Assuming that a transition t_i is fireable at time τ from state S = (M, U). Then the state S' = (M' , U'), reached from S by firing t_i at time τ can be computed in two steps.

1) M' is computed, for all places:

$$\forall p \in P: \quad M'(p) = M(p) - B(t, p) + F(t, p)$$

2) U' is updated, for all transitions:

$$\forall t \in \{t \in T \mid (\textit{t is mark-enabled}) \wedge ((U(t) = \phi) \vee (t = t_i))\}: \quad U'(t) = 0.0$$

$$\forall t \in \{t \in T \mid (\textit{t is disabled})\}: \; U'(t) = \phi$$

Note that step-1 is the only step required for Petri Nets, the step-2 is required to avoid violation of time bounds on the transitions by the control algorithm. The condition of $(t = t_i)$ in step-2, is required, because if t is the transition which was fired and is again mark-enabled then its age u_i should be reinitialized to pass its minimum period of α_i, before it is fireable. This is also in accordance with the restriction on simultaneous multiple enableness of transitions. When t is not the fired transition, the condition $(U(t) = \phi)$ is to confirm that only the age values of those transitions which are mark-enabled by the firing of t_i, are initialized.

For instance, for the net in Figure 1, as shown above, if $M_0 = [1, 0, 0, 1]$ and only t_1 satisfies the condition of being mark-enabled, then t_1 can fire any time between $\tau = 1.0$ to 6.0. Let it fire at $\tau = 4.25$, then the two states at same instant in order of occurrence are;

$$S_{1\,(4.25)} = (M_0, [4.25, \phi, \phi])$$

$$S_{2\,(4.25)} = ([0, 1, 1, 1], [\phi, 0.0, 0.0])$$

In the final state of $\tau = 4.25$, t_2 and t_3 are mark-enabled. The time margins for t_2 and t_3, to fire are

$$\tau = 4.25 + 1.0 \text{ to } 4.25 + 6.0 = 5.25 \text{ to } 10.25 \qquad \text{and}$$

$$\tau = 4.25 + 2.0 \text{ to } 4.25 + 4.0 = 6.25 \text{ to } 8.25 \qquad \text{respectively.}$$

However, these margins may be changed after the decision is made by the given control algorithm which may require to fire t_2 and t_3, simultaneously, or prioritize one over the other. Let the algorithm select t_2 to fire first, then its reduced time margin is from $\tau = 5.25$ to 8.25, because firing of t_3 can not be delayed more than $\tau = 8.25$, and if t_2 is to fire at the last instant of the reduced interval then t_3 will also fire at the same time instant.

Let t_2 fire at $\tau = 6.8$, the two consecutive states are:

$$S_{1\,(6.8)} = (\,[0, 1, 1, 1]\,, [\phi, 2.55, 2.55]\,)$$

$$S_{2\,(6.8)} = (\,[1, 0, 1, 1]\,, [0.0, \phi, 2.55]\,)$$

In $S_{2\,(6.8)}$, t_1 and t_3 are mark-enabled, but only t_3 is fireable. Again, to meet some performance objective, firing of t_3 may be delayed further for a period of 1.45 time units, and t_1 may fire at/after $\tau = 7.8$.

If t_1 fires at $\tau = 8.0$, then

$$S_{1\,(8.0)} = (\,[1, 0, 1, 1]\,, [1.2, \phi, 3.75]\,)$$

$$S_{2\,(8.0)} = (\,[0, 1, 2, 1]\,, [\phi, 0.0, 3.75]\,)$$

where only t_3 can fire between $\tau = 8.0$ to 8.25,

By firing t_3 at same instant $\tau = 8.0$,

$$S_{3\,(8.0)} = (\,[0, 1, 1, 1]\,, [\phi, 0.0, 0.0]\,)$$

and so on.

