Removing Timed Delays in Stochastic Automata*

Fida Kamal Dankar, Gregor V. Bochmann
School of Information Technology and Engineering (SITE)
University of Ottawa
{fdankar,bochmann}@site.uottawa.ca

Abstract: We present a method to remove timed delays (timed internal actions) from a subset of stochastic automata. After removing the internal actions, the steady state probability of a subset of the states of the automata is preserved. The stochastic automata considered in this paper have the property of being Markov regenerative processes.

1. Overview

Internal or invisible actions in a process often arise due to communication and synchronization between the different components of the process [7]. Removing these internal actions simplifies the system (to reduce the complication of performance analysis), makes it deterministic, and serves in many useful applications such as submodule construction [6].

We consider in this paper processes whose transitions are triggered by the occurrence of stochastically timed events, these processes may be described using generalized semi-Markov process algebra [1] or stochastic automata [2]. In these systems, removing the timed internal action is a challenging open problem. The approach taken so far to remove internal actions from these systems is to separate actions from their timing information by splitting every transition into two transitions: one describing the time delay and the other is a timeless internal action [1,5], that way removing internal actions from these systems becomes the trivial problem of removing timeless internal actions [7]. However, separating actions from their timing information doubles the number of transitions for the purpose of removing few of these transitions (the ones representing internal actions), so this approach does not lead to a simplification of the process. Hence the need for removing timed internal actions, which is a challenging and open problem so far.

In this paper, we present an algorithm to remove timed internal actions, also known as timed delays, from a subset of stochastic automata. While removing the internal actions, the steady state probability for a subset of the states of the automata is preserved. The

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

subset consists of all those states that do not have an incoming or outgoing internal transition. The performance measures that are preserved after removing the internal moves are the ones obtained from a reward model [4] that assigns zero to all the states that do not keep their steady state probability. In reliability analysis, a fail state is assigned the reward 0 while an up state is assigned the reward 1. So if the fail states and the recovery states are not reached from another state via an internal transition (which is usually the case) then dependability measures are preserved.

The class of stochastic automata considered in this paper have the property that at any time, the set of active clocks with a general distribution have the same elapsed lifetime, in other words they were all enabled at the same time instant, while clocks with an exponential delay distribution may be set at arbitrary times. The idea behind the algorithm will be presented in more details in subsection 2.2, but we will first present some background on Markov regenerative processes and stochastic automata.

2. Introduction

2.1. Markov regenerative processes and Stochastic Automata

We begin our introduction with some background on Markov regenerative process and stochastic automata as these models are heavily used in this paper. To provide a formal definition of a Markov regenerative process (MRGP), the notion of a renewal sequence must be introduced. The following definitions are taken from [8]:

A **Markov renewal sequence** is defined as the sequence of pairs of random variables (X_n, T_n) (usually X_i represents the state of the process that was entered at time T_i) for which the following properties hold:

$$P\{X_{n+1} = j, T_{n+1} - T_n \le t \mid X_n = i, T_n, X_{n-1}, T_{n-1}, ..., X_0, T_0\} =$$
$$P\{X_{n+1} = j, T_{n+1} - T_n \le t \mid X_n = i\} =$$
$$P\{X_1 = j, T_1 \le t \mid X_0 = i\}$$

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

where the first equality highlights the Markov property of the process, and the second shows it homogeneity. According to this definition, the current state of the process alone determines probabilistically the next state and the duration of time in the current state.

If a Markov renewal sequence (X_n, T_n) is associated with a stochastic process $Y(t)$, whose behavior between instants T_n and T_{n+1} is of any kind, but whose value in T_{n+1} is determined by X_n alone, this will be called a **Markov regenerative process**. Formally the following property holds for $Y(t)$:

$$P\{Y(T_n + t) = j \mid Y(u), 0 \le u \le T_n, X_n = i\} = P\{Y(T_n + t) = j \mid X_n = i\}$$
$$= P\{Y(t) = j \mid X_0 = i\}$$

So these processes behave like a Markov process relative to instants T_n, which we refer to as regeneration instants. But between these instants, the process can evolve in any way. From an intuitive point of view, it can be said that there are instants $T_0, T_1, ..., T_n, ...$ between which the behavior of the process is not affected by its previous history. As the process is homogenous, each of the cycles can be studied as if the point of regeneration from which the process is examined were $T_0 = 0$.

