Stationary analysis of FSPNs with mutually dependent discrete and continuous parts*

R. German[1], M. Gribaudo[2], G. Horváth[3], M. Telek[3]

[1] Dept. of Comp. Sci. 7, Friedrich-Alexander Universität Erlangen-Nürnberg
e-mail: german@informatik.uni-erlangen.de

[2] Dipartimento di Informatica, Università di Torino, Torino, Italy
e-mail: marcog@di.unito.it

[3] Dept. of Telecommunications, Technical University of Budapest, Hungary
e-mail: {hgabor,telek}@webspn.hit.bme.hu

Abstract

In this paper, we analyze the stationary behaviour of FSPNs with single fluid place and mutually dependent continuous and discrete part. The proposed analysis technique is restricted to the cases when the fluid rate does not change sign in a given marking.

We provide the set of ordinary differential equations and boundary conditions that determines the stationary behaviour and we discuss potential numerical methods that evaluate the stationary distribution based on this description. The numerical analysis of FSPN models of telecommunication systems demonstrate the applicability of the introduced analysis methods.

Key words: Fluid stochastic Petri nets, stationary distribution, ordinary differential equations.

1 Introduction

The fluid extension of the Petri net formalism [13] allows to describe stochastic models with combined discrete and continuous state space with a well

*This work is partially supported by OTKA grant n. T-34972.

established, intuitive graphical language. As a new and effective model description language Fluid stochastic Petri nets (FSPNs) gain importance in the stochastic modeling community, but the widespread application of FSPNs is still delayed by the lack of effective numerical analysis methods and automatic tools implementing the solution methods. This paper intends to extend the set of FSPN analysis methods. We consider a special analysis problem, the stationary analysis of FSPNs with a single finite fluid place, and propose numerical analysis methods for the computationally effective solution of this problem.

Various FSPN formalisms with subtle differences have been introduced in the literature [1, 13, 14, 7], but here we restrict our attention to first order FSPN models with ordinary input and output fluid arcs. The class of FSPN models considered here is identical with the class studied in [8] with the restrictions summarized below.

Our considered FSPN class is wider than the one in [13, 6] and the class of "stochastic fluid models" applied in telecommunication traffic engineering [2, 4], because in those models the evolution of the discrete part does not depend on the fluid part of the model, while in our FSPN formalism the discrete and the continuous part mutually affects each others evolution. This property has a significant consequence on the set of equations describing the underlying model behaviour and, as a consequence, on the numerical methods applicable for the solution of the considered FSPN models.

The majority of FSPN analysis papers, published so far, focuses on the transient analysis of FSPN models. The transient analysis techniques cope with one continuous variable for the time and as many further continuous variables as many fluid places are in the model. The solution of these models are usually based on equi-distance discretization of each continuous variable and the differences of these methods are in the applied approximations of the elementary step of the continuous range and the boundaries.

The stationary analysis of FSPN models allow to eliminate the time variable from the analytical description on the price of having no initial condition for the fluid distribution. The stationary analysis of FSPN models with only one fluid place, i.e., the analytical description with one continuous variable, allows to apply more sophisticated numerical techniques than the ones for the transient analysis. We consider some of these methods for effective analysis of FSPN models in this paper.

The analytical description and the numerical analysis presented in this paper are extensions of the ones presented in [6]. The main difference of this work with respect to [6] is that the transition rates of the discrete part and the fluid drift of the continuous part depends on the fluid level here, while they are constant in [6].

The class of FSPN models considered in this paper is equivalent with the class of models introduced in [8] with the following restrictions:

- there is one finite fluid place in the model,
- the aggregate fluid rate (input rate − output rate) can not change sign in the continuous range, i.e., the aggregate fluid rate is either strictly positive, or strictly negative, or constant zero,
- we allow discontinuity and sign change at the boundaries.

The rest of the paper is organized as follows. Section 2 presents the analytical description of the continuous range of the model together with its possible solution methods. Section 3 introduces the boundary equations and the obtained set of linear equations. Section 4 discusses the case when the fluid level remains constant in the continuous range in some discrete states. Numerical examples are presented in Section 5. Section 6 concludes the paper.

2 Stationary description of FSPNs with single finite fluid place

The formal definition of the considered FSPN class is provided in [8]. Here we only summarize the analytical description of the considered FSPN models. The $Z(t) = \{X(t), Y(t)\}$ process represents the state of a FSPN with single fluid place, where $X(t) \in \mathcal{S}$ is the discrete marking and $Y(t) \in [0, B]$ is the fluid level of the (only) fluid place of the FSPN at time t. $\mathcal{S}$ denotes the finite set of reachable discrete markings and B is the maximum fluid level. With the applied restriction of the fluid rates the distribution of the fluid level might not have probability masses between 0 and B. We define $\hat{\pi}_i(t,x)$, $\hat{l}_i(t)$ and $\hat{u}_i(t)$ to describe the fluid density and the probability masses at the lower and upper bound, respectively, i.e.

$$\hat{\pi}_i(t,x) = \lim_{\Delta \to 0} \frac{Pr(X(t) = i, x \leq Y(t) < x + \Delta)}{\Delta} \ ,$$

$$\hat{l}_i(t) = Pr(X(t) = i, Y(t) = 0) \ ,$$

$$\hat{u}_i(t) = Pr(X(t) = i, Y(t) = B) \ .$$

Assuming the system converges to a stationary solution, the stationary fluid density function and fluid mass functions are

$$\pi_i(x) = \lim_{t \to \infty} \hat{\pi}_i(t,x), \quad l_i = \lim_{t \to \infty} \hat{l}_i(t), \text{ and } \ u_i = \lim_{t \to \infty} \hat{u}_i(t).$$

Over the $(0, B)$ interval the stationary distribution, $\pi(x) = \{\pi_i(x)\}$, satisfies [8]:

$$\frac{d}{dx}\Big(\pi(x)\mathbf{R}(x)\Big) = \pi(x)\mathbf{Q}(x) \ , \tag{1}$$

where matrix $\mathbf{Q}(x)$ is the transition rate matrix of the discrete marking of the Petri net when the fluid level is x, and the diagonal matrix $\mathbf{R}(x) = \text{diag} < r_i(x) >$ is composed by the fluid rates $r_i(x)$ (the rate of change of the fluid level at the fluid place in discrete marking i and fluid level x). For the numerical solution of (1) we also need boundary conditions as it is discussed later.