Note that $\tau = 8.0$ has three states, and there is no fireable transition in its final state that is $S_{3(8.0)}$. For some TPN models this may not be the case and at a certain time there may always exist at least one fireable transition, and a given control algorithm may be locked into a loop of firings at this instant, or an independent time advancing mechanism may allow the algorithm to exit the loop with different states for different runs of the algorithm. To avoid such a condition, the structure of a TPN model may always be restricted from having any loop so that the sum of earliest firing times of all transitions in the loop is zero.

3 Reachability Algorithm for TPNs

The reachability tree of causal Petri nets [1][2], gives the set of all markings or states reachable from M_0. It follows from Definition 2, that for TPNs, without the firing of a transition, an infinite number of states can be reached. Hence, to obtain a meaningful reachability tree for TPNs, we need to distinguish the states immediately reached by firing of transitions, from the states reached by aging of transitions.

Definition 4

The state $S_{i(\tau)} = (M, U)$ for $i = 2, 3, 4 \cdots$, which occurs by firing of a transition, is called a *transited-state*, denoted as $\check{S}_j$ with $j = 1, 2, 3, \cdots$, that is $\check{S}_j = (M, U)$. ◆

Initial state $S_{(0)} = (M_O , U_O)$ is the initial transited-state R_O , assumed to be result of some firing before $\tau = 0$.

Definition 5

A transited-state $\check{S}_n$ is called *reachable* from a state $\check{S}_O$, if there exists a sequence of firings that transforms $\check{S}_O$ to $\check{S}_n$. ◆

A firing sequence is denoted by $\sigma = \check{S}_O\ t_1\ \check{S}_1\ t_2\ \check{S}_2\ t_3\ \check{S}_3\ \cdots\ t_n\ \check{S}_n$, or simply $\sigma = t_1\ t_2\ t_3\ \cdots\ t_n$. In this case, $\check{S}_O$ is reachable from $\check{S}_n$ and we write $\check{S}_O\ [\sigma > \check{S}_n$. The set of all possible transited states reachable from $\check{S}_0$, in a TPN '$\mathcal{T}$' is denoted as $R(\mathcal{T}, \check{S}_O)$ or simply $R(\check{S}_O)$, and the set of all possible firing sequences from $\check{S}_O$ is denoted by $L(\mathcal{T}, \check{S}_O)$ or simply $L(\check{S}_O)$.

Definition 6

Two transited-states $\check{S} = (M, U)$ and $\check{S}' = (M', U')$ are *identical* iff:

$$\forall p \in P : M(p) = M'(p)$$

and $\forall t \in T : U(t) = U'(t)$ ♦

An essential step for a state reachability algorithm is to check repeatedly for any unfired transitions from the previously enumerated states, which *may be selected next to fire*, that is *scheduleable*. For (causal) Petri nets we do not distinguish between a mark-enabled transition and a fireable transition, hence every mark-enabled transition is scheduleable. However, for time Petri nets we need the following definition.

Definition 7

A transition t_i is called *scheduleable*, iff;

$\forall t \in \{ t \in T \mid (t \text{ is mark-enabled}) \}: \ (\alpha(t_i) - U(t_i) \leq \beta(t) - U(t))$ ♦

The reason for above condition is that, if a transition is not fireable in the given state, due to its age being less than its α-value, then it still can be selected to fire provided that a valid state where t_i is fireable, can be obtained. The above condition is always true when $U(t_i) \geq \alpha(t_i)$, that is t_i is fireable in the given state. When $U(t_i) < \alpha(t_i)$ then to fire t_i, we need to age all mark-enabled transitions at least by $(\alpha(t_i) - U(t_i))$ to reach a state where t_i is fireable. The above condition confirms that this minimum increment will not let the age of any transition exceed its β-value.

