IEEE Conference on Computer Communications (INFOCOM'87), San Francisco, CA, March 31 - April 2, 1987, pp-721-730.

PREEMPTIVE D-TIMED PETRI NETS, TIMEOUTS, MODELLING AND ANALYSIS OF COMMUNICATION PROTOCOLS

W.M. Zuberek

Department of Computer Science, Memorial University of Newfoundland
St. John's, NL, Canada A1C–5S7

Abstract

Preemptive D-timed Petri nets are Petri nets with deterministic firing times and with generalized inhibitor arcs to interrupt firing transitions. A formalism is presented which represents the behavior of free-choice D-timed Petri nets by discrete-space discrete-time semi-Markov processes. Stationary probabilities of states can thus be determined by standard techniques used for analysis of Markov chains. A straightforward application of timed Petri nets is modelling and analysis of systems of asynchronous communicating processes, and in particular communication protocols. Places of Petri nets model queues of messages, transitions represent delays in communication networks, interrupt arcs conveniently model timeout mechanisms, and probabilities associated with free-choice classes correspond to relative frequencies of random events. Simple protocols are used as an illustration of modelling and analysis.

1. INTRODUCTION

The increasing trend to distribute functions of information-processing systems into different processes and processors results in significant growth of complexity of coordination in such systems. This is particularly acute for the interactions or protocols that specify how distributed processes or processors are synchronized and how they communicate with one another. Formal methods are gradually being developed to describe such interactions [6,9,14,25], and Petri nets [1,7,19,23] proved to be an interesting and advantageous example of such formalisms [8,9,13,15]. Petri nets have been successfully used in modelling [3,10,18], validation [4,13] and analysis [8,10,16] of systems of events in which it is possible for some events to occur concurrently, but there are constraints on the occurrence, precedence, or frequency of these occurrences. Basic Petri nets, however, are not complete enough for the performance studies since no assumption is made on the duration of systems events. Timed Petri nets have been introduced by Ramchandani [20] by assigning firing times to transitions of Petri nets (t-timed nets), and by Sifakis [24] by assigning time to places of a net (p-timed nets). Garg [11] used p-timed nets for specification of communication protocols. Merlin and Farber [16] discussed timed Petri nets where a time threshold and maximum delay were assigned to each transition of a net to allow the incorporation of timeouts into protocol models. Berthomieu and Menasche [5,15] used *state classes* to obtain finite representation of behavior of nets defined by Merlin and Farber; their description is sufficient for validation and verification studies, but requires further refinements for performance analysis since no distribution of firing times is assumed. Razouk [21] and Razouk and Phelps [22] discussed an interesting class of timed Petri nets with enabling as well as firing times (p&t-timed nets), and derived performance expressions for simple communication protocols; since the enabling and firing times correspond to the time thresholds of Merlin and Farber, such nets can easily model timeout mechanisms, however, the proposed formalisms requires some further refinements or extensions in cases of multi-level priorities and more general conflicts. A thorough study of general conflicts in t-timed nets was presented by Holliday and Vernon [12].

This paper describes a continuation of the approach originated by Ramchandani [20] and subsequently extended by inhibitor arcs and guarded places to model timeouts [26,27]. In inhibitor Petri nets, however, firing transitions cannot be]em interrupted, and timeout signals must be rather *neutralized* than *canceled* or *cleared*. In this paper, inhibitor Petri nets are augmented by *interrupt* arcs in order to *cancel* initialized firings of transitions, as required in strict modelling of timeouts. Similarly as in [20,26,27], constant (or deterministic) firing times are assigned to transitions of a Petri net, and a *state* description is introduced which is isomorphic to discrete-space discrete-time homogeneous semi-Markov processes. Standard techniques derived for analysis of Markov chains can thus be used to derive many performance measures such as utilization of systems components, average waiting times and turnaround times or average throughput rates, and at the same time preserves the simplicity of model specification and offers automatic generation of the state space.

This paper is organized in 4 main sections. Section 2 contains definitions of basic concepts for inhibitor free-choice Petri nets. Extended D-timed Petri nets are introduced in Section 3. Application of extended D-timed Petri nets to modelling and analysis of protocols is discussed in Section 4. Section 5 discusses enhanced D-timed Petri nets which combine two types of Petri nets, ordinary (i.e., without time) inhibitor nets and extended D-timed nets in order to reduce the state space by elimination of all *vanishing* states, i.e., those states which formally belong to the state space, but which do not contribute to the stationary behavior of a net.

2. INHIBITOR PETRI NETS

An inhibitor Petri net $\mathbf{N}$ is a quadruple $\mathbf{N} = (P, T, A, B)$ where:

P is a finite, nonempty set of places,

T is a finite, nonempty set of transitions,

A is a set of directed arcs which connect places with transitions and transitions with places such that for each transition there is at least one place connected with it: $\forall t \in T \ \exists p \in P \ : \ (p, t) \in A$,

B is a (possibly empty) set of inhibitor arcs which connect places with transitions, $B \subset P \times T$, and A and B are disjoint sets.

A place p is an input (or an output) place of a transition t iff there exists an arc (p, t) (or (t, p), respectively) in the set A. The sets of all input and output places of a transition t are denoted by $Inp(t)$ and $Out(t)$, respectively. Similarly, the sets of input and output transitions of a place p are denoted by $Inp(p)$ and $Out(p)$. Also, a place p is an inhibitor place of a transition t iff $(p, t)]inB$. The set of all inhibitor places of t is denoted by $Inh(t)$, and the set of transitions connected by inhibitor arcs with a place p is denoted by $Inh(p)$, $Inh(p) = \{t | p \in Inh(t)\}$. The notation is extended in the standard way to sets of places and transitions.

A marked Petri net $\mathbf{M}$ is a pair $\mathbf{M} = (\mathbf{N}, m_0)$ where:

$\mathbf{N}$ is an inhibitor Petri net, $\mathbf{N} = (P, T, A, B)$,

m_0 is an initial marking function which assigns a nonnegative integer number of so called tokens to each place of the net, $m_0 : P \rightarrow \{0, 1, ...\}$.

Let any function $m : P \rightarrow \{0, 1, ...\}$ be called a marking in a net $\mathbf{N} = (P, T, A, B)$.

A transition t is enabled by a marking m iff every input place of this transition contains at least one token and every inhibitor place of t contains zero tokens. The inhibitor arcs provide thus a *test if zero* condition, which is not available in basic Petri nets [1,7,19]. The set of all transitions enabled by a marking m is denoted by $En(m)$.

