Modeling and analysis of threshold queues with hysteresis using stochastic Petri nets: the monoclass case

Bruno Tuffin and Louis-Marie Le Ny
IRISA
Campus universitaire de Beaulieu
35042 Rennes cedex, France
E-mail:{btuffin,leny}@irisa.fr

Abstract

We consider a monoclass threshold queue with hysteresis. Many applications of such a system can be found in the transport protocols of communication networks. Thresholds help to smooth the variations of delay and throughput, and hysteresis limits the number of (expensive) switches of the control mechanism. The queues are modeled and analysed by means of Stochastic Petri Nets (SPNs), a powerful paradigm to represent complex systems. The representation is shown to be very simple and can aid in the study and configuration of the queue to conform to design specifications. Moreover, Fluid Stochastic Petri Nets (FSPNs) are used as an approximation which might be helpful to analyse high speed networks. Numerical illustrations are given in order to validate the choice of threshold queues with hysteresis and their representation by SPNs and FSPNs.

1 Introduction

We consider in this paper threshold queues with hysteresis and a single class of customers. These queues are controled by a sequence of forward thresholds and a sequence of reverse thresholds such that, when the content of the queue exceeds a forward threshold, the arrivals are slowed down and/or the services are speeded up. In the same way, when a reverse threshold is reached, the arrivals are speeded up and/or the services are slowed. Delay may happen for example when the sources are far away and cannot be informed immediately of the decision to slow down or speed up the arrival process or when some management is necessary before modifying the parameter of the queue services (such as the time for an additional server to arrive).

An example of threshold queue with hysteresis is the K-server queue where additional servers arrive when the buffer content exceeds the forward thresholds and leave when the buffer content decrease to reverse thresholds. This kind of queueing system has many applications in the dynamic control of computer systems and telecommunication networks. An example quoted in [13] is the case where several transport connections are multiplexed over a single network connection. In order to control the throughput and delay, the transport protocol can open (or close) further network connections. Another example in [13] is the telephony directory assistance where the number of attendants depends on the number of calls. In cable TV traffic management, control is also a key issue. Indeed, empty buffers can lead to an image break and full buffers occasion a loss of data [15].

Markovian threshold-based queueing systems have been extensively studied in the literature. In [14], Larsen and Agrawala conjecture that a threshold-policy optimizes system performance. Lin and Kumar [19] show that this conjecture is true when minimizing the mean sojourn time of customers in the system. In their model, the maximum number of servers is 2. Moreover, they say that the optimality result is difficult to generalize to the K-server queue with $K \geq 3$. Le Ny [15] obtains a closed form of the steady-state probabilities of a queue with two thresholds and hysteresis. Morrison [21] studies a two server queue with a threshold and focuses on the equilibrium density of the waiting time. In the same time, Li [18] studies the overload control of a system with one forward and one reverse threshold in the case where the queue is modeled by a birth-death or a quasi-birth-death process. An algorithm to compute the steady-state probabilities is given. In one of the first works dealing with more than two thresholds [13], Ibe and Keilson give a closed form for the occupancy distribution of a homogeneous multi-server queue with hysteresis using the Green's function method when the servers are identical or when there are only two non-identical servers (in the Markovian case). Then Lui and Golubchik study in [20] a K-server threshold-based queueing system with hysteresis using the method of stochastic complement analysis for uncoupling Markov chains. They give an explicit algorithm for computing the steady-state probabilities in the heterogeneous case but no compact closed-form. An algorithm for the bulk arrival case is also provided. More recently, Le Ny and Tuffin [17] give a closed-form for the steady-state probabilities of a heterogeneous multi-server threshold queue with hysteresis using the cuts method, which presents the advantage to be very simple in comparison with other mathematical methods.

The literature shows that an analytic analysis quickly becomes tricky. The goal of this paper is to model and to analyse threshold queues with hysteresis by using Stochastic Petri Nets (SPNs) (see for instance [1]). Indeed, SPNs are proven to be a powerful representation of complex systems, so that the representation of threshold-based queues is very simple and convenient. In the same way, Fluid Stochastic Petri Nets (FSPNs) [4, 12, 24] is a hybrid extension of SPNs where fluid can model continuous components or approximate discrete states to prevent the explosion of the state-space.

Many software packages are available in order to translate the SPN model to a CTMC and to solve it, to analyse it also in the case where it is a semi-Markov process or to simulate it otherwise. In this paper, we are going to use SPNP (for *Stochastic Petri Net Package*) [3, 11], a software developed at Duke University. It presents the advantage to also simulate FSPNs and non-Markovian SPNs.

The layout of this paper is the following. Section 2 presents the systems we are going to model and analyse. Section 3 is devoted to a description of SPNs (in a very general setting) and FSPNs. Then the modeling of Markovian, non-Markovian, delay and Fluid threshold queues by SPNs and FSPNs are described respectively in Sections 4, 5, 6, 7. Numerical analysis is provided in Section 8. Finally, we give some conclusions and directions for future work in Section 9. The case of threshold queues with several classes of customers (see for instance [6]) is a big challenge for example for DiffServ, the service differentiation which will drive the future Internet, and is the subject of our next research. To our knowledge, this case has never been studied using SPNs, except in the case of polling systems [10] (but no hysteresis was included).

