15-th ACM Annual Computer Science Conference (CSC'87); St. Louis, MO, 1987, February 17–19, 1987, pp.261–268.

MODIFIED M-TIMED PETRI NETS IN MODELLING AND PERFORMANCE EVALUATION OF SYSTEMS

W.M. Zuberek

Department of Computer Science, Memorial University
St.John's, NL, Canada A1B 3X5

Abstract

Modified M-timed Petri nets are Petri nets with exponentially distributed firing times and with generalized inhibitor arcs to interrupt firing transitions. It is shown that the behavior of modified free-choice M-timed Petri nets can be represented by probabilistic state graphs, stationary probabilities of states can thus be obtained by standard techniques used for analysis of continuous-time homogeneous Markov chains. An immediate application of such a model is performance analysis of queueing systems with exponentially distributed service and interarrival times, and with priority and/or preemptive scheduling disciplines. Simple models of computer systems with different scheduling strategies are used as an illustration of modelling and performance analysis.

1. INTRODUCTION

Petri nets [1,7,11] have been successfully used in modelling, validation and analysis 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 [6,11]. Basic Petri nets, however, are not complete enough for the study of systems performance since no assumption is made on the duration of systems events. Timed Petri nets have been introduced by Ramchandani [12] by assigning "firing times" to the transitions of Petri nets. Sifakis [14] proposed another definition of a timed Petri net by assigning "enabling times" to places of a net. Merlin and Farber [9] discussed timed Petri nets where a time threshold and maximum delay were assigned to each transition of a net to allow modelling of timeouts used to recover from failures in communication systems. Razouk [13] discussed yet another class of timed Petri nets with enabling as well as firing times, and demonstrated the derivation of (symbolic) performance expressions for communication protocols. Another approach to Petri nets "with time" assumes that the firing times are exponentially distributed random variables, and the corresponding rates are assigned to transitions of a net; such nets are called stochastic Petri nets [2,10]. In generalized stochastic Petri nets [2] the set of transitions is subdivided into two classes of transitions, timed and immediate ones. Timed transitions (as in stochastic nets) have exponentially distributed firing times, while immediate transitions fire in zero time, i.e., they are used to represent logical conditions which do not contribute to "delay" times. The memoryless property of the exponential distribution simplifies the "state" description and analysis of such nets, and, in fact, analysis of (basic and generalized) stochastic nets is based on the sets of reachable markings which are generated without timing constraints. Consequently, the stochastic approach can be used for only such models in which the state space is isomorphic to the space of reachable markings [19], and this may be a nontrivial task to check.

The formalism described in this paper is a continuation of the approach originated by Ramchandani [12,16] and subsequently extended by multiple and inhibitor arcs [17,18]. In (basic and extended) M-timed Petri nets, however, firing transitions cannot be interrupted, and therefore preemptive disciplines are difficult to model and analyze. In this paper, the basic Petri nets are enhanced by "interrupt" arcs in order to suspend the process of transition firing, as required in strict modelling of preemptions. Similarly as in [2,10,17,18,19], firing rates of exponentially distributed firing times are assigned to transitions of a Petri net, and a "state" description is derived which represents the behavior of modified free-choice M-timed Petri nets by continuous-time homogeneous Markov chains. The stationary probabilities of the states can thus be obtained by standard techniques, and this provides many performance measures such as utilization of systems components, average waiting times and turnaround times or average throughput rates, which can be derived automatically from model specifications.

This paper is organized in 3 main sections. Section 2 contains definitions of basic concepts for modified free-choice Petri nets. Modified M-timed Petri nets are introduced in Section 3. Application of modified M-timed Petri nets to modelling and performance evaluation is discussed in Section 4.

2. MODIFIED PETRI NETS

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

P is a finite, nonempty set of places,

T is a finite, nonempty set of transitions,

A is a nonempty 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

$$\forall t \in T \ \exists p \in P \ : \ (p, t) \in A,$$

C is a (possibly empty) set of interrupt arcs which connect places with transitions, $C \subset P \times T$, and A and C 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 interrupting place of a transition t iff $(p, t) \in C$. The set of all interrupting places of t is denoted by $Int(t)$, and the set of transitions connected by interrupt arcs with a place p is denoted by $Int(p)$, $Int(p) = \{t | p \in Int(t)\}$. The notation is extended on sets of places and transitions in the usual way.