To evaluate the solution of (1) between 0 and B we have

$$\frac{d}{dx}\pi(x) = \pi(x)\left(\mathbf{Q}(x) - \frac{d}{dx}\mathbf{R}(x)\right)\mathbf{R}^{-1}(x) = \pi(x)\mathbf{B}(x) \ , \tag{2}$$

where

$$\mathbf{B}(x) = \left(\mathbf{Q}(x) - \frac{d}{dx}\mathbf{R}(x)\right)\mathbf{R}^{-1}(x) \ .$$

The case when the fluid level remains constant in some discrete markings ($\exists i \in \mathcal{S}$ such that $r_i(x) = 0$) is discussed later in Section 4.

2.1 Solution based on series expansion

We denote the solution of (2) by

$$\pi(x) = \pi(0^+)\mathbf{W}(x) \ , \tag{3}$$

for $0 < x < B$, where $\mathbf{W}(x)$ satisfies

$$\frac{d}{dx}\mathbf{W}(x) = \mathbf{W}(x)\mathbf{B}(x) \ , \tag{4}$$

with initial condition $\mathbf{W}(0) = \mathbf{I}$. One way to find the solution of (3) is obtained from the power series representation of $\mathbf{W}(x)$ with coefficient matrices $\mathbf{M}_i$:

$$\mathbf{W}(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!}\ \mathbf{M}_i \ , \tag{5}$$

and the Taylor expansion of $\mathbf{B}(x)$ around 0:

$$\mathbf{B}(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!}\ \ \frac{d^i}{dx^i}\mathbf{B}(x)\Big|_{x=0^+} \ .$$

From (4) we have

$$\sum_{i=0}^{\infty} \frac{x^i}{i!}\, \mathbf{M}_{i+1} = \sum_{i=0}^{\infty} \frac{x^i}{i!}\, \mathbf{M}_i \cdot \sum_{j=0}^{\infty} \frac{x^j}{j!}\ \left.\frac{d^j}{dx^j}\mathbf{B}(x)\right|_{x=0^+} .$$

The equivalence of the coefficient matrices of the power series on the left and on the right hand sides results in the following recursive relation of the $\mathbf{M}_i$ coefficient matrices:

$$\mathbf{M}_{i+1} = \sum_{j=0}^{i} \binom{i}{j} \mathbf{M}_{i-j}\ \left.\frac{d^j}{dx^j}\mathbf{B}(x)\right|_{x=0^+} , \tag{6}$$

where, from the initial condition of $\mathbf{W}(x)$, $\mathbf{M}_0 = \mathbf{I}$.

For level independent $\mathbf{Q}(x)$ and $\mathbf{R}(x)$ (i.e., $\mathbf{Q}(x) = \mathbf{Q}$, $\mathbf{R}(x) = \mathbf{R}$) the applied expansion results in

$$\mathbf{W}(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!} \left(\mathbf{Q}\ \mathbf{R}^{-1}\right)^i = e^{x\ \mathbf{Q}\ \mathbf{R}^{-1}} .$$

In case of exponentially increasing fluid rates $\mathbf{R}(x) = e^x\mathbf{R}$ and fluid level independent transition rates $\mathbf{Q}(x) = \mathbf{Q}$ the $\mathbf{B}(x)$ matrix is $-\mathbf{I} + \mathbf{Q}\mathbf{R}^{-1}e^{-x}$, and the first $\mathbf{M}_i$ coefficients are: $\mathbf{M}_0 = \mathbf{I}$, $\mathbf{M}_1 = -\mathbf{I} + \mathbf{Q}\mathbf{R}^{-1}$, $\mathbf{M}_2 = \mathbf{M}_1^2 - \mathbf{Q}\mathbf{R}^{-1}$.

For numerical analysis we used the $\mathbf{N}_i = \mathbf{M}_i/i!$ coefficient matrices instead of $\mathbf{M}_i$, where $\mathbf{N}_0 = \mathbf{I}$ and

$$\mathbf{N}_{i+1} = \frac{1}{i+1} \sum_{j=0}^{i} \mathbf{N}_{i-j} \frac{1}{j!}\ \left.\frac{d^j}{dx^j}\mathbf{B}(x)\right|_{x=0^+} . \tag{7}$$

2.2 Numerical solution of differential equations

There are various numerical solution techniques available for the solution of ordinary differential equations (ODEs) like (2). Among others, we can classify them according to the following properties: order of approximation, explicit or implicit, fix or dynamic step size. For the details of numerical ODE solution methods (e.g., local truncation error) we refer to standard numerical analysis textbooks like [10].

We selected 3 numerical ODE solution methods to compare their properties. The TR (trapezoid rule) method is a first order implicit method, whose elementary step is:

$$\mathbf{W}(x_{n+1}) = \mathbf{W}(x_n) + \frac{h}{2}\mathbf{W}(x_n)\mathbf{B}(x_n) + \frac{h}{2}\mathbf{W}(x_{n+1})\mathbf{B}(x_{n+1}) . \tag{8}$$

Note, that the unknown $\mathbf{W}(x_{n+1})$ appears also on the rhs and that is why the

$$\mathbf{W}(x_{n+1}) = \mathbf{W}(x_n)\Big(\mathbf{I} + \frac{h}{2}\mathbf{B}(x_n)\Big)\Big(\mathbf{I} - \frac{h}{2}\mathbf{B}(x_{n+1})\Big)^{-1}$$

or the

$$\mathbf{W}(x_{n+1}) = \mathbf{W}(x_n)\Big(\mathbf{I} + \frac{h}{2}\mathbf{B}(x_n)\Big)\sum_{i=0}^{\infty}\frac{h}{2}\mathbf{B}^i(x_{n+1})$$

expression is used for the computation. The RK4 (Runge-Kutta) method is a fourth order explicit method, whose elementary step is:

$$\begin{aligned}
\mathbf{K}_1 &= h\mathbf{W}(x_n)\mathbf{B}(x_n), \\
\mathbf{K}_2 &= h\Big(\mathbf{W}(x_n) + \tfrac{1}{2}\,\mathbf{K}_1\Big)\mathbf{B}(x_n + \tfrac{h}{2}), \\
\mathbf{K}_3 &= h\Big(\mathbf{W}(x_n) + \tfrac{1}{2}\,\mathbf{K}_2\Big)\mathbf{B}(x_n + \tfrac{h}{2}), \\
\mathbf{K}_4 &= h\Big(\mathbf{W}(x_n) + \mathbf{K}_3\Big)\mathbf{B}(x_n + h), \\
\mathbf{W}(x_{n+1}) &= \mathbf{W}(x_n) + \frac{1}{6}\,\mathbf{K}_1 + \frac{1}{3}\,\mathbf{K}_2 + \frac{1}{3}\,\mathbf{K}_3 + \frac{1}{6}\,\mathbf{K}_4\ .
\end{aligned} \tag{9}$$