A place p is shared iff it is an input place for more than one transition. In inhibitor nets, a shared place p is guarded iff for each two different transitions t_i and t_j sharing p there exists another place p_k such that p_k is in the input set of one and in the inhibitor set of the other of these two transitions:

$$\forall t_i \in Out(p) \ \forall t_j \in Out(p) - \{t_i\} \ \exists p_k \in P - \{p\} \ : \ (p_k, t_i) \in A \wedge (p_k, t_j) \in B) \vee (p_k, t_i) \in B \wedge (p_k, t_j) \in A).$$

i.e., no two transitions from the set $Out(p)$ can be enabled by the same marking.

A shared place p is free-choice (or extended free-choice [7]) iff the input sets and inhibitor sets of all transitions sharing p are identical. An inhibitor net is free-choice iff all its shared places are either free-choice or guarded. Only free-choice nets are considered in this paper since in most cases free-choice nets are sufficient for modelling of random events, e.g., random faults in communication networks or random services with discrete distributions (another class of timed Petri nets is used for random events with continuous distributions [2,17,28]). Some other classes of nets can be described in a similar way.

Every transition enabled by a marking p can fire. When a transition fires, a token is removed from each of its input places (but not inhibitor places) and a token is added to each of its output places. This determines a new marking in a net, a new set of enabled transitions, and so on.

A marking m_j is directly reachable (or t_k-reachable) from a marking m_i in a net $\mathbf{N}$ iff there exists a transition t_k enabled by the marking m_i, $t \in En(m_i)$, such that

$$\forall p \in P) \ : \ m_j(p) = \begin{cases} m_i(p) - 1, \ \text{if } p \in Inp(t_k) - Out(t_k), \\ m_i(p) + 1, \ \text{if } p \in Out(t_k) - Inp(t_k), \\ m_i(p), \ \text{otherwise.} \end{cases}$$

A marking m_j is (generally) reachable from a marking m_i in a net $\mathbf{N}$ if there exists a sequence of markings $(m_{i_0} m_{i_1} m_{i_2} ... m_{i_k})$ such that $m_{i_0} = m_i$, $m_{i_k} = m_j$, and each marking m_{i_ℓ} is directly reachable from the marking $m_{i_{\ell-1}}$ for $\ell = 1, ..., k$.

A set $M(\mathbf{M})$ of reachable markings of a marked Petri net $\mathbf{M} = (\mathbf{N}, m_0)$ is the set of all markings which are reachable from the initial marking m_0 in the net $\mathbf{N}$.

A marked net $\mathbf{M}$ is bounded if there exists a positive integer k such that each marking in the set $M(\mathbf{M})$ assigns at most k tokens to each place of the net
$\exists k > 0 \ \forall m \in M(\mathbf{M}) \ \forall p \in P \ : \ m(p) \leq k$.

An obvious conclusion is that a marked net $\mathbf{M}$ is bounded iff its reachability set $M(bfM)$ is finite.

In analysis of timed nets it appears very convenient to have a concise notation that indicates all possibilities of firing transitions for a given marking m. The set of selection functions is introduced to describe all such possibilities.

A selection function g of a marking m in a net $\mathbf{N}$ is any function $g : T \rightarrow \{0, 1, ...\}$ which satisfies the following conditions:

(1) there exists a (finite) sequence of transitions $u = (t_{i_1}, t_{i_2}, ..., t_{i_k})$, such that $t_{i_j} \in En(m_{i_{j-1}})$ for $j = 1, ..., k$, where $m_{i_0} = m_0$ and
$$\forall p \in P \ : \ m_{i_j}(p) = m_{i_{j-1}}(p) - \begin{cases} 1, \ \text{if } p \in Inp(t_{i_j}), \\ 0, \ \text{otherwise ;} \end{cases}$$

(2) the set of transitions enabled by m_{i_k}, $En(m_{i_k})$, is empty;

(3) for each $tinT$, $g(t)$ is equal to the number of occurrences of t n the sequence u.

A selection function g is thus any function which indicates (by nonzero values) all those transitions which can simultaneously initiate their firings (and some transitions may initiate their firings *several times*).

The set of all selection functions of a marking m is denoted by $Sel(m)$.

A marked net $\mathbf{M}$ is singular iff all selection functions of all reachable markings indicate at most a single firing of a transition

$\forall m \in M(\mathbf{M})\ \forall g \in Sel(m))\ \forall t \in T)\ :\ g(t) \leq 1.$

To simplify the description of timed Petri nets, only singular free-choice nets are considered in this paper. Nonsingular nets can be described by a rather straightforward extension of the formalism given in the next section.

Example. The Petri net shown in Fig.1 (as usual, places are represented by circles, transitions by bars, inhibitor arcs by arcs with small circles instead of arrowheads, and the initial marking function is indicated by dots inside places) contains one inhibitor arc (p_5, t_5), one free-choice place p_3, and one guarded place p_4.

The initial part of the derivation of the set of reachable markings $M(\mathbf{M})$ is given in Tab.1; it should be observed that the set $M(\mathbf{M})$ is infinite since the sequence of firing transitions $(t_1, t_5, t_1, t_5, ...)$ can be continued *for ever*, creating consecutive markings $(m_1, m_3, m_7, ...)$ in which the number of tokens in p_2 increases. Similarly, the cyclic sequence of firing transitions $(t_1, t_2, t_4, t_1, ...)$ systematically increases the number of tokens in p_5. Consequently, the net is unbounded. □

i	m_i 1 2 3 4 5	t_k	j
0	1 0 0 0 0	1	1
1	0 1 0 1 0	2	2
		5	3
2	0 0 1 1 0	3	4
		4	5
		5	6
3	1 1 0 0 0	1	7
		2	6
4	0 0 0 1 0	5	0
5	1 0 0 1 1	1	8
		6	0
6	1 0 1 0 0	1	9
		3	0
		4	10
7	0 2 0 1 0	2	9
		5	11
8	0 1 0 2 1	2	12
		6	1
9	0 1 1 1 0	2	13
...	....	...	...

Tab.1. Initial markings reachable from $m_0 = [1, 0, 0, 0, 0]$.

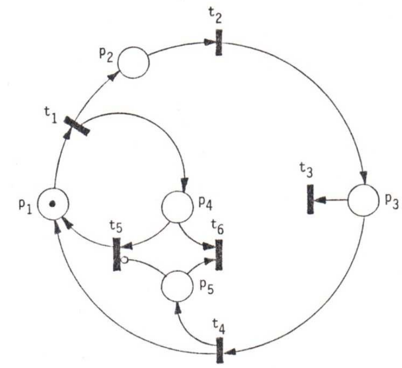

Fig.1. Marked Petri net $\mathbf{M}_1$.