A modified net $\mathbf{N} = (P, T, A, C)$ is singular iff the input sets of transitions with nonempty interrupting sets are disjoint with interrupting sets of other transitions

$$\forall t \in T \ : \ Int(t) = \emptyset \ \vee \ Int(Inp(t)) = \emptyset$$

where $\emptyset$ denotes the empty set. The description of non-singular nets must take into account "sequences" of consecutive interrupts in a net when one interrupted transition, through its input places, interrupts another transition(s). In singular nets there is no such "propagation" of interrupts.

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

$\mathbf{N}$ is a modified Petri net, $\mathbf{N} = (P, T, A, C)$,

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

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

A transition t is enabled by a marking m iff every input place of this transition contains at least one token and every interrupting place of t contains zero tokens. For ordinary nets (i.e., nets without time) the interrupt arcs are thus equivalent to inhibitor arcs [1,11,17].

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. A shared place p is guarded iff for each two different transitions t_i and t_j sharing p there is another place p_k such that p_k is in the input set of one of these transitions and in the interrupting set of the other one
$\forall t_i \in Out(p) \ \forall t_j \in Out(p) - \{t_i\} \ \exists p_k \in P - \{p\} \ :$
$((p_k, t_i) \in A \wedge (p_k, t_j) \in C) \ \vee \ ((p_k, t_j) \in A \wedge (p_k, t_i) \in C)$,
i.e., no two transitions from the set $Out(p)$ can be enabled by the same marking m. A net is conflict-free iff all its shared places are guarded.

A shared place p is free-choice (or extended free-choice [7]) iff the input sets and interrupting sets of all transitions sharing p are identical, i.e., iff:

$$\forall t_i, t_j \in Out(p)) \ : \ Inp(t{-}i) = Inp(t_j) \wedge Int(t_i) = Int(t_j).$$

A net is free-choice iff all its shared places are either guarded or free-choice. Only free-choice Petri nets are considered in this paper.

Every transition enabled by a marking m can fire. When a transition fires, a token is removed from each of its input places (but not from interrupting 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

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

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 (including m_0).

3. MODIFIED M-TIMED PETRI NETS

In timed Petri nets, each transition t takes a "real" time to fire, and therefore it is convenient to distinguish three phases of transition firings, its initiation, delay, and termination. When a transition t is enabled, a firing can be initiated by removing a token from each of t's input places. This token remains in the transition t for the "firing time", and then the firing terminates by adding a token to each of t's output places. Each of the firings is initiated in the same instant of time in which it becomes enabled. If a transition becomes enabled while it fires, a new, independent firing can be initiated. If a net contains conflicts, and there are several different possibilities of firing transitions for the same marking, the choice of actual transitions is assumed to be a random process which can be described by corresponding probabilities or probability distribution functions. In modified timed Petri nets, a firing of a transition may be interrupted if the set of transition interrupting places becomes nonempty. If, during a firing period of such transition t, all t's interrupting places become marked, 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. It should be noticed that the interrupting places of t must be empty to initiate a firing of t; an interrupt can thus occur only as a result of termination of another firing (or firings).

Since in timed nets all transition firings are initiated in the same instants of time in which the transitions become enabled, it is convenient to associate with each marking m a set of all possibilities of new firings (in nets with conflicts there are usually several such possibilities). The set of selection functions describes all such possibilities.

A selection function e of a marking m in a net $\mathbf{N}$ is any function $e : T \rightarrow \{0, 1, ...\}$ such that

(1) there exists a sequence of transitions $u = (t_{i_1}, t_{i_2}, ..., t_{i_k})$ in which $t_{i_j} \in En(m_{i_{j-1}})$ 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 transitions enabled by m_{i_k}, $En(m_{i_k})$, is empty,

(3) for each $tinT$, $e(t)$ is equal to the number of occurrences of t in the sequence u;

i.e., a selection function e indicates (by nonzero values) all those transitions which can initiate their firings simultaneously (and some transitions may initiate several firings). The set of all selection functions of a marking m is denoted by $Sel(m)$.

A marked net $\mathbf{M}$ is simple if all selection functions of all reachable markings assign at most one firing to each transition of $\mathbf{M}$, i.e.

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

Only simple nets are discussed in this paper. Non-simple nets can be described by a straightforward extension of simple nets [18,19] that takes into account combinations of multiple firings and their corresponding probabilities. It should be noticed that in M-timed nets without multiple arcs, non-simple behaviour can occur only in the initial states, when the initial marking function introduces "multiple enablings" of some transitions. Moreover, such initial markings can usually be converted to equivalent ones, which preserve the behaviour of a net, and which provide simple selection functions.

