Fluid Stochastic Petri Nets Augmented with Flush-Out Arcs: A Transient Analysis Technique

Marco Gribaudo and András Horváth

Abstract—Fluid Stochastic (or Hybrid) Petri Nets with flush-out arcs are Petri net-based models with two classes of places: discrete places that carry a natural number of distinct objects (tokens), and fluid places that hold a positive amount of fluid, represented by a real number. For this kind of formalisms, equations can be automatically derived from the model. Such equations, however, are often too complex to be solved analytically and simple discretization techniques usually can be successfully applied only to simple cases. In this paper, we present a particular solution technique for transient solution that makes use of Kronecker-algebra.

Index Terms—Non-Markovian models, fluid stochastic petri nets, numerical techniques.

1 Introduction

Fluid Stochastic Petri Nets (FSPN) or Hybrid Petri Nets (HPN) are Petri net-based models in which some places may hold a discrete number of tokens, and some places a continuous quantity represented by a nonnegative real number. Places that hold continuous quantities are referred to as *fluid* or continuous places, and the nonnegative real number is said to represent the fluid level in the place. Discrete tokens move along discrete arcs with the enabling and firing rules of standard Petri Nets (PN), while the fluid moves along special continuous (or fluid) arcs according to an assigned instantaneous flow rate. The formalism is augmented with flush-out arcs, which work as follows: If a flush-out arc connects a fluid place to a transition, then, if the transition fires, the level of the fluid place is set to 0. With this feature we can model situations in which fluid like quantities disappear in an immediate manner or in very short time.

Several different versions of FSPNs have been defined in the literature (see, for example [1], [2], [3], [4], [5], [6]) and, for many of them, a method to derive the equations that describe the underlying stochastic process has been provided. In general, the solution of these equations is not a trivial task, and this problem has been directly addressed in many papers. In particular, steady state solution for the case of FSPN in which there is no dependency of the fluid places has been considered in [4]. In that paper, a solution technique which requires *spectral decomposition* of a matrix has been presented. Transient analysis has also been considered in the same paper, proposing a technique called *upwind semidiscretization*. A more complex discretization technique for transient analysis of second order differential equations has been proposed in [6]. This technique uses an *implicit discretization* scheme which requires the solution of a linear system at every time step. Furthermore, it is possible to solve FSPN models using "ad hoc" discretization schemes based on the understanding of the dynamics of the model as in [7]. Other possible directions to solve FSPNs include Laplace transform methods [8] (these are limited to the cases when the distributions associated with the process have rational Laplace transform) and finite elements or finite volumes techniques [9].

Even if the proposed techniques are quite general, none of them is really appropriate to overcome the difficulties of the numerical solution of the equations describing a FSPN. Spectral decomposition, for example, may cause excessive roundoff errors if the matrix is large and, thus, could be successfully applied only to systems characterized by a small number of states. Pure upwind semidiscretization can instead lead to a system with a very high number of unknown variables, limiting the applicability of the algorithm only to systems with a small number of discrete states and one or two fluid places. Implicit techniques instead are not stable for systems which involve only first order differential equations such as the one describing common FSPNs.

In this paper, we present a numerical solution technique for transient analysis. The technique can be applied only to a special subclass of FSPNs (the restrictions that define this subclass are given in Section 4.1) and its purpose is to solve equations arising from models larger (in number of discrete states or in number of fluid places) than the ones that could be analyzed using conventional discretization techniques.

For even larger models, simulation remains the only possible approach. Simulation of FSPNs has been considered in [10], [11].

The rest of the paper is organized as follows: Section 2 introduces the considered FSPN formalism and the notations. Section 3 gives the set of equations that describe the evolution of the stochastic behavior of the FSPN. Section 4 gives a discretization scheme to compute transient probabilities of the underlying stochastic process. Numerical examples and analysis of the method are given in Section 5.

- *M. Gribaudo is with the Department of Informatics, University of Turin, Corso Svizzera 185, 10149 Torino, Italy. E-mail: marcog@di.unito.it.*
- *A. Horváth is with the Department of Telecommunications, Budapest University of Technology and Economics, BME Hiradástechnikai Tanszék, 1521 Budapest, Pf. 91., Hungary. E-mail: horvath@webspn.hit.bme.hu.*

Manuscript received 5 Jan. 2002; revised 15 May 2002; accepted 31 May 2002.
Recommended for acceptance by B. Haverkort.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 116815.

2 Definitions and Notations

The definition of the FSPN with flush-out arc is described in [2] with standard notation inherited from [12]. Because of space limitations, the FSPN formalism and its accompanying equations (Section 3) are explained only in brief. A comprehensive description is given in [2].

A FSPN is a tuple $\langle \mathcal{P}, \mathcal{T}, \mathcal{A}, B, F, W, R, M_0 \rangle$, where

- $\mathcal{P}$ is the set of places partitioned into a set of discrete places $\mathcal{P}_d = \{p_1, \ldots, p_{|\mathcal{P}_d|}\}$ and a set of continuous places $\mathcal{P}_c = \{c_1, \ldots, c_{|\mathcal{P}_c|}\}$ (with $\mathcal{P}_d \cap \mathcal{P}_c = \emptyset$ and $\mathcal{P}_d \cup \mathcal{P}_c = \mathcal{P}$). The discrete places may contain a natural number of tokens, while the marking of a continuous place is a nonnegative real number. In the graphical representation, a discrete place is drawn as a single circle while a continuous place is drawn with two concentric circles. The complete state (marking) of a FSPN is described by a pair of vectors $M = (\boldsymbol{m}, \boldsymbol{x})$, where the vector $\boldsymbol{m}$ of dimension $|\mathcal{P}_d|$ is the marking of the discrete part of the FSPN and the vector $\boldsymbol{x}$ of dimension $|\mathcal{P}_c|$ represents the fluid levels in the continuous places ($x_l \geq 0$ stands for the fluid level of place c_l). We denote the time by τ and we can think the marking M at time τ as the stochastic marking process $\mathcal{M}(\tau) = \{(\boldsymbol{m}(\tau), \boldsymbol{x}(\tau)), \tau \geq 0\}$.
- $\mathcal{T}$ is the set of transitions partitioned into a set of stochastically timed transitions $\mathcal{T}_e$ and a set of immediate transitions $\mathcal{T}_i$ (with $\mathcal{T}_e \cap \mathcal{T}_i = \emptyset$ and $\mathcal{T}_e \cup \mathcal{T}_i = \mathcal{T}$). A timed transition $T_j \in \mathcal{T}_e$ is drawn as a rectangle and has an instantaneous firing rate associated to it. An immediate transition $t_h \in \mathcal{T}_i$ is drawn with a thin bar and has a constant zero firing time. We denote the timed transitions with uppercase letters and the immediate transitions with lowercase letters.
- $\mathcal{A}$ is the set of arcs partitioned into four subsets: $\mathcal{A}_d$, $\mathcal{A}_h$, $\mathcal{A}_c$, and $\mathcal{A}_f$. The subset $\mathcal{A}_d$ contains the discrete arcs which can be seen as a function $A_d : ((\mathcal{P}_d \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P}_d)) \to I\!N$. The arcs $\mathcal{A}_d$ are drawn as single arrows. The subset $\mathcal{A}_h$ contains the inhibitor arcs, $A_h : (\mathcal{P}_d \times \mathcal{T}) \to I\!N$. These arcs are drawn with a small circle at the end. The subset $\mathcal{A}_c$ defines the continuous arcs. These arcs are drawn as double arrows to suggest a pipe. $\mathcal{A}_c$ is a subset of $(\mathcal{P}_c \times \mathcal{T}_e) \cup (\mathcal{T}_e \times \mathcal{P}_c)$, i.e., a continuous arc can connect a fluid place to a timed transition or it can connect a timed transition to a fluid place. The subset $\mathcal{A}_f$ contains the *flush-out* arcs. $\mathcal{A}_f$ is a subset of $(\mathcal{P}_c \times \mathcal{T}_e)$. These arcs (introduced in [7], [13]) connect continuous places to timed transitions and describe the fact that the transition empties in zero time the existing fluid from a continuous place when it fires. The arcs $\mathcal{A}_f$ are drawn as thick single arrows.
- The function $B : \mathcal{P}_c \to I\!R^+ \cup \{\infty\}$ describes the fluid upper bounds on each continuous place. This bound has no effect when it is set to infinity. Each fluid place has an implicit lower bound at level 0.
- The firing rate function F is defined for timed transitions $\mathcal{T}_e$ so that $F : \mathcal{T}_e \times \mathcal{S} \to I\!R^+$. Therefore, a timed transition T_j enabled at time τ in a discrete marking $\boldsymbol{m}(\tau)$ with fluid level $\boldsymbol{x}(\tau)$, may fire with rate $F(T_j, \boldsymbol{m}(\tau), \boldsymbol{x}(\tau))$, that is:

$$\lim_{\Delta\tau \to 0} P\{T_j \text{ fires in } (\tau, \tau + \Delta\tau) | \mathcal{M}(\tau) = (\boldsymbol{m}(\tau), \boldsymbol{x}(\tau))\} / \Delta\tau = F(T_j, \boldsymbol{m}, \boldsymbol{x}).$$

  We also use as a shorthand notation $F(T_j, M)$, where $M = (\boldsymbol{m}, \boldsymbol{x})$).
- The weight function W for immediate transitions $\mathcal{T}_i$ ($W : \mathcal{T}_i \times \mathcal{S}_d \to I\!R^+$) has the usual meaning and it may depend on the discrete part [12].
- The function $R : \mathcal{A}_c \times \mathcal{S} \to I\!R^+ \cup \{0\}$ is called the *flow rate function* and describes the marking dependent flow of fluid across continuous arcs.
- The initial state of the FSPN is denoted by the pair $M_0 = (\boldsymbol{m}_0, \boldsymbol{x}_0)$.