3. EXTENDED D-TIMED PETRI NETS

In timed Petri nets each transition fires in *real time*, i.e., there is a *firing time* associated with each transition of a net. The firing times can be defined in several ways. In D-timed Petri nets [26,27] they are deterministic (or constant), i.e., there is a nonnegative number assigned to each transition of a net which determines the duration of transition's firings. In M-timed Petri nets [28] (or stochastic Petri nets [2,17]), the firing times are exponentially distributed random variables, and the corresponding firing rates are assigned to transitions of a net.

Since in timed Petri nets the firing of a transition is not an instantaneous event, the inhibitor arcs can be generalized. The *proper* inhibitor arcs affect the transitions only at the beginning of their firings because they participate in enabling of transitions. The generalized inhibitor arcs, called interrupt arcs, affect a transition not only at the beginning of firing, but also during its firing; they can *interrupt* firing transitions and preempt the *resources* acquired at the beginning of firing. Interrupt arcs are necessary to model preempting scheduling disciplines, to represent properly timeout mechanisms, or to model unreliable processors which can *fail* during processing of user jobs. In some cases such interrupts and preemptions can be represented by inhibitor nets [27], but usually such models and their behavior become unnecessarily complicated.

An extended D-timed free-choice Petri net $\mathbf{T}$ is a triple $\mathbf{T} = (\mathbf{M}, c, f)$ where:

M is an extended free-choice marked Petri net, $\mathbf{M} = (\mathbf{N}, m_0)$, $\mathbf{N} = (P, T, A, B, C)$, and C is a set of interrupt arcs, $C \subseteq B$,

c is a choice function which assigns a *free-choice* probability to each transition t of the net in such a way that for each free-choice place p of $\mathbf{N}$:

$$\sum_{t \in Out(p)} c(t) = 1,$$

and for all remaining transitions $c(t) = 1$,

f is a firing time function which assigns a nonnegative real number $f(t)$ to each transition t of the net, $f : T \to \mathbf{R}^+$ and $\mathbf{R}^+$ denotes the set of nonnegative real numbers.

In ordinary nets (i.e., nets without time), interrupt arcs are equivalent to inhibitor arcs since the firing of an enabled transition is an instantaneous event. In extended timed Petri nets, the firing of a transition may be *discontinued* by any one of interrupt arcs associated with this transition. If, during a firing period of a transition t, one of places connected with p by interrupt arcs becomes nonempty (i.e., it contains at least one token), the firing of t ceases and the tokens removed from

t's input places at the beginning of firing are *returned* to their original places.

In extended nets, a place @p@ is an interrupting place of a transition t iff $(p,t) \in C$. The set of interrupting places of t is denoted by $Int(t)$, and the set of transitions connected with p by interrupt arcs is denoted by $Int(p)$, $Int(p) = \{t \in T \mid pinInt(t)\}$.

Moreover, an extended Petri net is simple if the input sets of transitions with nonempty interrupting sets are disjoint with interrupting sets of other transitions, i.e.

$$\forall t \in T \ : \ Int(t) = \emptyset \vee Inp(t) \cap Int(T - \{t\} = \emptyset,$$

where $\emptyset$ denotes the empty set. Simple nets eliminate *propagation* of interrupts when one interrupted transition, through its input set, interrupts another transition.

In order to simplify the description of net behavior, only simple nets are considered in this paper.

The behavior of an extended D-timed Petri net can be represented by a sequence of *states* where each *state* describes the current marking as well as the firing transitions of a net. Each termination of a transition firing changes the state of a net.

A state s of an extended D-timed Petri net $\mathbf{T}$ is a triple $s = (m, n, r)$ where:

m is a marking function, $m : P \to \{0, 1, ...\}$,

n is a firing-rank function which indicates the number of active firings (i.e., the number of firings which have been initiated but are not yet terminated) for each transition of the net, $n : T \to \{0, 1, ...\}$,

r is a remaining-firing-time function which assigns the remaining firing time to each independent firing (if any) of a transition, i.e., if the firing rank of a transition t is equal to $k > 0$, $n(t) = k$, the remaining-firing-time function $r(t)$ is a vector of k nonnegative nondecreasing real numbers denoted by $r(t)[1]$, $r(t)[2]$, ..., $r(t)[k]$; r is a partial function and it is undefined for all those transitions t for which $n(t) = 0$.

An initial state s_i of a net $\mathbf{T}$ is a triple $s_i = (m_i, n_i, r_i)$ where n_i is a selection function from the set $Sel(m_0)$, $n_i \in Sel(m_0)$, the remaining-firing-time function is equal to the firing times $f(t)$ for all those transitions t for which $n_i > 0$:

$$\forall t \in T \ : \ r(i)(t) = \begin{cases} f(t), & \text{if } n_i(t) > 0, \\ \text{undefined}, & \text{otherwise;} \end{cases}$$

and the marking m_i is defined as:

$$\forall p \in P \ : \ m_i(p) = m_0(p) - \textstyle\sum_{t \in Out(p)} n_i(t) \ .$$

An extended free-choice D-timed net $\mathbf{T}$ may have several different initial states.

A state $s_j = (m_j, n_j, r_j)$ is directly reachable (or g_k–reachable) from the state $s_i = (m_i, n_i, r_i)$, iff the following conditions are satisfied:

(1) $g_k \in Sel(m_{iji})$,

(2) $\forall p \in P \ : \ m_j(p) = m_{iji}(p) - \sum_{t \in Out(p)} g_k(t)$,

(3) $\forall t \in T \ : \ n_j(t) = n_i(t) - a_i(t) - d_i(t) + g_k(t)$,

(4) $\forall t \in T \ : \ r_j(t)[\ell] = \begin{cases} r_i(t)[\ell + a_i(t) + d_i(t)] - h_i, \\ \text{if } 1 \le \ell \le n_i(t) - a_i(t) - d_i(t), \\ f(t), \ \text{if } g_k(t) > 0 \wedge \ell = n_j(t), \end{cases}$

where:

(5) $\forall p \in P \ : \ m_{iji}(p) = m_{ij}(p) + \sum_{t \in Out(p)} d_i(t)$,

(6) $\forall tinT \ : \ d_i(t) = min(n_i(t) - a_i(t), \sum_{p \in Int(t)} m_{ij}(p))$,

7) $\forall p \in P \ : \ m_{ij}(p) = m_i(p) + \sum_{t \in Inp(p)} a_i(t)$,

(8) $\forall t \in T \ : \ a_i(t) = \begin{cases} 1, & \text{if } n_i(t) > 0 \wedge r_i(t)[1] = h_i, \\ 0, & \text{otherwise;} \end{cases}$

(9) $h_i = min_{t \in T}(r_i(t)[1])$.