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

$\mathbf{M}$ is a modified free-choice marked Petri net, $\mathbf{M} = (\mathbf{N}, m_0)$, $\mathbf{N} = (P, T, A, C)$,

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:

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

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

r is a firing rate function which assigns a positive real number $r(t)$ to each transition t of the net, $r : T- > tarrow\mathbf{R}^+$, and $\mathbf{R}^+$ denotes the set of positive real numbers; the firing time of a transition t is a random variable $v(t)$ with the distribution function

$$\text{Prob}(v(t) > x) = e^{-x*r(t)}, x > 0.$$

The memoryless (or Markov) property of the (negative) exponential distribution is the basic factor in descriptions of M-timed Petri nets (the exponential distribution is the only continuous distribution with the memoryless property). It means that if the duration v of a certain activity (e.g., the firing time) is distributed exponentially with parameter r, and if that activity is observed at time y after its beginning, then the remaining duration of the activity is independent of y and is also distributed exponentially with parameter r:

$$\text{Prob}(v > y + x | v > y) = \text{Prob}(v > x) = e^{-r*x}.$$

Consequently, the state descriptions of M-timed nets represent only the actual configuration of a net, and completely igmores the history, or the sequence of previous states.

The behavior of an M-timed Petri net can be described by a set of states (or configurations of a net), with state transitions corresponding to terminations of firings (and initiations of new firings).

A state $ss@$ of an M-timed Petri net $\mathbf{T}$ is a pair of functions $s = (m, f)$ where:

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

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

An initial state s of a net $\mathbf{T}$ is a pair $s = (m)i, f_i)$ where f_i is a selection function from the set $Sel(m_0)$, and the marking m_i is defined by

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

A free-choice M-timed net $\mathbf{T}$ may have several different initial states.

A state $s_j = (m_j, f_j)$ is directly reachable (or (t_k, e_ℓ)-reachable) from the state $s_i = (m_i, f_i)$ iff:

1. $f_i(t_k) > 0$,
2. $e_\ell \in Sel(m_{ikj})$,
3. $\forall p \in P) \ : \ m_j(p) = m_{ikj}(p) - \sum_{t \in Out(p)} e_\ell(t)$,
4. $\forall t \in T \ : \ f_j(t) = f_{ik}(t) - d_{ik}(t) + e_\ell(t)$,

where

5. $\forall p \in P \ : \ m_{ikj}(p) = m_{ik}(p) + \sum_{t \in Out(p)} d_{ik}(t)$,
6. $\forall t \in T \ : \ d_{ik}(t) = min(f_{ik}(t), min_{p \in Int(t)}(m_{ik}(p))$.
7. $\forall p \in P \ : \ m_{ik}(p) = m_i(p) + \begin{cases} 1, & \text{if } p \in Out(t_k), \\ 0, & \text{otherwise,} \end{cases}$
8. $\forall t \in T \ : \ f_{ik}(t) = f_i(t) - \begin{cases} 1, & \text{if } t = t_k, \\ 0, & \text{otherwise.} \end{cases}$

The state s_j which is (t_k, e_ℓ)-reachable from the state s_i is thus obtained by the termination of a t_k firing (1), updating the marking and firing functions (7,8), performing all the interrupts (if any) of firing transitions which correspond to the updated marking, mik (5 and 6), and then initiating new firings (if any) which are determined by a selection function e_ℓ from the set $Sel(m_{ikj})$ (2,3 and 4). It should be noticed that the formula (6) is a direct consequence of singularity and does not take "propagation" of interrupts into account. For nonsingular nets, the description of state reachability becomes a little more convoluted.

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, the 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 $\mathbf{T}$ (including the initial state(s)).

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

u is a transition-rate function which assigns the rate of transition from s_i to s_j to each arc (s_i, s_j) in the set D, $u : D \rightarrow \mathbf{R}^+$, in such a way that if s_j is (t_k, e_ℓ)-reachable from s_i and $s_i = (m_i, f_i)$, then

$$u(s_i, s_j) = r(t_k) * f_i(t_k) * \prod_{t \in T} c(t)^{e_\ell(t)}.$$

It should be observed that the state graph of a modified free-choice M-timed Petri net is a continuous-time homogeneous Markov chain. The stationary probabilities $x(s)$ of the states $s \in S(\mathbf{T})$ can be obtained by solving a system of simultaneous linear equations [5,8]

$$\begin{cases} \displaystyle\sum_{(s_j, s_i) \in D} u(s_j, s_i) * x(s_j) = x(s_i) * \sum_{(s_i, s_j) \in D} u(s_i, s_j); \\ \qquad\qquad\qquad \text{for } i = 1, ..., K-1 \\ \displaystyle\sum_{1 \le i \le K} x(s_i) = 1 \end{cases}$$

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

Example. The M-timed Petri net shown in Fig.1a (as usual, places are represented by circles, transitions by bars, interrupt arcs by small dots instead of arrowheads, the initial marking by dots inside places, and the firing rate function and the choice function are given as an additional descriptions of transitions) net contains one guarded place, p_1. The net is singular since it contains only one transition (t_3) with nonempty interrupting set. The choice function, c, assigns the probability 1 to all transitions since this net does not contain free-choice places.