For a more detailed explanation of the previous sets and of the dynamics of a FSPN with flush-outs, see [2].

3 Stochastic Behavior of the FSPN

In this section, we derive the equations for the joint process $\mathcal{M}(\tau) = (\boldsymbol{m}(\tau), \boldsymbol{x}(\tau))$ that describes the dynamic behavior of the FSPN model as a function of time.

3.1 The Infinitesimal Generator

The marking process $\mathcal{M}(\tau)$ is characterized by a matrix $\boldsymbol{Q}$, that we call *infinitesimal generator*. The set $\mathcal{S} = (\mathcal{S}_d \times \mathcal{S}_c)$ of all the states is decomposed in two parts, where $\mathcal{S}_d$ represents the discrete component of the state space and $\mathcal{S}_c$ the continuous component.

In order to derive the complete equations, we start investigating the behavior of the discrete part of the system. Since fluid arcs and flush-out arcs do not change the enabling condition of a transition, standard analysis techniques can be applied to the discrete marking process $\boldsymbol{m}(\tau)$ [12]. These techniques split the discrete state space into two disjoint subsets, called, respectively, the *tangible marking* set and the *vanishing marking* set. Since the process spends no time in vanishing markings, they can be removed and their effect can be included in the transitions between tangible markings. From this point on, we will consider only tangible markings.

In a FSPN with flush-outs, the infinitesimal generator matrix $\boldsymbol{Q}$ depends on the fluid component of the state $\boldsymbol{x}$ and on the power set of the fluid places (because any subset can be flushed out as a result of a state transition): $\boldsymbol{Q}(\boldsymbol{x}, \mathrm{s})$, where $\mathrm{s} \in 2^{\mathcal{P}_c}$. The matrix $\boldsymbol{Q}(\boldsymbol{x}, \emptyset)$ accounts for the transition rates among tangible states when no flush-out occurs and $\boldsymbol{Q}(\boldsymbol{x}, \{c_l\})$ accounts for the transition rates among tangible states when flush-out of place c_l does occur. If a transition flushes out two fluid places c_l and c_k, its effect is included in $\boldsymbol{Q}(\boldsymbol{x}, \{c_l, c_k\})$, etc. In particular, we define

$$\boldsymbol{Q}(\boldsymbol{x}, \mathrm{s}) = [\mathrm{q_{ij}}(\boldsymbol{x}, \mathrm{s})],$$

where $q_{ij}(\boldsymbol{x}, \mathrm{s})$ represents the transition rate from state $\boldsymbol{m}_i$ to state $\boldsymbol{m}_j$ when the level of the fluid is $\boldsymbol{x}$ and the considered transition flushes out the fluid places belonging to the (possibly empty) set s, that is

$$q_{ij}(\boldsymbol{x},\mathrm{s}) = \sum_{\substack{T_k \in \mathcal{K}_i \ \mid \\ \boldsymbol{m}_i \xrightarrow{T_k} \boldsymbol{m}_j \wedge \\ c_l \in \mathrm{s} \Leftrightarrow (c_l, T_k) \in \mathcal{A}_f}} \mathrm{F}(\mathrm{T_k}, \boldsymbol{m}_\mathrm{i}, \boldsymbol{x}), \quad \mathrm{i} \neq \mathrm{j}.$$

The summation considers the transition rate of all the transitions T_k that have concession in marking $\boldsymbol{m}_i$ (denoted by $T_k \in \mathcal{K}_i$) and bring the net from state $\boldsymbol{m}_i$ to $\boldsymbol{m}_j$ (denoted by $\boldsymbol{m}_i \xrightarrow{T_k} \boldsymbol{m}_j$), flushing out exactly all the fluid places specified in the (possibly empty) set s. In the standard equations that describe a CTMC, the terms on the diagonal of the infinitesimal generator account for the probability of exiting from a state. Here, we have to consider not only standard transitions, but also changes of state that cause a flush-out. We denote by

$$q_i(\boldsymbol{x}) = \sum_{\boldsymbol{m}_j \in \mathcal{S}_d} \sum_{\mathrm{s} \in 2^{\mathcal{P}_c}} q_{ij}(\boldsymbol{x},\mathrm{s}) \tag{1}$$

the total exit rate from state $\boldsymbol{m}_i$ when the fluid level is $\boldsymbol{x}$. This function takes into account the sum of the rates from state $\boldsymbol{m}_i$ to any state $\boldsymbol{m}_j$, with any combination of flush-outs. The diagonal element defined in (1) is included in the matrix $\boldsymbol{Q}(\boldsymbol{x},\emptyset)$ and, hence,

$$q_{ii}(\boldsymbol{x},\emptyset) = -q_i(\boldsymbol{x}). \tag{2}$$

The above defined matrix $\boldsymbol{Q}(\boldsymbol{x},\mathrm{s})$ of dimensions $|\mathcal{S}_d| \times |\mathcal{S}_d|$ is a proper infinitesimal generator.

3.2 Equations of the Model

In order to derive the equations that describe the underlying stochastic process, we recall that $\boldsymbol{x} = (x_1, x_2, \ldots, x_{|\mathcal{P}_c|})$ is the vector whose component x_l represents the fluid level in the continuous place c_l. We denote by $2^{\mathcal{P}_c}$ the power set of the fluid place set. Let $\mathrm{s} \in 2^{\mathcal{P}_c}$ be a subset (possibly the empty set) of $\mathcal{P}_c$.

We collect all the possible actual flow rates in a diagonal matrix $\boldsymbol{R}(\{c_l\},\boldsymbol{x})$, with $\{c_l\} \in \mathcal{P}_c$. The element $r_{jj}(\{c_l\},\boldsymbol{x})$ of $\boldsymbol{R}(\{c_l\},\boldsymbol{x})$ represents the fluid flow rate of continuous place c_l in discrete state j conditioned to the fluid level $\boldsymbol{x}$, that is $r_{jj}(\{c_l\},\boldsymbol{x}) = r_l(M = (\boldsymbol{m}_j,\boldsymbol{x}))$. (For the complete definition of $\boldsymbol{R}(\{c_l\},\boldsymbol{x})$, see [2].)

Matrix $\boldsymbol{Q}$ together with matrix $\boldsymbol{R}$ completely describe the stochastic process. Also, since we need to consider the case of multiple flush-outs induced by a single transition, the equation that describes the system should have an integral term for each possible combination of fluid places flushed out together.

Following [2], we define a Dirac's delta extended to a set

$$\boldsymbol{\delta}(\boldsymbol{x},\mathrm{s}) = \begin{cases} 1, & \mathrm{s} = \emptyset, \\ \prod_{c_l \in \mathrm{s}} \delta(x_l), & \mathrm{s} \neq \emptyset, \end{cases}$$

the integral extended to a set

$$\begin{aligned} &\int_0^\infty F(\cdots) d\mathrm{s} = \\ &\begin{cases} F(\cdots), & \mathrm{s} = \emptyset, \\ \int_0^\infty \int_0^\infty \cdots \int_0^\infty F(\cdots) dx'_{i_1} dx'_{i_2} \cdots dx'_{i_{|\mathrm{s}|}}, & \mathrm{s} \neq \emptyset. \end{cases} \end{aligned}$$

This symbol is a shorthand notation used to describe the fact that each flush-out may happen at any level of the fluid places that will be emptied with the transition. This behavior is caught by integrating the solution over each fluid component which represents a continuous place involved in the flush-out.

The projection operator is defined as

$$\boldsymbol{\sigma}(\boldsymbol{x},\mathrm{s}) = (\sigma_1, \sigma_2, \ldots, \sigma_{|\mathcal{P}_c|}) \quad \sigma_l = \begin{cases} x_l, & c_l \notin \mathrm{s} \\ x'_l, & c_l \in \mathrm{s}. \end{cases}$$

The projection operator will be substituted into the latter equations in a symbolic manner. The purpose of this operator is just to select the correct integration variables when using the extended integral notation.

We denote by $\boldsymbol{\pi}(\tau,\boldsymbol{x})$ the probability density vector at time τ with level $\boldsymbol{x}$; its ith entry corresponds to the ith tangible markings.

Using this notation, the equation for the model is

$$\begin{aligned} &\frac{\partial \boldsymbol{\pi}(\tau,\boldsymbol{x})}{\partial \tau} + \sum_{c_l \in \mathcal{P}_c} \frac{\partial(\boldsymbol{R}(c_l,\boldsymbol{x})\boldsymbol{\pi}(\tau,\boldsymbol{x}))}{\partial x_l} = \\ &\sum_{\mathrm{s} \in 2^{\mathcal{P}_c}} \boldsymbol{\delta}(\boldsymbol{x},\mathrm{s}) \int_0^\infty \boldsymbol{\pi}(\tau,\boldsymbol{\sigma}(\boldsymbol{x},\mathrm{s}))\boldsymbol{Q}(\boldsymbol{\sigma}(\boldsymbol{x},\mathrm{s}),\mathrm{s}) d\mathrm{s}. \end{aligned} \tag{3}$$

The first term of (3) represents the time and the second accounts for the fluid flow in all the fluid places. Each continuous place has a term in the summation which represents its instantaneous fluid change in each discrete state. The term in the right-hand side of (3) accounts for the probability change due to discrete state change. Each term of the summation corresponds to transitions that flush-out a particular subset of fluid places. The derivation of (3) is based on the supplementary variable approach [8].

4 Transient Analysis

In this section, we present an explicit discretization scheme of the equations described in Section 3.2 in order to carry out the transient analysis of FSPNs. The matrices that describe the evolution of the probabilities between two successive steps of the transient analysis are expressed as Kronecker-expressions of suitable matrices and vectors.