The state s_j which is g_k-reachable from the state s_i is thus obtained by the termination of the *next* firings (i.e., those firings for which the remaining firing time is the smallest one; this time is denoted by h_i and is called the holding time or the sojourn time of state s_i) (8,9), updating the marking of a net (7), checking if updated interrupting sets discontinue any active firing and performing all required modifications (5,6), and then initiating new firings (if any) which correspond to the selection function g_k from the set $Sel(miji)$ (1,2,3 and 4). It should be observed that the number of interrupted firings is determined by the total number of tokens in the set of interrupting places (6).

Similarly as for marked nets, a state s_j is (generally) reachable from a state s_i if there is a sequence of directly reachable states from the state s_i to the state s_j. Also, a set $S(\mathbf{T})$ of reachable states is defined as the set of all states of a net $\mathbf{T}$ which are reachable from the initial states of the net $\mathbf{T}$.

A state graph $\mathbf{G}$ of a D-timed Petri net $\mathbf{T}$ is a labeled directed graph $\mathbf{G}(\mathbf{T}) = (V, D, b)$ where:

V is a set of vertices which is equal to the set of reachable states of the net $\mathbf{T}$, $V = S(\mathbf{T})$,

D is a set of directed arcs, $D \subset V \times V$, such that (s_i, s_j) is in D iff s_j is directly reachable from s_i,

b is a labeling function which assigns the probability of transition from s_i to s_j to each arc (s_i, s_j) in the set D, $b : D \to [0, 1]$, in such a way that if s_j is g_k-reachable from s_i, then

$$b(s_i, s_j) = \prod_{t \in T} c(t)^{g_k(t)}.$$

Example. The D-timed Petri net shown in Fig.2 (the interrupt arcs have small dots instead of arrowheads and the firing time function as well as the choice function are given as additional descriptions of transitions) is a refinement of the net from Fig.1; it contains one interrupt arc (p_5, t_5), one free-choice place p_3, and one guarded place p_4.

The state graph for this net is shown in Fig.3, and the derivation of the set $S(\mathbf{T})$ of reachable states is given in Tab.2. □

s_i	m_i 1 2 3 4 5	n_i 1 2 3 4 5 6	h_i	m_{ij} 1 2 3 4 5	g_k 1 2 3 4 5 6	s_j	$b(s_i, s_j)$
1	0 0 0 0 0	1 0 0 0 0 0	1.0	0 1 0 1 0	0 1 0 0 1 0	2	1.00
2	0 0 0 0 0	0 1 0 0 1 0	2.0	0 0 1 0 0	0 0 0 1 0 0	3	0.90
					0 0 1 0 0 0	4	0.10
3	0 0 0 0 0	0 0 0 1 1 0	2.0	1 0 0 0 1	1 0 0 0 0 1	5	1.00
4	0 0 0 0 0	0 0 1 0 1 0	0.0	0 0 0 0 0	0 0 0 0 0 0	6	1.00
5	0 0 0 0 0	1 0 0 0 0 1	0.0	0 0 0 0 0	0 0 0 0 0 0	1	1.00
6	0 0 0 0 0	0 0 0 0 1 0	3.0	1 0 0 0 0	1 0 0 0 0 0	1	1.00

Tab.2. The set of reachable states for $\mathbf{T}_1$.

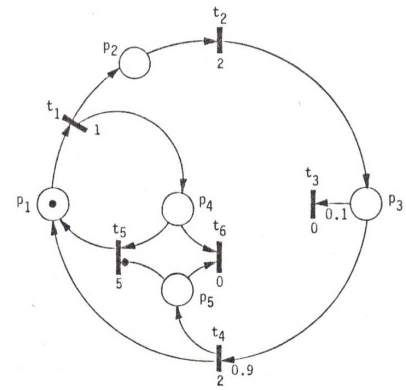

Fig.2. D-timed Petri net $\mathbf{T}_1$.

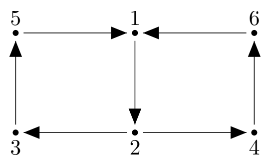

Fig.3. State graph $\mathbf{G}(\mathbf{T}_1)$.

A D-timed Petri net $\mathbf{T}$ is p-bounded if there exists a positive integer k_p such that for each state $s = (m, n, r)$ in $S(bfT)$, the marking function m assigns at most k_p tokens to each place of the net

$$\exists k_p > 0 \ \forall (m, n, r) \in S(\mathbf{T}) \ \forall p \in P \ : \ m(p) \leq k_p \ .$$

Also, $\mathbf{T}$ is t-bounded if there exists a positive integer k_t such that for each state $s = (m, n, r)$ in $S(\mathbf{T})$, the firing-rank function n assigns at most k_t firings to each transition of the net

$$\exists k_t > 0 \ \forall (m, n, r) \in S(\mathbf{T}) \ \forall t \in T \ : \ n(t) \leq k_t \ .$$

Finally, $\mathbf{T}$ is bounded iff it is p-bounded and t-bounded.

Similarly as for marked nets, $\mathbf{T}$ is bounded iff its set of reachable states $S(\mathbf{T})$ is finite.

Only bounded D-timed Petri nets are considered in this paper.

It should be observed that in general case there is no direct relationship between boundedness of a D-timed Petri net $\mathbf{T} = (\mathbf{M}, c, f)$ and its underlying marked net $\mathbf{M}$. The D-timed net from Fig.2 is bounded (Tab.2) while its underlying marked net (Fig.1) in unbounded (Tab.1).

4. PROTOCOL MODELLING AND ANALYSIS

The Petri net shown in Fig.2 is a model of a very simple protocol in which messages are exchanged between a sender (place p_1) and a receiver (place p_3), and each received message is confirmed by an acknowledgement sent back to the sender (in the loop $p_1, t_1, p_2, t_2, p_3, t_4, p_1$). There is a nonzero probability that the system can lose (or distort) a message or an acknowledgement; the place p_3 is a free-choice place, and the transition t_3 models a message/acknowledgement *sink*; the probability associated with t_3, $c(t_3)$, represents thus the probability of losing a message or an acknowledgement (or shortly a *token*) in the system. A *timeout* is used to recover from lost *tokens*. It works in the following way. An event of *sending a message* is modelled by the transition t_1. When it fires, single tokens are deposited in p_2 (a *message*) and in p_4 (a *timeout*). A token in p_4 immediately starts a firing of the *timeout* transition t_5 (since p_5 is empty). The firing time associated with t_5 is large enough to allow the transfer of a message and an acknowledgement. If there is no loss of tokens, i.e., if t_4 is selected for firing (according to its probability), the transition t_4 will finish its firing before t_5, and then a token in the place p_5 interrupts and cancels the timeout (i.e., the firing of t_5), the *timeout* token is returned to p_4, and then t_6 fires and removes the tokens from p_4 and p_5 (t_6 is another token *sink*). If, however, a message or acknowledgement has been lost (i.e., if t_3 has been selected for firing instead of t_4), the timeout t_5 ends its firing without interruption, and regenerates the *lost* token in p_1, i.e., the message is retransmitted to the receiver.