Assuming that t_i is scheduleable in $\check{S}_{(\tau)}$, there may exist an infinite number of states where t_i is fireable without requiring to fire any other transition before it, during the time interval bounded by $(\tau + v_1)$ and $(\tau + v_2)$, where

$$v_1 = \max(0, \alpha_i - u_i) \tag{2}$$

$$v_2 = \min(\ \forall t \in \{ t \in T \mid (t \text{ is mark-enabled in } \check{S}) \}: \ \beta(t) - U(t)\ ) \tag{3}$$

and, u_i and $U \in \check{S}$

Since v_1 and v_2 depend only upon the age vector in the given state $\check{S}_{(\tau)}$ and not the absolute time τ, hence we will refer to this interval as (v_1 , v_2) or *scheduling interval*, for a given scheduleable transition in $\check{S}$. Any of the possibly infinite number of values 'v' in the interval (v_1 , v_2) can be added to the age vector of $\check{S}$, to obtain a fireable state S for t_i, as follows;

$$\forall t \in \{t \in T \mid (t \textit{ is mark-enabled in } \check{S})\ \}:\ U_S(t) = U_{\check{S}}(t) + v \tag{4}$$

The distinction of fireable and scheduleable transitions in TPNs, therefore, require the addition of Step 2.4.1 in the following transited-state reachability algorithm, which otherwise is similar to the algorithm for causal Petri nets, given by Murata [2].

Algorithm for Transited-States Reachability Tree of a TPN

Step 1) Label the initial state $\check{S}_0$ as the root and tag it 'new'.
Step 2) While 'new' transited-states exist, do the following;
- *Step 2.1) Select a new transited-state $\check{S}$.*
- *Step 2.2) If $\check{S}$ is identical to a transited-state on the path from the root to $\check{S}$, then tag $\check{S}$ 'old', and go to another new transited-state.*
- *Step 2.3) If no transitions are scheduleable at $\check{S}$, tag $\check{S}$ 'dead-end'.*
- *Step 2.4) While there exist scheduleable transitions at $\check{S}$, do the following for each scheduleable transition t at $\check{S}$:*
  - *Step 2.4.1) Obtain the state S, from $\check{S}$, where t should fire.*
  - *Step 2.4.2) Obtain the transited-state $\check{S}'$ that results from firing t at S.*
  - *Step 2.4.3) Introduce $\check{S}'$ as a node, draw an arc with label t from $\check{S}$ to $\check{S}'$, and tag $\check{S}'$ 'new'.*

Figure 2, shows the (partial) transited-state reachability tree, representing the arbitrary choices we made in Section 2.2, for the TPN of Figure 1. The arc labels represent the fired transitions and the increment 'v' chosen to reach its fireable state. Each occurrence of the symbol 'v' represents a different value, depending upon its parent transited-state and the transition being fired.

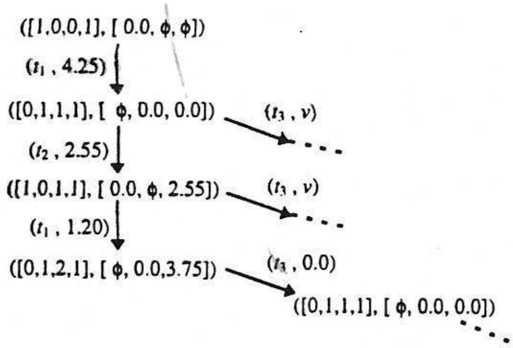


Figure 2. A transited-state reachability tree.

