International Journal of Systems Science
volume 34, numbers 10–11, 15 August–15 September 2003, pages 615–626

Admission and routing control with partial information and limited buffers

E. ALTMAN†, R. MARQUEZ‡ and U. YECHIALI§*

Problems of admission and routing control for loss systems comprised of a controller and C down-stream servers are studied. We focus on problems in which control actions have to be taken with either delayed or with no information on the state of the down-stream servers. We first consider a problem of routing into C servers and compare the performance of two policies: a static round-robin policy, which does not wait for the delayed information at a risk of losing customers at the busy servers, and a Wait policy, that avoids losses at the servers but risks losses at the controller buffer. We identify regions in which each of the policies performs better. We then study the problem with no information on down-stream servers and propose a timer mechanism to decide when to dispatch an arriving customer. We optimize the value of the timer's parameter. Our study is accompanied with numerical investigations.

1. Introduction

In high speed networks, propagation delay of information cannot be neglected with respect to transmission delays. This is particularly the case in geosatellite satellite networks in which round-trip information delays are around 250 ms. In addition, large random time varying delays are often incurred due to queueing. Many network control problems (such as routing and admission control) therefore have to take into account the information delay. In such cases, we either have to take decisions without waiting for the delay or have to evaluate the impact of having to wait for the delays on the system performance.

This paper focuses on admission and routing problems occurring in loss systems, in which state information is either delayed or non-available. The common objectives in the problems that we pose is to minimize losses (or equivalently, maximize the throughput).

Received 15 July 2002. Revised 25 December 2002. Accepted 1 June 2003.

† INRIA, 2004 route des Lucioles, B.P. 93, F-06902 Sophia Antipolis, France, and CESIMO, Universidad de Los Andes, Mérida, Venezuela.

‡ Facultad de Ingenieria, Universidad de Los Andes, Mérida, Venezuela.

§ Department of Statistics and Operations Research, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel.

* To whom correspondence should be addressed.
e-mail: uriy@post.tau.ac.il

We first consider a problem of routing into C servers and compare the performance of two policies:

(1) Static round-robin policy, which does not wait till the delayed information on service completion arrives; it dispatches each arriving customer according to the round-robin policy at the risk of loss of that customer at the server, if it has not completed its service of the previous customer there.
(2) Wait policy, which only dispatches a job to a server once it receives the information that the server has completed service. Customers that arrive when the Wait policy is used have to queue in a finite queueing facility till they are dispatched. This policy avoids losses at the servers but results in losses when a customer arrives and finds the queueing facility full.

We evaluate the performance of both policies and show that for large delays, the round robin outperforms the Wait policy, and for low delays, the situation is reversed. This suggests the existence of a threshold such that for delays larger than the threshold it is better to use the round robin policies, and for delays lower than the threshold it is better do use the Wait policy. Through an extensive numerical investigation, we validate the existence of such a threshold and study its properties.

We then study the problem with no information on down-stream servers and propose a timer mechanism to decide when to dispatch an arriving customer. We optimize the value of the timer's parameter. Our study is accompanied with numerical investigations.

The structure of the paper is as follows. We introduce in Section 2 a brief (non-exhaustive) survey of control problems in telecommunications with delayed information. We then introduce in Section 3 the general model. Then we study in Section 4 the performance of the Wait policy and that of the round-robin policies. The comparison between the policies and the existence of a threshold, obtained numerically, are the subject of Section 5. Finally the timer model is presented, analysed and optimized in Section 6.

2. Related work

We briefly review work on control problems with delayed information in telecommunications. Flow control with delayed information has been studied in Altman and Nain (1992), Altman and Stidham (1995), Kuri and Kumar (1997) by transforming the problem into an equivalent MDP with full information. The first paper has been extended to noisy delayed information in Altman and Koole (1995). Two types of flow control have been studied. The first type is a rate-base flow control, in which the rate of transmission of packets is directly controlled. The second type is a window-based flow control, in which the controller adjusts its window dynamically; a window stands for the number of packets that can be sent before acknowledgements to the source arrive from the destination. Work on rate-based flow control with delay in the framework of linear-quadratic control (linear dynamics and quadratic cost) has appeared in Altman *et al.* (1999) and references therein. The impact of delay on window-based flow control in the framework of Jackson network is analysed in Bovopoulos and Lazar (1991). A problem of optimal priority assignment for access to a single channel with delay has been investigated in Altman *et al.* (1995). Routing with delayed information has been investigated in Artiges (1995), Kuri and Kumar (1992) and Litvak and Yechiali (2001).

The model in our paper is closely related to that in Litvak and Yechiali (2001) who also compares the performance of policies that wait for information and policies that ignore the information. The framework is however of an infinite queue and the performance measure studied is expected delays. This is in contrast to our framework in which we study finite buffers and are interested in maximizing throughputs and minimizing loss probabilities.

We finally mention some works on control of communication with delayed information in the case of several decentralized controllers. The reference uses a framework known as 'delay sharing information', in which the state space can be decomposed to several parts, each corresponding to another controller. Now each control has an immediate information on his own part of the state space, but a delayed information on the parts corresponding to other controllers. In Schoute (1978), a decentralized control in packet switched satellite communication is studied, whereas a decentralized control problem for multiaccess broadcast networks have been studied in Grizzle *et al.* (1982). In both examples, each controller has to decide whether to transmit or not, without knowing if packets have arrived in the current time unit to other nodes. If they did, then packets from other nodes could be scheduled for transmission at the same time and collisions could occur.

3. Model

A single controller accepts arriving messages (jobs) and dispatches them to C down-stream servers.