The state graph representing the behavior of this model contains 6 states, as shown in Fig.3 (and Tab.2). It contains one *decision* node (the state s_2) with two outgoing branches and branch probabilities q and $(1-q)$, corresponding to a lost and regenerated token, and a correct transfer, respectively (in the example $q = c(t_3) = 0.1$). It can be observed that the state graph can be reduced (by simply *aggregating* the paths connecting the *decision* node) to a single-node graph shown in Fig.4 in which one branch represents *a loss and recovery of a token* (and replaces the states s_1, s_2, s_4 and s_6), and the second branch represents correct transfers (i.e., it replaces the states s_1, s_2, s_3 and s_5).

Fig.4. Reduced transition graph for $\mathbf{T}_1$.

The times associated with the branches (i.e., the total times of *aggregated* paths) correspond to the *transition times*, and are equal to 6 and 5, respectively. The (reduced) transition graph from Fig.4 can be transformed to an equivalent state graph [27] shown in Fig.5, with the state holding times equal to $h_1 = 6$ and $h_2 = 5$ and transition probabilities q and $1 - q$.

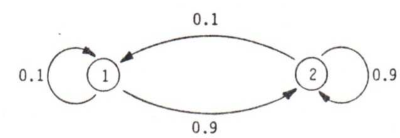

Fig.5. Reduced state graph for $\mathbf{T}_1$.

The stationary probabilities x_1 and x_2 of the reduced states s_1 and s_2 (Fig.5) can be obtained as the solution of two equations

$$x_1 = q * x_1 + q * x - 2,$$
$$x_1 + x_2 = 1,$$

from which $x_1 = q$ and $x_2 = 1 - q$. In general case, the stationary probabilities $x(s)$ of the reduced states $s \in S_r(\mathbf{T})$ are obtained by solving a system of simultaneous linear equations

$$\begin{cases} \sum_{(s_j,s_i)\in D_r} b(s_j, s_i) * x(s_j) = x(s_i), \quad i = 1, ..., K-1; \\ \sum_{1\le i\le K} x(s_i) = 1; \end{cases}$$

where K is the number of states in the reduced set of states $S_r(\mathbf{T})$.

Since the stationary probability of correct transfers is equal to $1 - q = 0.9$, and one correct transfer requires $f(t_1) + f(t_2) + f(t_4) = 5$ time units, the throughput ρ is equal to $0.9 * 5/(0.9 * 5 + 0.1 * 6)/5 = 0.174$ messages per time unit.

The total time required for one correct transfer is composed of the message sending time ($f(t_1)$) and two transmission delays, one for a message ($f(t_2)$) and the second for acknowledgement ($f(t_4)$); the acknowledgement is usually very short, so its sending time is negligible. The message sending time is directly proportional to the length of a message, and inversely proportional to the data signaling rate (or the speed of transmission) v. For a fixed message length, the transfer time can thus be reduced by increasing the data signaling rate v; then, however, the probability of errors, q, is increased. Assuming (for simplicity of consideration) that q is a linear function of v

$$q(v) = q_0 + q_1 * v$$

and that the total transfer time $d(v)$ can be approximated by the following simple function

$$d(v) = d_0 + d_1/v,$$

the throughput ρ as a function of v is

$$\rho(v) = (1 - q(v))/d(v) = (1 - q_0 - q_1 * v)/(d_0 + d_1/v).$$

The function $\rho(v)$ has a maximum between 0 and $(1 - q_0)/q_1$, as shown in Fig.6 (for $q_0 = 0$, $q_1 = 0.1$, $d_0 = 3$ and $d_1 = 2$ the maximum throughput is equal to 0.2 messages per time unit, and it corresponds to $v = 2$). It should be noticed that the initial part of this curve is very steep.

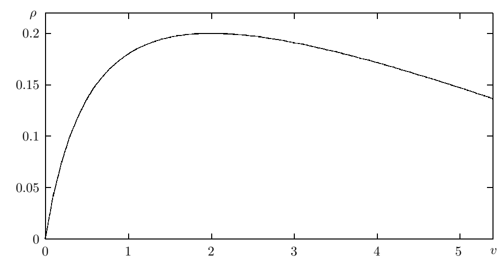

Fig.6. Throughput ρ as a function of v.

Further improvement of throughput can be obtained by optimizing the length of messages [26].

5. ENHANCED D-TIMED PETRI NETS

Extended D-timed Petri nets usually contain some *auxiliary* transitions which have zero firing times (e.g., t_3 and t_6 in Fig.2), i.e., transitions which fire *instantaneously.* Such transitions do not contribute to the *timed behavior* of a net, but they increase the state space generating states with holding time equal to zero. Enhanced Petri nets eliminate such *vanishing* states during generation of the set of reachable states, and this can considerably reduce the state space as well as simplify the performance analysis (based on stationary probabilities). Enhanced Petri nets combine two different classes of Petri nets, *immediate* nets which are in fact ordinary (i.e., without time) inhibitor free-choice *straight* Petri nets, and *timed* nets which are extended free-choice simple D-timed Petri nets.

A net $\mathbf{N}_i = (P_i, T_i, A_i, B_i, C_i)$ is a T_i-implied subnet of an extended net $\mathbf{N} = (P, T, A, B, C)$ iff

(1) $T_i \subset T$,

(2) $A_i = A \cap (P \times T_i \cup T_i \times P)$,

(3) $B_i = B \cap (P \times T_i)$,

(4) $C_i = C \cap (P \times T_i)$,

(5) $P_i = Inp(T_i) \cup Out(T_i) \cup Inh(T_i) \cup Int(T_i)$.

A net $\mathbf{N}$ is straight iff for all initial markings m_0, the set of reachable markings $M(\mathbf{M})$, $\mathbf{M} = (\mathbf{N}, m_0)$, can be ordered in such a way that for all pairs of markings m_i and m_j from the set $M(\mathbf{M})$, m_j can be (generally) derived from m_i only if i is smaller than j, i.e., iff the marking graph of $\mathbf{M}$ [19,28] is acyclic. In straight nets all firing sequences of (finite) markings are finite. If a graph of a net $\mathbf{N} = (P, T, A, B, C)$ does not contain cycles, i.e., if a directed graph $(P \cup T, A)$ is acyclic, the net is obviously straight, but many cyclic nets are also straight.