2 Model definition

We consider a system where the number of services and inter-arrival distributions are governed by a forward threshold sequence $(S_1, \cdots, S_{K-1})$ and a reverse threshold sequence $(s_1, \cdots, s_{K-1})$ such that, without loss of generality, $S_k < S_{k+1}$ and $s_k < s_{k+1}$ for all $1 \leq k \leq K-2$.

The system behaves as follows. When the number of customers in the queue goes beyond the forward threshold S_k ($1 \leq k \leq K-1$), service and inter-arrival distribution parameters are changed by slowing down the arrivals and/or speeding up the services in order to keep the buffer-content in a given range. In the same way, if the number of customers in the queue falls below s_k, the parameters are changed back to the previous values. In [13, 17, 20], the inter-arrivals and services follow exponential distributions such that thresholds do not affect inter-arrivals but only service rates by activating or removing additional servers.

Hysteresis comes from the fact that forward and reverse thresholds do not have the same values. Then, the number of costful switches (in management operations) is expected to decrease. We will investigate this point.

Inter-arrivals and services will first be assumed to be exponential. The thresholds will then drive the rate values. Even if this exponential assumption is valid for some applications such as telephone networks, many application areas need more general distributions. In the ATM and Internet networks for instance, services should be assumed to be determistic; moreover, arrivals often present self-similar or heavy-tailed behavior (see for instance [5]). We will also study non-Markovian threshold-queues in Section 5.

The delay between the decision to switch the parameters and its application will first be assumed to be null. Nevertheless, when we want to notify the sender to slow down or to speed up his flow of information, it might require some delay. Moreover, in the case of K-server systems [13, 17, 20], the time for the additional server to arrive and to be operational may be strictly positive. This will be investigated in Section 6.

3 Stochastic Petri Nets, Stochastic Reward Nets and Fluid Stochastic Petri Nets

Petri nets is a formal graph useful to represent systems that exhibit concurrency, synchronisation, sequencing and conflict. We describe here an extension of Stochastic Petri Nets (SPNs) that is actually a mixture of Stochastic Reward Nets and a kind of ESPNs (general distributions are possible). We will abusively call this model SPN. We will then define FSPNs.

An SPN is a bipartite directed graph composed of two kinds of nodes: places (represented by circles) and transitions (represented by bars for immediate transitions and rectangle for timed transitions). Directed arcs can connect places to transitions (they are then called input arcs) or transitions to places (they are then called output arcs). Places may contain tokens (represented by dots). The *marking* m of the SPN is defined by the number of tokens in each place. The initial marking is called m^0.

The dynamics of SPNs is the following. Each arc is given by its multiplicity (which may be marking-dependent). A transition is said to be *enabled* if each of its input places contains as many tokens as the multiplicity of the corresponding arc. Two other properties need to be verified for a transition to be enabled. First the *inhibitor arcs* (if any), directed from a place to a transition and represented by an arc terminated by a small circle, set the condition that each inhibitor place must contain a number of tokens less than the (possibly marking-dependent) multiplicity of the corresponding arc. The second required property is that the (marking-dependent) guard function of the transition (if any) is valued to 1. Guard function is a generalization of inhibitor arcs, saying whether or not a transition is enabled in the current marking. Nevertheless we still also consider inhibitor arcs for their graphical usefulness and for sake of generality. Among all enabled transitions, the one with the smallest firing time is said to *fire*. The firing times of transitions follow random distributions which can be general (i.e., not only exponential). When it fires, the transition removes a number of tokens from each of its input places equal to the multiplicity of the corresponding arc and it deposits in each of its output place a number of tokens equal to the multiplicity of the corresponding arc, leading then to a new marking. When several transitions have the same firing time (which can occur for instance when several immediate transitions are concurrent), a weight function is used to choose the one which will fire (each transition is associated to a probability to fire equal to the individual weight divided by the sum of weights of possible transitions). When a transition fires, some other transitions can become disabled. When they become enabled again, there are several possibilities for static resampling policy: PRI (preemptive repeat identical), PRD (preemptive repeat different) and PRS (preemptive resume). In the same way, each transition may be affected by the firing of another transition, even if the considered transition remains enabled. PRI, PRS and PRD are also possible for this affecting policy.

An FSPN [4, 12, 24] is an extension of an SPN where some places contain fluid which can flow through transitions following given rules. Fluid places are represented by two concentric circles and fluid arcs by directed thick arcs. The dynamics of an FSPN is then governed by discrete events (firing of transitions modifying the discrete and fluid marking) and continuous flows in fluid places between these firings as described in the following. When the transitions related to fluid arcs (input or output) are enabled, fluid flows along the arc with the constraint that the fluid level remains between 0 and a prespecified upper-bound. Like in [4], we assume that the fluid flow rate is at most linear: for each fluid place p, the fluid level $x_p(t)$ changes according to the equation [4]

$$\frac{dx_p(t)}{dt} = a(m)x_p(t) + b(m)$$

where m is the current discrete marking and a and b are some functions of the discrete marking. Moreover, when a transition fires, some fluid can immediately be put in or removed from fluid places. The input and output arcs related to these marking-dependent fluid impulse of the connecting transitions and fluid places are represented by thin directed arcs like the arcs connecting transitions and discrete places. Of course, the initial marking x^0 in fluid places, needs also to be specified.