Assumptions:

(1) Arrivals: the external arrival is Poisson (λ) with interarrival times IA $\sim \text{Exp}(\lambda)$ having Laplace–Stieltjes Transform (LST) $E[\exp\{-s\cdot \text{IA}\}] = \widetilde{\text{IA}}(s)$.

(2) Controller: the controller has a buffer of size $N_c + 1$ (where $0 \leq N_c \leq \infty$), i.e. if $N_c = 0$, then only one job can reside in the controller's buffer. If there are $N_c + 1$ jobs in the controller's buffer and arrival occurs, it is lost.

(3) Servers: the buffer of each server is of size $N_s + 1$. The service time B of each individual job is distributed $\text{Exp}(\mu)$.

(4) Information: information about service completion reaches the controller only after a random delay $V \sim \text{Exp}(\gamma)$.

We assume that all interarrival times, service times and information delays are independent.

We propose below several policies for the controller and compare the performances, under different assumptions on the information delay.

4. Model 1

4.1. *Wait option*

We assume here that the controller waits until he receives information on service completion before dispatching a job (if available) to a server. (In such a case there could be at most one job in each server's buffer.) This leads to the following Markovian model.

Due to the delayed information on service completions, the controller does not know the real number of jobs present in the system. We shall adopt here the view that the controller considers a job to be 'in the system' until the information on the departure of that job becomes known to the controller. Let X denote the number of jobs 'in the system' and let J denote the number of actually operating (servicing) servers. To illustrate the transitions between states consider the case $N_c = 0$. Assume that there are $J = j < C$ actual operating servers and that the controller considers there to be $X = n$ jobs in the system (clearly $n \geq j$). We denote this situation as state (j, n). When $n = C$ and a new job arrives the job is kept at the controller's buffer and the state becomes $(j, C+1)$. At that time there are $C - j$ servers that are free, although this information is not yet available to the controller. The remaining time till the first information on a new server becoming free arrives at the controller is exponentially distributed with parameter $(C - j)\gamma$. As soon as this information becomes available, the controller immediately dispatches the job he holds in his buffer, bringing the state of the system to $(j+1, C)$. When $n < C$ and a job arrives it is immediately dispatched to one of the $C - n$ available servers, bringing the state of the system to $(j+1, n+1)$. Finally, when the controller counts $n = C + 1$, any new arrival is lost.

It should be noted that according to this policy, there are no losses at the servers' side.

Let P_{jn} be the probability that there are j operating servers and total n jobs 'in the system' as counted by the controller ($j \leq \min(n, C)$; $n \leq C + N_c + 1$).

The rate-of-transition diagram for $N_c = 0$ is depicted in figure 1, where the vertical axis denotes the number of operating servers, J, and the horizontal axis depicts the total number of jobs 'in the system', X.

Balance equations, $N_c \geq 0$:

When $N_c \geq 0$, the balance equations for the state probabilities P_{jn} are the following:

$j = 0$: (1)

$$\begin{aligned}
\lambda P_{00} &= \gamma P_{01}, && n = 0\\
(\lambda + n\gamma)P_{0n} &= (n+1)\gamma P_{0,n+1} + \mu P_{1n}, && 1 \leq n \leq C-1\\
(\lambda + C\gamma)P_{0C} &= \mu P_{1C}, && n = C\\
(\lambda + C\gamma)P_{0n} &= \mu P_{1,n+1} + \lambda P_{0,n-1}, && n = C+1, \ldots, C+N_c\\
C\gamma P_{0,C+N_c+1} &= \mu P_{1,C+N_c+1} + \lambda P_{0,C+N_c}, && n = C+N_c+1.
\end{aligned}$$