The advantage of the presented procedure is that it leads to a relatively simple and memory-efficient implementation of the discretization scheme and provides satisfactorily precise results in a wide range of cases. However, to apply the method some assumptions have to be made.

4.1 Assumptions

There are three main assumptions on the FSPN we work with:

1. The fluid rates do not depend on the continuous part of the marking. In other words, the flow rate of a fluid place can depend neither on the fluid level of the place itself nor on the fluid level of other places.
2. The dependencies of the firing rates on the continuous part of the marking may be expressed in a product form manner. The contribution of fluid level x_k in marking $\boldsymbol{m_i}$ to the firing rate of transition T_j is

denoted by $\lambda_{ij}^{(k)}(x_k)$. In other words, in the presence of n fluid places with levels $x_k, 1 \leq k \leq n$, the rate of transition T_j in marking $\boldsymbol{m_i}$ may be expressed as a product as

$$F(T_j, \boldsymbol{m_i}, \boldsymbol{x}) = \prod_{k=1}^{n} \lambda_{ij}^{(k)}(x_k).$$

If in $\boldsymbol{m_i}$ the firing rate of T_j does not depend on fluid level x_k one could simply set $\lambda_{ij}^{(k)}(x_k) = const,\ \forall x_k$.

3. In order to have bounded state space, we assume that the number of discrete markings $|\mathcal{S}_d|$ is finite and all the fluid places are bounded.

Even if these assumptions are fairly serious, we believe that there are still a wide range of applications satisfying these restrictions. In Section 4.4, some ideas will be given about how to relax the first two assumptions.

We require that it is possible to choose the discretization steps of the level of the fluid places and the discretization step of the transient time in such a way that, during a sojourn in any discrete marking, the change of fluid level will be an integer multiple of the discretization step. As a result of our first assumption, this condition does not hold only if the modeler defines a "strange" fluid rate structure. This condition is necessary to avoid step-by-step error accumulation. Without this assumption, the procedure could result in high inaccuracy.

We assume as well that a state jump from discrete marking $\boldsymbol{m_i}$ to discrete marking $\boldsymbol{m_j}$ always flushes out fluid place $c_k,\ 1 \leq k \leq |\mathcal{P}_c|$ or never flushes it out. This restriction could be easily relaxed; however, we apply this assumption because it makes the description of the procedure simpler.

Before starting to describe the discretization scheme, some notation has to be introduced. This is done in the next section.

4.2 Notation

The upper bounds of the fluid places are denoted by $B_i,\ 1 \leq i \leq |\mathcal{P}_c|$. The discretization step of fluid place c_i is δ_i; the number of points of discretization along fluid level x_i is $\lceil B_i/\delta_i \rceil$. The discretization step of the transient time is denoted by δ_{time}. Since we assume that fluid rates do not depend on the fluid levels, the fluid rate of place c_j in marking $\boldsymbol{m_i}$ may be denoted simply by $r_i^{(j)}$.

The set of fluid places that may be nonempty in discrete marking $\boldsymbol{m_i}$ is denoted by $\mathcal{N}_i$. To determine precisely the sets $\mathcal{N}_i,\ 1 \leq i \leq |\mathcal{S}_d|$ is, in general, not a trivial task and is beyond the scope of this paper. Some trivial observations, however, can be made. For example, a fluid place is always empty in a discrete marking $\boldsymbol{m_i}$ if it is always flushed out when a jump to $\boldsymbol{m_i}$ occurs and $r_i \leq 0$. The size of $\mathcal{N}_i$ is significant from the point of view of the discretization scheme since, as we will see later, the less elements $\mathcal{N}_i$ has, the smaller the discretized state space describing $\boldsymbol{m_i}$ is. Nevertheless, the proposed procedure can be applied without determining the sets $\mathcal{N}_i,\ 1 \leq i \leq |\mathcal{S}_d|$ precisely, but the memory consumption can be much higher than necessary.

In order to describe the discretization procedure, we need to differentiate the transitions that are enabled to move discrete tokens from those that are enabled to change only the continuous marking of the net. To this end, by $\mathcal{K}_i$, we denote the set of transitions that have concession in marking $\boldsymbol{m_i}$ (i.e., those that are enabled in $\boldsymbol{m_i}$ to move discrete tokens of the net).

4.3 Discretization Scheme

In order to describe the discretization scheme, we introduce the following vectors and matrices.

4.3.1 Vector of Firing Rates

Given a transition T_j that has concession in marking $\boldsymbol{m_i}$, the vector $\boldsymbol{l}_{ij}^{(k)},\ 1 \leq k \leq |\mathcal{P}_c|$ describes how the level of a fluid place c_k contributes to the firing rate of T_j in $\boldsymbol{m_i}$ at different points of the discretized fluid range

$$\boldsymbol{l}_{ij}^{(k)} = \begin{bmatrix} \lambda_{ij}^{(k)}(0) \\ \lambda_{ij}^{(k)}(\delta_k) \\ \lambda_{ij}^{(k)}(2\delta_k) \\ \vdots \\ \lambda_{ij}^{(k)}(h_i^{(k)}\delta_k) \end{bmatrix},$$

$$\text{where } h_i^{(k)} = \begin{cases} 0, & \text{if } c_k \notin \mathcal{N}_i, \\ \lceil B_k/\delta_k \rceil, & \text{if } c_k \in \mathcal{N}_i. \end{cases}$$

As one could observe the length of $\boldsymbol{l}_{ij}^{(k)}$ is $\lceil B_k/\delta_k \rceil + 1$ if c_k can be nonempty in $\boldsymbol{m_i}$, otherwise it contains a single value.

4.3.2 Matrix to Describe Fluid Rates

Next, we define the matrix $\boldsymbol{G}_\ell^{(k)}$ that describes how a fluid level x_k changes in a discrete marking $\boldsymbol{m_\ell}$ during a period of δ_{time}. If c_k may be nonempty in $\boldsymbol{m_\ell}$, the matrix is of size $(\lceil B_k/\delta_k \rceil + 1) \times (\lceil B_k/\delta_k \rceil + 1)$ and its entries are given by

$$\left[\boldsymbol{G}_\ell^{(k)}\right]_{ij} =$$

$$\begin{cases} 1, & \text{if } j - i = \frac{r_\ell^{(k)}\delta_{time}}{\delta_k}, \\ 1, & \text{if } j = 0, r_l^{(k)} < 0, i \leq \frac{|r_\ell^{(k)}|\delta_{time}}{\delta_k}, \\ 1, & \text{if } j = \lceil B_k/\delta_k \rceil,\ r_\ell^{(k)} > 0,\ i \geq \lceil B_k/\delta_k \rceil - \frac{r_\ell^{(k)}\delta_{time}}{\delta_k}, \\ 0, & \text{otherwise}, \end{cases}$$

where i (j) corresponds to the fluid level before (after) the period of length δ_{time}. If fluid place c_k is always empty in $\boldsymbol{m_\ell}$, then $\boldsymbol{G}_\ell^{(k)} = [1]$. As mentioned in Section 4.1, we assume that $r_\ell^{(k)}\delta_{time}/\delta_k$ is an integer. If this condition does not hold, the procedure can be still applied using the closest integer to $r_\ell^{(k)}\delta_{time}/\delta_k$. To give an example, let us assume that $\delta_{time} = 10\delta_1$ and $r_1^{(1)} = 0.2$, in this case during a sojourn in marking $\boldsymbol{m_1}$ the fluid level increases by $2\delta_1$ in an interval of length δ_{time}; this is described by

$$\boldsymbol{G}_1^{(1)} = \begin{bmatrix} 0 & 0 & 1 & 0 & \cdots & & \\ 0 & 0 & 0 & 1 & 0 & \cdots & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & & & & \\ & & & \cdots & 0 & 0 & 1 \\ & & & & \cdots & 0 & 1 \\ & & & & & \cdots & 1 \end{bmatrix}.$$

4.3.3 Matrix to Describe Fluid Dynamics at State Jumps

The matrix denoted by $\boldsymbol{J}_{kl}^{(m)}$ describes the evolution of fluid level x_m during an interval of length δ_{time} in which a state jump occurs from discrete marking $\boldsymbol{m_k}$ to discrete marking $\boldsymbol{m_\ell}$. The entries of matrix $\boldsymbol{J}_{kl}^{(m)}$ have the meaning

$$\begin{aligned} &\left[\boldsymbol{J}_{kl}^{(m)}\right]_{ij} \cong \\ &P\{x_m(t+\delta_{time}) \in [(j-1/2)\delta_m,(j+1/2)\delta_m] \mid \\ &\boldsymbol{m}(t)=\boldsymbol{m_k}, \boldsymbol{m}(t+\delta_{time})=\boldsymbol{m_\ell}, x_m(t)=i\delta_m, \\ &\text{one jump occurs in } [t,t+\delta_{time}]\}. \end{aligned} \tag{4}$$

In general, when the firing rates depend on one or more fluid levels, the probability defined in (4) also depends on the fluid levels. We will assume, however, that the firing rates do not change during the interval of length δ_{time} and, as a result of this assumption, the single state jump occurs uniformly distributed in the interval. Hence, the probability given in (4) is not computed precisely, it is only approximated. For this reason, $\cong$ is used instead of $=$.

We note as well that $\boldsymbol{J}_{kl}^{(m)}$ is not necessarily a square matrix. If $c_m \in \mathcal{N}_k$ but $\notin \mathcal{N}_\ell$ (or vice versa), then $\boldsymbol{J}_{kl}^{(m)}$ is a column vector of length $(\lceil B_m/\delta_m \rceil + 1)$ (or a row vector of length $(\lceil B_m/\delta_m \rceil + 1)$).