4 Markovian queues

We consider first the case of a Markovian threshold-queue with hysteresis and capacity C. We have a sequence $S_1 < \cdots < S_{K-1}$ of forward thresholds and a sequence $s_1 < \cdots < s_{K-1}$ of reverse thresholds (see Section 2). Figure 1 represents the SPN related to this single queue, with guard functions described in Table 1 and rate functions described in Table 2. In Figure 1, Place $Psyst$

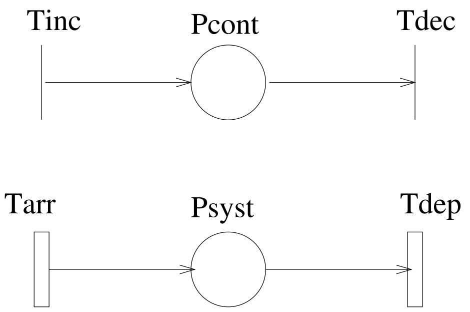

Figure 1. Representation of a threshold queue with hysteresis by an SPN. Guard and rate functions are respectively given in Table 1 and 2

Transition	guard
$Tinc$	$\#Psyst > S_{\#Pcont}$ && $\#Pcont < K$
$Tdec$	$\#Psyst \leq s_{\#Pcont-1}$
$Tarr$	$\#Psyst < C$

Table 1. Guard functions of the SPN of Figure 1

Transition	Rate
$Tarr$	$\lambda(\#Pcont)$
$Tdep$	$\mu(\#Pcont)$

Table 2. Rate functions of the SPN of Figure 1

gives the total number of customers (or packets in networking terminology) in the queue. Place $Pcont$ gives the rate control state, which changes using immediate transitions when thresholds are reached (events given by the guard functions of transitions $Tinc$ and $Tdec$). In this case, the rate modifications are instantaneous. The arrival rates are $\lambda(\#Pcont)$ and the service rates $\mu(\#Pcont)$. They are such that $\lambda(1) \geq \cdots \geq \lambda(K)$ and $\mu(1) \leq \cdots \leq \mu(K)$.

5 Non-Markovian queues

In the non-Markovian case, marking-dependent exponential distributions of transitions $Tarr$ and $Tdep$ are replaced by marking-dependent general distributions. Two cases are then possible: either the sampled time of the next arrival/service is kept and the next will be sampled with new parameters, or the time is resampled immediately with new parameters. Even if the model of Figure 1 (with general distribution) can be used to show the behavior of the system, it does not allow to compute loss probabilities because we do not have the PASTA property like with Markovian models. Thus we slightly modify the SPN of Figure 1 and replace it by the one of Figure 2 where arrivals are not blocked anymore by transition $Tarr$ guard function if the buffer is full. When a new customer arrives, he goes in Place $Pentrance$ and immediate transition $Tloss$ (resp. Tok) fires if he is lost, i.e., if the buffer is full, (resp. accepted). The previous model is then supplemented by the guard function given in Table 3 (and no more guard function is put on transition $Tarr$). The loss probability is then given by the ratio of throughputs TH_{Tloss}/TH_{Tarr}.

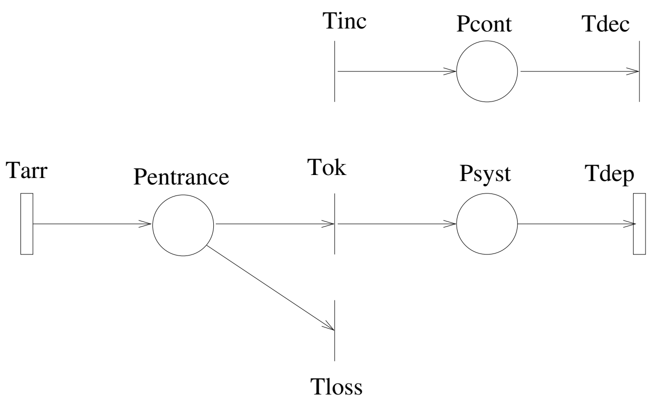

Figure 2. Representation of a threshold queue with hysteresis and general probability distributions by an SPN.

Transition	guard
$Tloss$	$\#Psyst = C$
Tok	$\#Psyst < C$

Table 3. Supplementary Guard functions of the SPN of Figure 2

6 Queues with delay

An important way to generalize the previous system is when the decision to slow down the arrivals or to speed up the services are not always immediately effective. Some delay needs then to be introduced. Figure 3 represents the SPN related to a queue with delay, with guard functions described in Table 4 and rate functions described in Table 5.