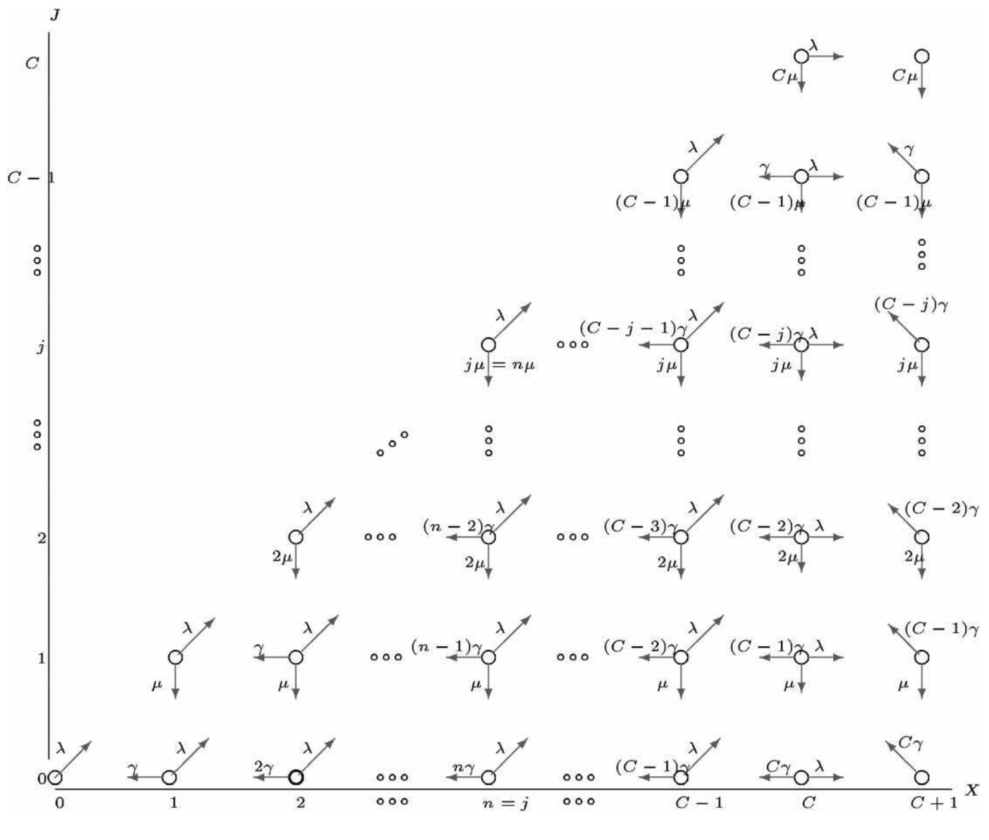

Figure 1. Transition rate diagram for $N_c = 0$.

$1 \le j \le C-1:$ (2)

$$\begin{aligned}
(\lambda+j\mu)P_{jj} &= \gamma P_{j,j+1}+\lambda P_{j-1,j-1} \quad n=j\\
(\lambda+(n-j)\gamma+j\mu)P_{jn} &= (n+1-j)\gamma P_{j,n+1}\\
&\quad +(j+1)\mu P_{j+1,n}+\lambda P_{j-1,n-1}\\
&\quad j+1\le n\le C-1\\
(\lambda+(C-j)\gamma+j\mu)P_{jC} &= (j+1)\mu P_{j+1,C}+\lambda P_{j-1,C-1}\\
&\quad +(C-j+1)\gamma P_{j-1,C+1}\\
&\quad n=C\\
(\lambda+(C-j)\gamma+j\mu)P_{jn} &= (j+1)\mu P_{j+1,n}+\lambda P_{j,n-1}\\
&\quad +(C-j+1)\gamma P_{C-1,n+1}\\
&\quad n=C+1,\ldots,C+N_c\\
((C-j)\gamma+j\mu)P_{j,C+N_c+1} &= (j+1)\mu P_{j+1,C+N_c+1}+\lambda P_{j,C+N_c}\\
&\quad n=C+N_c+1.
\end{aligned}$$

$j=C:$ (3)

$$\begin{aligned}
(\lambda+C\mu)P_{CC} &= \lambda P_{C-1,C-1}+\gamma P_{C-1,C+1} \quad n=C\\
(\lambda+C\mu)P_{Cn} &= \lambda P_{C,n-1}+\gamma P_{C-1,n+1}\\
&\quad n=C+1,\ldots,C+N_c\\
C\mu P_{C,C+N_c+1} &= \lambda P_{C,C+N_c} \quad n=C+N_c+1.
\end{aligned}$$

Now, for this 'wait' policy of the controller, whenever there are $n=C+N_c+1$ jobs in the system (i.e. the controller holds N_c+1 jobs in his buffer) each new arrival will be lost. Thus, the probability of loss is given by

$$P_{\text{loss}}(\text{wait}) = \sum_{j=0}^{C} P_{j,C+N_c+1} =: P_{\bullet,C+N_c+1}.$$

The mean number of losses per unit time (i.e. loss rate) is

$$\lambda P_{\text{loss}}(\text{wait}) = \lambda P_{\bullet,C+N_c+1}.$$

Limiting case: Suppose $1/\gamma \to 0$. That is, the server obtains information on service completions with no delay. The state space collapses to a one dimensional space (that denotes the number of jobs in the system) and the transition diagram, for $N_c > 0$, is depicted in figure 2.

Denoting $a=\lambda/\mu$, the balance equations are:

$$P_n = \frac{1}{n!}a^n P_0, \qquad n=0,1,2,\ldots,C \tag{4}$$

$$P_{C+k} = \left(\frac{a}{C}\right)^k P_C, \quad k=1,2,\ldots,N_c+1. \tag{5}$$

As the probabilities sum to one, we get

$$P_0^{-1} = \sum_{n=0}^{C}\frac{a^n}{n!} + \frac{a^C}{C!}\sum_{k=1}^{N_c+1}\left(\frac{a}{C}\right)^k.$$

Thus,

$$\begin{aligned}
P_{\text{loss}}(\text{wait}) &= P_{C+N_c+1}\\
&= \frac{(a^C/C!)(a/C)^{N_c+1}}{\sum_{n=0}^{C}(a^n/n!) + (a^C/C!)\sum_{k=1}^{N_c+1}(a/C)^k}.
\end{aligned}$$

The expected number of losses per unit time is $\lambda P_{\text{loss}}(\text{wait}) = \lambda P_{C+N_c+1}$.