An enhanced Petri net $\mathbf{H}$ is a 6-tuple $\mathbf{H} = (P, T_t, T_0, A, B, C)$ where:

$(P, T_t \cup T_0, A, B, C)$ is an extended free-choice Petri net,

T_t is a set of timed transitions,

T_0 is a set of immediate transitions such that the T_0-implied subnet of $(P, T_t \cup T_0, A, B, C)$ is straight, the sets T_t and T_0 are disjoint, and for each free-choice place p:

$$Out(p) \subseteq T_t \vee Out(p) \subseteq T_0,$$

i.e., each free-choice class of transitions must belong either to the set of timed or to the set of immediate transitions.

The set of all transitions is denoted by T, $T = T_t \cup T_0$.

An enhanced marked Petri net $\mathbf{M}$ is a pair $\mathbf{M} = (\mathbf{H}, m_0)$ where $\mathbf{H}$ is a an enhanced Petri net and m_0 is the initial marking function.

Similarly as before, the set of transitions enabled by a marking m is denoted by $En(m)$. Moreover, $En_0(m)$ denotes the set of immediate transitions enabled by m, and $En_t(m)$ the set of timed transitions enabled by m.

An enhance function e of a marking m in a net $\mathbf{M}$ is any function $e : T \to \{0, 1, ...\}$ such that:

(1) there exists a finite (possibly empty) firing sequence of immediate transitions $u = (t_{i_1}, t_{i_2}, ..., t_{i_k})$ which transforms the marking m into a marking m_k, and the set of immediate transitions enabled by m_k, $En_0(m_k)$, is empty, and

(2) for each immediate transition $t \in T_0$ the number of occurrences of t in the sequence u is equal to $e(t)$, while for each timed transition $t \in T_t$, $e(t) = 0$.

The set of all enhance functions of a marking m is denoted by $Enh(m)$.

Since the immediate subnet (i.e., the T_0-implied subnet) is straight, for each marking m, the set $Enh(m)$ is finite.

In enhanced nets, selection functions are defined similarly as in extended (or inhibitor) nets but with respect to the timed transitions only. A selection function of a marking m in an enhanced net $\mathbf{H}$ is thus any such function $g : T \to \{0, 1, ...\}$ that:

(1) there exists a sequence of timed transitions $w = (t_{i_1}, t_{i_2}, ..., t_{i_k})$ in which $t_{i_j} \in En_t(m_{i_{j-1}})$ for $j = 1, ...k$ and for $m_{i_0} = m$, where:

$$\forall p \in P \;:\; m_{i_j}(p) = m_{i_{j-1}}(p) - \begin{cases} 1, & \text{if } p \in Inp(t_{i_j}), \\ 0, & \text{otherwise}; \end{cases}$$

(2) the set of timed transitions enabled by m_{i_k}, $En_t(m_{i_k})$, is empty;

(3) for each timed transition $t \in T_t$ the number of occurrences of t in the sequence w is equal to $g(t)$, while for each immediate transition $t \in T_0$, $g(t) = 0$.

It should be observed that there are two basic differences between enhance and selection functions:

(1) enhance functions are (effectively) defined for immediate transitions while selection functions for timed transitions,

(2) enhance functions describe sequences of (complete) firings while selection functions indicate initiations of (timed) firings.

An enhanced free-choice D-timed Petri net $\mathbf{T}$ is a triple $\mathbf{T} = (\mathbf{M}, c, f)$ where:

M is a marked enhanced free-choice Petri net, $\mathbf{M} = (\mathbf{H}, m_0)$, $bfH = (P, T_t, T_0, A, B, C)$,

c is a choice function which assigns a *free-choice* probability to each transition of a net in such a way that for each free-choice place p:

$$\sum_{t \in Out(p)} c(t) = 1,$$

and for all remaining transitions $c(t) = 1$,

f is a firing time function which assigns the time of firing $f(t)$ to each timed transition t of the net, $f : T_t \to \mathbf{R}^+$.

A state s of a D-timed Petri net $\mathbf{T}$ is a triple $s = (m, n, r)$ where:

m is a marking function, $m : P \to \{0, 1, ...\}$,

n is a firing-rank function, $n : T_t \to \{0, 1, ...\}$,

r is a remaining-firing-time function, $r : T_t \to (\mathbf{R}^+)^n$.

An initial state s_i of an enhanced net $\mathbf{T}$ is a triple $s_i = (m_-i, n_i, r_i)$ where n_i is a selection function from the set $Sel(m_k)$, $n_i \in Sel(m_k)$, the marking function $m - i$ is the residual marking after initiating the firings indicated by n_i, the remaining firing time function r_i is equal to the firing times $f(t)$ for all those transitions $t \in T_t$ for which n_i is nonzero, and the (intermediate) marking m_k is determined by an enhance function e_k from the set $Enh(m_0)$, $e_k \in Enh(m_0)$, in the following way:

$$\forall p \in P \;:\; m_k(p) = m_0(p) + \sum_{t \in Inp(p)} e_k(t) - \sum_{t \in Out(p)} e_k(t).$$

An enhanced free-choice net $\mathbf{T}$ may have several different initial states.

A state $s_j = (m_j, n_j, r_j)$ is directly reachable (or (e_k, g_ℓ)-reachable) from the state $s_i = (m_i, n_i, r_i)$ iff:

(1) $e_k \in Enh(m_{iji})$;

(2) $g_\ell \in Sel(m_{ijk})$;

(3) $\forall p \in P \;:\; m_j(p) = m_{ijk}(p) - \sum_{t \in Out(p)} g_\ell(t)$;

(4) $\forall t \in T \;:\; n_j(t) = n_i(t) - d_i(t) + g_\ell(t)$;

(5) $\forall t \in T \;:\; r_j(t)[z] = \begin{cases} r_i(t)[z + d_i(t)] - h_i, \\ \qquad \text{if } 1 \le z \le n_i(t) - d_i(t), \\ f(t), \text{ if } g_\ell(t) > 0 \wedge z = n_j(t); \end{cases}$

where:

(6) $\forall p \in P) \;:\; m_{ijk}(p) = m_{iji}(p) - \sum_{t \in Out(p)} e_k(t) + \sum_{t \in Inp(p)} e_k(t)$;

(7) $\forall p \in P \;:\; m_{iji}(p) = m_{ij}(p) + \sum_{t \in Out(p)} d_i(t)$;

(8) $\forall t \in T \ : \ d_i(t) = \begin{cases} min(n_i(t) - a_i(t), \sum_{p \in Int(t)} m_{ij}(p)), & \text{if } t \in T_t, \\ 0, & \text{otherwise;} \end{cases}$

(9) $\forall p \in P \ : \ m_{ij}(p) = m_i(p) + \sum_{t \in Inp(p)} a_i(t);$

(10) $\forall t \in T \ : \ a_i(t) = \begin{cases} 1, & \text{if } n_i(t) > 0 \wedge r_i(t)[1] = h_i, \\ 0, & \text{otherwise;} \end{cases}$

(11) $h_i = min_{t \in T \wedge n_i(t) > 0}(r_i(t)[1])$.