Two quantities capable of describing the evolution of the MRGP are defined:

The **local kernel** $E(t)$ and the **global kernel** $K(t)$. Where $E_{ij}(t) = P(Y(t) = j \wedge T_1 > t \mid X_0 = i)$ describes the evolution of the process between two regeneration instants, and $K_{ij}(t) = P(X(t) = j \wedge T_1 \le t \mid X_0 = i)$ describes the evolution of the process at the regeneration instants themselves. For more details refer to [8]

Stochastic automaton (SA) is a state automaton whose transitions are triggered by the occurrence of stochastically timed events. We begin first by enumerating the components of a SA. A SA is a tuple $(S, s_0, C, A, \mapsto, k, F)$ where:

- S is a nonempty set of states with s_0 being the initial state.
- C is a set of clocks,
- A is a set of actions,

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

- $\mapsto \subseteq S \times (A \times \wp_{fin}(C)) \times \wp_{fin}(C) \times S$ is the set of edges (where $\wp_{fin}(C)$ represents the set of finite subsets of C), an element of $\mapsto$ is represented as $s \xrightarrow{a,C}_E s'$ where $(a,C) \in A \times \wp_{fin}(C)$ and $E \in \wp_{fin}(C)$,
- $k : S \to \wp_{fin}(C)$ is the clock setting function which represents all the clocks that are initialized when we reach a state, and
- $F : C \times S \to (\Re \to [0,1])$ (where $\Re$ represents the set of positive real numbers) is the clock distribution function such that $F_s(c)(t) = 0$ for $t < 0$ and $\lim_{t\to\infty} F_s(c)(t) = 1$. For simplicity of notation, we will denote $F_s(c)(t)$ by $c^s(t)$. (Note that the distribution of the clock depends on the state it was initialized in).

As soon as state s is entered, all clocks c in $k(s)$ are initialized according to their probability distribution function $F_s(c)$. Once initialized, clocks start counting down until they either expire or are disabled. A clock expires if it reaches the value 0. The occurrence of an action is controlled by the expiration of clocks. Thus whenever there is a transition $s \xrightarrow{a,C}_E s'$ and the system is in state s, action a can happen as soon as all clocks in set C expire, clocks in E are then disabled and the automaton moves to state s'. The active clocks of a state s are formed from the active clocks of the previous state that have not expired or disabled together with the set of clocks that are initialized in s: $k(s)$. So if $\Delta(s)$ denotes the active clocks of state s, the active clocks in state s' would then be $\Delta(s) - E \cup C + k(s')$. The clocks in the set $\Delta(s) - E \cup C$ are not restarted in s', they rather keep whatever is remaining of their lifetime, while events in $k(s')$ are assigned a new lifetime according to their distributions. In the rest of the paper, when describing a transition, we will not include the disabled clocks as they can be deduced from the active clocks in the states. For more information on SA refer to [2].

2.2. Introduction to the Removal of Timed Delays

As mentioned before we will consider a special case of stochastic automata that we refer to as concurrent generalized stochastic automata or CGSA. CGSA are SA with the following restrictions:

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

1. At most $\mathrm{M}, (\text{for some } \mathrm{M} \geq 0)$ generally distributed clocks are enabled simultaneously, and no other generally distributed clock can be enabled until they are all disabled. So in any state of a CGSA, if two (or more) generally distributed clocks are active, then these clocks have the same elapsed lifetime. In other words they were activated at the same time instant.
2. All clocks have a continuous distribution.
3. The expiration of any single clock induces a transition, and it induces exactly one transition.

(Note that the composition of two CGSAs might not result in a CGSA)

Puliafito et al. [8] proved that with these properties (1-4) the automata is in fact a Markov regenerative process. To be able to see this, we consider all the states in the CGSA where

1. At least one generally distributed clock is initialized or,
2. no generally distributed clock is active (only exponentially distributed clocks are)

Then this set of states clearly forms the embedded renewal sequence. We call the set of these states regenerative states (RS) because once you reach one of them no knowledge of the process history is needed to predict the future.