The TR-BDF2 method is a combination of the TR method and the BDF2 (2nd order backwards difference formula) method which is a second order implicit method. The TR-BDF2 method is found to be effective and accurate in the analysis of homogeneous CTMCs [12, 11]. An elementary step of the method is composed by a TR step and a BDF2 step:

$$\begin{aligned}
&\mathbf{W}(x_n + \gamma h) = \mathbf{W}(x_n) + \frac{\gamma h}{2}\mathbf{W}(x_n)\mathbf{B}(x_n) + \frac{\gamma h}{2}\mathbf{W}(x_n + \gamma h)\mathbf{B}(x_n + \gamma h), \\
&\mathbf{W}(x_{n+1}) = \\
&\quad -\frac{(1-\gamma)^2}{\gamma(2-\gamma)}\mathbf{W}(x_n) + \frac{1}{\gamma(2-\gamma)}\mathbf{W}(x_n + \gamma h) + \frac{h(1-\gamma)}{2-\gamma}\mathbf{W}(x_{n+1})\mathbf{B}(x_{n+1}),
\end{aligned} \tag{10}$$

where γ $(0 < \gamma < 1)$ is a constant determining the internal point of the h interval. Its typical value falls in the $0.5 \leq \gamma \leq 0.7$ range. Similar to the TR method a matrix inversion is necessary in both sub-steps of the method.

2.3 Step size control

The numerical ODE solution methods can be used with fix and with dynamically varying step size. There are several proposals to dynamically modify the step size, but all of these methods are composed by two main elements, the estimation of the local truncation error (LTE) and the policy of step size adjustment.

We used the following approximations of the LTE:

$$LTE_a = \frac{1}{h} ||\mathbf{W}(x_{n+1}) - \mathbf{W}(x_n)|| \ ,$$

$$LTE_b = \frac{h(-3\gamma^2 + 4\gamma - 2)}{6(2-\gamma)} \left|\left| -\frac{1}{\gamma}\mathbf{W}(x_n) + \frac{1}{\gamma(1-\gamma)}\mathbf{W}(x_n + \gamma h) - \frac{1}{1-\gamma}\mathbf{W}(x_{n+1}) \right|\right| \ ,$$

and

$$LTE_c = ||\mathbf{W}(x_{n+1}) - \mathbf{W}'(x_{n+1})|| \ ,$$

where $\mathbf{W}'(x_{n+1})$ is calculated in 2 steps using step size $h/2$. LTE_b is applicable only with the TR-BDF method. The calculation of LTE_a and LTE_b is based on known quantities while the analysis of $\mathbf{W}'(x_{n+1})$ cost 2 extra elementary steps for each LTE calculation.

We compared two step size adjustment policies the CI (continuous increase) policy and the HY (hysteresis) policy. The CI policy is based on the local precision requirement (ϵ) and it increases the step size ($h_{new} = ah_{old}$) in each steps of the method as long as $LTE < \epsilon$ and otherwise the step size is decreased to $h_{old}/2$. The typical value of a is in the range of $(1.0001, 1.5)$.

The HY policy is based on the local precision requirement (ϵ) and the minimal local precision requirement (ϵ_{min}). The step size remains unchanged as long as $\epsilon_{min} < LTE < \epsilon$ and the step size is multiplied (divided) by 2 if $\epsilon < LTE$ ($LTE < \epsilon_{min}$). In case of computationally expensive LTE calculation it is possible to reduce the number of LTE checks of the HY policy for one per every N steps (e.g., $N = 2, 5, 10, 20, 50, \ldots$).

3 Boundary conditions

For the boundary conditions two different cases arises, depending on the direction of the fluid flow [3].

Case a) absorbing states: The first case is the one in which the fluid flow is directed towards the bound, that is $r_j(0^+) < 0$ (for the lower bound), or $r_j(B^-) > 0$ (for the upper bound). In this case, independently on the sign of the rate at the discontinuity, probability mass builds up at the bound. In this case we can characterize the bound by a single equation:

$$l_j q_{jj}(0) - \pi_j(0^+) r_j(0^+) + \sum_{k \neq j} l_k q_{kj}(0) = 0,$$

for the lower bound, and for the upper bound:

$$u_j q_{jj}(B) + \pi_j(B^-) r_j(B^-) + \sum_{k \neq j} u_k q_{kj}(B) = 0.$$

Case b) emitting states: The second case is the one in which the fluid flow is directed in the opposite direction with respect to the bound, that is $r_j(0^+) > 0$ (for the lower bound), or $r_j(B^-) < 0$ (for the upper bound). In this case we may have two different behavior, depending on the sign of the rate at boundary. No probability mass will be formed if the sign of the fluid rate at the boundary is identical to the one immediately after ($sign(r_j(0)) = sign(r_j(0^+))$ and if $sign(r_j(B)) = sign(r_j(B^-))$). (It is the most common case, but we also consider the special case when $r_j(x)$ is not continuous at 0 and/or at B for completeness.) Probability mass will build up if the sing of the rate at the boundary is zero or opposite to the fluid flow next to the boundary. In any case we will have two equations per boundary, that are:

$$\begin{aligned}
\text{if } sign(r_j(0)) = sign(r_j(0^+)): \quad & 0 = l_j q_{jj}(0), \\
& \pi_j(0^+) r_j(0^+) = \sum_{k \neq j} l_k q_{kj}(0), \\
\text{if } sign(r_j(0)) \neq sign(r_j(0^+)): \quad & 0 = l_j q_{jj}(0) + \sum_{k \neq j} l_k q_{kj}(0), \\
& \pi_j(0^+) r_j(0^+) = 0,
\end{aligned}$$

for the lower bound, and for the upper bound:

$$\begin{aligned}
\text{if } sign(r_j(B)) = sign(r_j(B^-)): \quad & 0 = u_j q_{jj}(B), \\
& -\pi_j(B^-) r_j(B^-) = \sum_{k \neq j} u_k q_{kj}(B), \\
\text{if } sign(r_j(B)) \neq sign(r_j(B^-)): \quad & 0 = u_j q_{jj}(B) + \sum_{k \neq j} u_k q_{kj}(B), \\
& -\pi_j(B^-) r_j(B^-) = 0.
\end{aligned}$$

3.1 The set of equations

If the number of discrete states of the model is $|\mathcal{S}|$, then the number of unknowns is $4|\mathcal{S}|$, i.e., vectors l, u, $\pi(0^+)$ and $\pi(B^-)$. By the considerations provided in the previous sections we have got exactly $4|\mathcal{S}|$ equations to evaluate these unknowns. Eq. (3) for $x = B^-$ represents $|\mathcal{S}|$ equations. The states with positive (negative) fluid rate over the $(0, B)$ interval are emitting (absorbing) states at the lower bound and absorbing (emitting) states at the upper bound, hence there are 3 further equations per state.