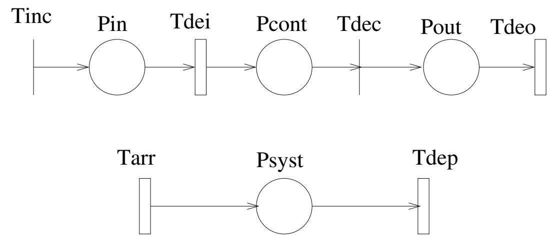

Figure 3. Representation of a threshold queue with hysteresis and delay by an SPN. Guard and rate functions are respectively given in Table 4 and 5.

In Figure 3, in addition to the notations used in Figure 1, the

Transition	Rate
$Tarr$	$\lambda(\#Pcont + \#Pout)$
$Tdep$	$\mu(\#Pcont + \#Pout)$
$Tdei$	λ_{dei}
$Tdeo$	λ_{deo}

Table 5. Rate functions of the SPN of Figure 3

number of tokens in place Pin (resp. place $Pout$) is for the number of thresholds reached forward (resp. reverse) and for which the changes are still not in effect due to delay. The rates λ_{dei} and λ_{deo} of transitions $Tdei$ and $Tdeo$, representing the rates at which the control decisions are effective, are here assumed to be constant. Nevertheless, we can easily make them dependent of the state of the system. Of course, exponential distributions can here also be changed to other distributions like in the previous subsection. Note also that

Transition	guard
$Tinc$	$\#Psyst > S_{\#Pin+\#Pcont}$ && $\#Pin + \#Pcont + \#Pout < K$
$Tdec$	$\#Psyst \leq s_{\#Pin+\#Pcont-1}$ && $\#Pin + \#Pcont > 0$
$Tarr$	$\#Psyst < C$
$Tdep$	$\#Pcont + \#Pout > 0$

Table 4. Guard functions of the SPN of Figure 3

the arrival and service rates depend on $\#Pcont + \#Pout$ and not only on $\#Pcont$ because the number of tokens in place $Pout$ are for the upper-levels still not cancelled.

Other cases with delay can also be considered. For example, we may assume that modifications of rates are effective immediately when reverse thresholds are reached but not when forward thresholds are reached. Moreover, it might be possible to cancel a modification call still not applied rather than canceling an applied one. This can happen when the queue is a multi-server queue and additional servers are added each time forward thresholds are hit and removed each time reverse thresholds are hit (and where arrival rates are constant, which is not assumed in Table 7 for generality sake). This has been studied in [16] when $K = 2$. Removing a server can be assumed to be immediate but adding one can need some time. Figure 4 represents the SPN related to this queue, with guard functions described in Table 6 and rate functions described in Table 7.

In Figure 4, transition $T1+$ is then added such that, when

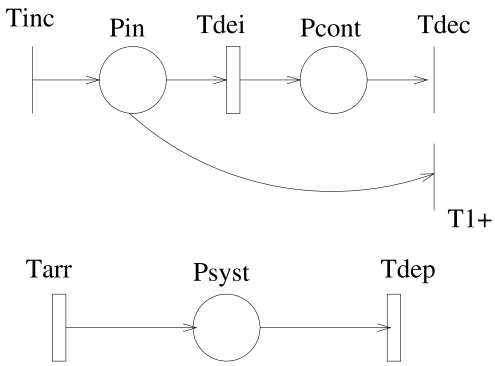

Figure 4. Representation of a threshold queue with hysteresis and delay by an SPN. Guard and rate functions are respectively given in Table 6 and 7.

a reverse threshold is reached and a modification call due to reaching a forward threshold is still not effective (i.e., $\#Pin > 0$), then this modification call is immedialety cancelled by firing transition $T1+$. Other modifications of this model with delay can easily be devised.

Transition	Rate
$Tarr$	$\lambda(\#Pcont)$
$Tdep$	$\mu(\#Pcont)$
$Tdei$	λ_{dei}

Table 7. Rate functions of the SPN of Figure 4

7 Fluid queues

When the queue is dealing with a lot of informations (i.e., a lot of tokens), it is relevant to consider the arrivals and services like a continuous flow. This approximation can simplify the analysis because, in the discrete analysis, the state space can become very large and a discrete-event simulation could require a huge number of operations to simulate even very short intervals of time. Even if second order FSPNs [27] could represent a better approximation, we will consider the first order FSPNs that we have described in Section 3 because they are well matched to the ATM environment [6]. Figure 5 represents the (deterministic) FSPN related to the single queue. It is aimed to be used in the following as a component of a stochastic system where the fluid arrival rates are driven by a random process. Place $Psyst$

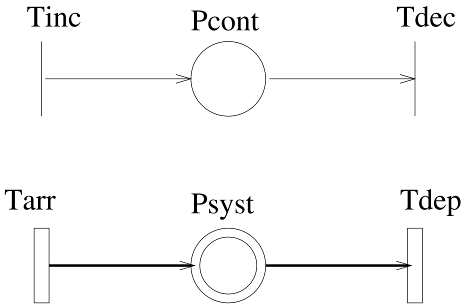

Figure 5. Representation of a threshold queue with hysteresis by an FSPN.

is then continuous. Delay can be added exactly like in the discrete case without any difficulty. Using fluid places, the model can be seen at a different scale. Whereas in previous