The state s_j which is (e_k, g_ℓ)-reachable from the state s_i is obtained by the termination of the *next* firings (10,11) and updating the marking of a net (9), performing all interrupts (if any) as determined by the function d_i (7,8), then performing all immediate firings which correspond to an enhance function e_k from the set $Enh(m_{ij})$ (1,6), and finally initiating new firings (if any) indicated by the selection function g_ℓ from the set $Sel(m_{ijk})$ (2,3,4 and 5).

Similarly as for reachable markings, a state s_j is reachable from a state s_i if there is a sequence of directly reachable states from the state s_i to the state s_j. Also, a set $S(\mathbf{T})$ of reachable states is defined as the set of all states of a net $\mathbf{T}$ which are reachable from the initial states (including the initial states). For enhanced free-choice bounded D-timed nets the sets of reachable states are finite.

A state graph $\mathbf{G}$ of a D-timed Petri net $\mathbf{T}$ is a labeled directed graph $\mathbf{G}(\mathbf{T}) = (V, D, b)$ where:

V is a set of vertices which is equal to the set of reachable states of the net $\mathbf{T}$, $V = S(\mathbf{T})$;

D is a set of directed arcs, $D \subseteq V \times V$, such that (s_i, s_j) is in D iff s_j is directly reachable from s_i in $\mathbf{T}$;

b is a labeling function which assigns the probability of transitions from s_i to s_j to each arc (s_i, s_j) in the set D, $b : D \to [0, 1]$, in such a way that if s_j is (e_k, g_ℓ)-reachable from s_i, then

$$b(s_i, s_j) = \prod_{t \in T} c(t)^{e_k(t) + g_]ell(t)}.$$

Example. The enhanced net $\mathbf{T}_2$ shown in Fig.7 is yet another model of the protocol from Fig.2 (the timed transitions are represented by solid bars while immediate transitions by bars). $\mathbf{T}_2$ uses immediate transitions as a *replacement* for all transitions with zero firing times (an additional immediate transition t_7 is used to satisfy the requirement that the whole free-choice classes must be either immediate or timed).

The derivation of the state space for $\mathbf{T}_2$ is shown in Tab.3.

s_i	m_i 1 2 3 4 5 6	n_i 1 2 4 5	h_i	e_k 3 6 7	g_ℓ 1 2 4 5	s_j
1	0 0 0 0 0 0	1 0 0 0	1.0	0 0 0	0 1 0 1	2
2	0 0 0 0 0 0	0 1 0 1	2.0	0 0 1	0 0 1 0	3
				1 0 0	0 0 0 0	4
3	0 0 0 0 0 0	0 0 1 1	2.0	0 1 0	1 0 0 0	1
4	0 0 0 0 0 0	0 0 0 1	3.0	0 0 0	1 0 0 0	1

Tab.3. The set of reachable states for $\mathbf{T}_2$.

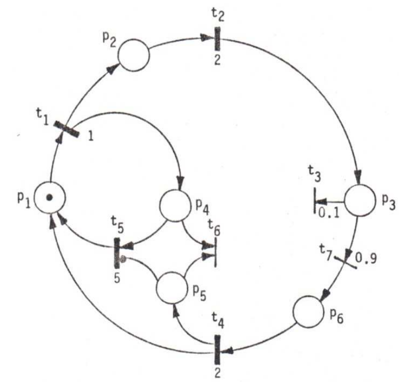

Fig.7. D-timed Petri net $\mathbf{T}_2$.

Tab.3 contains only 4 states. It should be observed that the entries with nonzero values of h_i are identical in Tab.3 and in Tab.2.

The idea of reducing the number of reachable states by changing some of transitions into immediate ones is used in an enhanced free-choice D-timed Petri net $\mathbf{T}_3$ shown in Fig.8. $\mathbf{T}_3$ is a Petri net model of the *sliding window* protocol.

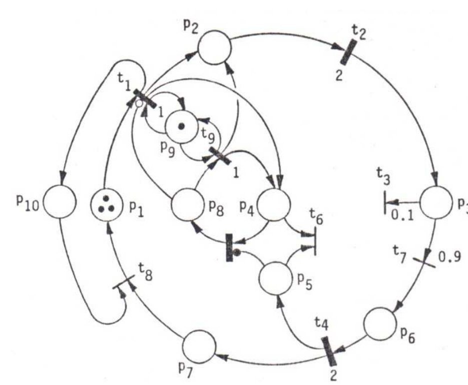

Fig.8. D-timed Petri net $\mathbf{T}_3$.

The simple communication protocol discussed earlier (Fig.2, Fig.7) may be adequate for short distances, but is very inefficient for long distance communication since to send another message, the sender must wait the whole delay of the message as well as acknowledgement. For long distance communication, a more efficient method is to allow the sender to transmit several consecutive messages without waiting for an acknowledgement; since the transmitted messages may be lost or damaged in transit, the sender must store them in a buffer for possible retransmission. The concept of a *window* is used to denote the messages stored in the buffer, and *window size* determines the maximum number of outstanding unacknowledged messages. When the sender eventually receives acknowledgement for a buffered message, it releases the corresponding space in the buffer and transmits another message storing it in the released buffer space (the protocol's window *slides* by one message). For retransmission of lost messages or acknowledgements, a timeout mechanism is used, similarly as before. Since messages may be lost in transit, each message contains a unique sequence number which identifies the message. It is assumed that the messages may be be sent and received in order not necessarily following the sequence numbers.

The sender and receiver are represented in Fig.8 by p_1 and p_3, respectively. The sender's buffer is modelled by p_{10}. The window size is 3 in this example ($m_0(p_1)$). t_1 and p_9 send consecutive messages into communication channel (p_2 and t_2), store them in the buffer (p_{10}), and start the timeout (p_4 and t_5). A correct transfer is acknowledged by firing t_7 and then t_4, which cancels (one) timeout (p_5, t_5 and t_6), and releases one buffer section (t_8). If a message is lost or damaged (t_3), eventually the timeout terminates (t_5 and p_8), and retransmits the message (t_9). In Fig.8 retransmission of timed-out messages has priority over transmission of consecutive messages (inhibitor arc (p_8, t_1) blocks t_1 if p_8 is nonempty).

The net $\mathbf{T}_3$ has 64 states, 26 reduced states, and its throughput is equal to 0.524 messages per time unit, so it is three times greater than the throughput of previous protocols (Fig.2, Fig.7). It should be observed that the protocol from Fig.8 is equivalent to those from Fig.2 and Fig.7 for window size equal to 1 (i.e., $m_0(p_1) = 1$).