We can represent the obtained set of equations using matrix notation. $\mathcal{S}^+$ ($\mathcal{S}^-$) is the subset of states with with positive (negative) fluid rate over the $(0, B)$ interval. With proper numbering of the states $\mathbf{Q}(x)$ and $\mathbf{R}(x)$ have

the form

$$\mathbf{Q}(x) = \begin{array}{|c|c|}\hline \mathbf{Q}^{--}(x) & \mathbf{Q}^{-+}(x) \\ \hline \mathbf{Q}^{+-}(x) & \mathbf{Q}^{--}(x) \\ \hline \end{array} \quad \text{and} \quad \mathbf{R}(x) = \begin{array}{|c|c|}\hline \mathbf{R}^{-}(x) & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{R}^{+}(x) \\ \hline \end{array}.$$

The vectors associated with $\mathcal{S}^+$ and $\mathcal{S}^-$ are denoted with $+$ and $-$ superscript, respectively. With these notations the boundary conditions are:

- for absorbing states
  - lower bound: $-\pi^-(0)\mathbf{R}^-(0) + l^-\mathbf{Q}^{--}(0) + l^+\mathbf{Q}^{+-}(0) = 0$
  - upper bound: $\pi^+(B)\mathbf{R}^+(B) + u^-\mathbf{Q}^{-+}(B) + u^+\mathbf{Q}^{++}(B) = 0$
- for emitting states (without sign change at the boundary)
  - lower bound: $l^+ = 0$
    $-\pi^+(0)\mathbf{R}^+(0) + l^-\mathbf{Q}^{-+}(0) + l^+\mathbf{Q}^{++}(0) = 0$
  - upper bound: $u^- = 0$
    $\pi^-(B)\mathbf{R}^-(B) + u^-\mathbf{Q}^{--}(B) + u^+\mathbf{Q}^{+-}(B) = 0$

Using also the partitioned form of the $\mathbf{W}(B)$ matrix the $z\mathbf{Z} = 0$ linear equation needs to be solved where $z = \{l^-, l^+, \pi^-(0), \pi^+(0), \pi^-(B), \pi^+(B), u^-, u^+\}$ and matrix $\mathbf{Z}$ is provided in Figure 1.

	l^-	l^+	$\pi^-(0)$	$\pi^+(0)$	$\pi^-(B)$	$\pi^+(B)$	u^-	u^+
l^-	$\mathbf{Q}^{--}(0)$			$\mathbf{Q}^{-+}(0)$				
l^+	$\mathbf{Q}^{+-}(0)$	I		$\mathbf{Q}^{++}(0)$				
$\pi^-(0)$	$-\mathbf{R}^-(0)$		$\mathbf{W}^{--}(B)$			$\mathbf{W}^{-+}(B)$		
$\pi^+(0)$			$\mathbf{W}^{+-}(B)$	$-\mathbf{R}^+(0)$		$\mathbf{W}^{++}(B)$		
$\pi^-(B)$			-I		$\mathbf{R}^-(B)$			
$\pi^+(B)$						-I		$\mathbf{R}^+(B)$
u^-					$\mathbf{Q}^{--}(B)$		I	$\mathbf{Q}^{-+}(B)$
u^+					$\mathbf{Q}^{+-}(B)$			$\mathbf{Q}^{++}(B)$

Figure 1: The structure of the coefficient matrix

3.2 Normalizing condition

Since the $4n$ equations presented above are linearly dependent we need to define a normalizing condition to determine the required solution. The normalizing condition is obtained from the fluid distribution and the probability masses at the bounds as

$$1 \;=\; \sum_{j\in\mathcal{S}} l_j + u_j + \int_0^B \pi_j(x)\; dx = l\mathbf{1} + u\mathbf{1} + \int_0^B \pi(x)\mathbf{1}\; dx, \qquad (11)$$

where $\mathbf{1}$ is the column vector of ones with the appropriate size. Unfortunately, the normalizing condition can not be evaluated based on the $\mathbf{W}(B)$ matrix, but it requires the evaluation of the $\int_0^B \pi(x)\mathbf{1}\; dx = \pi(0)\int_0^B \mathbf{W}(x)\mathbf{1}\; dx$ integral.

To obtain the accumulated measure $\mathbf{L}(x) = \int_0^x \mathbf{W}(u)du$ the following solution methods can be used:

a) Numerical integration of $\mathbf{W}(x)$ during the numerical solution of ODE (2):

$$\mathbf{L}(x_{n+1}) = \mathbf{L}(x_n) + \frac{h}{2}\Big(\mathbf{W}(x_n) + \mathbf{W}(x_{n+1})\Big). \qquad (12)$$

In case of the TR-BDF2 method we can utilize the solution at the internal point $x_n + \gamma h$ as well:

$$\mathbf{L}(x_{n+1}) = \mathbf{L}(x_n) + \frac{h}{2}\Big(\gamma\mathbf{W}(x_n) + \mathbf{W}(x_n+\gamma h) + (1-\gamma)\mathbf{W}(x_{n+1}\Big). \qquad (13)$$

b) Series expansion:

$$\mathbf{L}(x) = \sum_{i=0}^{\infty} \frac{x^{i+1}}{(i+1)!}\; \mathbf{M}_i \qquad (14)$$

c) Numerical solution of the ODE describing the accumulated measure:

$$\frac{d}{dx}\mathbf{L}(x) = \mathbf{I} + \mathbf{L}(x)\mathbf{B}(x) - \int_{u=0}^{x} \mathbf{L}(u)\mathbf{B}'(u)\; du\;. \qquad (15)$$

If the $\int_0^B \pi(x)\mathbf{1}\; dx$ integral is calculated together with matrix $\mathbf{W}(x)$ we need to calculate matrix $\mathbf{L}(x)$. In contrast, if $\mathbf{W}(x)$ and $\pi(0)$ (actually $\pi(0)$ times a constant) is calculated prior to the calculation of $\int_0^B \pi(x)\mathbf{1}\; dx$ it is enough to calculate the $\pi(0)\mathbf{L}(x)$ vector instead of the $\mathbf{L}(x)$ matrix.

3.3 The overall analysis method

To conclude this section we summarize the main steps of the overall numerical procedure in Figure 2. The two versions of the procedure differs in the calculation if the normalizing condition. The first version (on the left) calculates the $\mathbf{L}(B)$ matrix, while the second version calculates the $\pi(0)\mathbf{L}(B)$ vector.