In [8], Puliafito et al. presented a method to derive the transient and steady state probabilities of such processes from their global and local kernels E_{ij} and K_{ij}.

In this paper, given a CGSA M with internal timed transitions, we would like to eliminate these transitions from the automata while keeping some kind of equivalence. We will proceed by eliminating the internal transitions one by one, after each elimination, the equivalence is preserved. So given an internal transition from a state s as follows: $s \xrightarrow{\tau,c} s'$ (where τ is the label that represents an internal action), we have noticed (see Section 4.1) that if state s and all of its direct successors are regenerative, then we can remove the τ transition and obtain a CGSA that is weakly bissimilar to the original one. Moreover, we designed an algorithm to transform any state in a CGSA into a regenerative one. And this transformation preserves the steady state probability.

So the τ elimination will be done in two steps:

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

1. First we transform some of the states of M into regenerative states, we obtain a CGSA M', with M and M' having the same steady state probabilities. In this step we will benefit from the fact that the transient and steady state probabilities of the CGSA are given (as mentioned before, they are calculated using the kernels E_{ij} and K_{ij}).
2. Then we will remove the τ transitions from M'; the CGSA M'' obtained will be weakly bissimilar to M'.

The resulting equivalence between M and M'' will be denoted by equilibrium-equivalence. The equivalence definitions will be presented in the next Section together with some preliminary definitions. Then the τ elimination will be presented in Section 4.

3. *Definitions*

Definition 1. Successor, invisible successor, and level successor

Let s be a state in a CGSA, we call

- **successor** of s, written $Succ(s)$, is the set of all states in the CGSA that can be reached from s by one transition
- **invisible successor** of s or $ISucc(s)$, all the states in the CGSA that can be reached from s by one transition involving an invisible action.
- **Level successor** of s or $LSucc(s)$, is the set $Succ(s) \bigcup_{r \in ISucc(s)} Succ(r)$.

Definition 2. Local trace

A **local trace** in a CGSA is a trace:

$s_1 \xrightarrow{a_1, c_1} s_2 \ldots s_{n-1} \xrightarrow{a_{n-1}, c_{n-1}} s_n$ where s_1 is regenerative and $s_2, \ldots, s_n$ are not.

Definition 3. Preceding regenerative states of s

Let s be a non regenerative state in a CGSA. We define $R(s)$ as the maximal set of **regenerative states** from which s is reachable through a local trace.

Definition 4. Residual distribution

Let s be a state in a CGSA, let $c \in C$ be a clock such that c is active in s. The **residual distribution** of clock c in state s: $\mathrm{Re}\, s_s^M c(x)$ is the probability that clock c will expire between $[0, x]$ time units after reaching state s (Note that the calculation of this quantity is a complex and difficult task as it depends on the trace of transitions performed until we reach state s). When there is no confusion about the CGSA we will simply write $\mathrm{Re}\, s_s c(x)$. Note that if $c \in k(s)$ or if c is exponentially distributed, then $\mathrm{Re}\, s_s c(x) = c^s(x)$

Definition 5. Structural traces

An **actual trace** in a CGSA is a trace of the following form: $s_0 \xrightarrow{a_1*,t_1} s_1 \xrightarrow{a_2*,t_2} s_2 ...... \xrightarrow{a_n*,t_n} s_n$ where t_i is the time when state s_i was entered, a_i* is the action a_i preceded by any number of τ transitions with $a_i \neq \tau$ for all $i \in \{1,...,n\}$.

Definition 6. Weak bisimulation

Let $M = (S, s_0, C, A, \mapsto, k, F)$ and $M' = (S', s'_0, C', A', \mapsto, k', F')$ be two CGSA.