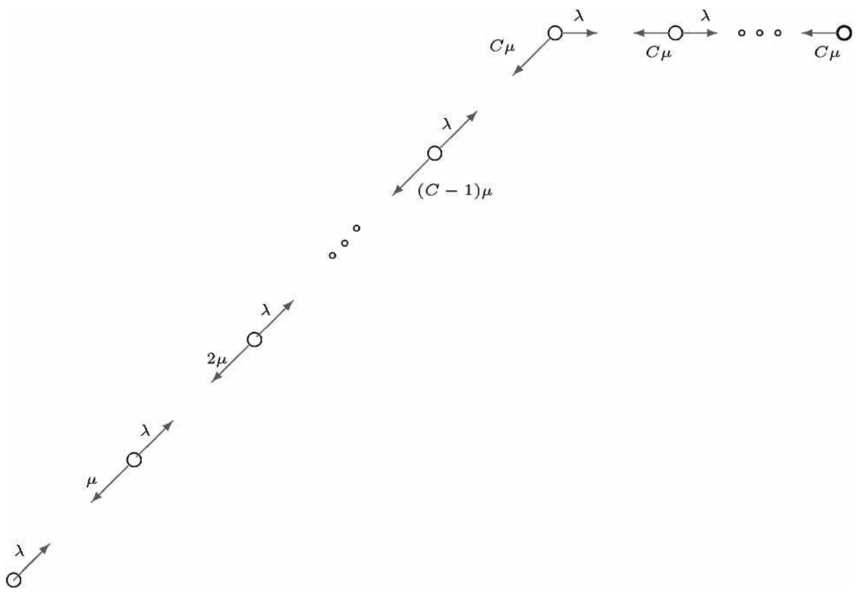

Figure 2. Transition rate diagram for $1/\gamma = 0$ and $N_c > 0$.

4.2. *No wait: the round robin policy*

According to this policy the controller dispatches jobs following the round robin (RR) mechanism, that is, arrival number $kC + i$ is sent to server no. i ($k = 0, 1, 2, \cdots$; $1 \le i \le C$). Thus, the inter-arrival time to each server is $\text{Erlang}(C, \lambda)$ with mean C/λ. We assume that $N_s = 0$ for each server.

The probability of a loss $P_{\text{loss}}(\text{RR})$ at a given server is the probability that the interarrival time is shorter than the service time B, i.e.

$$P_{\text{loss}}(\text{RR}) = P[\text{Erlang}(C, \lambda) < B] = \left(\frac{\lambda}{\lambda + \mu}\right)^C = [\mathrm{I}\tilde{\mathrm{A}}(\mu)]^C$$

Thus, the expected number of losses per unit time is

$$\lambda P_{\text{loss(RR)}} = \lambda \left(\frac{\lambda}{\lambda + \mu}\right)^C.$$

5. Comparison between Wait and RR policies

5.1. *Extreme cases: γ small and large*

For the limiting case $1/\gamma \to 0$, i.e. when full information is available,

$$P_{\text{loss(RR)}} > P_{\text{loss}}(\text{wait}) \ \text{ iff } \left(\frac{\lambda}{\lambda+\mu}\right)^C = \left(\frac{a}{a+1}\right)^C > \frac{(a^C/C!)(a/C)^{N_c+1}}{\sum_{n=0}^{C}(a^n/n!) + (a^C/C!)\sum_{k=1}^{N_c+1}(a/C)^k}.$$

When $N_c = 0$ this is equivalent to

$$\sum_{n=0}^{C} \frac{a^n}{n!} + \frac{a}{C} \cdot \frac{a^C}{C!} > \frac{a}{C \cdot C!}(a+1)^C,$$

or to,

$$\sum_{n=0}^{C} \frac{a^n}{n!} > \frac{a}{C \cdot C!}\left[(a+1)^C - a^C\right]. \qquad (6)$$

Proposition 5.1: *For the limiting case* $1/\gamma \to 0$ *we have*: $P_{loss}(RR) > P_{loss}(wait)$ for all $C \ge 1$.

The proof follows directly from the next two Lemmas.

Lemma 5.1: *When* $N_c = 0$, $P_{loss}(RR) > P_{loss}(wait)$ for all $C \ge 1$.

Proof: Writing $(a+1)^C = \sum_{n=0}^{C}\binom{C}{n}a^n$, the right hand side of (6) becomes $(1/C)\sum_{n=1}^{C} a^n/((C+1-n)!(n-1)!)$. It is now easy to check that the coefficient of each power of a on the left hand side of (6) is greater than the corresponding coefficient on the right hand side. ■

Lemma 5.2: $P_{loss}(wait)$ *is monotone decreasing in* N_c.

Proof: Fix a value $N_c \ge 0$ and denote by X the random variable corresponding to the number of customers in the system. Denote by X' the random variable corresponding to the number of customers in another system which differs from the original only by the fact that $N_c' = N_c + 1$. Define $Y = \max(X' - 1, 0)$ and note that Y and X have the same range of $(0, 1, 2, \ldots, C + N_c + 1)$. Let

$$r_X(0) = 0, \quad r_X(n) = P(X = n-1)/P(X = n), \quad n = 1, 2, 3, \ldots, C + N_c + 1.$$

We define in the same way r_Y. Then:

$$r_X(n) = \begin{cases} n/a & n = 1\ldots, C, \\ C/a & n = C+1, \ldots, C + N_c + 1, \end{cases}$$

$$r_Y(n) = \begin{cases} 2(1+a)/a^2 & n = 1, \\ (n+1)/a & n = 2\ldots, C-1, \\ C/a & n = C, \ldots, C + N_c + 1. \end{cases}$$

It is easy to check that for all $n = 1, 2, \ldots, N_c + C + 1$ we have

$$r_X(n) \le r_Y(n).$$

It then follows (e.g. Ross and Yao 1990, equation 4) that $X \ge Y$ in the likelihood ratio and in the stochastic order ratio, which implies that

$$P(X = C + N_c + 1) \ge P(Y = C + N_c + 1)$$

or equivalently,

$$P(X = C + N_c + 1) \ge P(X' = C + N_c + 2).$$

This establishes the proof. ■

Remark 5.1: If $1/\gamma \to \infty$ then the controller, if waits, never dispatches jobs to the C channels and all losses are incurred by the controller, so $P_{\text{loss(RR)}} < P_{\text{loss(Wait)}}$.