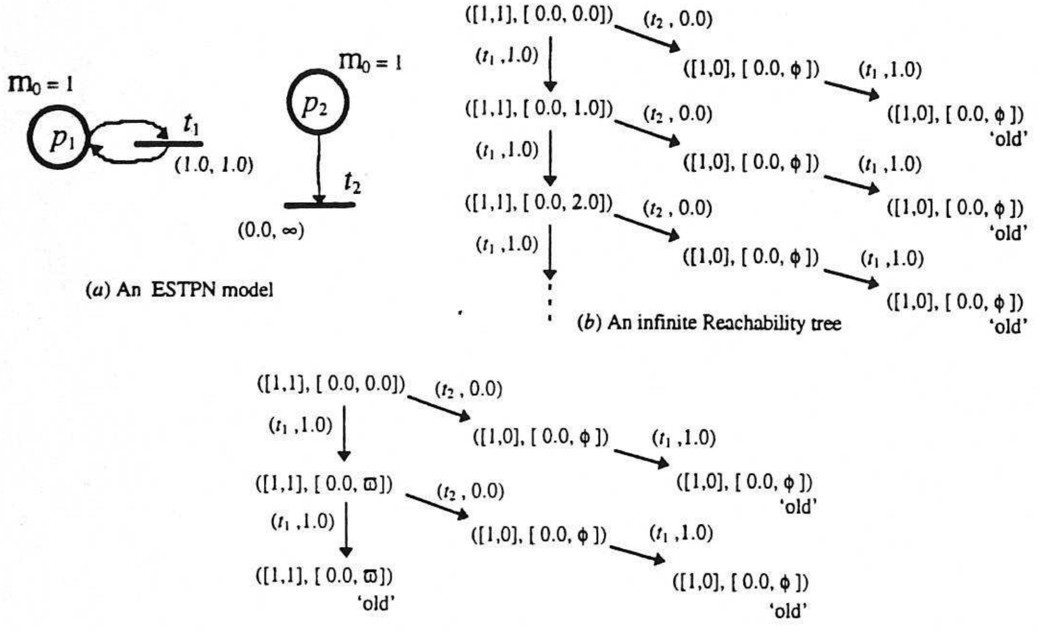
4 Deterministic Scheduling Time Petri Nets

The example of Figure 2, shows that by arbitrary choices in the scheduling interval an infinite number of different transited states can be enumerated for the same firing sequence from the initial state. This makes the tasks of finding schedules or verifying most of the system properties by its reachability tree, unrealizable. However, these tasks can be performed for a TPN model which has some restriction about selection of time in the scheduling interval. This is, particularly, fruitful for the problem of finding schedules to optimize some regular performance measure. A performance measure is called regular, if it can not be improved by an arbitrary delay in the occurrence of any event [13], which means that we can find an optimal schedule from the reachability tree drawn by firing every transition at the earliest time in the scheduling interval, that is v_1. Application of any rule governing choices from the scheduling intervals, reduces the non-determinism of the TPN model with respect to the scheduling time of events. Particularly, when the rule is such that there exists only one instant in the scheduling interval for the selected transition, the scheduling time is decided as soon as a scheduleable transition is selected. The model may, therefore, be called *Deterministic Scheduling Time Petri Net* (DSTPN). Note that DSTPNs are still non-deterministic as compared with TdPNs [3], for which the scheduling time is decided as soon as the transition is mark-enabled. It is obvious that an unlimited number of different rules can be described to produce different DSTPNs.

For the reachability tree of any DSTPN to be a useful analysis tool it should be of a finite size, but as we will see in the next section that, though a DSTPN has reduced nondeterminism, it can still generate an infinite number of transited-states. Finiteness of the reachability tree of a DSTPN is the problem which we address in the next section. Due to their practical importance our discussion will be limited to DSTPNs with earliest firing rule (to be called *earliest scheduling TPNs* or ESTPNs for brevity). However, our arguments can be found valid for most of the other DSTPNs.

4.1 Finiteness of Transited-State Reachability Tree

A transited-state reachability tree may be infinite, that is the state enumeration process does not terminate, if the number of distinguishable marking or age vectors is infinite and the check for identical states (Definition 6) fails to keep the tree bounded. Figure 3 shows an example where an infinite number of age vectors may be generated by the above reachability algorithm. By drawing an arc from transition t_1 to p_2, in the example, the reachability tree will also have an infinite number of marking vectors. However, the number of age vectors in the transited-states (for brevity we will call them transited-age vectors) for the ESTPN in Figure 3, is finite if the transition t_2 has finite value of β. Alternatively, the tree can be made finite by extending the set of nonnegative real numbers with symbol 'ϖ', which can be thought of as "infinity" having the properties that for each real number r: $\varpi > r$, $\varpi \pm r = \varpi$, and $\varpi \geq \varpi$. We know that the reachability tree is constructed by finding and firing the scheduleable transitions (Definition 7) in each transited-state. The number and fireability of the scheduleable transitions in a given state, remain unaffected if the age of any transition with infinite value of β, is replaced by the symbol 'ϖ', whenever it is greater than the corresponding α-value. This is illustrated for the given example in Figure 3(c).