An equivalence relation $R \subset S \times S'$ is a **weak bisimulation** if whenever $s_1 R s'_1$ then

- $s_1 \xrightarrow{a*,c} s_2$ implies that there exists s'_2 such that $s'_1 \xrightarrow{a*,c'} s'_2$ and $s_2 R s'_2$
- $s'_1 \xrightarrow{a*,c'} s'_2$ implies that there exists s_2 such that $s_1 \xrightarrow{a*,c} s_2$ and $s_2 R s'_2$
- Moreover, if two actual traces $T = s_0 \xrightarrow{a_1*,t_1} s_1 \xrightarrow{a_2*,t_2} s_2 ...... \xrightarrow{a_n*,t_n} s_n$ and $T' = s'_0 \xrightarrow{a_1*,t_1} s'_1 \xrightarrow{a_2*,t_2} s'_2 ...... \xrightarrow{a_n*,t_n} s'_n$ were followed in M and M' before we reached s_1 and s'_1, respectively, such that $s_i R s'_i$ for all $i = 1,...,n$, then the probability $\{\text{in}\, M\}$, that the transition $s_1 \xrightarrow{a*,c} s_2$ will be done within t time units after reaching s_1 is equal to the probability $\{\text{in}\, M'\}$, that transition $s'_1 \xrightarrow{a*,c'} s'_2$ will be done within t time units after reaching s'_1.

Definition 8. Equilibrium Equivalence

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

Let $M = (S, s_0, C, A, \mapsto, k, F)$ and $M' = (S', s'_0, C', A', \mapsto, k', F')$ be two CGSA with $S' \subseteq S$. Let $\Gamma \subseteq S'$, then M and M' are said to be **equilibrium equivalent over** Γ, or $M \approx_\Gamma M'$, if for all $s \in \Gamma$, the steady state probability of s in M and M' is the same.

4. Algorithm for Removing Timed Delays

In this section, we will present the method to remove timed delays.

If $s \xrightarrow{\tau,c} s'$ is a τ transition in a CGSA, then as discussed before, the removal of the τ transition will be done in two steps: first we transform $LSucc(s)$ and s into regenerative states, then we delete the τ transition. In the next subsection we will present the method to transform a non-regenerative state into a regenerative one, then in Subsection 4.2, we will present the algorithm to delete the τ transition with the assumption that $LSucc(s)$ and s are regenerative.

4.1. From non-Regenerative to Regenerative

Let s be a non-regenerative state in a CGSA $M = (S, s_0, C, A, \mapsto, k, F)$, let $R(s) = \{r_1, ..., r_n\}$ and let $\{g_1, ..., g_m, e_1, ..., e_l\}$ be the active clocks of state s, where the g_i have a general distribution and the e_i have an exponential distribution. Our aim in this section is to transform s into a regenerative state. In other words, we need to find the expected distribution of clocks $\{g_1, ..., g_m\}$ in state s, i.e. we need to determine $\mathrm{Res}_s g_j(t)$ for $j \in \{1, ..., m\}$. For that, we assume that a steady state probability π exists for the CGSA and that for all $r \in S, \pi(r) \neq 0$.

Theorem 1.

Let s be a non-regenerative state in a CGSA $M = (S, s_0, C, A, \mapsto, k, F)$, let $R(s) = \{r_1, ..., r_n\}$ and let $\{g_1, ..., g_m, e_1, ..., e_l\}$ be the active clocks of state s, where the g_i have a general distribution and the e_i have an exponential distribution.

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

$$\operatorname{Re} s_s g_j(t) = \frac{\sum_{i=1}^{n} \lambda(r_i) \int_0^{\infty} \frac{dE_{r_i s}(\varepsilon)}{d\varepsilon} \left(\frac{g_j^{r_i}(t+\varepsilon) - g_j^{r_i}(\varepsilon)}{1 - g_j^{r_i}(\varepsilon)} \right) d\varepsilon}{\sum_{i=1}^{n} \lambda(r_i) E_{r_i s}(\infty)}$$

where $E_{r_i s}(t)$ is the local kernel $E_{r_i s}(t) = P(s(t) \wedge T_1 > t \mid r_i(0))$, and $\lambda(r_i)$ is the rate of entering state r_i in equilibrium.

Proof.

We start first with some notation:

- "$s^{\downarrow}(\infty)$" means that we enter state s at equilibrium
- "$r_i \underset{NR}{\longrightarrow} s$" means that we travel from r_i to s and no regenerative state is visited in between.
- "$r_i \overset{\varepsilon}{\underset{NR}{\longrightarrow}} s$" means that s is reached in the interval $[\varepsilon, \varepsilon + d\varepsilon]$ given we entered r_i at time 0, and that no regenerative state is visited in between.
- "$r_i \overset{\varepsilon}{\underset{N\{s_1,\ldots,s_h\}}{\longrightarrow}} s$"means that s is reached in the interval $[\varepsilon, \varepsilon + d\varepsilon]$ given we entered r_i at time 0, and that no state among $\{s_1,\ldots,s_h\}$ is visited on the way.