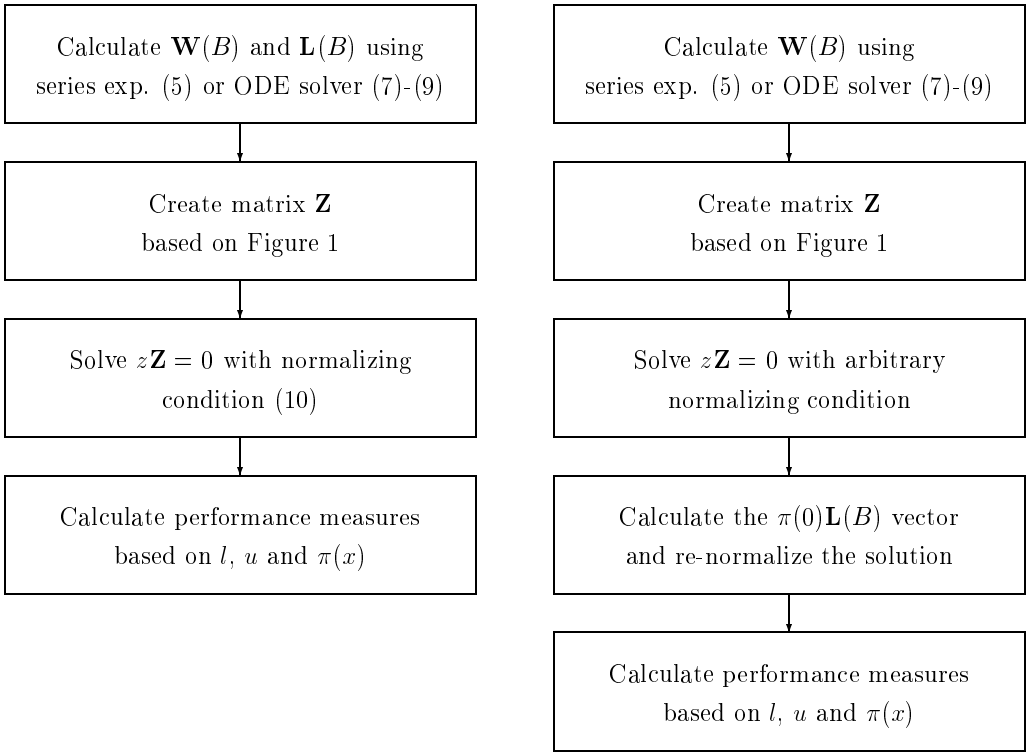

Figure 2: The main steps of the solution method

4 Extension to states with zero rate

In the previous sections we have always considered $r_j(x) \neq 0$ for the $(0, B)$ interval. In many practical situation however, there are cases where $r_j(x) = 0$. When there are this kind of *zero states*, both the continuous part and the boundary conditions change accordingly.

4.1 Zero states at continuous domain

Let us partition the state space ($\mathcal{S}$) into the sets of non-zero ($\mathcal{S}^{\emptyset}$) and zero states ($\mathcal{S}^{0}$). With proper numbering of the states the generator and the fluid rate matrix takes the form

$$\mathbf{Q}(x) = \begin{array}{|c|c|}\hline \mathbf{Q}^{\emptyset\emptyset}(x) & \mathbf{Q}^{\emptyset 0}(x) \\ \hline \mathbf{Q}^{0\emptyset}(x) & \mathbf{Q}^{00}(x) \\ \hline \end{array} \quad \text{and} \quad \mathbf{R}(x) = \begin{array}{|c|c|}\hline \mathbf{R}^{\emptyset}(x) & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \\ \hline \end{array}.$$

Let $\pi^0(x)$ be the vector of the probability density of the states in $\mathcal{S}^0$ (zero states), and $\pi^{\emptyset}(x)$ the probability density of the other states in $\mathcal{S}^{\emptyset}$ (non-zero states). We can rewrite equation (1) as:

$$\frac{d}{dx}\Big(\pi^{\emptyset}(x)\mathbf{R}^{\emptyset}(x)\Big) = \pi^{\emptyset}(x)\mathbf{Q}^{\emptyset\emptyset}(x) + \pi^{0}(x)\mathbf{Q}^{0\emptyset}(x)\ , \tag{16}$$

$$0 = \pi^{\emptyset}(x)\mathbf{Q}^{\emptyset 0}(x) + \pi^{0}(x)\mathbf{Q}^{00}(x)\ . \tag{17}$$

From Equation (16) and (17) we obtain:

$$\frac{d}{dx}\Big(\pi^{\emptyset}(x)\mathbf{R}^{\emptyset}(x)\Big) = \pi^{\emptyset}(x)\left[\mathbf{Q}^{\emptyset\emptyset}(x) - \mathbf{Q}^{\emptyset 0}(x)\mathbf{Q}^{00^{-1}}(x)\mathbf{Q}^{0\emptyset}(x)\right]\ , \tag{18}$$

$$\pi^{0}(x) = -\pi^{\emptyset}(x)\mathbf{Q}^{\emptyset 0}(x)\mathbf{Q}^{00^{-1}}(x)\ , \tag{19}$$

Note that

$$\widehat{\mathbf{Q}}(x) = \mathbf{Q}^{\emptyset\emptyset}(x) - \mathbf{Q}^{\emptyset 0}(x)\mathbf{Q}^{00^{-1}}(x)\mathbf{Q}^{0\emptyset}(x)$$

is the generator of the discrete state process restricted to the non-zero states at fluid level x.

Using $\widehat{\mathbf{Q}}(x)$ and $\widehat{\mathbf{R}}(x) = \mathbf{R}^{\emptyset}(x)$ equation (18) is the same kind as (1) and the same solution methods are applicable.