The entries of $\boldsymbol{J}_{kl}^{(m)}$ are approximated the following way. Let us assume that the state jump occurs at $a \leq \delta_{time}$ time instant, the jump does not cause the fluid place c_m to be flushed out, and $x_m(0) = i\delta_m$. At time instant δ_{time} the fluid level $x_m(\delta_{time})$ is

$$x_m(\delta_{time}) = \begin{cases} \max(\max(0, i\delta_m + ar_k^{(m)}) + (\delta_{time} - a)r_\ell^{(m)}, 0), \\ \qquad \text{if } r_k^{(m)} \leq 0, r_\ell^{(m)} \leq 0, \\ \min(\max(0, i\delta_m + ar_k^{(m)}) + (\delta_{time} - a)r_\ell^{(m)}, B_m), \\ \qquad \text{if } r_k^{(m)} \leq 0, r_\ell^{(m)} > 0, \\ \max(\min(B_m, i\delta_m + ar_k^{(m)}) + (\delta_{time} - a)r_\ell^{(m)}, 0), \\ \qquad \text{if } r_k^{(m)} > 0, r_\ell^{(m)} \leq 0, \\ \min(\min(B_m, i\delta_m + ar_k^{(m)}) + (\delta_{time} - a)r_\ell^{(m)}, B_m), \\ \qquad \text{if } r_k^{(m)} > 0, r_\ell^{(m)} > 0. \end{cases} \tag{5}$$

If the state jump from $\boldsymbol{m_k}$ to $\boldsymbol{m_\ell}$ flushes out c_m, the above equations simplify to

$$x_m(\delta_{time}) = \begin{cases} \max((\delta_{time} - a)r_\ell^{(m)}, 0), & \text{if } r_\ell^{(m)} \leq 0, \\ \min((\delta_{time} - a)r_\ell^{(m)}, B_m), & \text{if } r_\ell^{(m)} > 0. \end{cases} \tag{6}$$

Based on (5) and (4), one can determine the interval $[a_1, a_2]$ for which

$$\text{if } a \in [a_1, a_2], \text{ then } x_m(\delta_{time}) \in [(j-1/2)\delta_m, (j+1/2)\delta_m].$$

Having determined the interval $[a_1, a_2]$, the (i, j) entry of $\boldsymbol{J}_{kl}^{(m)}$ is simply approximated as

$$\left[\boldsymbol{J}_{kl}^{(m)}\right]_{ij} = \frac{a_2 - a_1}{\delta_{time}}.$$

The expression is exact in case of those state jumps that are caused by a transition whose firing rate does not depend on the fluid levels. The approximation is satisfactory for those transitions whose firing rate does not vary in a large scale by small changes in the fluid levels. If the firing rate changes a lot (the extremity is the transition that fires in a deterministic manner when a given fluid level reaches a value), the discretization step of the transient time (δ_{time}) has to be small in order to avoid inaccuracy.

The matrices $\boldsymbol{J}_{kl}^{(m)}$, $1 \leq k, l \leq |\mathcal{S}_d|$, $1 \leq m \leq |\mathcal{P}_c|$ have an important role in the procedure from the point of view of precision. Having a state jump in an interval of length δ_{time} we will not assume that the jump occurred at the beginning (or at the end) of the interval. Instead, the probabilities will be distributed among several points of the discretized state space according to when the state jump could occur. As a result, the dynamics of the fluid will be followed fairly precisely in the presence of state jumps as well.

4.3.4 Vector of Transient Probabilities

During the calculations the transient probabilities of the system are stored in the vectors $\boldsymbol{p}_i$, $1 \leq i \leq |\mathcal{S}_d|$; $\boldsymbol{p}_i$ contains the discretized common probability density function of the fluid levels in discrete marking $\boldsymbol{m_i}$. Accordingly, the length of $\boldsymbol{p}_i$ is given by

$$\prod_{j, c_j \in \mathcal{N}_i} \left(\left\lceil \frac{B_j}{\delta_j} \right\rceil + 1 \right).$$

In the vectors $\boldsymbol{p}_i$, $1 \leq i \leq |\mathcal{S}_d|$ every position corresponds to a combination of fluid levels. In the following, we describe how to find a given fluid level combination in the vectors. We use a so-called mixed-based numbering scheme which is closely related to the Kronecker-product operator. Let us use the notation

$$n_i^{(k,l)} = \prod_{j=k}^{l} s_i^{(j)},$$

$$\text{where } s_i^{(j)} = \begin{cases} 1, & \text{if } c_j \notin \mathcal{N}_i \\ \lceil B_j/\delta_j \rceil + 1, & \text{if } c_j \in \mathcal{N}_i. \end{cases}$$

Then, in the discretized state space, the probability that the process is in discrete marking $\boldsymbol{m_i}$ and the vector describing the fluid levels

$$\boldsymbol{x} = [l_1\delta_1, l_2\delta_2, \ldots, l_{|\mathcal{P}_c|}\delta_{|\mathcal{P}_c|}],$$

$$\text{where } 0 \leq l_k \leq \begin{cases} 0, & \text{if } c_k \notin \mathcal{N}_i, \\ \lceil B_k/\delta_k \rceil, & \text{if } c_k \in \mathcal{N}_i, \end{cases}$$

is given by the mth entry of the vector $\boldsymbol{p}_i$ with

$$\begin{aligned} m =& (\ldots(l_1 s_i^{(2)} + l_2)s_i^{(3)} \ldots)s_i^{(|\mathcal{P}_c|)} + l_{|\mathcal{P}_c|} = \\ & \sum_{k=1}^{|\mathcal{P}_c|} l_k n_i^{(k+1,|\mathcal{P}_c|)}, \end{aligned}$$

where $n_i^{(|\mathcal{P}_c|+1,|\mathcal{P}_c|)} = 1$.

Having defined the above vectors and matrices, hereinafter we describe how the transient probabilities are computed from one step to another. Let us denote the emergent probability vectors by $\boldsymbol{p}'_i,\ 1 \leq i \leq |\mathcal{S}_d|$.

In every step, the evolution of the probabilities during an interval of length δ_{time} is determined. We assume that 0 or 1 state jump occurs during the interval. If δ_{time} is low enough, this assumption does not lead to significant inaccuracy. Being in $\boldsymbol{m_i}$ with continuous marking $\boldsymbol{x} = [l_1\delta_1, l_2\delta_2, \dots, l_{|\mathcal{P}_c|}\delta_{|\mathcal{P}_c|}]$, the probability that a transition $T_j \in \mathcal{K}_i$ fires in the next δ_{time} interval is approximated by

$$\delta_{time} \prod_{k=1}^{|\mathcal{P}_c|} \lambda_{ij}^{(k)}(l_k\delta_k). \tag{7}$$

The same comments may be repeated on the goodness of the approximation given in (7) as the ones that were given on the goodness of the approximation given for the matrices $\boldsymbol{J}_{kl}^{(m)}$. The approximation in (7) is precise for those firing rates that do not vary much as a result of small changes in the fluid levels.

In every step, at first, we follow the case when no state jump occurs during the observed interval

$$\begin{aligned} \boldsymbol{p}'_i = \boldsymbol{p}_i \left(\boldsymbol{I} - \delta_{time} \sum_{j,\, T_j \,\in\, \mathcal{K}_i} \mathrm{diag}\left(\bigotimes_{k, c_k \,\in\, \mathcal{N}_i} \boldsymbol{l}_{ij}^{(k)} \right) \right) \\ \times \left(\bigotimes_{k, c_k \,\in\, \mathcal{N}_i} \boldsymbol{G}_i^{(k)} \right), \ \text{for } 1 \leq i \leq |\mathcal{S}_d|, \end{aligned} \tag{8}$$

where $\mathrm{diag}(\boldsymbol{v})$ represents the square matrix with vector $\boldsymbol{v}$ in its diagonal. According to (8), the entries of $\boldsymbol{p}_i$ are reduced by the probability that the process jumps out of $\boldsymbol{m_i}$ (described by the entries of the vectors $\boldsymbol{l}_{ij}^{(k)}$) and then are shifted in accordance with the fluid rates (described by the matrices $\boldsymbol{G}_i^{(k)}$). This shift corresponds to the discretization of the second term of the left-hand side of (3).

The state jumps and their effect on the fluid levels are considered by

$$\begin{aligned} \boldsymbol{p}'_i \mathrel{+}= \delta_{time} \sum_{j=1}^{|\mathcal{S}_d|} \boldsymbol{p}_j \left(\sum_{k,\, T_k \,\in\, \mathcal{K}_j} b_{ji}^k \ \mathrm{diag}\left(\bigotimes_{\ell, c_\ell \,\in\, \mathcal{N}_i} \boldsymbol{l}_{jk}^{(\ell)} \right) \right) \\ \times \left(\bigotimes_{\ell, c_\ell \,\in\, \mathcal{N}_i} \boldsymbol{J}_{ji}^{(\ell)} \right), \ \text{for } 1 \leq i \leq |\mathcal{S}_d|, \end{aligned} \tag{9}$$

where b_{ji}^k denotes the probability that the firing of T_k in $\boldsymbol{m_i}$ leads to $\boldsymbol{m_j}$. The quantities b_{ji}^k, which are used to take into account the effect of immediate transitions, can be determined directly from the net. In (9), we use $\mathrel{+}=$ because the right-hand side has to be added to the right-hand side of (8).

In (9), as in (8), the probability of a state jump is calculated based on the values of the vectors $\boldsymbol{l}_{jk}^{(\ell)}$, and the fluid level in the next state is distributed according to the matrices $\boldsymbol{J}_{ji}^{(k)}$.

The different kinds of performance indexes described in [2] can be calculated during the transient analysis by substituting the integrals over the state space by suitable summations over the discretized state space.

4.4 Extensions

In this section, we give some ideas on how to relax the assumptions listed in Section 4.1.