Being in state s at equilibrium implies that the last regenerative state visited was r_1, or r_2,…., or r_n, and that s was reached ε time units after entering one of the states r_i's, where ε is finite.

$\operatorname{Re} s_s g_j(t) = P(g_j$ has expires within t time units of entering $s \mid s^{\downarrow}(\infty))$

Since the state s was entered at equilibrium after having been in state r_1, or r_2,…., or r_n, we have

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

$\operatorname{Re} s_s g_j(t) = \sum_i P(g_j^{r_i}$ will expire within t time units after entering s and

$r_i \xrightarrow[NR]{} s \mid s^{\downarrow}(\infty))$

But this means that s is reached in the interval $[\varepsilon, \varepsilon + d\varepsilon]$ given we entered one of the r_i's at time $0, \varepsilon$ being finite. So

$\operatorname{Re} s_s g_j(t) = \int_0^\infty \sum_i \{P(g_j^{r_i}$ will expire within t time units after entering s and

$r_i \xrightarrow[NR]{\varepsilon} s \mid s(\infty))\}$

$= \int_0^\infty \sum_i P(g_j^{r_i}$ will expire within t time units after entering $s \mid \ s(\infty)$ and

$r_i \xrightarrow[NR]{\varepsilon} s). \dfrac{P(r_i \xrightarrow[NR]{\varepsilon} s \mid s^{\downarrow}(\infty)}{d\varepsilon}) d\varepsilon$

$$= \frac{\sum_{i=1}^{n} \int_0^\infty \left(\frac{g_j^{r_i}(t+\varepsilon) - g_j^{r_i}(\varepsilon)}{1 - g_j^{r_i}(\varepsilon)}\right) \lambda(r_i) P(s^{\downarrow}(\varepsilon) \wedge T_1 > \varepsilon \mid r_i(0)) d\varepsilon}{\sum_{i=1}^{n} \lambda(r_i) E_{r_i s}(\infty)}$$

$$= \frac{\sum_{i=1}^{n} \lambda(r_i) \int_0^\infty \frac{dE_{r_i s}(\varepsilon)}{d\varepsilon} \left(\frac{g_j^{r_i}(t+\varepsilon) - g_j^{r_i}(\varepsilon)}{1 - g_j^{r_i}(\varepsilon)}\right) d\varepsilon}{\sum_{i=1}^{n} \lambda(r_i) E_{r_i s}(\infty)}$$

(Note that $\lambda(r_i)$ which is the rate of entering state r_i can be calculated using the global kernel and without resorting to calculate the steady state probability of the CGSA: if we consider the SMP underlying the CGSA (who is defined by the global kernel), then $\lambda(r_i) = \dfrac{\pi(r_i)}{d}$ where $\pi(r_i)$ is the steady state probability of state r_i in the SMP, and d is the average time we stay in state r_i once we enter it)

Theorem 2.

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

Let s be a non-regenerative state in a CGSA $M = (S, s_0, C, A, \mapsto, k, F)$, let $\{g_1,..., g_m\}$ be the set of active clocks of s. Let $M' = (S, s_0, C, A, \mapsto, k, F')$ be the CGSA obtained by transforming s into a regenerative state as follows: $k'(s) = \{g_1,..., g_n\}$ and $F'_s(g_i) = \mathrm{Res}_s g_i(t)$. Then M and M' have the same steady state probability π_M and $\pi_{M'}$ respectively.

Proof of theorem 2.

Let $r \in S$ and assume that g is an active clock of r. We will prove first that $\mathrm{Res}_r^{M'}(g)(t) = \mathrm{Res}_r^{M}(g)(t)$. Note that M' has one additional regenerative state s, so we need to study its effect on the residual time of clock g. If $RS_r^M = \{r_1,..., r_n\}$ then either $RS_r^{M'} = \{r_1,..., r_n, s\}$ or $RS_r^{M'} = \{r_1,..., r_n\}$. We will assume without loss of generality that $RS_r^{M'} = \{r_1,..., r_n, s\}$.