4.2 Zero states at the bounds

When $r_j(x) = 0$ over the $(0, B)$ interval the value of $r_j(0)$ and $r_j(B)$ does not play role. Probability mass can develop at the bounds in any case according to

$$\begin{aligned} 0 &= l_j q_{jj}(0) + \sum_{k\neq j} l_k q_{kj}(0), \\ 0 &= u_j q_{jj}(B) + \sum_{k\neq j} u_k q_{kj}(B). \end{aligned} \tag{20}$$

In matrix form (20) is

$$\begin{aligned} 0 &= l^{\emptyset}\mathbf{Q}^{\emptyset 0}(0) + l^{0}\mathbf{Q}^{00}(0), \\ 0 &= u^{\emptyset}\mathbf{Q}^{\emptyset 0}(B) + u^{0}\mathbf{Q}^{00}(B). \end{aligned} \tag{21}$$

4.3 Normalizing condition

The normalizing condition is the same as before, but utilizing to the partitioning of the state space we can calculate the normalizing condition by the measures of $\mathcal{S}^{\emptyset}$:

$$\begin{aligned} 1 &= l\mathbf{1} + u\mathbf{1} + \int_0^B \pi(x)\mathbf{1}\, dx \\ &= l^{\emptyset}\Big(\mathbf{I}^{\emptyset\emptyset}\mathbf{1} - \mathbf{Q}^{\emptyset 0}(0)\mathbf{Q}^{00^{-1}}(0)\mathbf{1}\Big) + u^{\emptyset}\Big(\mathbf{I}^{\emptyset\emptyset}\mathbf{1} - \mathbf{Q}^{\emptyset 0}(B)\mathbf{Q}^{00^{-1}}(B)\mathbf{1}\Big) \\ &\quad + \int_0^B \pi^{\emptyset}(x)\Big(\mathbf{I}^{\emptyset\emptyset}\mathbf{1} - \mathbf{Q}^{\emptyset 0}(x)\mathbf{Q}^{00^{-1}}(x)\mathbf{1}\Big)\, dx, \end{aligned} \tag{22}$$

which is easier to evaluate using the numerical integration of $\mathbf{W}(x)$ during the numerical solution of the ODE (1) (i.e., eq. (12) or (13)).

5 Numerical examples

In this section we introduce the FSPN models of two telecommunication systems and analyze them with the proposed numerical methods.

5.1 Example 1: Cable modem subnetwork

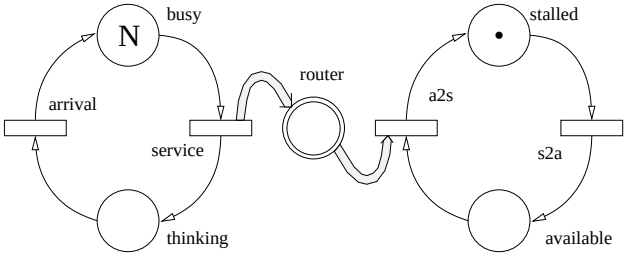

Figure 3: The FSPN model of the cable modem system

We consider a subnetwork, that access the internet using a shared channel like a cable modem. N users work in the subnetwork (left block of the FSPN in Figure 3). Each of them produce packets with bandwidth r (through the fluid arc from **sercive** to **router**) when they are active (indicated by a token in **busy**) and they do not send any packets during their thinking phase (indicated by a token in **thinking**). When the router holds the channel (there is a token in **available**), packets can be transmitted through the channel at bandwidth c (represented by the fluid arc from **router** to **a2s**).

number of users N	8
arrival rate α	$0,2$ per user
service rate $\lambda(x)$	$0.4(1-p(x))$ per user
transmission rate $r(x)$	$0.3(1-p(x))$ per user
router output bandwidth C	3
router buffer size B	2
rate of **a2s**	0.5
rate of **s2a**	0.5

Table 1: Parameter of the cable modem example

When the channel is held by another subnetwork (token in **stalled**), packets need to be enqueued in the router buffer (fluid place **router**). The channel capacity is high enough to satisfy the instantaneous demand of users, that is $c > Nr$. (This condition ensures that the drift does not change sign in the $0, B$ range in any discrete state.) In order to provide better performance, a pro-active routing algorithm, like RED (random early detection) is used at the router. That is, packets coming to the router buffer are discarded with a probability ($p(x)$) that linearly increases with the buffer size. The actual input rate of **router** is $r(1-p(x))$ times the number of active users. At the same time the pro-active routing algorithm slows down the service of the users and they busy periods increase proportionally, since they have to retransmit the packets they have lost. The transition rate of **service** is $\lambda(1-p(x))$. Transitions **service** and **arrival** have infinite server semantic. In the discrete state ($\#\mathbf{thinking} = N, \#\mathbf{stalled} = 1$) the fluid rate remains constant, hence we applied the technique described in Section 4.

We used 4 different $p(x)$ functions ($= 0$, $= x/(1.2B)$, $= x/(1.01B)$, $= 0.99(1-e^x)/(1-e^B)$) together with the model parameters summarized in Table 1. The resulted fluid level distribution is depicted in Figure 4. The numerical differences of the applied solution methods are negligible. The computation time of our Matlab implementation was less than 10 seconds for each curve with the ODE solvers more than one minute with the series expansion. The number of elementary steps (S) and the number of step size reductions (M) of the ODE solution methods with continuous step size increase (CI) policy are summarized in Table 2, where the applied step increase coefficient was $a = 1.2$ and we used LTE_c.

5.2 Example 2: Parallel WEB server

Our second example is an extension of the parallel WEB server system studied in [5] and [6]. Figure 5 represents the FSPN model of the system.

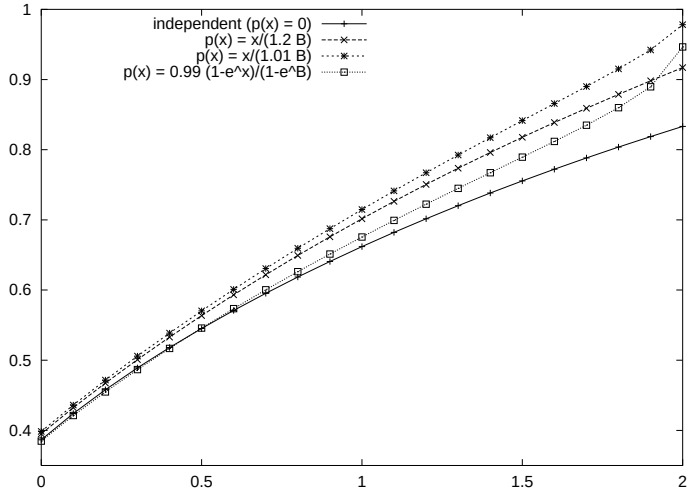

Figure 4: Distribution of the buffer content with different $p(x)$ functions

LTE=10^{-5}	TR		RK4		TR-BDF2	
	S	M	S	M	S	M
$p(x) = 0$	772	13	153	2	241	6
$p(x) = \frac{x}{1.2B}$	441	22	185	8	286	8
$p(x) = \frac{x}{1.01B}$	638	60	289	21	344	25
$p(x) = 0.99\frac{1-e^x}{1-e^B}$	845	89	169	13	560	31