(c) A finite Reachability tree

Figure 3

The step 2.4 of the reachability algorithm is therefore modified as follows;

Step 2.4) While there exist scheduleable transitions at Š, do the following for each scheduleable transition t at Š:

Step 2.4.1) Obtain the state S, from Š, where t should fire.

Step 2.4.2) Obtain the transited-state Š' that results from firing t at S.

Step 2.4.3) Replace U' (t) by ϖ for each transition t for which β(t)= ∞ and U' (t) > α(t).

Step 2.4.4) Introduce Š' as a node, draw an arc with label t from Š to Š', and tag Š' 'new'.

Due to our inability to prove the finiteness of the transited-age vectors when either of the α and β values is an irrational number, we will restrict the following discussion for ESTPNs with α and β values from the set of rational numbers. However, examples may be given of ESTPNs where the time bounds are irrational numbers and the number of transited-age vectors is finite.

Lemma 1

For a finite set of rational numbers there exists a sizable (non-infinitesimal) number 'x' so that each member of the set is an integer multiple of 'x', and difference of any two members of the set is also an integer multiple of 'x'.

Proof:

A rational number can be written as a quotient $^p/_q$, where p and q are integers. If there are n members in the set then each member 'i' can be written as

$$\frac{p_i \prod_{j=1, j\neq i}^{n} q_j}{\prod_{j=1}^{n} q_j} \qquad \text{so that} \qquad x = \frac{1}{\prod_{j=1}^{n} q_j}$$

■

Theorem 1

For a t-safe ESTPN with an extended set of nonnegative real numbers (the nonnegative real numbers plus the symbol ϖ), the number of transited-age vectors is finite.

Proof:

In the reachability tree of a t-safe ESTPN, at a given time and with any marking vector, the entries in the transited-age vectors are either null (that is 'ϕ'), or nonnegative real numbers from a bounded domain of 0 and β values, or ϖ. That is, there are either a finite number of symbols or real numbers from a bounded domain, to appear at any entry in the transited-age vector. The number of distinguishable instances of such vector can be infinite if and only *if* the difference of entries of any two instances is infinitesimal. We know that the difference of 'ϕ' and a real number 'r' is undefined. The difference of 'ϖ'and a finite real number 'r' is 'ϖ', which is also not infinitesimal. The difference of two entries both with finite real numbers can be infinitesimal in the space of states of a ESTPN but not in the space of the transited states, because starting from the initial state the age vectors of all successive transited-age vectors are generated by applying (2) and (4). That is the real number values in all transited-age vectors are an integer multiple of the non-infinitesimal real number 'x' (Lemma 1), hence the difference of finite real numbers at the corresponding entries in any two vectors is also an integer multiple of 'x'. Hence, the number of transited-age vectors is finite. ■

As mentioned above, the second cause which makes the reachability tree infinite is the number of distinguishable marking vectors. We know that in the enumeration process of causal Petri nets, the marking of a place(s) is decided unbounded if there exists a firing sequence which can occur infinitely often, and each occurrence of the sequence produces the marking vectors where the marking of the place(s) is increased and the recurrence of the sequence is possible. The state from where starts the first occurrence of the sequence is called *coverable* by the state which has increased number of tokens and cause recurrence of the sequence. For causal Petri nets presence of coverable states (marking vectors) is checked as follows on the path from the root marking to every lately generated state (marking vector) M', [1];