Transition	guard
$Tinc$	$\#Psyst > S_{\#Pin+\#Pcont}$ && $\#Pin + \#Pcont < K$
$Tdec$	$\#Psyst \leq s_{\#Pin+\#Pcont-1}$ && $\#Pin + \#Pcont > 0$ && $\#Pin = 0$
$T1+$	$\#Psyst \leq s_{\#Pin+\#Pcont-1}$ && $\#Pin + \#Pcont > 0$ && $\#Pin > 0$
$Tarr$	$\#Psyst < C$
$Tdep$	$\#Pcont > 0$

Table 6. Guard functions of the SPN of Figure 4

Transition	guard
$Tinc$	$\#Psyst > S_{\#Pcont}$ && $\#Pcont < K$ && $r(Tarr) > r(Tdep)$
$Tdec$	$\#Psyst \leq s_{\#Pcont-1}$ && $r(Tarr) < r(Tdep)$

Table 8. Guard functions of the SPN of Figure 6. Fluid rates $r(Tarr)$ and $r(Tdep)$ are given in Table 9

subsections we were describing the queue at the cell-scale, we are now able to describe the system at the burst-scale. Figure 6 describes the same fluid queue where the input fluid rate is driven by the state of another stochastic process, a set of N On/Off sources. This can be a representation of a queue where the N sources are represented at the cell-scale by interrupted Poisson processes or other interrupted processes. The guard functions are described in Table 8 and fluid rate functions are described in Table 9. The fluid input rate is modified by the start/end of a burst of an On-Off source. Our only restriction here is that the flow rates are linear as described in Section 3. Guard functions are the same than in the discrete case, with, of course, the difference that all values are real numbers instead of integers. In the same way, the rates are no more representing the rates at which customers (i.e., tokens) are arriving at the queue, but the continuous flow rates arriving and leaving the queue. Moreover, as indicated in Table 9, arrivals cannot exceed services when the buffer is full and services cannot exceed arrivals when it is empty. In this example, we have considered a K-server queue where the number of active servers is given by place $Pcont$.

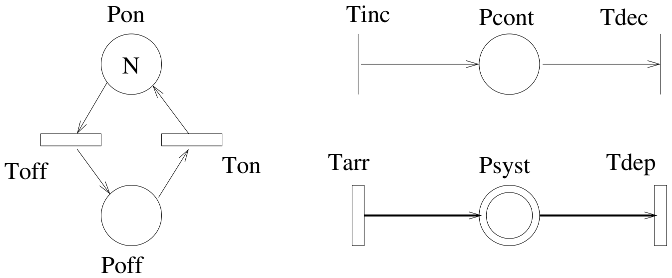

Figure 6. Representation of a threshold queue with hysteresis by an FSPN driven by On-Off sources.

8 Analysis

8.1 Solution methods

SPNs and FSPNs are implemented in Stochastic Petri Net Package (SPNP) [3, 11], where they are described in CSPL, a library of the ANSI C language. The capacity modeling in SPNP is very powerful and is the one described in Section 3.

The solution methods we are using are the following and are implemented in SPNP. Markovian SPNs of the previous sections are solved analytically-numerically. The reachability graph is first generated from the SPN and then converted to a Markov chain (or more exactly to a Markov reward model) which is solved numerically. For steady-state evaluation, we will choose Steady-State SOR (Successive Overrelaxation [23]), the fastest method, but Steady-State Gauss-Seidel [23], and Steady-State Power method [26] are also avaible in SPNP and are used when SSOR does not converge. For transient-state solution of the CTMC, we use standard uniformization or uniformization using the Fox and Glynn method for computing the Poisson probabilities.

On the other hand, non-Markovian SPNs and FSPNs are solved by simulation. Steady-state simulation will be performed using simulation with batches or regenerative simulation [7] (in this last case, our SPN needs to be in the class of MRSPNs [2]). Transient simulation will be performed by crude independent replications. However, for information, there are other simulation methods available in SPNP: importance splitting techniques (Restart and splitting) [25], importance sampling [7, 8, 22], regenerative simulation with importance sampling [9, 22], all suitable to estimate rare events.

Transition	Fluid rate $r(\text{Transition})$
$Tarr$	$\lambda \# Pon\ 1_{[\# Psyst<C]} + \min(\lambda \# Pon, \mu \# Pcont) 1_{[\# Psyst=C]}$
$Tdep$	$\mu \# Pcont\ 1_{[\# Psyst>0]} + \min(\lambda \# Pon, \mu \# Pcont)) 1_{[\# Psyst=0]}$