The first assumption may be refined the following way. Instead of assuming that the fluid rates do not depend on the continuous marking, we assume only that a fluid rate depends on other fluid rates in a piecewise constant manner, i.e., the functions $\lambda_{ij}^{(k)}(x)$ are piecewise constant. In this case, the state space of a discrete marking may be partitioned into subsets in which the fluid rates are constant and a very similar procedure can be applied.

The second assumption may be relaxed as well. If the firing rates cannot be expressed as the product of functions, the matrices that are described in (8) and (9) by Kronecker-operators as

$$\mathrm{diag}\left(\bigotimes_{\ell, c_\ell \,\in\, \mathcal{N}_i} \boldsymbol{l}_{jk}^{(\ell)} \right)$$

have to computed by other (possibly more complicated) Kronecker expressions or have to be constructed elementwise "by hand."

4.5 Complexity

One step of the transient analysis can be viewed as a vector-matrix multiplication where the vector contains the transient probabilities of the underlying discretized state space. Thus, the applicability of the presented discretization scheme is determined by the size of the whole discretized state space. It is given by

$$\sum_{i=1}^{|\mathcal{S}_d|} \prod_{j, c_j \,\in\, \mathcal{N}_i} \left(\left\lceil \frac{B_j}{\delta_j} + 1 \right\rceil \right). \tag{10}$$

The following implications are shown by (10):

- If the model has a large number of discrete states, the proposed discretization can be applied only if there is only one fluid place in the model and there are only few tangible markings in which the fluid places can be nonempty.
- If the fluid places are nonempty in most of the tangible markings, the number of tangible markings has to be low.

In other words, the largeness of a model depends neither on the number of tangible markings nor on the number of fluid places. It depends on the size of the underlying discretized state space.

The advantage of the presented algorithm is that, as a result of the modular representation, even if the state space is huge, the matrices describing the evolution of the process can be stored in a memory-efficient way. This is due to two reasons:

- Some descriptors of the discretization scheme, i.e., the vectors $\boldsymbol{l}$ and the matrices $\boldsymbol{G}$, $\boldsymbol{J}$, can be identical for different indices. For example, the firing rate of a transition may be the same in different markings, or a fluid level may evolve in the same way in several markings. In this case, as natural, the descriptor is stored only once which allows us to exploit the symmetries of the net. If there is a lot of symmetry in the model noticeable amount of memory can be saved this way.

- The assumptions we made allow us to handle the fluid levels in a separate manner. The evolution of the fluid levels is described by the Kronecker-expression of relatively small matrices with as many terms as many fluid places may be nonempty in the given marking. Instead, if the evolution of fluid levels was described in a nonmodular manner, then we should store a matrix with as many entries as the product of the number of entries of the terms of the Kronecker-expression.

The complexity of performing the Kronecker-operations given in (8) and (9) depends on the chosen algorithm. Several different possibilities with their complexity are described and compared in [14]. Using the algorithms given in [14], one can exploit the sparsity of the matrix and the fact that the same entries of the matrices are used several times during the computation of the Kronecker-products.

4.6 Error Estimation

Precise error estimation of the presented discretization scheme is not straightforward and is out of the scope of this paper. This is because the dependency of the firing rates on the fluid levels gives rise to differential equations that are hard to analyze. As future work, it is planned to identify more restricted cases for which thorough error estimation is possible.

Even if giving precise error estimation is a difficult task, it is easy to see which steps of the transient analysis introduce inaccuracies:

- Firing intensities are calculated in (7). This expression assumes that the firing rate of the transition is constant during the δ_{time} interval and gives a first order approximation of the probability of firing. This can be inaccurate if the firing intensity varies a lot as a result of a small change in the fluid levels or if the chosen δ_{time} is not small enough. The proposed scheme does not consider multiple discrete state jumps in a δ_{time} interval. If δ_{time} is not small enough, this feature can result in inaccuracies as well.
- Another source of error is in the matrices $J_{kl}^{(m)}$ which describe what happens to the fluid levels in a δ_{time} interval when state jump occurs. As for the firing rates, the error caused by this approximation is big if the firing intensity varies a lot as a result of a small change in the fluid levels or if the discretization steps on the fluid levels are not small enough.

Further analysis of the proposed discretization technique will be given in connection with the numerical examples in the next section.

5 Application of the Discretization Scheme

In this section, first, we illustrate the applicability of the proposed discretization scheme by performing transient analysis of a FSPN with one fluid place that models a system with garbage collection. Then, using the introduced example, we analyze the memory and time consumption of the scheme and compare it to other methods like semi-discretization and simulation. In Section 5.3, an example with two fluid places is given.

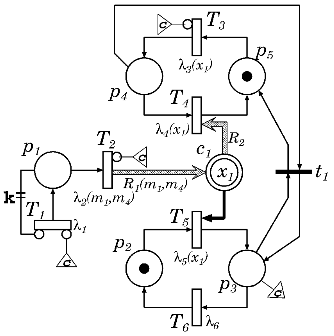

Fig. 1. A system with garbage collection.

5.1 FSPN Model of a System with Garbage Collection

Consider a multiuser system with garbage collection, such as a Java servlet running under a Java virtual machine. Fig. 1, models such a system. The model has five discrete places and a single fluid place. Place p_1 models the load of the network: The number of tokens in place p_1 corresponds to the number of users in the system. We assume that the maximum number of users present in the system is bounded by a constant k. Transition T_1 models the entrance of a new user in the system. The inhibitor arc with multiplicity k ensures the bound on the number of users concurrently working. We assume a constant arrival rate λ_1. Place p_4 and p_5 model the garbage collector: When the token is in p_5, the garbage collector is inactive; if place p_4 is marked instead, the system is undergoing a garbage collection. Fluid place c_1 models the amount of allocated memory. Transition T_2 corresponds to the service being offered to the users. It fires when a user finishes its service and leaves the system. The transition rate of T_2, $\lambda_2(m_1, m_4)$ depends both on the load of the system (marking of place p_1) and on whether the garbage collector is active or not. We expect transition T_2 to grow almost linearly with the number of customers in the system (marking of place p_1) and to fall down dramatically during the garbage collection. We assume that each process requires a quantity of memory that grows linearly with its service time and we model this with a fluid arc connecting T_2 to c_1. We expect the flow rate of this arc $R_1(m_1, m_4)$ to be proportional to the firing rate of T_2 and, since, as required by the proposed algorithm, the fluid rate depends only on the discrete part of the marking. Transition T_3 models the start of the garbage collection activity and depends on the actual memory allocated x_1. We expect a higher probability of starting the garbage collection action when the allocated memory reaches an upper threshold $\mathrm{M}_{\mathrm{high}}$. Garbage collection stops with the firing of transition T_4. Also, in this case, we imagine it being dependent on the quantity of allocated memory x_1. In this case, we assume a high probability of ending the garbage collection when the allocated memory falls below a given threshold $\mathrm{M}_{\mathrm{low}}$. During garbage collection, memory is freed at a constant speed R_2 through the fluid arc connecting fluid place c_1 to transition T_4. If the allocated memory reaches $\mathrm{M}_{\mathrm{crash}} = 1$, i.e., when there is no more memory to allocate, the system crashes. Place p_2 models the correct behavior of the system, while place p_3 denotes a crash state. Transition T_5 models the crash event and is a Dirac delta depending on the total allocated memory x_1 and placed on $\mathrm{M}_{\mathrm{crash}}$.[1] Transition T_6 models the system recovery and it is exponentially distributed with constant rate λ_6. Inhibitor arcs connecting place p_3 to transitions T_1, T_2, and T_3 prevents the system from work during the recovery phase. Immediate transition t_1, models the event of a crash occurring during a garbage collection phase.

The firing intensity of transition T_5 depends in a deterministic manner on fluid level x_1. This fact is reflected during the discretization by setting

$$\left[\boldsymbol{l}_{i5}^{(1)}\right]_j = \begin{cases} 0, & \text{if } j < \lceil \mathrm{M}_{\mathrm{crash}}/\delta_1 \rceil, \\ \frac{1}{\delta_{time}}, & \text{if } j = \lceil \mathrm{M}_{\mathrm{crash}}/\delta_1 \rceil, \end{cases}$$

for $\forall i$ such that $T_5 \in \mathcal{K}_i$. While, for the sake of conserving probability, all other transitions are "switched off" if the fluid level reached $\mathrm{M}_{\mathrm{crash}}$, i.e.,

$$\left[\boldsymbol{l}_{ij}^{(1)}\right]_{\lceil \mathrm{M}_{\mathrm{crash}}/\delta_1 \rceil} = 0,$$

for $\forall i, j$ such that $T_5 \in \mathcal{K}_i, T_j \in \mathcal{K}_i, j \neq 5$.

The example was analyzed using the following set of parameters:

$$k = 5 \text{ or } 7, \quad \mathrm{M}_{\mathrm{crash}} = 1, \quad \mathrm{M}_{\mathrm{high}} = 0.8 \text{ or } 0.95, \quad \mathrm{M}_{\mathrm{low}} = 0.3,$$

$$\lambda_1 = 2, \quad \lambda_2(m_1, m_4) = \frac{k}{m_1\,(m_4 + 1)},$$

$$\lambda_3(x_1) = \begin{cases} 0, & x_1 < \mathrm{M}_{\mathrm{low}} \\ 1, & \mathrm{M}_{\mathrm{low}} \leq \mathrm{x}_1 < \mathrm{M}_{\mathrm{high}} \\ 10, & \mathrm{M}_{\mathrm{high}} \leq \mathrm{x}_1 \end{cases},$$

$$\lambda_4(x_1) = \begin{cases} 1, & x_1 > \mathrm{M}_{\mathrm{low}} \\ 10, & x_1 \leq \mathrm{M}_{\mathrm{low}} \end{cases},$$

$$\lambda_5(x_1) = \delta(1 - x_1), \quad \lambda_6 = 0.25,$$

$$R_1(m_1, m_4) = 0.05 \frac{m_1}{m_4 + 1}, \quad R_2 = 0.2\,. \tag{11}$$

As mentioned in Section 4.1, such discretization steps (δ_1 and δ_{time}) have to be found with which, in any of the markings, the change in the fluid level during a sojourn of length δ_{time} is the integer multiple of δ_1. In order to assist the choice of appropriate δ_1 and δ_{time}, the rate of the fluid flow in different markings is given in Table 1. The values appearing in the table are computed based on (11).