For causal Petri nets, a marking M is coverable by marking M' iff:

i) $M' \neq M$

ii) $\forall p \in P: \; M'(p) \geq M(p)$

Since mark-enableness of a transition in causal Petri nets, does not guarantee that it will fire, hence the above definition does not check whether the increase in marking can mark-enable some transition. However, due to bound on the maximum age of a transition in TPNs, decision of unboundedness of any place(s), require to ensure that the increase in marking will not enable any transition, which has not fired yet in the recurring sequence, because firing of any new transition in the previously recurring sequence may change the course of future firings and keep a place from being bounded or unbounded. Accordingly, we extend the definition of coverability for TPNs in the following;

Definition 8

A transited-state $\check{S} = (M, U)$, of a Time Petri Net, is called *coverable* by another transited-state $\check{S}' = (M', U')$ if all of the following conditions hold;

i) $\check{S}' \neq \check{S}$

ii) $\forall p \in P: \; M'(p) \geq M(p)$

iii) $\forall p \in \{p \in P \mid M'(p) > M(p)\} \; \forall t \in \{ t \in T \mid B(t, p) > 0\}: \; (\beta(t) = \infty) \vee (M(p) \geq B(t, p))$

iv) $\forall t \in T: \; U'(t) = U(t)$

◆

The condition (iv) is required to confirm that $\check{S}'$ generates the same firing sequence as $\check{S}'$. Note that for Petri nets the condition (iii) is always true and the coverability may be decided, as for the corresponding causal Petri nets, under the earliest firing rule.

The number of marking vectors for reachability tree of ESTPNs can be made finite, in similar way as for causal Petri nets [1] by extending the set of nonnegative integers with symbol 'ω', which can be thought of as "infinity" having the properties that for each integer n: $\omega > n$, $\omega \pm n = \omega$, and $\omega \geq \omega$. The step 2.4 of the reachability algorithm is further modified as follows;

> *Step 2.4) While there exist scheduleable transitions at $\check{S}$, do the following for each scheduleable transition t at $\check{S}$:*
> *Step 2.4.1) Obtain the state S, from $\check{S}$, where t is fireable.*
> *Step 2.4.2) Obtain the transited-state $\check{S}'$ that results from firing t at S.*
> *Step 2.4.3) Replace $U'(t)$ by ϖ for each transition t for which $\beta(t)=\infty$ and $U'(t) > \alpha(t)$.*
> *Step 2.4.4) On the path from the root to $\check{S}'$ if there exists an state $\check{S}''$, which is coverable by $\check{S}'$ then replace $M'(t)$ by ω for each p considered in the condition (iii) of coverability.*
> *Step 2.4.5) Introduce $\check{S}'$ as a node, draw an arc with label t from $\check{S}$ to $\check{S}'$, and tag $\check{S}'$ 'new'.*

We can now prove the finiteness of the transited-state reachability tree by following lemma.

Lemma 2

For a t-safe ESTPN, the ordered set of different transited-age vectors reachable for a marking vector M, in a transited-state reachability tree is the same as that for the marking vector M', whenever M and M' satisfy the following conditions

i) $\forall p \in P: \; M'(p) \geq M(p)$

ii) $\forall p \in \{p \in P \mid M'(p) > M(p)\} \; \forall t \in \{ t \in T \mid B(t, p) > 0\}: \; M(p) \geq B(t, p)$

Proof:

The given conditions assert that both M and M' has a number of tokens equal to, or greater than the minimum number of tokens required to mark-enable a certain sub-set of transitions. Since mark-enableness (Definition 1) of transitions in t-safe ESTPNs, is unaffected by the excess of tokens, hence the null 'ϕ' entries in the two sets of the age vectors remain the same. Similarly, the real number entries in the two ordered sets are obtained from (2) and (4), which also does not depend upon the number of tokens contained by the input places of the mark-enabled transitions. Hence, the two ordered sets of the transited-age vectors are same. ■

Theorem 2

For a t-safe ESTPN with extended set of nonnegative integers (the nonnegative integers plus the symbol ω), the number of marking vectors on any path in the reachability tree is finite.