The state graph $\mathbf{G}(\mathbf{T}_1)$ is shown in Fig.1b, and the derivation of the set of reachable states $S(\mathbf{T}_1)$ is given in Tab.1 which also contains the stationary probabilities $x(s_i)$ of the states $s_i \in S(\mathbf{T}_1)$. □.

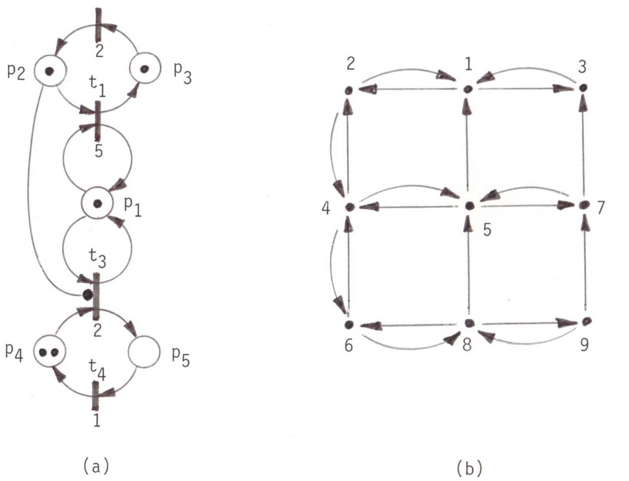

Fig.1. M-timed net $\mathbf{T}_1$ (a) and its state graph (b).

		m_i	f_i		d_{ik}	m_{ikj}	e_ℓ		
s_i	$x(s_i)$	1 2 3 4 5	1 2 3 4	t_k	1 2 3 4	1 2 3 4 5	1 2 3 4	s_j	$u(s_i, s_j)$
1	0.184	0 0 0 2 0	1 1 0 0	1	0 0 0 0	1 0 1 2 0	0 1 1 0 0	2	5.0
				2	0 0 0 0	0 1 0 2 0	0 0 0 0	3	2.0
2	0.184	0 0 0 1 0	0 2 1 0	2	0 0 1 0	1 1 0 2 0	1 0 0 0	1	4.0
				3	0 0 0 0	1 0 0 1 1	0 0 1 1	4	2.0
3	0.084	0 1 0 2 0	1 0 0 0	1	0 0 0 0	1 1 1 2 0	1 1 0 0	1	5.0
4	0.180	0 0 0 0 0	0 2 1 1	2	0 0 1 0	1 1 0 1 0	1 0 0 0	5	4.0
				3	0 0 0 0	1 0 0 0 1	0 0 0 1	6	2.0
				4	0 0 0 0	0 0 0 1 0	0 0 0 0	2	1.0
5	0.136	0 0 0 1 0	1 1 0 1	1	0 0 0 0	1 0 1 1 0	0 1 1 0	4	5.0
				2	0 0 0 0	0 1 0 1 0	0 0 0 0	7	2.0
				4	0 0 0 0	0 0 0 2 0	0 0 0 0	1	1.0
6	0.107	1 0 0 0 0	0 2 0 2	2	0 0 0 0	1 1 0 0 0	1 0 0 0	8	4.0
				4	0 0 0 0	1 0 0 1 0	0 0 1 0	4	2.0
7	0.051	0 1 0 1 0	1 0 0 1	1	0 0 0 0	1 1 1 1 0	1 1 0 0	5	5.0
				4	0 0 0 0	0 1 0 2 0	0 0 0 0	3	1.0
8	0.057	0 0 0 0 0	1 1 0 2	1	0 0 0 0	1 0 1 0 0	0 1 0 0	6	5.0
				2	0 0 0 0	0 1 0 0 0	0 0 0 0	9	2.0
				4	0 0 0 0	0 0 0 1 0	0 0 0 0	5	2.0
9	0.016	0 1 0 0 0	1 0 0 2	1	0 0 0 0	1 1 1 0 0	1 1 0 0	8	5.0
				4	0 0 0 0	0 1 0 1 0	0 0 0 0	7	2.0

Tab.1. The set of reachable states for $\mathbf{T}_1$.