We have that:

In both M and M', if we enter state r in equilibrium, then:

- Either we were in one of the states $\{r_1,..., r_n\}$ then we reached state r without passing by states $r_1,..., r_n$ and s.
- Or we are coming from state s then we reached state r without passing by states $r_1,..., r_n$ and s.

And both cases will produce the same residual time for clock g in both M and M'.

Hence $\mathrm{Res}_r^{M'}(g)(t) = \mathrm{Res}_r^{M}(g)(t)$. But this implies that the sojourn time distribution in every state is unchanged, and that the probability of going from one state to another in equilibrium is unchanged. And that implies that the steady state probabilities are preserved.

4.2. Removing delays

Let s be a state in CGSA $M = (S, s_0, C, A, \mapsto, k, F)$ and assume that s and $LSucc(s)$ are all regenerative states. Assume that $s \xrightarrow{\tau, c} s'$. In this subsection, we will present a method to remove this τ transition from the automata and obtain a CGSA $M'(S', s'_0, C', A', \mapsto, k', F')$ that is weakly bissimilar to the original one.

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

First, assume that s' has transitions leading to states: $\{r_1,...,r_m\}$ and s has transitions leading to states $\{s', l_1,...,l_n\}$, and that $k(s)=\{c, c_1,...,c_n\}$ (refer to Figure 1).

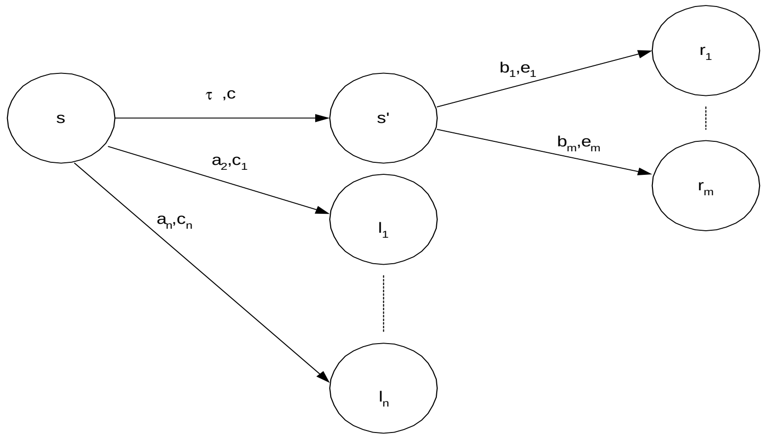

Figure 1. CGSA M

Before proceeding to remove the τ transition, we note that the probability of going from s to r_i within t time units of entering s, $p(M(t)=r_i / M(0)=s)$ or simply $P^M_{s,r_i}(t)$, is equal to

$$P^M_{s,r_i}(t)=\int_0^t\{\int_0^{t_1}\frac{dc^s(\varepsilon)}{d\varepsilon}\frac{de_i^{s'}(t_1-\varepsilon)}{dt_1}\prod_{j=1}^{n}(1-c_j{}^s(\varepsilon))\prod_{\substack{j=1\\ j\neq i}}^{n}(1-e_j{}^{s'}(t_1-\varepsilon))d\varepsilon\}dt_1 \qquad \textbf{(1)}$$

which means that, if clock c expires at time ε and clock e_i expires at time $t_1-\varepsilon$ where $\varepsilon\in[0,t_1]$ and $t_1\in[0,t]$ then all active clocks in state s (i.e. $c_1,...,c_n$) have to expire after time ε, and all active clocks of state s' (i.e. $e_1,...,e_{i-1},e_{i+1},...,e_m$) have to expire after time $t_1-\varepsilon$.

Similarly, the probability of going from s to l_i within t time units of entering s is:

$$P^M_{s,l_i}(t)=\int_0^t\frac{dc_i{}^s(t_1)}{dt_1}(1-c^s(t_1))\prod_{\substack{j=1\\ j\neq i}}^{n}(1-c_j{}^s(t_1))dt_1 \qquad \textbf{(2)}$$

To remove the τ transition, we need to create m new transitions out of state s: $s\xrightarrow{b_i,e'_i}r_i$, $i\in\{1,...,m\}$ (see Figure 2). Moreover, we need to change the distribution of

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

the clocks in the originally existing transitions $s \xrightarrow{b_i, c_i} l_i$ to $s \xrightarrow{b_i, c'_i} l_i$, $i \in \{1,...,n\}$ (refer to Figure 2).