Proof:

First assuming that all places of given ESTPN are bounded. That is there is a lower and upper bound for the integers to occur at any entry of the n-dimensional marking vector, hence the number of instances of such vector are finite. Now we assume that on a given path from the root of the reachability tree, some of the places are unbounded, so that the marking vector can be considered as composed of two sub-marking vectors one of bounded places and the other of unbounded places. The number of marking vectors on any path can be infinite if and only if the number of unbounded sub-marking vectors are infinite, that is new sub-marking vectors are generated where markings of the places are increased and at least one of the coverability conditions always fail, so that marking of the unbounded places can not be replaced by symbol 'ω'. But: Condition (i) of coverability is always true because a marking vector with increased number of tokens represent a different transited-state. Condition (ii) is always true for marking vectors with equal or increased number of tokens at all places. Given that the weights of all backward incidence arcs *from* any *transition*, are finite numbers, the Condition (iii) also becomes persistently true after the marking of unbounded places exceeds a finite number. Finally, due to Lemma 2, the ordered set of transited-age vectors is same for two marking vectors satisfying conditions (ii) and (iii), hence Condition (iv) is always satisfied at each step in the recurrence of any firing sequence. Since at least one entry of the unbounded sub-marking vector can be replaced by 'ω' after a finite number of increased markings, hence all finite number of entries will be replaced by 'ω' after a finite number of the unbounded sub-marking vectors are generated. Since the number of both sub-marking vectors are finite, hence the number of marking vectors on any path is finite. ■

To this point, following (1) we have restricted the (ES)TPNs to be *t*-safe, for convenience in the discussion and to keep the notations simple. However, its easing off may be desirable in some applications. The multiple mark-enablness of a transition means that we have multiple records or instances of the transition in the age vector, each of which is invoked, possibly at different times by arrival of the minimum number of extra tokens required for mark-enableness (Definition 1). Our previous discussion can easily be extended for *t*-bounded (ES)TPNs, and the arguments remain valid because the age vector require only finite number of additional records for any transition. The case of *t-unbounded* (ES)TPNs however, has a peculiar annotation; a transition can be unbounded if and only if all of its input places are unbounded, but a place can not be declared unbounded if it does not have the minimum number of tokens required to enable all instances of its output transitions. This means that unless the term 'unboundedness' of a transition implies the unboundedness of its input places, it can not be applied to a transition and its input places at the same time. Hence, we need only consider t-bounded case, by making k greater than the largest number specified for any k-bounded transition.

Theorem 3

For an ESTPN with extended set of nonnegative integers (the nonnegative integers plus the symbol ω) for the marking vectors, and the extended set of nonnegative real numbers (the nonnegative real numbers plus the symbol ϖ) for the age vectors, the transited-state reachability tree is finite.

Proof:

The number of transited-states is the product of the number of marking vectors and transited-age vectors. Hence, it follows from Theorem 1 and Theorem 2 (both being valid for *t-bounded* ESTPNs) that the number of transited-states on any path of the tree is finite. It can be shown that for a tree with a maximum of n number of nodes on any path, and m number of branches from any node (the number of scheduleable transitions from the transited-state), the maximum number of nodes in the tree is given by

$$\sum_{i=0}^{n} m^{i}$$

■

4.2 Properties of ESTPNs

The transited-state reachability algorithm produced after the modifications has strong similarities with that of causal Petri nets. This means that we can decide at least the same properties of ESTPNs as of causal Petri nets, by reachability analysis [1]. Boundedness of buffers in manufacturing and communication systems, is one of the important properties which can be tested using the reachability tree. *An ESTPN or any place p, is bounded if and only if the symbol ω never appears for M(p) in the coverability tree.* Conversely, if the ESTPN is *unbounded*, then the number of reachable transited-states is infinite. Since the reachability tree is finite, the symbol 'ω' must appear to represent the infinite number of marking vectors. However, as for causal Petri nets, since ω represents unlimited number of markings at a place, hence reachability of a particular marking can not be decided from the reachability tree.