4. PERFORMANCE EVALUATION

A very simple closed-network model of an interactive system with 2 classes of users (and jobs) and with a priority preemptive scheduling is shown in Fig.2a. It contains one central server P_c with two queues of waiting jobs, Q_1 and Q_2 (for class-1 and class-2 jobs, respectively), and n_1 (active) users in class-1 and n_2 (active) users in class-2. All class-1 jobs have higher priority than the class-2 ones, i.e., the class-1 jobs receive the service before the jobs from class-2.

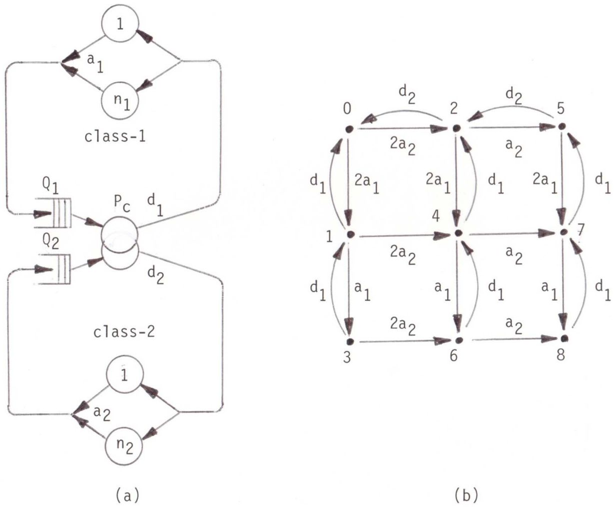

Fig.2. Closed network model os an interactive systems (a) and its transition-rate diagram (b).

Suppose that the jobs of the same class are statistically identical, that they are served by the First-Come-First-Served discipline, and that the service times as well as the user terminal (or "thinking") times are exponentially distributed. Under these assumptions the number of jobs in the system (i.e., in the server and its waiting queues) is a finite continuous-time homogeneous Markov chain [5,8]. For $n_1 = n_2 = 2$, there are 9 states of this Markov chain:

0: no jobs in the system;
1: a class-1 job in P_c, empty queues;
2: a class-2 job in P_c, empty queues;
3: a class-1 job in P_c, 1 job in the queue Q_1;
4: a class-1 job in P_c, 1 job in the queue Q_2;
5: a class-2 job in P_c, 1 job in the queue Q_2;
6: a class-1 job in P_c, 1 job in Q_1, 1 job in Q_2;
7: a class-1 job in P_c, 2 jobs in the queue Q_2;
8: a class-1 job in P_c, 1 job in Q_1, 2 jobs in Q_2.

The corresponding transition-rate diagram [6] is shown in Fig.2b where a_1, a_2, d_1 and d_2 denote the terminal rates for class-1 and class-2, and the service rates for class-1 and class-2 jobs, respectively.

The same system can be modeled by the M-timed Petri net shown in Fig.1a. The transitions t_1 and t_3 correspond to the central server processing class-1 jobs (t_1) and class-2 jobs (t_3) with the service rates (or the firing rates) equal to 5 and 2, respectively. The places p_2 and p_4 model the waiting queues (Q_1 and Q_2, respectively). The transitions t_2 and t_4 correspond to the class-1 and class-2 users (with the terminal rates equal to 2 for class-1 and 1 for class-2). The initial number of tokens in the places p_2 and p_3 represents the number of users in the class-1, n_1, and the total initial number of tokens assigned to places p_4 and p_5 determines the number of users in the class-2, n_2. The initial number of tokens in the place p_1 represents the number of servers (or server channels), in this case 1.

For the initial marking $m_0 = [1, 1, 1, 2, 0]$ (i.e., $m_0(p_1) = 1$, $m_0)(p_2) = 1$, etc.) there are 9 states of the net (Tab.1) and 9 states of the Markov chain (Fig.2b). The correspondence between the Petri net states and the states of the Markov chain is as follows:

Petri net	1	2	3	4	5	6	7	8	9
Markov chain	7	5	8	2	4	0	6	1	3

It can be observed that for $a_1 = 2$, $a_2 = 1$, $d_1 = 5$ and $d_2 = 2$, the Markov chain from Fig.2b is isomorphic to the state graph from Fig.1b.