5.2. *Threshold policy: numerical results*

Having seen that in the extreme cases RR is better when delays are large and Wait is better for short delays, we could expect there to be a threshold $\hat{\gamma}^*(\mu, \lambda)$ on the delay parameter such that RR is better than Wait for $\gamma < \hat{\gamma}^*(\mu, \lambda)$ and Wait is better than RR for $\gamma > \hat{\gamma}^*(\mu, \lambda)$. The existence of such a threshold will be supported by our numerical investigation. Since we can rescale time (by redefining what is a basic time unit), the threshold will be of the form:

$$\hat{\gamma}^*(\mu, \lambda) = \lambda\gamma^*(\mu/\lambda).$$

Without loss of generality we can thus choose $\lambda = 1$ and check the dependence of γ^* on μ.

Figures 3–5 analyse the case of $N_c = 0$ and $\lambda = 1$. Using Matlab, we did an exhaustive numerical study of the performance of both RR and Wait policies as a function of the parameters by solving equation (1–3). We consider the case of two, three and four servers (figures 3–5, respectively).

We let γ (horizontal axis) vary from 0.01 to 10. We take six values of μ: 0.1000, 0.3162, 1.0000, 3.1623, 10.0000 and 31.6228. The vertical axis in the figures corresponds to the loss probability, and the horizontal axis to the value of the parameter γ. For each fixed μ there is one pointed horizontal line that gives the loss probability under the RR policy, and there is also a curve of the loss probability as a function of γ under the Wait policy. We see that for each value of μ, the curve describing the Wait policy intersects once with the horizontal line describing the RR policy. This shows that there is indeed a threshold $\gamma^*(\mu)$ which is obtained as the intersection point.

We further see from the figures that the threshold is *increasing* in μ. The threshold is almost linear on the log-log scale of the figure. For example, for $C = 3$ it is approximated by the empirical relation:

$$\log(P_{loss}(RR)) = -3.8253 \times \log(\gamma) - 0.9972.$$

If we did not take a log-log scale we would see clearly that the threshold as a function of μ is almost constant for μ in the range of 0.2–4 and its value is close to one.

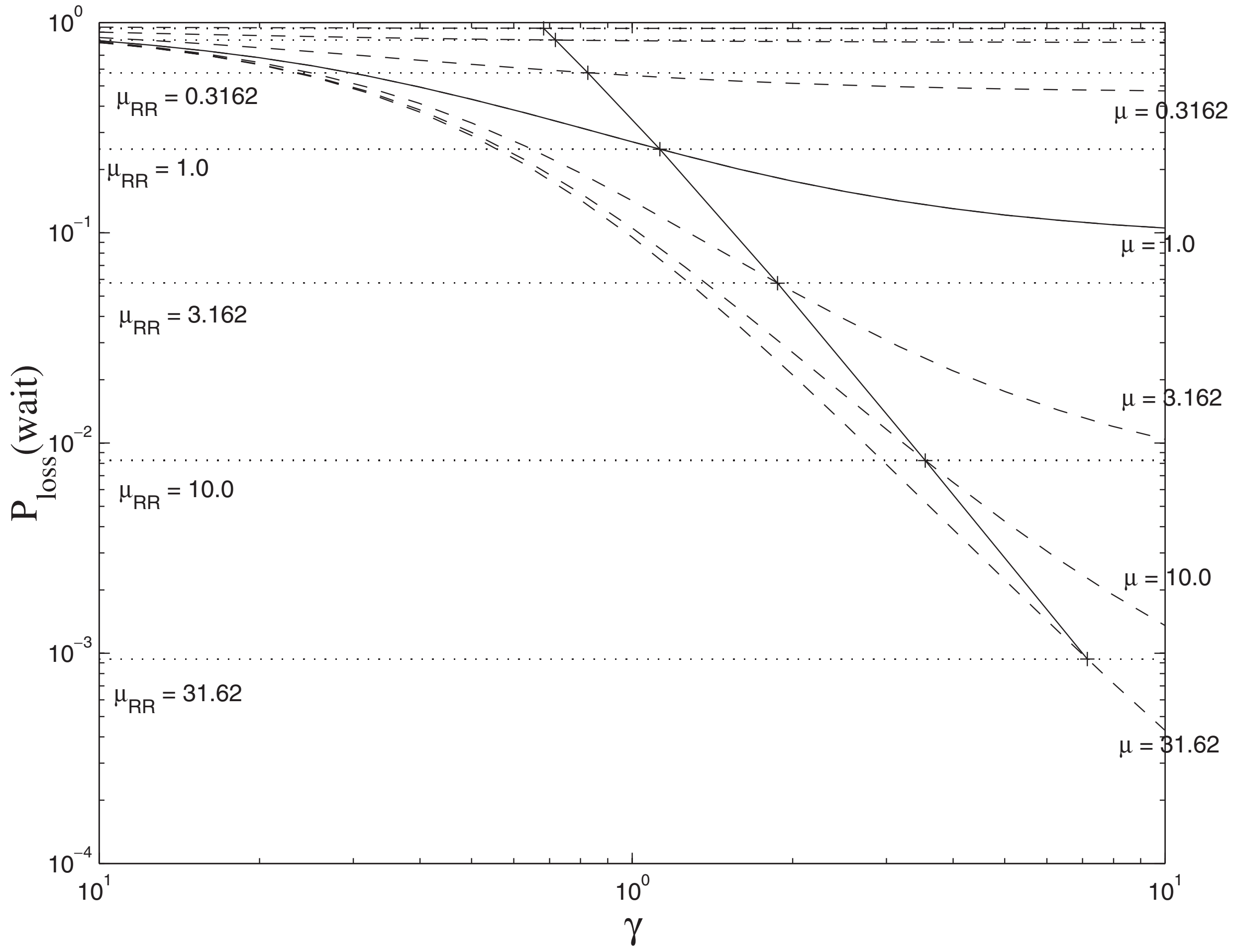

Figure 3. Numerical analysis of loss probabilities as a function of γ and μ for both RR and Wait policies, $c = 2$ servers, $N_c = 0$ and $\lambda = 1$. The dotted horizontal lines correspond to loss probabilities under the RR policies with $P_{loss}(RR) = (\lambda/(\lambda + \mu))^c = (1/(1 + \mu))^2$. The curved lines correspond to the Wait policies.