Table 9. Rate functions of the SPN of Figure 6

8.2 Numerical results

We consider as a first illustration a Markovian queue with one forward and one backward threshold and a finite buffer of capacity $C = 64$. We consider a threshold queue with hysteresis as described in Figure 1 that we compare with a threshold queue without hysteresis as well as with a queue without thresholds. We assume that arrivals follow a Poisson process with constant rate 2.0 and that (for the threshold queues) a server is added/removed each time thresholds are reached. For the threshold queues, we assume that each server provides a service following an exponential distribution with rate 1.5. To obtain relevant comparisons, we assume that the mean service rate at the queue without threshold is the same than the mean available service rate at threshold queues $1.5P(1 \text{ server is working} \mid \text{at least 1 is working}) + 3P(2 \text{ servers working} \mid \text{at least 1 is working})$, i.e., about 2.0. The threshold queue without hysteresis is assumed to have only one threshold with value $s_1 = S_1 = 32$. The threshold queue with hysteresis is assumed to have one forward threshold of value $S_1 = 44$ and one reverse threshold of value $s_1 = 20$. It remains now to define the cost (or reward) function on which we will compare the results. This function is decomposed in two components, one is the user cost function, representing the effects perceived by the customer, and the other one is the network cost function which encompass the management costs of the queue. We consider that the quality of service required is to have very few losses but also no starvations. It is for instance the case in video transmission where packet losses should be rare and where we do not expect any image interruption. Define α_u (resp. α_d) as the mean number of server activation (resp. disactivation) per unit time (note that $\alpha_u = \alpha_d$ in steady-state). Define also an hyperbolic function f depending on the number m of customers in the queue,

$$f(m) = \begin{cases} \frac{64}{3m+16} & \text{if } m \leq 16, \\ \frac{64}{3(64-m)+16} & \text{if } m \geq 48, \\ 1 & elsewhere. \end{cases}$$

The user cost function (representing what is felt by the user application) is

$$C_{user} = c_1 P_{loss} + c_2 P(\text{queue empty}) + c_3 f(m)$$

and the network cost function (representing the additional costs to improve the QoS without any perception for the user) is

$$C_{net} = c_4 \mu_1 P(\text{server 1 used}) + c_5 \mu_2 P(\text{server 2 used}) + c_6 \alpha_u + c_7 \alpha_d.$$

For our example, we arbitrarily choose the weights to be $c_1 = 20$, $c_2 = 15$, $c_3 = 0.5$, $c_4 = 0.05$, $c_5 = 0.05$, $c_6 = 0.4$ and $c_7 = 0.4$. By using the hyperbolic function $f(m)$, we introduce a larger cost when having a small or a big number of customers in the queue. Moreover we add other costs when the queue is empty or when there is a loss.

Using SSOR method to analyse this model, we obtain (in about 0.1 second) the results displayed in Table 10. The results illustrate the fact that introducing thresholds reduces the user cost and the total cost. Of course, it increases the network cost (because management is more complicate) but at a much smaller expense than the benefit on user cost. When comparing the results with or without hysteresis, we can check that there is a small difference in user cost but that the network cost is significantly reduced when using hysteresis, reducing by then the overall cost.

Now, we consider a non-Markovian queue. We consider the model of Figure 2 where the inter-arrivals follow a Pareto distribution with smallest value $0.05/2.1$ and shape parameter 1.05. This kind of distribution is often used to model Web traffic [5]. The service duration (at transition T_{dep}) is assumed to be $2/3$ (constant) for each server. We assume here that there is a synchronisation between servers: when a new server is called (i.e., when a threshold is reached), this new server is beginning its service only when the working ones finish their own service. This is modeled by the PRS resampling policy. Using this, the service parameter of transition T_{dep} remains constant at $2/3$, but the multiplicity of the arc form place P_{syst} to transition T_{dep} is made equal to the number of tokens in place P_{cont}. We use the same cost function than for the Markovian net, where the rates μ_i are replaced by the inverse of expectations of the transition firing times. The results are displayed in Table 11. For comparison sake, we have taken 0.5283 as the constant service times for the model without thresholds (which is about the mean service for the cases with thresholds). Table 11 gives 95% confidence intervals obtained by the batching simulation method. Here again, thresholds and hysteresis are shown to be useful.

We now consider the queues with delay. We compute the same steady-state costs than for the Markovian queue, with the same parameters (in order to compare with the results of Table 10). The only difference is that we have introduced

	No thresh.	Thresh. but no hyster.	Thres. and hyster.
User cost	1.27127166362	0.500998438111	0.505629885157
Network cost	0.0984615384615	0.366680228407	0.110668898195
Total cost	1.36973320208	0.867678666518	0.616298783352

Table 10. Results for a Markovian queue for (1) a queue without thresholds, (2) a threshold-queue without hysteresis and (3) a threshold queue with hysteresis.

	No thresh.	Thresh. but no hyster.	Thres. and hyster.
User cost	[2.6490e+01,2.7037e+01]	[1.6287e+01,1.6557e+01]	[1.5836e+01,1.6094e+01]
Network cost	[6.7224e-02,6.8804e-02]	[1.7378e-01,1.7910e-01]	[1.5873e-01,1.6359e-01]
Total cost	[2.6557e+01,2.7106e+01]	[1.6464e+01,1.6733e+01]	[1.5997e+01,1.6255e+01]

Table 11. Results for the non-Markovian queue for (1) a queue without thresholds, (2) a threshold-queue without hysteresis and (3) a threshold queue with hysteresis.