Many performance measures can be derived from the stationary probabilities $x(s)$ of the states $s \in S(\mathbf{T})$. Since the server is idle in the state s_6 ($m_6(p_1) = 1$, and also Markov state q_0), the stationary probability that the system is idle is equal to the stationary probability $x(s_6) = 0.107$ (Tab.1). Then the utilization of the system is immediately 1-0.107=0.893 or 89.3($x(s_2) + x(s_4)$ since $f_2(t_3) = f_4(t_3) = 1$) and 0.529 for class-1 jobs. The average throughput rates can be obtained from the server's loads. Since the average service time for class-1 jobs is equal to 0.2 time units, and the server utilization for this class is 0.529, then the average throughput rate for class-1 jobs is equal to 0.529/0.2=2.645 jobs per time unit, and the average turnaround time is equal to 2/2.645=0.756 time units. Since the average terminal time and the average service time for class-1 jobs are equal to 0.5 and 0.2, respectively, the class-1 jobs spend, on average, 0.756-0.5-02=0.056 time units in the waiting queue (when the server is processing another class-1 job; the states s_3, s_7 and s_9). Similarly, for class-2 jobs, the average throughput rate, the average turnaround time and the average waiting time are equal to 0.728, 2.747 and 1.247, respectively.

Fig.3 shows slightly modified M-timed Petri net in which the additional place p_6 with the interrupt arc (p_6, t_3) effectively "blocks" interrupts from p_2 since p_6 is always "empty" ($m_0(p_6) = 0$ and $Inp(p_6) = \emptyset$). Consequently, t_3 cannot be preempted (all interrupting places of a firing transition must be nonempty for an interrupt), but the arc (p_2, t_3) still provides priority for class-1 jobs waiting for the server; the scheduling discipline is thus a non-preemptive priority one.

The derivation of the state space for $\mathbf{T}_2$ is shown in Tab.2, and the same performance measures for the net $\mathbf{T}_2$ are as follows:

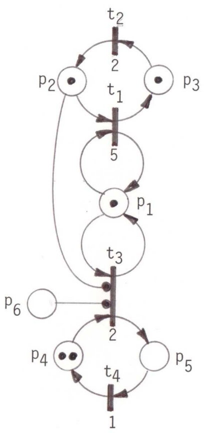

Fig.3. M-timed Petri net $\mathbf{T}_2$.

s_i	$x(s_i)$	m_i 1 2 3 4 5 6	f_i 1 2 3 4	t_k	s_j	$u(s_i, s_j)$
1	0.035	0 0 0 2 0 0	1 1 0 0	1	2	5.0
				2	3	2.0
2	0.048	0 0 0 1 0 0	0 2 1 0	2	4	4.0
				3	5	2.0
3	0.030	0 1 0 2 0 0	1 0 0 0	1	1	5.0
4	0.071	0 1 0 1 0 0	0 1 1 0	2	6	2.0
				3	7	2.0
5	0.116	0 0 0 0 0 0	0 2 1 1	2	8	4.0
				3	9	2.0
6	0.102	0 2 0 1 0 0	0 0 1 0	3	10	2.0
7	0.093	0 0 0 1 0 0	1 1 0 1	1	5	5.0
				2	10	2.0
				4	1	1.0
8	0.092	0 1 0 0 0 0	0 1 1 1	2	11	2.0
				3	12	2.0
				4	4	1.0
9	0.122	1 0 0 0 0 0	0 2 0 2	2	12	4.0
				4	5	2.0
10	0.081	0 1 0 1 0 0	1 0 0 1	1	7	5.0
				4	3	1.0
11	0.062	0 2 0 0 0 0	0 0 1 1	3	13	2.0
				4	6	1.0
12	0.101	0 0 0 0 0 0	1 1 0 2	1	9	5.0
				2	13	2.0
				4	7	2.0
13	0.056	0 1 0 0 0 0	1 0 0 2	1	12	5.0
				4	10	2.0

Tab.2. The set of reachable states for $\mathbf{T}_2$.

the utilization of the system	0.878
the average throughput rate	2.917
the class-1 utilization of the system	0.387
the average class-1 throughput rate	1.935
the average class-1 turnaround time	1.034
the average class-1 waiting time ...	0.334
the class-2 utilization of the system	0.491
the average class-2 throughput rate	0.982
the average class-2 turnaround time	2.037
the average class-2 waiting time ...	0.537

It should be noticed that the pair of interrupt arcs in Fig.3 is equivalent to an inhibitor arc [1,11,17] from p_2 to t_3. In fact, modified nets easily represent inhibitor arcs since interrupts occur only when all interrupting places of a transition are nonempty; an additional "empty" place (like p_6 in Fig.3) is sufficient to "block" the interrupts. This, however, slightly complicates "multilevel" preemption structures. Fig.4 shows an extension of the net from Fig.1, which represents three classes of jobs, the highest priority for class-1 (subnet t_1, p_3, t_2, p_2), the lowest priority for class-3 (subnet t_5, p_7, t_6, p_6). It should be observed that the net $\mathbf{T}_3$ in Fig.4 is non-singular because $Int(t_3) = \{p_2\}$ and $Int(Inp(t_3)) = \{t_5\}$.