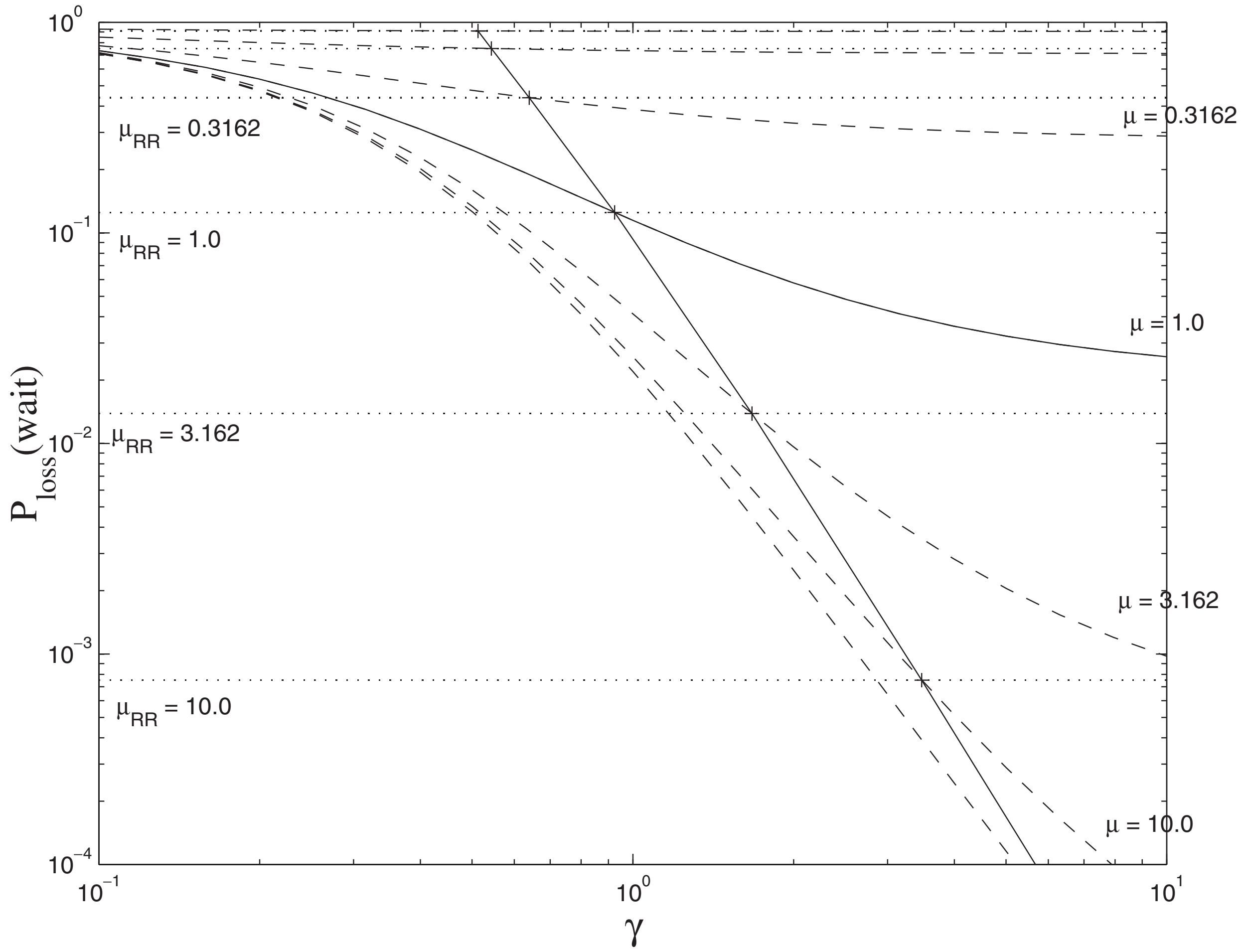

Figure 4. Numerical analysis of loss probabilities as a function of γ and μ for both RR and Wait policies, $c = 3$ servers, $N_c = 0$ and $\lambda = 1$.

6. Model 2: a timer

We introduce the following Timer policy. As soon as the controller dispatches a job to a server, he activates a Timer, having a random duration T.

We consider the case where the controller obtains no information on service completions. The first arrival during T (if occurs) is held in the controller's buffer and released for service at time T. Subsequent jobs within T (if any) are lost. If the first arrival after a dispatching occurs beyond T, it is sent immediately to one of the servers and a new Timer is activated. Moreover, if a job is dispatched to a server and the latter is busy, the job is lost.

The problem is to find the value (or the distribution) of T so as to minimize the total rate of losses, both at the controller's and the servers' side.

$C = 1$ servers: We consider first the case $C = 1$ and assume that the down-stream server may hold only one job, i.e. $N_s = 0$. We further assume that the controller can hold only one job: $N_c = 0$.