Table 2: Number of elementary ODE solution steps

Requests arrive in burst to the web server. The left subnet models the bursty generation of the requests by an interrupted poisson process. Places **on** and **off** model the state of the process and transition **arrival** models the requests arrival. Transitions **off2on** and **on2off** model respectively the beginning and the end of a burst. Interruption of the arrival is modeled by the inhibitor arc that connects place **off** to transition **arrival**. The parallel web server has a common buffer where all the requests are queued. This buffer is modeled by place **insystem**. The request buffer has a finite capacity K, which is modeled by place **free**. The number of tokens in place **free** represents the position still available in the buffer. Since WWW pages are typically composed of text and images, the data transmitted back to the client has a very big variance. This is approximated by the two immediate transitions **long** and **short** which correspond respectively to the transmission of a small (i.e. an HTML page) or a large (i.e. an image) amount of data. The parallelism of the web server is represented by the number of tokens p contained in place **processors**. Each time a new request arrives, it is

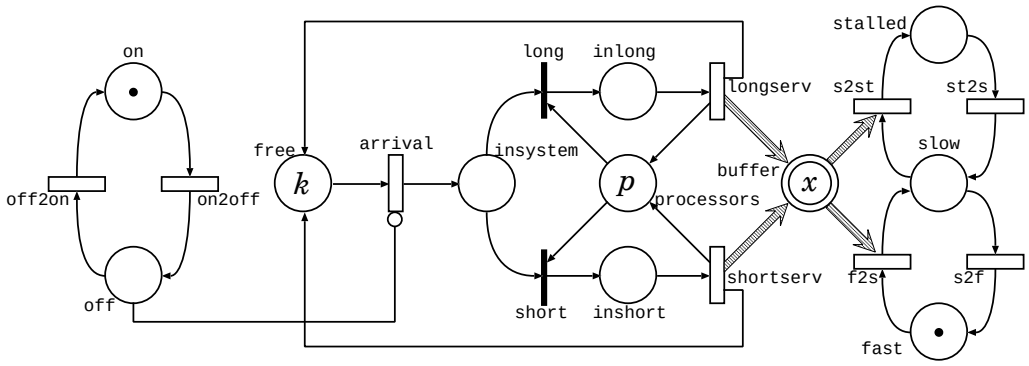

Figure 5: The FSPN model of the parallel WEB server

served immediately if there is a free processor, otherwise it is enqueued. The actual service of the request is modeled by transitions **longsrv** and **shortsrv**. During this service period, packets are transferred using a LAN to the buffer of the router. The server applies a traffic control procedure to reduce the probability of buffer overflow. The transmission rate of each processors depends on the fluid level and the firing rate of transitions **longsrv** and **shortsrv** changes according to the transmission rate. As soon as a service has finished, the processor becomes available and can serve another request. Packets are approximated by fluid, and the router buffer is modeled by fluid place **buffer**. This buffer has a finite capacity B. If a packet arrives when the buffer is full, it is lost. The speed with which data is transmitted by the network is fluctuating during transmission, due to varying network load. This is modeled by the subnetwork on the right. The traffic generated by the WEB server is a significant part of the network load, hence transition rates of the subnetwork on the right is a function of the buffer content. The network can be in three different states: at full speed (modeled by place **fast**), at reduced speed (place **slow**) or stalled (place **stalled**). Transitions **f2s**, **s2f**, **s2st** and **st2s** represent the changes of the network state.

The parameters of the model are given in Table 3. Transitions **shortsrv** and **fastsrv** have infinite server semantic. The fluid arcs connecting transition **shortsrv** and **fastsrv** to the fluid buffer have a marking dependent flow rate: the amount of fluid transferred to the buffer is a function of the number of processors serving long (resp. short) requests and the fluid level. The flow rate of the fluid arc connecting **buffer** to transition **f2s** corresponds to the network speed in the fast state, and the flow rate of the arc connecting **buffer** to transition **s2st** corresponds to the network speed in the slow state. The rate of **on2off** is given by $1/BL$, the rate of **on2off** $1/(BL(\Theta-1))$, and the rate of **arrival** by $\bar{\lambda}\Theta$.

mean arrival rate $\bar{\lambda}$	1800 requests per hour
burstiness Θ	2
mean burst length BL	1 h
small data size	1 kB (rate:$1/(1-p(x))$)
large data size	39 kB (rate:$1/39(1-p(x))$)
fast transfer rate	128 kB/s
slow transfer rate	64 kB/s
network transfer rate	$9.6(1-p(x))$ kB/s
change of network condition	0.8 s
request buffer size k	10
packet buffer size B	16 kB
number of processors p	2

Table 3: Parameter of the parallel web server model

LTE=10^{-4}	TR		RK4		TR-BDF2	
	S	M	S	M	S	M
$p(x)=0$	318	20	148	5	166	10
$p(x)=\frac{x}{1.2B}$	493	15	152	6	254	8
$p(x)=\frac{x}{1.01B}$	535	29	204	14	270	14
$p(x)=0.99\frac{1-e^x}{1-e^B}$	585	31	196	13	311	18

Table 4: Number of elementary ODE solution steps

This model also has some zero states (when the requests buffer is empty and the system is stalled), so we applied the technique described in Section 4. The considered model has 180 states and its solution with our Matlab implementation takes less than 90 seconds on a $800Mhz$ machine with the ODE solvers. The series expansion method did converge in reasonable time. We used several built in Matlab routines, and the majority of the computation time is devoted to the script processing for matrix construction with both solution approaches, the ODE solver and the series expansion. Most probably this part of the computation could be significantly faster in a C implementation. The results are provided in Figure 6 and the number of elementary steps in Table 4.

5.3 Numerical experiences

Here we summarize the experiences obtained from our Matlab implementation of the proposed methods. Other implementations may result in the same tendencies with different numbers.