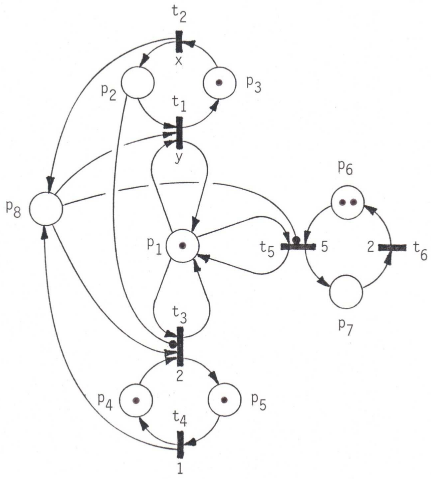

Fig.4. M-timed Petri net $\mathbf{T}_3$.

Moreover, the net from Fig.4 can be used as a model of a system with an "unreliable" (central) processor and two classes of users; the subnet (t_1, p_3, t_2, p_2) represents exponentially distributed processor operative and inoperative periods of time with the rates $r(t_2)$ and $r(t_1)$, respectively (it is important in this case that the initial marking of p_1 is equal to the joint initial marking of p_2 and p_3). And since this subnet has the highest priority, it can preempt all classes of users (such a preemption corresponds to a failure of the central processor during processing of one of user jobs).

Fig.5 shows a Petri net model of yet another interactive system in which there is only one class of users (the transition t_1) but there are two classes of submitted jobs, the "short" and the "long" jobs, with the average service times equal to 0.2 (the transition t_2) and 0.5 (the transition t_3) time units, respectively. The probabilities of short and long jobs are equal to 0.34 ($c(t_2)$) and 0.66 ($c(t_3)$), i.e., they correspond to relative frequencies of class-1 and class-2 jobs in the model from Fig.3 (the relative frequency of class-1 jobs is equal to the class-1 throughput rate divided by the total throughput rate, or $1.935/2.917 = wig 0.66$). Also, the average terminal time corresponds to these relative frequencies, $r(t_1) = 1.5 = wig 1/(0.66 * 0.5 + 0.34 * 1.0)$.

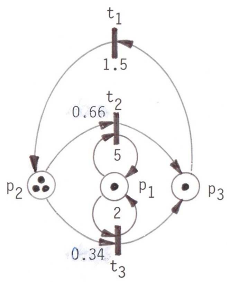

Fig.5. M-timed Petri net $\mathbf{T}_4$.

The derivation of the state space @S(bold T sub 4)@ is shown in Tab.3.

		m_i	f_i		e_ℓ		
s_i	$x(s_i)$	1 2 3	1 2 3	t_k	1 2 3	s_j	$u(s_i, s_j)$
1	0.152	0 2 0	1 0 1	1	0 0 0	3	1.500
				3	1 0 1	4	0.680
					1 1 0	5	1.320
2	0.100	0 2 0	1 1 0	1	0 0 0	6	1.500
				2	1 0 1	4	1.700
					1 1 0	5	3.300
3	0.114	0 3 0	0 0 1	3	1 0 1	1	0.680
					1 1 0	2	1.320
4	0.135	0 1 0	2 0 1	1	0 0 0	1	3.000
				3	1 0 1	7	0.680
					1 1 0	8	1.320
5	0.133	0 1 0	2 1 0	1	0 0 0	2	3.000
				2	1 0 1	7	1.700
					1 1 0	8	3.300
6	0.030	0 3 0	0 1 0	2	1 0 1	1	1.700
					1 1 0	2	3.300
7	0.089	0 0 0	3 0 1	1	0 0 0	4	4.500
				3	1 0 0	9	2.000
8	0.118	0 0 0	3 1 0	1	0 0 0	5	4.500
				2	1 0 0	9	5.000
9	0.128	1 0 0	4 0 0	1	0 0 1	7	2.040
					0 1 0	8	3.960

Tab.3. The set of reachable states for $\mathbf{T}_4$.

Some performance results for the net $\mathbf{T}_4$ are as follows:

the utilization of the system	0.872
the average throughput rate	2.887
the short-job utilization of the system	0.381
the average class-1 throughput rate ..	1.905
the long-job utilization of the system	0.491
the average class-2 throughput rate ..	0.982

Since the performance indices are practically the same as for the previous model, in many cases the modelling nets can be simplified replacing original subnets with simpler subnets equivalent in terms of operational analysis [3].

Some other examples and results are given in [19,20].