Let τ be the time between two consecutive dispatches of jobs to the server. Let R be the time interval from the moment of dispatching till the first arrival thereafter occurs. R is either the full interarrival time (if the moment of dispatching occurs immediately upon arrival), or it is the residual interarrival time (if the moment of dispatching occurs when the timer had expired previously and there was a job in the controller's buffer). In both cases, due to the Poisson arrival, R has an exponential distribution with parameter λ. Thus, $\tau = \max(R, T)$.

The probability of loss at the server is given by

$$P_{\text{loss}}(\text{server}) = P(\tau < B) = \tilde{\tau}(\mu),$$

where $\tilde{\tau}$ is the Laplace–Stieltjes transform of τ. Since the rate of arrival to the server is $1/E[\tau]$, the rate of losses at the server's barrier is $\tilde{\tau}(\mu)/E[\tau]$. On the other hand, the rate of losses at the controller's entrance is $\lambda - 1/E[\tau]$.

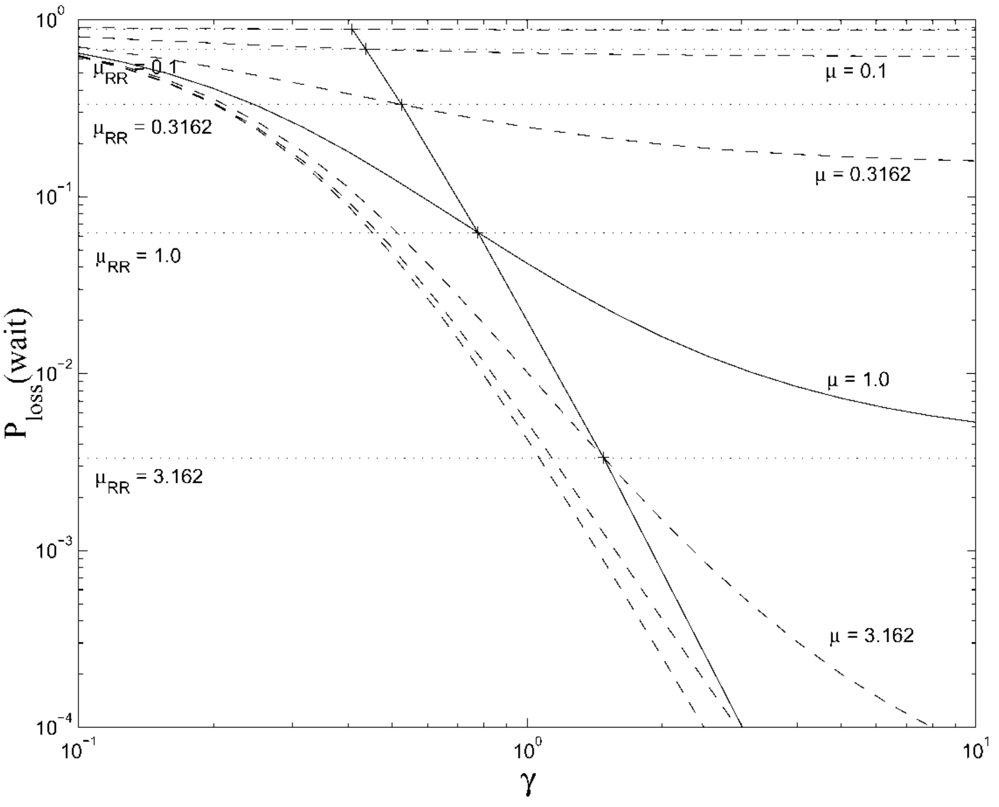

Figure 5. Numerical analysis of loss probabilities as a function of γ and μ for both RR and Wait policies, $c = 4$ servers, $N_c = 0$ and $\lambda = 1$.

The total rate of losses is thus

$$\frac{\tilde{\tau}(\mu)}{E[\tau]} + \lambda - \frac{1}{E[\tau]}.$$

The throughput is then

$$\text{THP} = \lambda - \{\text{total rate of losses}\} = \frac{1 - \tilde{\tau}(\mu)}{E[\tau]}.$$

We now wish to find T that maximizes THP.

Deterministic timer, $C = 1$: When T is a fixed constant $T = T_0$, then

$$P(\tau \le t) = \begin{cases} 0 & \text{for } t < T_0, \\ 1 - \exp(-\lambda t) & \text{for } t \ge T_0. \end{cases}$$

Hence

$$\tilde{\tau}(\mu) = \int_0^\infty e^{-\mu t} \mathrm{d}P(\tau \le t) = \int_{t=T_0}^\infty e^{-\mu t} \lambda e^{-\lambda t} \mathrm{dt}$$
$$= \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)T_0},$$

and

$$E[\tau] = \int_0^\infty [1 - P(\tau \le t)]\mathrm{dt} = T_0 + \frac{1}{\lambda} e^{-\lambda T_0}.$$

So

$$\text{THP} = \frac{1 - \tilde{\tau}(\mu)}{E[\tau]} = \frac{1 - [\lambda/(\lambda + \mu)]e^{-(\lambda+\mu)T_0}}{T_0 + (1/\lambda)e^{-\lambda T_0}}. \quad (7)$$

To obtain the maximum throughput, we compute the derivative of THP at zero and obtain the condition

$$\left(\lambda T_0 + \frac{\lambda}{\lambda + \mu}\right) e^{-(\lambda+\mu)T_0} + \frac{\mu}{\lambda + \mu} e^{-(2\lambda+\mu)T_0} = 1 - e^{-\lambda T_0}.$$

One can easily see that this equation has a unique finite solution $T_0 > 0$.