	Thresh. but no hyster.	Thres. and hyster.
User cost	0.50105289419	0.505922354577
Network cost	0.29188117199	0.110795248309
Total cost	0.792934066183	0.616717602886

Table 12. Results for the queue with delay at arrivals and departures for (1) a threshold-queue without hysteresis, (2) a threshold-queue with hysteresis.

	Thresh. but no hyster.	Thres. and hyster.
User cost	0.500938287674	0.50570389706
Network cost	0.36931432904	0.110626328949
Total cost	0.870252616718	0.616330226009

Table 13. Results for the queue with delay only at arrivals for (1) a threshold-queue without hysteresis, (2) a threshold-queue with hysteresis.

the delays for the servers to arrive and to leave, which are not assumed to be instantaneous but to follow an exponential distribution with rate 5.0. The results for the queue with delay at both arrivals and departures (Figure 3) are given in Table 12 and for the queue with delay only at arrivals (Figure 4) in Table 13. It shows that (except in one case) delay slightly increases the costs, but thresholds and hysteresis are still efficient.

Consider now a fluid queue with hysteresis fed by 10 independent On-Off sources as described in Figure 6. The guards and fluid rates are the ones described in Tables 8 and 9. We take here $\lambda = 0.004$ and $\mu = 0.011$. On and Off durations at each source are assumed to follow an exponential distribution with rate 1.0 (but we can easily devise more general cases). We take here $S_1 = 1.5$, $s_1 = 0.5$ and $C = 2$. Table 14 compares the results obtained for the fluid model with a discrete model where one fluid unit represents 100 tokens (or discrete units). The fluid flow rates are then changed to rates of exponential distributions, but multiplied by 100 in order to get the same mean flow. For both fluid and discrete simulations, we use the batch method with 5000 blocks. The same kind of cost function in the fluid case than in the discrete case, but the definition of some costs are modified to fit the continuous case as described now. Consider first the network cost. The switching cost (to release or to call a server) remain unchanged. The other component of the network cost, the server use cost $c_4\mu_1 P(\text{server 1 used}) + c_5\mu_2 P(\text{server 2 used})$, is replaced by $c_4\mu * 100 * E(\#Pcont)$ which represents the mean fluid rate (multiplied by 100 to be compared to the discrete case). Considering now the user cost, the cost of having an empty queue and the loss cost are not changed. The loss probability is computed as the ratio of the mean feeding fluid rate $\lambda E(\#Pon)$ minus the mean of $r(Tarr)$ on $\lambda E(\#Pon)$ (and the empty probability is computed using the same principle). The hyperbolic cost is the one which has to undergo numerous adaptations. We precomputed the different possibilities of integrals of $f(\#Psyst)$ between two instants of time and added them to the cost at each step of the discrete event simulation. The hyperbolic curves are on intervals $[0, 0.4]$ and $[1.6, 2]$ instead of $[0, 40]$ and $[160, 200]$. The values are shown to be close, so that the approximation is quite good. Moreover, even with this small order of magnitude, we can note a substantial reduction in simulation time (and even a variance reduction in this case). We have considered here a small example in order to be able to compare the results of all the methods. Nevertheless, if the state-space increases, the generation of the reachability graph becomes impossible and simulation is the only solution method. Moreover, if inter-arrival and service rates increases (as it does in high-speed network), the simulation of the discrete SPN is becoming very long because a lot of events need to be generated to simulate the evolution of the system even during short periods of time. For instance, if we multiply by 100 the arrival and service rates in the previous example, then the ratio of the time to perform the fluid simulation divided by the time to perform the discrete one (i.e., to simulate the same period of time) is becoming 0.027. Then, when considering high-speed networks, the fluid model has to be considered.

9 Conclusion

Modeling and analysis of threshold-based queues using SPNs and FSPNs is then very convenient and tools (like SPNP for instance) allow to do it quickly. This can aid in the study and configuration of the queue to conform to design specifications. In this paper, we have been able to analyse the monoclass queue when it is Markovian, non-Markovian, with or without delay to supply the service resources. In each case, we have shown the interest of thresholds (with respect to a cost function) as well as the interest of hysteresis. We have also used a fluid model as an approximation of a node of a high-speed network and illustrated the gain in time of this approximation.

As directions for future work, we can enhance:

- Extend our modeling to a network of queues.
- Study the multi-class case to differentiate services as expected in next Internet generation. We have begun such a study.
- Test and devise protocols where the nodes of the network notify to the sources when and how to speed up or slow down their transmission rates, particularly when dealing with a network of queues.

References

[1] M. Ajmone Marsan, G. Conte, and G. Balbo. A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM Transactions on Computer Systems*, 2(2):93–122, May 1984.

[2] H. Choi, V. Kulkarni, and K. Trivedi. Markov regenerative stochastic petri nets. *Performance Evaluation*, 20(1-3):337–357, 1993.

[3] G. Ciardo, J. Muppala, and K. Trivedi. SPNP: Stochastic Petri net Package. In *Proc. Third International Workshop on Petri Nets and Performance Models, PNPM'89*, pages 142–151, 1989.

[4] G. Ciardo, D. Nicol, and K. Trivedi. Discrete-Event Simulation of Fluid Stochastic Petri-Nets. *IEEE Transactions on Software Engineering*, 25(2):207–217, 1999.