CONCLUDING REMARKS

It has been shown that the behavior of modified free-choice M-timed Petri nets can be represented by state graphs which are continuous-time homogeneous Markov chains. For bounded nets the corresponding chains are finite. The state space can easily be derived from net specifications, and many performance measures can be obtained from stationary probabilities of states using standard rules of operational analysis.

The interrupt arcs introduced in this paper are generalized inhibitor arcs. In ordinary (i.e., without time) nets they are equivalent to inhibitor arcs. In timed nets they allow to interrupts firing transitions, i.e., they provide a simple mechanism for modelling preemption of servers and preemptive scheduling disciplines which are not available in other classes of timed Petri nets. It should be noticed that the interrupt arcs defined in this paper are sufficient to represent both inhibitor and interrupt conditions. Fig.6 shows a more general situation where interrupt arcs are used as a replacement of inhibitor arcs, and it can be observed that a relatively simple representation may become quite complicated. Therefore, it may be more convenient to use two different classes of arcs, inhibitor and interrupt arcs, as in [19,20].

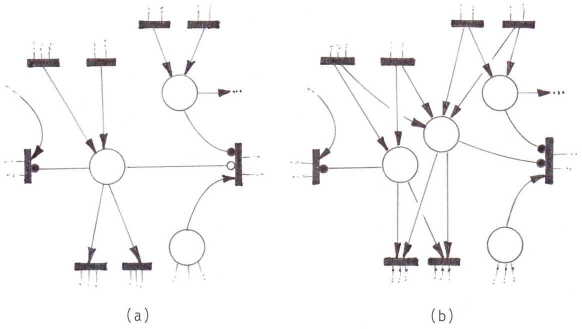

Fig.6. Conversion of inhibitor arcs (a) into interrupt arcs (b).

The class of timed Petri nets discussed in this paper is restricted in several ways (simple free-choice bounded nets), some of the restrictions, however, can be removed by appropriate extensions of the presented formalism. In fact, non-simple nets with more general conflicts can be described in a very similar way provided that the probabilities of conflicts are known and included in the net description. Some other generalizations are discussed in [17] (limited-choice), [18] (multiple arcs) and [19] (enhanced 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.P. Buzen, Fundamental operational laws of computer system performance; Acta Informatica, vol.7, no.2, pp.167-182, 1976.
4. D. Ferrari, **Computer systems performance evaluation**; Prentice-Hall 1978.
5. E. Gelenbe, I. Mitrani, **Analysis and synthesis of computer systems**; Academic Press 1980.
6. C. Girault, W. Reisig (eds.), **Application and theory of Petri nets** (Informatik-Fachberichte 52); Springer Verlag 1982.
7. M. Jantzen, R. Valk, Formal properties of place/transition nets; in: **Net Theory and Applications**, W. Brauer (ed.), (Lecture Notes in Computer Science 84), pp.165-212, Springer Verlag 1982.
8. L. Kleinrock, **Queueing systems**, vol.1: Theory, vol.2: Computer applications; J. Wiley & Sons 1975, 1976.
9. 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.
10. M.K. Molloy, Performance analysis using stochastic Petri nets; IEEE Trans. on Computers, vol.31, no.9, pp.913-917, 1982.
11. J.L. Peterson, **Petri net theory and the modeling of systems**, Prentice-Hall 1981.
12. C. Ramchandani, Analysis of asynchronous concurrent systems by timed Petri nets; Project MAC Technical Report TR-120, Massachusetts Institute of Technology, Cambridge MA, 1974.
13. 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.
14. J. Sifakis, Use of Petri nets for performance evaluation; in: **Measuring, modelling and evaluating computer systems**, pp.75-93, North-Holland 1977.
15. J. Sifakis, Performance evaluation of systems using netsU; in: **Net Theory and Applications**, W.Brauer (ed.), (Lecture Notes in Computer Science 84), pp.307-319, Springer Verlag 1980.
16. W.M. Zuberek, Timed Petri nets and preliminary performance evaluation; Proc. IEEE 7-th Annual Symp. on Computer Architecture, La Baule, France, pp.88-96, 1980.
17. W.M. Zuberek, Performance evaluation using extended timed Petri nets; Proc. Int. Workshop on Timed Petri Nets, Torino, Italy, pp.37-42, 1985.
18. W.M. Zuberek, Generalized M-timed Petri nets and performance evaluation of computer systems; INFOR Journal, vol.23, no.3, pp.344-362, 1985.
19. 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.
20. W.M. Zuberek, On modelling and evaluation of multiprocessor systems using extended M-timed Petri nets; Technical Report #8605, Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada A1C 5S7, 1986.