TPNs have the ability to test a single place for zero, hence they can simulate Turing machine. This capability lead Jones et. al. [12] to prove the reachability and boundedness of TPNs undecidable. They argue that these properties can be decided if and only if the halting problem of TPNs is decided, since TPNs simulate Turing machines for which the halting problem is undecidable hence the given properties are also undecidable. It is obvious that by restricting TPN transitions to fire at their earliest firing time, we do not lose its modeling power, hence the argument of Jones et. al. applies to boundedness of ESTPNs as well, which we have shown decidable. This contradiction is due to an invalid assumption in [12]. The reachability problem considered in Petri net literature, may be stated as: 'is a given marking reach*able*?' or '*can* a given marking be reached?', and the boundedness problem about a place: 'does there exist a finite natural number to be specified as the maximum number of tokens which the place *can* have?' It is obvious that the two problems are of determining the capability of the model, whereas the halting problem is a problem of predicting the future behavior of a Turing machine or TPN, which may be stated as: '*will* it ever halt?' Jones et. al. assume the logical similarity of the two questions, which is not true. If the '*can*' and '*will*' questions were equivalent then halting problem become trivially decidable, because the former asks about the capability of a Turing machine to halt, which it is known to have.

5 Conclusion

There may exist an unlimited number of DSTPNs for any TPN. Computational cost aside, the reachability analysis can be performed, at least for some of the DSTPNs. Similar analysis for TPNs is apparently not possible due to non-determinism of the scheduling time of transitions, but the undecidability of its properties is not confirmed and it is very probable that some of them can be defined as a function of the decidable properties of a finite number of DSTPNs.

REFERENCES

[1] J.L. Peterson, *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Englewood Cliffs, N.J., 1981.

[2] T. Murata, "Petri nets: properties, analysis, and applications," *Proc.of the IEEE*, vol. 77, pp. 541-580, Apr. 1989.

[3] C. Ramchandani, "Analysis of asynchronous concurrent systems by Petri nets," Ph.D. dissertation, L.C.S./M.I.T., Cambridge, MA, Feb. 1974.

[4] P. Merlin, "A methodology for the design and implementation of communication protocols," *IEEE Trans. Commun.*, vol. COM-24, pp. 614-621, 1976.

[5] M.K. Molloy, "Performance modeling using stochastic Petri nets," *IEEE Trans. Comput.*, vol. C-31, pp. 913-917, 1982.

[6] P. Freedman, "Time, Petri nets, and robotics," *IEEE Trans. on Robotics and Automat.*, vol. 7, pp. 417-433, 1991.

[7] C. Ramamoorthy and G. Ho, "Performance evaluation of asynchronous concurrent systems using Petri nets," *IEEE Trans. Software Eng.*, vol. SE-6, pp. 440-449, 1980.

[8] J. Carlier and P. Chretienne, "Timed Petri net schedules," *Advances in Petri nets, Lecture Notes in Computer Science*, vol. 340, pp. 62-84, 1988.

[9] B. Berthomieu and M. Diaz, "Modeling and verification of time dependent systems using Time Petri nets," *IEEE Trans. Software Eng.*, vol. 17, pp. 259-273, 1991.

[10] T. Yoneda and Y. Tohma, "Acceleration of timing verification method based on Time Petri nets," *Systems and Computers in Japan*, vol. 22, pp. 37-52. 1991.

[11] N.G. Leveson and J.L. Stolzy, "Safety analysis using Petri nets," *IEEE Trans. on Software Eng.*, vol. 13, pp. 386-397, 1987.

[12] N.D. Jones, L.H. Landweber, and Y.E. Lien, "Complexity of some problems in Petri nets," *Theoretical Computer Science*, vol. 4, pp. 277-299, 1977.

[13] S. French, *Sequencing and Scheduling - An Introduction to the Mathematics of the Job-shop*. Ellis Horwood Ltd., Chichester, 1982.

SHEFFIELD UNIV. LIBRARY ST. GEORGE'S