6. CONCLUDING REMARKS

It has been shown that the behavior of a class of D-timed Petri nets with interrupt arcs can be represented by discrete-state discrete-time homogeneous semi-Markov processes. Interrupt arcs (which are generalized inhibitor arcs) provide a simple mechanism to discontinue the firing of transitions, and this can be used for strict modelling of timeouts in communication protocols. Moreover, enhanced D-timed nets which combine two classes of nets, ordinary inhibitor nets and D-timed extended nets, reduce the state space of modelling nets by removing all those states which do not contribute to the timed behavior of a model, but which are formally needed for *intermediate* state transitions. Consequently, D-timed protocol models are usually quite simple (in fact, they are simpler than in many other approaches), and their parameters correspond in a very natural way to components or activities of modelled systems (e.g., the numbers of messages, timeout signals, etc.). The results can be obtained in a rather general form, and repeated analyses with different sets of parameters can indicate the effects of performance changes corresponding to different queueing disciplines, or different priorities of messages, etc. The state space can easily be generated from model specifications, and since the states of the modelling net directly correspond to the *states* of the modelled system, a verification step is provided which is not available in simulation or analytical modelling.

The class of timed Petri nets discussed in the paper is restricted in several ways (simple free-choice singular nets), some of the restrictions, however, can be removed by rather straightforward extensions of the presented formalism.

The class of timed Petri nets described in this paper can be used to represent *time* Petri nets proposed by Merlin and Farber [5,13] since the interrupt arcs are flexible enough to model timeout mechanisms, and minimum as well as maximum times assigned to *time* transitions can be represented by free-choice classes of transitions. Consequently, the *extreme* (or worst-case) conditions can be analyzed in the modelled nets.

Acknowledgement

The Natural Sciences and Engineering Research Council of Canada partially supported this research through Operating Grant A8222.

REFERENCES

1. T. Agerwala, Putting Petri nets to work; IEEE Computer Magazine, vol.12, no.12, pp.85-94, 1979.
2. M. Ajmone Marsan, G. Conte, G. Balbo, A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems; ACM Trans. on Computer Systems, vol.2, no.2, pp.93-122, 1984.
3. J-L. Baer, G. Gardarin, C. Girault, G. Roucairol, The two-step commitment protocol - modeling, specification and proof methodology; Proc. 5-th Int. Conf. on Software Engineering, San Diego CA, pp.363-373, 1981.
4. G. Berthelot, R. Terrat, Petri net theory for the correctness of protocols; IEEE Trans. on Communications, vol.30, no.12, pp.2497-2505, 1982.
5. B. Berthomieu, M. Menasche, An enumerative approach for analyzing time Petri nets; Information Processing 83, R.E.A. Mason (ed.), pp.41-45, IFIP 1983.
6. G.V. Bochmann, A general transition model for protocols and communication services; IEEE Trans. on Communications, vol.28, no.4, pp.643-650, 1980.
7. W. Brauer (ed.), **Net theory and applications**; Proc. of the Advanced Course on General Net Theory of Processes and Systems, Hamburg 1979; Lecture Notes in Computer Science 84, Springer Verlag 1980.
8. J.P. Courtiat, J.M. Ayache, B. Algayres, Petri nets are good for protocols; Computer Communication Review, vol.14, no.2, pp.66-74, 1984.
9. A. Danthine, Protocol representation with finite-state models; IEEE Trans. on Communications, vol.28, no.4, pp.632-643, 1980.
10. M. Diaz, Modeling and analysis of communication and cooperation protocols using Petri net based models; Computer Networks, vol.6, no.6, pp.419-441, 1982.
11. K. Garg, An approach to performance specification of communication protocols using timed Petri nets; Proc. 4-th International Conf. on Distributed Computer Systems, San Francisco CA, 1984.

12. M.A. Holliday, M.K. Vernon, A generalized timed Petri net model for performance evaluation; Proc. Int. Workshop on Timed Petri Nets, Torino, Italy, pp.181-190, 1985. 13. W. Juergensen, S.T. Vuong, Formal specification and validation of ISO transport protocol component using Petri nets; Computer Communication Review, vol.14, no.2, pp.75-82, 1984.

14. S.S. Lam, A.U. Shankar, Protocol verification via projections; IEEE Trans. on Software Engineering, vol.10, no.4, pp.325-342, 1984.

15. M. Menasche, B. Berthomieu, Time Petri nets for analyzing and verifying time dependent communication protocols; in: **Protocol Specification, Testing and Verification, III**, pp.161-172, North-Holland 1983.

16. P.M. Merlin, D.J. Farber, Recoverability of communication protocols - implications of a theoretical study; IEEE Trans. on Communications, vol.24, no.9, pp.1036-1049, 1976.

17. M.K. Molloy, Performance analysis using stochastic Petri nets; IEEE Trans. on Computers, vol.31, no.9, pp.913-917, 1982.

18. A. Pattavina, S. Trigila, Combined use of finite-state machines and Petri nets for modelling communicating processes; Electronic Letters, vol.20, no.22, pp.915-916, 1984.

19. J.L. Peterson, **Petri net theory and the modeling of systems**, Prentice-Hall 1981.

20. C. Ramchandani, Analysis of asynchronous concurrent systems by timed Petri nets; Project MAC Technical Report MAC-TR-120, Massachusetts Institute of Technology, Cambridge MA, 1974.

21. R.R. Razouk, The derivation of performance expressions for communication protocols from timed Petri nets; Computer Communication Review, vol.14, no.2, pp.210-217, 1984.

22. R.R. Razouk, C.V. Phelps, Performance analysis using timed Petri nets; Proc. Int. Conf. on Parallel Processing, Columbus OH, pp.126-128, 1984.

23. W. Reisig, **Petri nets - an introduction**; Springer Verlag 1985.

24. J. Sifakis, Use of Petri nets for performance evaluation; in: **Measuring, modelling and evaluating computer systems**, pp.75-93, North-Holland 1977.

25. P. Zafiropulo, C.H. West, D.D. Cowan, D. Brand, Towards analyzing and synthesizing protocols; IEEE Trans. on Communications, vol.28, no.4, pp.651-661, 1980.

26. W.M. Zuberek, Extended D-timed Petri nets, timeouts, and analysis of communication protocols; Proc. ACM Annual Conference, Denver CO, 1985, pp.10-15.

27. W.M. Zuberek, Inhibitor D-timed Petri nets and performance analysis of communication protocols; INFOR Journal, vol.24, no.3, pp.231-249, 1986.

28. W.M. Zuberek, M-timed Petri nets, priorities, preemptions, and performance evaluation of systems; in: **Advances in Petri Nets 1985** (Lecture Notes in Computer Science 222), G. Rozenberg (ed.), pp.478-498, Springer Verlag 1986.