[5] M. Crovella. Performance Characteristics of the World Wide Web. In G. H. et al., editor, *Performance Evaluation*, volume 1769 of *Lecture Notes in Computer Science*, pages 219–232. Springer-Verlag, 2000.

Model type	Fluid, simulation	Discrete, SSOR	Discrete, simulation
User cost	[4.9991e-01,5.0019e-01]	0.509297082031	[5.0705e-01,5.2447e-01]
Network cost	[1.0082e-01,1.0183e-01]	0.100760232542	[9.9290e-02,1.0099e-01]
Total cost	[6.0086e-01,6.0190e-01]	0.610057314573	[6.0714e-01,6.2467e-01]
Comp. time (sec.)	91.2	74.22	129.35

Table 14. Results for the fluid queue.

[6] A. Elwalid and D. Mitra. Statistical Multiplexing with Loss Priorities in Rate-Based Congestion Congestion of High-Speed Networks. *IEEE Transactions on Communication*, 42(11):2989–3002, 1994.

[7] G. Fishman. *Monte Carlo: Concepts, Algorithms and Applications*. Springer-Verlag, 1997.

[8] P. W. Glynn and D. L. Iglehart. Importance Sampling for Stochastic Simulations. *Management Science*, 35(11):1367–1392, November 1989.

[9] A. Goyal, P. Shahabuddin, P. Heidelberger, V. Nicola, and P. Glynn. A Unified Framework for Simulating Markovian Models of Highly Dependable Systems. *IEEE Transactions on Computers*, 41(1):36–51, January 1992.

[10] B. Haverkort. *Performance of Computer Communication Systems*. John Wiley and Sons, 1998.

[11] C. Hirel, B. Tuffin, and K. Trivedi. SPNP Version 6.0. In B. Haverkort, H. Bohnenkamp, and C. Smith, editors, *Computer performance evaluation: Modelling tools and techniques; 11th International Conference; TOOLS 2000, Schaumburg, Il., USA*, volume 1786 of *Lecture Notes in Computer Science*, pages 354–357. Springer Verlag, 2000.

[12] G. Horton, V. Kulkarni, D. Nicol, and K. Trivedi. Fluid Stochastic Petri nets: Theory, Application and Solution. *European Journal of Operational Research*, 105:184–201, 1998.

[13] O. Ibe and J. Keilson. Multi-server threshold queues with hysteresis. *Performance Evaluation*, 21:185–213, 1995.

[14] R. Larsen and A. Agrawala. Control of a Heterogeneous Two-Server Exponential Queueing System. *IEEE Transactions on Software Engineering*, SE-9(4):522–526, 1983.

[15] L.-M. Le Ny. Probabilité stationnaire d'une file d'attente à 2 seuils. Technical Report 160, IRISA, 1987.

[16] L.-M. Le Ny. Exact analysis of a threshold-based queue with hysteresis and a delayed additional server. In *Proceedings of the 9th International Conference on Telecommunication Systems*, pages 387–398, Mar. 2001.

[17] L.-M. Le Ny and B. Tuffin. A simple analysis of heterogeneous multi-server threshold queues with hysteresis. Technical Report 1333, IRISA, 2000.

[18] S. Li. Overload Control in a Finite Message Storage Buffer. *IEEE Transactions on Communications*, 37(12), December 1989.

[19] W. Lin and P. Kumar. Optimal Control of a Queueing System with Two Heterogeneous Servers. *IEEE Transactions on Automatic Control*, AC-29(8):696–703, 1984.

[20] J. Lui and L. Golubchik. Stochastic complement analysis of multi-server threshold queues with hysteresis. *Performance Evaluation*, 35:185–213, 1999.

[21] J. Morrison. Two-server queue with one server idle below a threshold. *Queueing Systems: Theory and Applications*, 7:325–336, 1990.

[22] V. F. Nicola, M. K. Nakayama, P. Heidelberger, and A. Goyal. Fast Simulation of Highly Dependable Systems with General Failure and Repair Processes. *IEEE Transactions on Computers*, 42(12):1440–1452, December 1993.

[23] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[24] K. S. Trivedi and V. G. Kulkarni. FSPNs: Fluid Stochistic Petri Nets. In *14th International Conference on Applications and Theory of Petri Nets*, pages 24–31, 1993.

[25] B. Tuffin and K. Trivedi. Implementation of importance splitting techniques in stochastic petri net package. In B. Haverkort, H. Bohnenkamp, and C. Smith, editors, *Computer performance evaluation: Modelling tools and techniques; 11th International Conference; TOOLS 2000, Schaumburg, Il., USA*, volume 1786 of *Lecture Notes in Computer Science*, pages 216–229. Springer Verlag, 2000.

[26] W. B. van den Hout. *The Power-Series Algorithm: A Numerical Approach to Markov Processes*. Tilburg University, 1996.

[27] K. Wolter. Second Order Fluid Stochastic Petri Nets: an Extension of GSPNs for Approximate and Continuous Modelling. In *Proceedings of the World Congress on Systems Simulation*, 1997.