The results presented in this section were calculated with discretization steps

$$\delta_1 = 0.000125, \quad \text{and} \quad \delta_{time} = 0.005.$$

TABLE 1
Rate of Change of Allocated Memory in Different States of the System

Number of users	Rate		
	inactive collector	active collector	crash
0	0	-0.2	0
1	0.05	-0.175	0
2	0.1	-0.15	0
3	0.15	-0.125	0
⋮	⋮	⋮	⋮

These discretization steps fulfill the assumptions (Section 4.1). In Section 5.2, we will investigate the effect of the choice of the discretization steps on the memory requirements and time consumptions.

Following the classification of performance measures given in [2], hereinafter some examples of different types of performance measures are given. As an example of *discrete state measures*, we compute the probability of being in working and crash state versus time (depicted for different cases in Figs. 2 and 3). The probability that the system is working and not undergoing garbage collection at time τ is given by

$$P\{\text{working at time } \tau\} = \sum_{\boldsymbol{m}_i : m_4 = 0, m_3 = 0} \int_0^{\mathrm{M}_{\mathrm{crash}}} \pi_i(\tau, x_1) dx_1.$$

A *discrete throughput measure* is the number of users served by the system divided by the elapsed time. This measure is depicted in Fig. 4 for the different cases and may be obtained at time τ by calculating

$$\frac{\int_0^{\tau} \sum_{\substack{\boldsymbol{m}_i \in \mathcal{S}_d \text{ such that} \\ T_2 \text{is enabled}}} \int_0^{\mathrm{M}_{\mathrm{crash}}} \pi_i(\tau, x_1) dx_1\ \lambda_2(m_1, m_4)\ d\tau}{\tau}.$$

As a combination of *discrete* and *continuous state measures*, it is possible to compute the probability of a situation that is defined by conditions on both the discrete and the fluid part

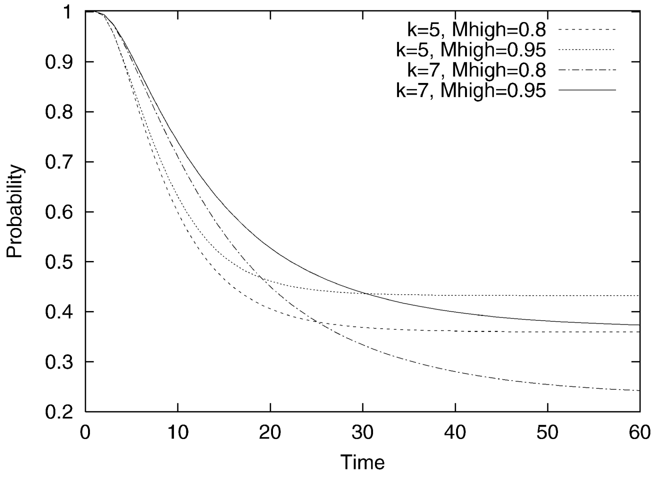

Fig. 2. Probability of working (garbage collector is inactive) versus time.

[1] Here, the use of Dirac delta function does not corresponds to its mathematical definition. However, because fluid level dependent immediate transitions are not included in the formalism, we use it to reflect the infinity of the firing rate.

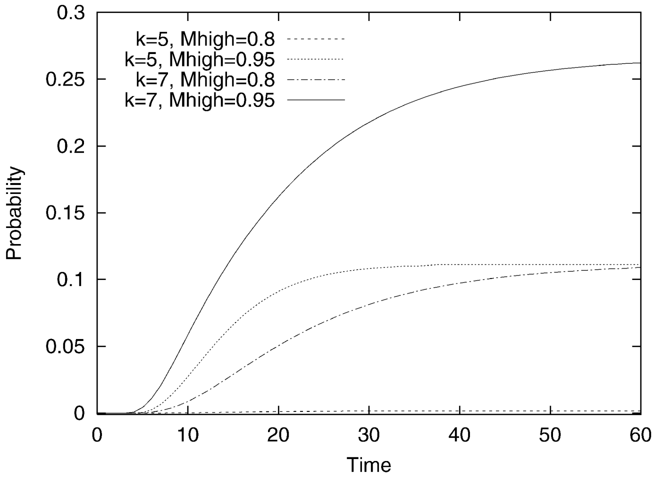

Fig. 3. Probability of crash versus time.

of the net. Fig. 5 shows the probability of the occurrence of a "dangerous situation" when the number of users in the system is higher than a given value and the amount of allocated memory is higher than $\mathrm{M_{high}}$. The probability of this situation at time τ may be expressed as:

$$P\{m_1 \geq N \text{ and } x_1 \geq \mathrm{M_{high}} \text{ at time } \tau\} = \sum_{\boldsymbol{m}_i : m_1 \geq N} \int_{\mathrm{M_{high}}}^{\mathrm{M_{crash}}} \pi_i(\tau, x_1) dx_1.$$

As an example of a pure *fluid state measure* the probability density function (*pdf*) of the distribution of the fluid level at different time instants for $k = 5, \mathrm{M_{high}} = 0.8$ is depicted on Fig. 6. The *pdf* of the fluid level at time τ is given by

$$\sum_{\boldsymbol{m}_i \in \mathcal{S}_d} \pi_i(\tau, x_1).$$

The precision of the above presented transient results obtained by the method described in Section 4 were validated by simulation.

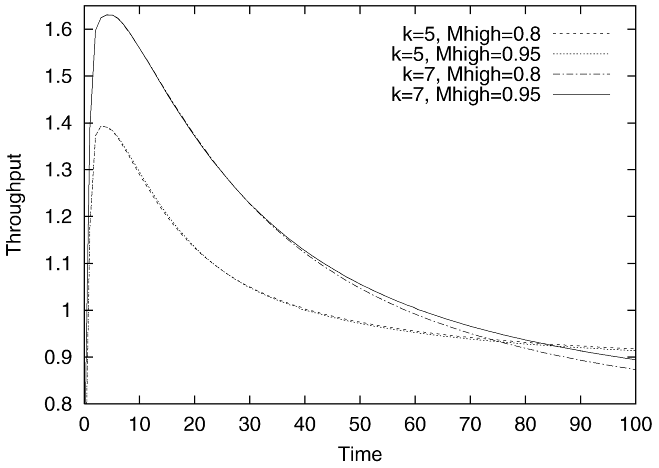

Fig 4. Average number of users served by the system divided by the elapsed time versus time.

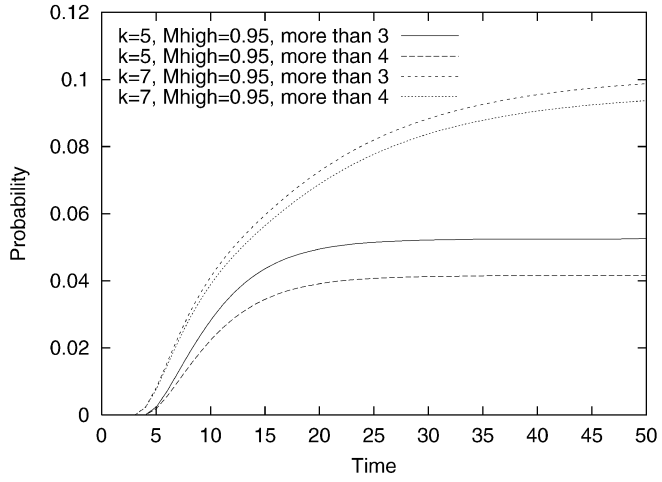

Fig. 5. Probability of having more than three or four users and fluid level higher than $\mathrm{M_{high}}$ versus time.

5.2 Analysis of the Discretization Scheme

Throughout this section, we concentrate on the case $k = 5$ and $\mathrm{M_{high}} = 0.8$.

In this example, it is easy to verify that the fluid place is empty when the system is in one of the markings representing crash (token in place p_3). In all other markings, the fluid level has the upper bound $\mathrm{M_{crash}}$ (reaching this level the system gets into crash). The number of markings corresponding to the working state (inactive garbage collector) is $k+1$. Similarly, we have $k+1$ markings corresponding to garbage collecting and $k+1$ markings representing crash situations. Accordingly, the size of the discretized state space is

$$2(k+1) \times \left(\frac{1}{\delta_1} + 1\right) + k + 1,$$

which gives 120,018 for $k=5$.

In the case $\delta_1 = 0.000125, \ \delta_{time} = 0.005$, the proposed modular description of the applied explicit discretization scheme requires the storage of matrices with 10^6 entries altogether. The explicit storage of the scheme would require

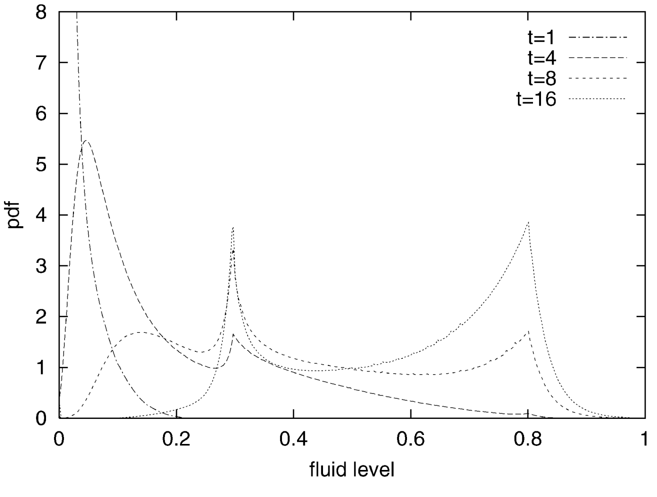

Fig. 6. Density function of fluid level distribution at different time instants.