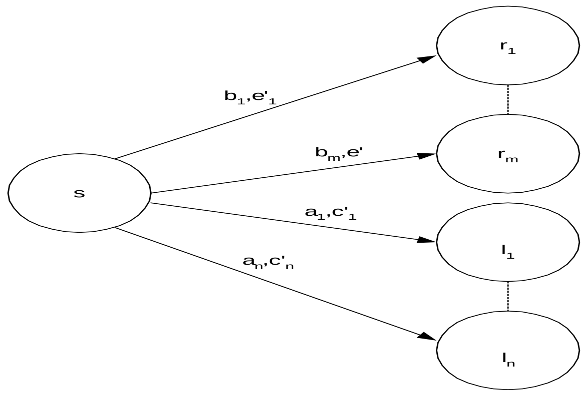

Figure 2. CGSA M'

So the remaining task would be to determine the new distributions for the clocks $e'_i, i \in \{1,...,m\}$ and $c'_i, i \in \{1,...n\}$ in a way that preserves weak bissimulation between the two automata. In other words, we need to preserve the probability of going from state s to state r_i (l_i respectively) within t time units of entering s. This means, we have $P^M_{s,r_i}(t) = P^{M'}_{s,r_i}(t)$ and $P^M_{s,l_i}(t) = P^{M'}_{s,l_i}(t)$ where:

$$P^{M'}_{s,r_i}(t) = \int_0^t \frac{de'^{\,s}_i(t_1)}{dt_1} \prod_{j=1}^{n}(1 - c'^{\,s}_j(t_1)) \prod_{\substack{j=1 \\ j \neq i}}^{n}(1 - e'^{\,s}_j(t_1))dt_1$$ and

$$P^{M'}_{s,l_i}(t) = \int_0^t \frac{dc'^{\,s}_i(t_1)}{dt_1} \prod_{j=1}^{n}(1 - e'^{\,s}_j(t_1)) \prod_{\substack{j=1 \\ j \neq i}}^{n}(1 - c'^{\,s}_j(t_1))dt_1$$

Hence

$$\int_0^t \frac{de'^{\,s}_i(t_1)}{dt_1} \prod_{j=1}^{n}(1 - c'^{\,s}_j(t_1)) \prod_{\substack{j=1 \\ j \neq i}}^{n}(1 - e'^{\,s}_j(t_1))dt_1 = P^M_{s,r_i}(t) \qquad \textbf{(1}_\textbf{i}\textbf{)}$$

$$\int_0^t \frac{dc'^{\,s}_i(t_1)}{dt_1} \prod_{j=1}^{n}(1 - e'^{\,s}_j(t_1)) \prod_{\substack{j=1 \\ j \neq i}}^{n}(1 - c'^{\,s}_j(t_1))dt_1 = P^M_{s,l_i}(t) \qquad \textbf{(2}_\textbf{i}\textbf{)}$$

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

So we have $n+m$ non-linear equations for $n+m$ unknowns ((1_i) for $i \in \{1,...,n\}$ and (2_i) for $i \in \{1,...,m\}$). For ensuring the existence of solutions, the distributions $P^M_{s,r_i}(t)$, $i \in \{1,...,n\}$ and $P^M_{s,l_i}(t)$, $i \in \{1,...m\}$ should be locally integrable.

However, since states $r_i, i \in \{1,...,n\}$ and $l_i, i \in \{1,...,m\}$ are regenerative, no information about clocks $e'_i, i \in \{1,...,m\}$ and $c'_i, i \in \{1,...,n\}$ is needed once we leave state s. In other words, these clocks are only active in s and their only role is to determine the next state once we reach state s. So the distributions of the clocks $e'_i, i \in \{1,...,m\}$ and $c'_i, i \in \{1,...,n\}$ are only used for determining the probabilities of reaching states r_i or l_i, that is $P^{M'}_{s,l_i}(t), i \in \{1,...,m\}$ and $P^{M'}_{s,r_i}(t), i \in \{1,...,n\}$, respectively. However, these probabilities were already determined by equations (1) and (2), therefore we will not resort here to solving the system of equations (1_i), (2_i) in order to find the clock probabilities.