From numerical point of view the most crucial model feature is the order

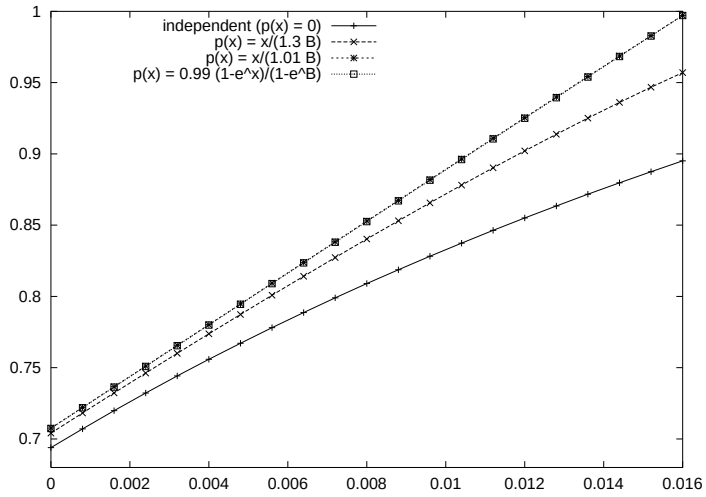

Figure 6: Distribution of the WEB server buffer content

of magnitude of $\mathbf{W}(b)$. When $\mathbf{W}(b)$ is too large ($> 10^{20}$) the linear system (which is partially composed by the elements of $\mathbf{W}(b)$) provides inaccurate result. To estimate the order of magnitude of $\mathbf{W}(b)$ in advance the ODE solution we used the following approximation. We calculated $\lambda_{max}(x)$ the eigenvalue of $\mathbf{B}(x)$ with the maximal real part at some points (x_i) in the $(0, B)$ range and used $\max_i e^{\lambda_{max}(x_i)B}$ to approximate the order of magnitude of $\mathbf{W}(b)$. According to our experiences, below the $\max_i \lambda_{max}(x_i)B = 35$ limit the applied numerical procedures are stable and rather insensitive to the precision requirement of the ODE solver (in a wide range of ϵ) and the applied dynamic step size control policy.

The fact that we do not have any other analysis methods for the numerical solution of the studied systems than the ones presented in this paper made difficult to validate our results and to judge the precision of the methods. We can compare our numerical procedures with "independent" solution methods only in the special case when $\mathbf{Q}(x)$ and $\mathbf{R}(x)$ are constant. In these cases we used the method of matrix exponentiation presented in [6] and the spectral decomposition method presented in [9] for comparison. We experienced a perfect matching of the results (at least to the first 4 meaningful digits) in the evaluated cases with all the 3 ODE solutions and a wide range of reasonable precision requirements.

In the majority of our evaluated examples, especially with larger state space, the absolute value of $\lambda_{max}(x)$ (≥ 0) and $\lambda_{min}(x)$ (the eigenvalue with the minimal real part, (≤ 0)) were close to each other (with less than 10% difference). Even though, in those cases when $|\lambda_{max}(x)| >> |\lambda_{min}(x)|$ we can

	TR	RK4	TR-BDF
inversion	1	0	2
multiplication	2	4	4
summation	1	7	2

Table 5: Number of matrix operations

extend the applicability of the numerical method by the $y = B - x$ variable substitution, which replaces the role of the upper and the lower bounds and reverses the direction of the continuous access. By this substitution the role of $\lambda_{max}(x)$ and $\lambda_{min}(x)$ interchanges as well and we get a lower $\lambda_{max}(y)B$ product.

The computational complexity of the proposed ODE solution procedures depends on the complexity of an elementary ODE solution step and the number of the elementary steps. The complexity of the elementary steps is summarized in Table 5 in term of $|\mathcal{S}| \times |\mathcal{S}|$ matrix operation. The calculation of the LTE further increase the complexity of an elementary step, e.g., the calculation of LTE_c doubles the complexity of the elementary step. The number of elementary step is predictable (based on the number of discrete states, the bound of the fluid place (B), and the step size) with fix step size, instead, it is a function of the model parameters with adaptive step size control. The computational complexity of the series expansion method depends on the number of calculated coefficient matrices, which depends on the model parameters in an unpredictable way. The complexity of the final step of the analysis, the solution of the linear system, is equivalent with the inversion of a $3|\mathcal{S}| \times 3|\mathcal{S}|$ matrix.

6 Conclusions

The stationary analysis of FSPNs with single finite fluid place and with mutually dependent continuous and fluid parts is considered in this paper. The dependence of the transition rate and the fluid drift on the fluid level complicates the analytical description of the underlying stochastic process a bit and, as a consequence, prevents the application of previously proposed solution techniques. Based on the analytical description of the model behaviour we presented numerical analysis techniques using series expansion and ODE solution methods applicable for the solution of the considered system.

We evaluated the FSPN models of two telecommunication systems, a subnetwork with cable model and a parallel WEB server, using the proposed numerical methods. The experienced numerical properties of the solution

methods are summarized together with the results of the studied examples.

References

[1] H. Alla and R. David. Continuous and Hybrid Petri Nets. *Journal of Systems Circuits and Computers*, 8(1), Feb 1998.

[2] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic Theory of a Data-Handling System. *Bell Sys. Thech. J*, 61(8):1871–1894, Oct 1982.

[3] D.-Y. Chen, Y. Hong, and K. S. Trivedi. Second order stochastic fluid flow models with fluid dependent flow rates. *Performance Evaluation*, 49(1-4):341–358, 2002.

[4] A. I. Elwalid and D. Mitra. Statistical Multiplexing with Loss Priorities in Rate-Based Congestion Control of High-Speed Networks. *IEEE Transaction on Communications*, 42(11):2989–3002, November 1994.

[5] R. German. *Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets*. John Wiley & Sons, 2000.

[6] M. Gribaudo and R. German. Numerical solution of bounded fluid models using matrix exponentiation. In *Proc. 11th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB)*, Aachen, Germany, Sep 2001. VDE Verlag.

[7] M. Gribaudo, M. Sereno, A. Bobbio, and A. Horvath. Fluid Stochastic Petri Nets augmented with Flush-out arcs: Modelling and Analysis. *Discrete Event Dynamic Systems*, 11(1 & 2), 2001.

[8] G. Horton, V. G. Kulkarni, D. M. Nicol, and K. S. Trivedi. Fluid stochastic Petri Nets: Theory, Application, and Solution Techniques. *European Journal of Operations Research*, 105(1):184–201, Feb 1998.

[9] V. G. Kulkarni. Fluid models for single buffer systems. In J. H. Dshalalow, editor, *Models and Applications in Science and Engineering*, Frontiers in Queueing, pages 321–338. CRC Press, 1997.

[10] L. Lapidus and J. Seinfeld. *Numerical solution of ordinary differential equations*. Academic Press, 1971.

[11] A. Reibman, R. Smith, and K.S. Trivedi. Markov and Markov reward model transient analysis: an overview of numerical approaches. *European Journal of Operational Research*, 40:257–267, 1989.

[12] A. Reibman and K.S. Trivedi. Numerical transient analysis of Markov models. *Computers and Operations Research*, 15:19–36, 1988.

[13] K. Trivedi and V. Kulkarni. FSPNs: Fluid Stochastic Petri nets. In *Application and Theory of Petri Nets 1993, Proc.* 14^{th} *Intern. Conference*, LNCS, Chicago, USA, June 1993. Springer Verlag.

[14] K. Wolter. Second order fluid stochastic petri nets: an extension of gspns for approximate and continuous modelling. In *Proc. of World Congress on System Simulation*, pages 328–332, Singapore, Sep 1997.