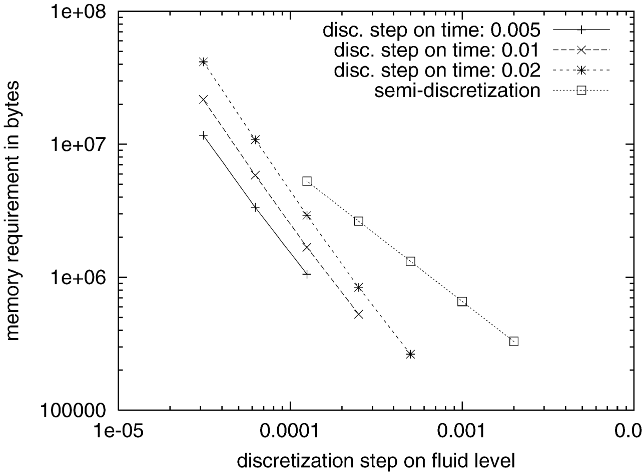

Fig. 7. Memory requirement.

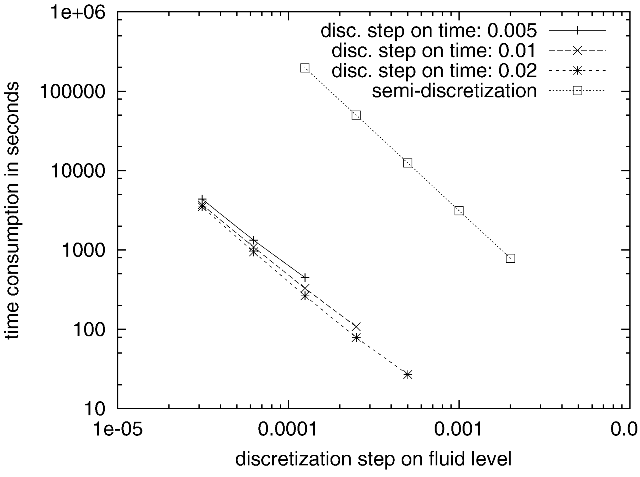

Fig. 8. Seconds needed to compute transient results until time t = 16.

storing a single matrix with approximately 2.5×10^6 entries. Let us refer here in short to Section 4.5: The above gain is achieved by exploiting the symmetries of the net. The advantages of separate handling of fluid places obviously do not appear in this example. According to our experience, for the model presented in Section 5.1, independently of the chosen discretization steps, the storage of the scheme using the modular description requires about factor 2.5 less memory than storing the scheme explicitly. In general, the amount of memory saved by using the Kronecker-based discretization scheme depends very much on the model under analysis.

Fig. 7 depicts the memory required to store the discretization scheme for different values of the discretization steps. The figure shows also the amount of memory required to store the matrix that is needed to analyze the system by semidiscretization (see, for example, [15]). We used the Forward Euler scheme to solve the system of linear equations obtained from the semidiscretization. In order to have stable calculations and precise results, the time step of the Forward Euler scheme has to be kept small considering the largest fluid rate of the model; to this aim, we set $\delta_{time} = \delta_1/rate_{max}/10$ (for example, $\delta_{time} = 0.0008$ when $\delta_1 = 0.002$). It can be observed in Fig. 7 that, for a given value of δ_1, the amount of necessary memory increases as δ_{time} increases. The reason is that, as δ_{time} increases, the number of nonzero entries of the matrices $\boldsymbol{J}$ increases as well. It can also be seen that the memory requirement of the proposed method is lower than that of the semidiscretization in the range $\delta_1 \in [10^{-4}, 10^{-3}]$. Below this range, as it can be predicted by extrapolation, the amount of memory needed by semidiscretization is lower. The reason is found again in the number of nonzero entries of the matrices $\boldsymbol{J}$: the matrices $\boldsymbol{J}$ fill up faster than the matrix of the semidiscretization.

Fig. 8 depicts the time needed to compute the transient results until 16 units of time both using the proposed method and semidiscretization. The experiences were performed on a computer running at 800 MHz with 512 Mbytes of main memory. Even if the memory requirement for smaller values of δ_{time} is smaller and the computation of one step ahead in transient time takes less time, the time needed to compute the transient results is higher because for smaller values of δ_{time} the scheme has to be applied more times. As it can be seen, since the time step of semidiscretization has to be kept smaller than δ_{time} of the proposed technique, semidiscretization requires significantly more time.

Fig. 9 depicts the pdf of the fluid level at transient time 1 with three customers present initially in the system. In this case, if neither a new customer arrives nor a customer leaves the system until time 1, the fluid level equals 0.15, so that there is a discontinuity in the pdf. As suggested by the figure, semidiscretization does not capture this discontinuity, it distributes the probability around 0.15. Instead, the proposed scheme reflects the discontinuity. Furthermore, semidiscretization assigns some probability to having fluid level higher than 0.25. This is in contrast with the property that after one time unit the highest possible value of the fluid level is 0.25. The proposed technique conserves this property of the model. The two approaches were used with the same discretization step of the fluid level $\delta_1 = 1.25 \times 10^{-4}$. The discretization step of the transient time was $\delta_{time} = 5 \times 10^{-5}$ in case of semidiscretization, while it was 5×10^{-3} in case of the proposed discretization technique.

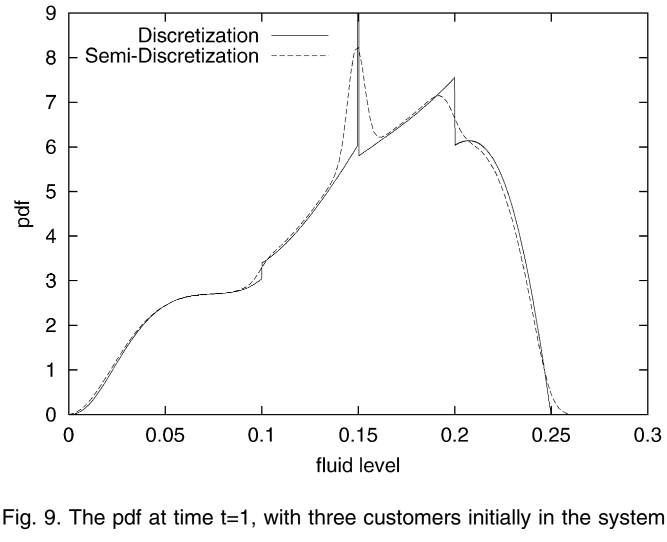

Fig. 9. The pdf at time t=1, with three customers initially in the system.

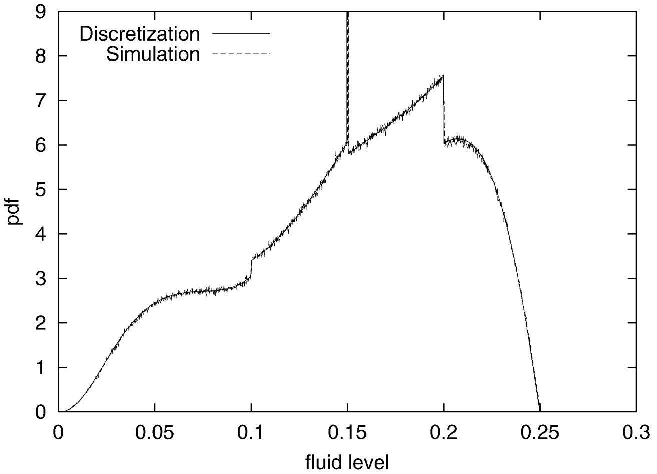

Fig. 10. The pdf at time t = 1, with three customers initially in the system compared with simulation.

Fig. 10 gives the above described pdf as obtained by simulation. The results obtained by simulation correspond very well with the results given by applying the proposed technique. However, obtaining such precise results by simulation requires about 100 times more time.

5.3 FSPN Model with Two Fluid Places

In this section, in order to present an example with two fluid places, we add another fluid place to the example presented in Section 5.1. This is done by duplicating the system. We assume that there are two kinds of users and the two kinds use two separate memories to run their applications. The only connection between the two subsystems is that when the amount of allocated memory of any of the them reaches $\mathrm{M_{crash}}$ both subsystems get into crash state. The model is depicted in Fig. 11.

The model was solved with the following parameters (parameters connected to the additional part of the net are indexed with an additional 2):

$$k = 2, \;\; \mathrm{M_{crash}} = 1,$$

$$\mathrm{M_{high}} = 0.8, \;\; \mathrm{M_{low}} = 0.3,$$

$$\mathrm{M_{high2}} = 0.7, \;\; \mathrm{M_{low2}} = 0.2,$$

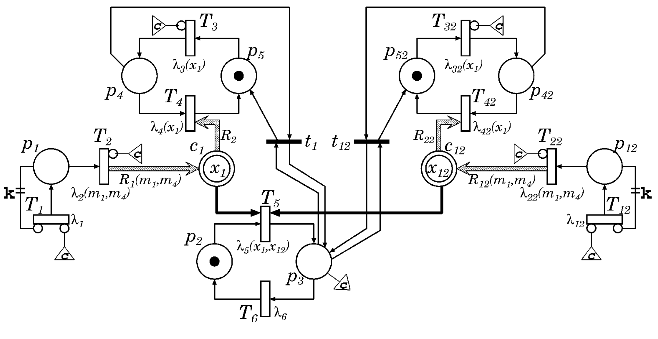

Fig. 11. Garbage collection with separate memories.