Claim1. Let s be a state in CGSA $M = (S, s_0, C, A, \mapsto, k, F)$ and assume that s and $LSucc(s)$ are all regenerative states. Assume that $s \xrightarrow{\tau,c} s'$ is a transition in M. Let $M' = (S, s_0, C', A, \mapsto', k', F')$ be the CGSA obtained from M by removing the τ move using the method above. Then all states in $S - \{s \cup ISucc(s)\}$ keep their steady state probability after the transformation.

Idea behind the proof. Note that if $r \in S - \{s \cup ISucc(s)\}$, and if Γ is a trace from s_0 to r in M then there exists a trace Γ' from s_0 to r in M' such that Γ and Γ' have the same visible actions (they also end with the same transition whether visible or not) and they both have the same probability distribution.

Theorem 3.

Let $M = (S, s_0, C, A, \mapsto, k, F)$ be a CGSA, let $M' = (S', s_0, C', A, \mapsto', k', F')$ be the CGSA obtained from M by removing the timed internal transitions following the algorithm in the previous subsections. Then M and M' are equilibrium equivalent.

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

The proof can be easily deduced from Claim 1 and Theorem 2.

One way to study performance in automata is using reward models. Reward models are obtained by assigning a reward to every state of the automaton. A state reward is an integer representing the desirability of being in that state. Using these rewards and the steady state probability of the automaton the expected reward rate in steady state is calculated. For more information refer to [9].

In the CGSA M' of Theorem 3, the performance measures that are preserved are the ones obtained from a reward model that assigns zero to all the states that do not keep their steady state probability.

In reliability Analysis, a fail state is assigned the reward 0 while 1 is assigned for the up states. If in the automata M' the fail states and the recovery states can not be directly reached through an internal transition then dependability measures are preserved between M and M'.

5. *Conclusion*

The issue of removing internal transitions from stochastic processes has been an open problem for quite a while. In this paper, we have presented a solution for this problem in the case of concurrent generalized stochastic automata. While removing the internal transitions, the steady state probability of a subset of the states of the automata is preserved. The subset consists of all states that have no incoming or outgoing internal transition. As a future work, we would like to generalize this method to cover a broader subclass of stochastic automata.

The τ-elimination presented in this paper, could also be used as a basis for state aggregation in Markov regenerative processes.

References

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.

1. .M. Bravetti, M. Bernardo, R. Gorrieri, *Towards Performance Evaluation with General Distributions in Process Algebra*, LNCS, vol. 1466, Springer, 2002
2. P. D'Argenio, J-P Katoen, E. Brinksma, *A Compositional Approach to Generalized Semi-Markov Processes*, WODES98, 1998.
3. P.R. D'Argenio, J.-P. Katoen, and E. Brinksma. *An algebraic approach to the specification of stochastic systems (Extended abstract)*. In Proceedings PROCOMET'98. Chapman & Hall, 1998.
4. Hillston, J., *A compositional Approach to Performance Modelling*, Distinguished Dissertation in Computer Science. In: Cambridge University Press, 1996.
5. J-P Katoen, P.R. D'Argenio, *General Distributions in Process Algebra*, LNCS, vol. 2090, Springer, 2001
6. P. Merlin, G.V. Bochmann, *On the Construction of Submodule Specifications and Communication Protocols*, ACM Transactions on Programming Languages and Systems (TOPLAS), 1983
7. R. Milner, *Communication and Concurrency*, Prentice-Hall, 1989.
8. A. Puliafito, M. Scarpa, and K. S. Trivedi, *Petri Nets with k-Simultaneously Enabled Generally Distributed Timed Transitions*, Performance Evaluation, 1998.
9. K. Trivedi , and k. Goseva-Popstojanova, *Stochastic Modelling Formalisms for Dependability, Performance and Performability*, Lecture Notes in Computer Science 1769, 2000.

*An extended abstract of this paper appeared in proceedings of the 7th International Workshop on Performability Modeling of Computer and Communication Systems: PMCCS-7,2005.