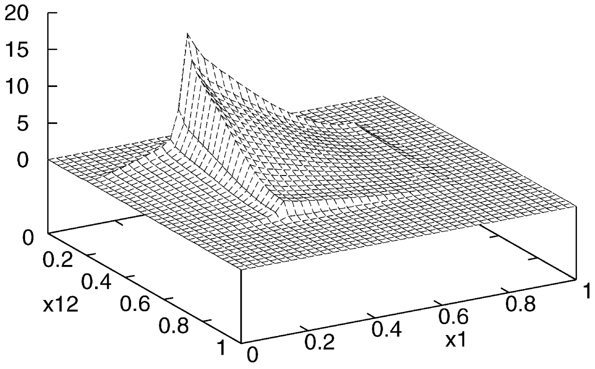

Fig. 12. Density of fluid distribution at transient time $t = 5$.

$$\lambda_1 = 2, \;\; \lambda_2(m_1, m_4) = \frac{k}{m_1 \; (m_4 + 1)} \; ,$$

$$\lambda_{12} = 2, \;\; \lambda_{22}(m_{12}, m_{42}) = \frac{k}{m_{12} \; (m_{42} + 1)} \; ,$$

$$\lambda_3(x_1) = \begin{cases} 0, & x_1 < \mathrm{M_{low}} \\ 1, & \mathrm{M_{low}} \leq \mathrm{x_1} < \mathrm{M_{high}} \\ 10, & \mathrm{M_{high}} \leq \mathrm{x_1} \end{cases} ,$$

$$\lambda_{32}(x_{12}) = \begin{cases} 0, & x_{12} < \mathrm{M_{low2}} \\ 1, & \mathrm{M_{low2}} \leq \mathrm{x_{12}} < \mathrm{M_{high2}} \\ 10, & \mathrm{M_{high2}} \leq \mathrm{x_{12}} \end{cases} ,$$

$$\lambda_4(x_1) = \begin{cases} 1, & x_1 > \mathrm{M_{low}} \\ 10, & x_1 \leq \mathrm{M_{low}} \end{cases} ,$$

$$\lambda_{42}(x_{12}) = \begin{cases} 1, & x_{12} > \mathrm{M_{low2}} \\ 10, & x_{12} \leq \mathrm{M_{low2}} \end{cases} ,$$

$$\lambda_5(x_1) = \delta(x_1 - 1, x_{12} - 1), \;\; \lambda_6 = 0.25,$$

$$R_1(m_1, m_4) = 0.1 \frac{m_1}{m_4 + 1}, \;\; R_2 = 0.2,$$

$$R_{12}(m_{12}, m_{42}) = 0.1 \frac{m_{12}}{m_{42} + 1}, \;\; R_{22} = 0.2,$$

where $\delta(x_1 - 1, x_{12} - 1)$ is used to reflect that firing rate of T_5 is infinite if any of the fluid places reach $\mathrm{M_{crash}}$.

The model was solved with discretization steps $\delta_{time} = 5 \times 10^{-2}, \delta_1 = \delta_{12} = 1.25 \times 10^{-3}$. The number of tangible markings is 45. The discretized state space has about 2.3×10^7 points. Using the proposed technique the storage of the scheme requires the storage of about 7×10^4 numbers. Explicit storage would require storing approximately 7×10^9 numbers. Memory savings are much higher in this case because this example shows the advantages of separate handling of the fluid places as well which did not appear in the first example. One step ahead in the transient analysis takes about 100 seconds.

Figs. 12 and 13 depict the common probability density function of the levels of the fluid places at transient time 5 and 15.

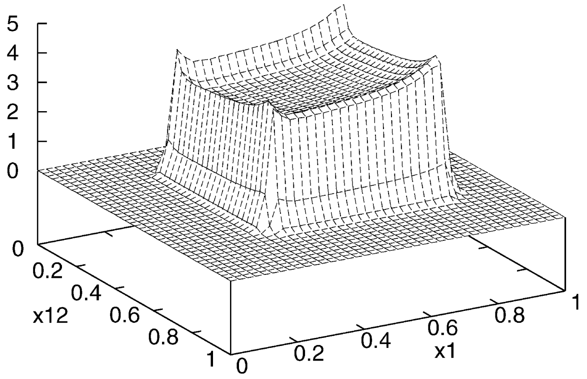

Fig. 13. Density of fluid distribution at transient time $t = 15$.

Our implementation of semidiscretization and simulation is not suitable for more than one fluid places. So that, the results presented in this section are not checked against other methods. However, since the discretization method gave quite accurate results for the example with one fluid place using $\delta_{time} = 5 \times 10^{-2}, \delta_1 = 1.25 \times 10^{-3}$ as discretization steps, we believe that the results given in this section are quite accurate as well.

6 Conclusions

In this paper, a numerical solution technique for the transient analysis of Fluid Stochastic Petri Nets with flush-out arcs has been presented. The transient solution is carried out by applying explicit discretization of the differential equations that describe the transient evolution of the model. Instead of storing explicitly a matrix, the discretization scheme is given by suitable Kronecker expressions of vectors and matrices of relatively small size. This modular description proved to be memory-efficient compared to explicit storage of the discretization scheme.

Because of the high level of modeling freedom given by the formalism, error estimation is not straightforward. As future work, we plan to investigate this issue.

The applicability of the transient solution was shown through the modeling of a multiuser system with garbage collection. The example was used also to compare the proposed method with semidiscretization and simulation. As opposed to semidiscretization, the technique proved to be superior from the point of view of tracking the movement of probability masses and sharp changes in probability density functions.

Acknowledgments

This work was supported in part by the Italian Ministry for University and Scientific Research through the Planet-IP project.

References

[1] H. Alla and R. David, "Continuous and Hybrid Petri Nets," *J. Systems Circuits and Computers,* vol. 8, no. 1, Feb. 1998

[2] M. Gribaudo, M. Sereno, A. Horváth, and A. Bobbio, "Fluid Stochastic Petri Nets Augmented with Flush-Out Arcs: Modeling and Analysis," *Discrete Event Dynamic Systems,* vol. 11, no. 1/2, pp. 97-117, Jan. 2001.

[3] G. Horton, "Computation of the Distribution of Accumulated Reward with Fluid Stochastic Petri Nets," *Proc. Second Int'l Computer Performance & Dependability Symp. (IPDS '96),* 1996.

[4] G. Horton, V.G. Kulkarni, D.M. Nicol, and K.S. Trivedi, "Fluid Stochastic Petri Nets: Theory, Application, and Solution Techniques," *European J. Operations Research,* vol. 105, no. 1, pp. 184-201, Feb 1998.

[5] K. Trivedi and V. Kulkarni, "FSPNs: Fluid Stochastic Petri Nets," *Proc. 14th Int'l Conf. Application and Theory of Petri Nets,* pp. 24-31, 1993.

[6] K. Wolter, "Second Order Fluid Stochastic Petri Nets: An Extension of GSPNs for Approximate and Continuous Modeling," *Proc. World Congress on System Simulation,* pp. 328-332, Sept. 1997.

[7] A. Bobbio, S. Garg, M. Gribaudo, A. Horváth, M. Sereno, and M. Telek, "Modeling Software Systems with Rejuvenation, Restoration and Checkpointing Through Fluid Stochastic Petri Nets," *Proc. Eighth Int'l Workshop Petri Nets and Performance Models,* Sept. 1999.

[8] D.R. Cox, "The Analysis of Non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables," *Proc. Cambridge Phylosophical Society,* vol. 51, pp. 433-440, 1955.

[9] S.V. Patankar, *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill, 1980.

[10] G. Ciardo, D.M. Nicol, and K.S. Trivedi, "Discrete-Event Simulation of Fluid Stochastic Petri Nets," *IEEE Trans. Software Eng.,* vol. 25, no. 2, pp. 207-217, Mar./Apr. 1999.

[11] M. Gribaudo and M. Sereno, "Simulation of Fluid Stochastic Petri Nets," *Proc. Eighth Int'l Symp. Modeling, Analysis and Simulation of Computer and Telecomm. Systems,* pp. 231-239, Aug. 2000.

[12] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modeling with Generalized Stochastic Petri Nets.* John Wiley & Sons, 1995.

[13] M. Gribaudo, M. Sereno, and A. Bobbio, "Fluid Stochastic Petri Nets: An Extended Formalism to Include Non-Markovian Models," *Proc. Eighth Int'l Conf. Petri Nets and Performance Models (PNPM '99),* pp. 71-82, 1999.

[14] P. Buchholz, G. Ciardo, P. Kemper, and S. Donatelli, "Complexity of Memory-Efficient Kronecker Operations with Applications to the Solution of Markov Models," *INFORMS J. Computing,* vol. 13, no. 3, pp. 203-222, 2000.

[15] R.J. LeVeque, *Numerical Methods for Conversation Laws.* Birkhauser Verlag, 1992.

Marco Gribaudo received the Masters degree in computer science from the University of Turin (Italy), in 1997. In February 2002, he also received the PhD degree from the same university. Now, he is a full time researcher in the of computer science in the University of Torino, Italy. His research interests are queuing networks with blocking, fluid stochastic Petri nets and hybrid systems, non-Markovian stochastic process, object oriented languages for system modeling, numerical solution of partial derivative differential equations, self-similar and autocorrelated arrival processes. He is currently working on fluid analysis of TCP flows.

András Horváth received the MSc degree in computer science from the University of Technology and Economics, Budapest, Hungary. Currently, he is a PhD student supervised by Miklós Telek at the same university. From 1999 to 2001, he held a two-year temporary researcher position at the University of Turin (Italy). His research interests are in the area of stochastic processes including performance analysis of non-Markovian systems and modeling issues of communication networks.