Exponential timer, $C = 1$: In case T is exponentially distributed with parameter ξ. We have

$$P(\tau \le t) = P(\max(T, \text{IA}) \le t) = P(T \le t)P(\text{IA} \le t)$$
$$= 1 - e^{-\lambda t} - e^{-\xi t} + e^{-(\lambda+\xi)t}.$$

Thus

$$\tilde{\tau}(\mu) = \int_0^\infty e^{-\mu t} \mathrm{d}P(\tau \le t) = \frac{\lambda}{\lambda+\mu} + \frac{\xi}{\xi+\mu} - \frac{\lambda+\xi}{\lambda+\xi+\mu},$$

and

$$E[\tau] = \frac{1}{\lambda} + \frac{1}{\xi} - \frac{1}{\lambda+\xi} = \frac{1}{\lambda} + \frac{\lambda}{\xi(\lambda+\xi)}.$$

Hence the throughput is given by

$$\begin{aligned}\mathrm{THP} &= \frac{1-\tilde{\tau}(\mu)}{E[\tau]}\\ &= \frac{[\mu/(\lambda+\mu)] - [\xi/(\xi+\mu)] + [(\lambda+\xi)/(\lambda+\xi+\mu)]}{1/\lambda + \lambda/(\xi(\lambda+\xi))}.\end{aligned} \tag{8}$$

Note that when there is no timer ($\xi \to \infty$) then

$$\lim_{\xi\to\infty} \mathrm{THP}\ (\xi) = \frac{1}{(1/\lambda) + (1/\mu)}.$$

Indeed, the expected interval between two successive job-departures equals $(1/\lambda) + (1/\mu)$ since any job sent to the server during service is lost, so that after a service completion (having mean $1/\mu$) it takes, on average, $1/\lambda$ units of time for the next arrival.

On the other hand, if $\xi \to 0$ then $E[\tau]$ tends to infinity and, obviously, the throughput tends to zero.

Numerical results: In figure 6, we plot the optimal value of the timer T_0 and of the exponentially average timer value $T_0 = \xi^{-1}$ as a function of μ. We see that it decreases in μ, and becomes almost constant for $\mu \ge \lambda/10$. We also depict the throughputs obtained under the optimal timer. We clearly see that the deterministic timer always outperforms the exponential one.

$C > 1$ servers: When $C > 1$, the controller dispatches arriving jobs to the various servers in a cyclic (i.e. Round Robin) fashion. After each dispatch he activates a Timer T. If an arrival occurs before T, the controller keeps it in its buffer. All subsequent arrivals within T are lost. If there is a job in the controller's buffer at time T, it is dispatched according to the RR policy. If not, the first arrival thereafter is immediately dispatched and the controller activates a new Timer.

Let τ be the time between two consecutive dispatches. As before, $\tau = \max(R, T)$ where R is the time interval from a moment of dispatching until first arrival thereafter. Recall that R is exponentially distributed with parameter λ.

The rate of loss at the controller's entrance is, as before, $\lambda - 1/E[\tau]$.

The rate of loss at the servers is calculated as follows. Since service times B are Exponential (μ),

$$P_{\text{loss}}(\text{any single server}) = P\left(\sum_{j=1}^{C} \tau_j < B\right) = [\tilde{\tau}(\mu)]^C.$$

As the rate of arrival to any single sever is $1/(CE[\tau])$, the total rate of loss for all C servers and the controller is

$$\frac{C[\tilde{\tau}(\mu)]^C}{CE[\tau]} + \left(\lambda - \frac{1}{E[\tau]}\right).$$

The throughput is the external arrival rate, λ, minus the total loss rate:

$$\mathrm{THP} = \frac{1}{E[\tau]} - \frac{[\tilde{\tau}(\mu)]^C}{E[\tau]} = \frac{1-[\tilde{\tau}(\mu)]^C}{E[\tau]}.$$

As examples, for T exponential with parameter ξ we obtain

$$[\tilde{\tau}(\mu)]^C = \left[\frac{\lambda}{\lambda+\mu} + \frac{\xi}{\xi+\mu} - \frac{\lambda+\xi}{\lambda+\xi+\mu}\right]^C$$

and as before,

$$E[\tau] = \frac{1}{\lambda} + \frac{\lambda}{\xi(\lambda+\xi)}$$

so that

$$\begin{aligned}&\mathrm{THP}\,(exp)\\ &= \frac{1 - [(\lambda/(\lambda+\mu)) + (\xi/(\xi+\mu)) - ((\lambda+\xi)/(\lambda+\xi+\mu))]^C}{1/\lambda + (\lambda/\xi(\lambda+\xi))}.\end{aligned} \tag{9}$$

When $C = 1$, equation (9) reduces to (8).

For $T = T_0$ deterministic, we have

$$[\tilde{\tau}(\mu)]^C = \left(\frac{\lambda}{\lambda+\mu}\right)^C e^{-C(\lambda+\mu)T_0},$$

and

$$E[\tau] = T_0 + \frac{1}{\lambda} e^{-\lambda T_0},$$

so

$$\mathrm{THP}\,(\text{deterministic}) = \frac{1 - (\lambda/\lambda+\mu)^C e^{-C(\lambda+\mu)T_0}}{T_0 + (1/\lambda)e^{-\lambda T_0}}. \tag{10}$$

Again, when $C = 1$ equation (10) reduces to (7). Also, as for the case $C = 1$, the optimal value T_0 can be calculated by differentiation.

Numerical results: In figures 7–9, we plot the optimal value of the timer T_0 and of the exponentially average timer value $T_0 = \xi^{-1}$ as a function of μ, for the cases

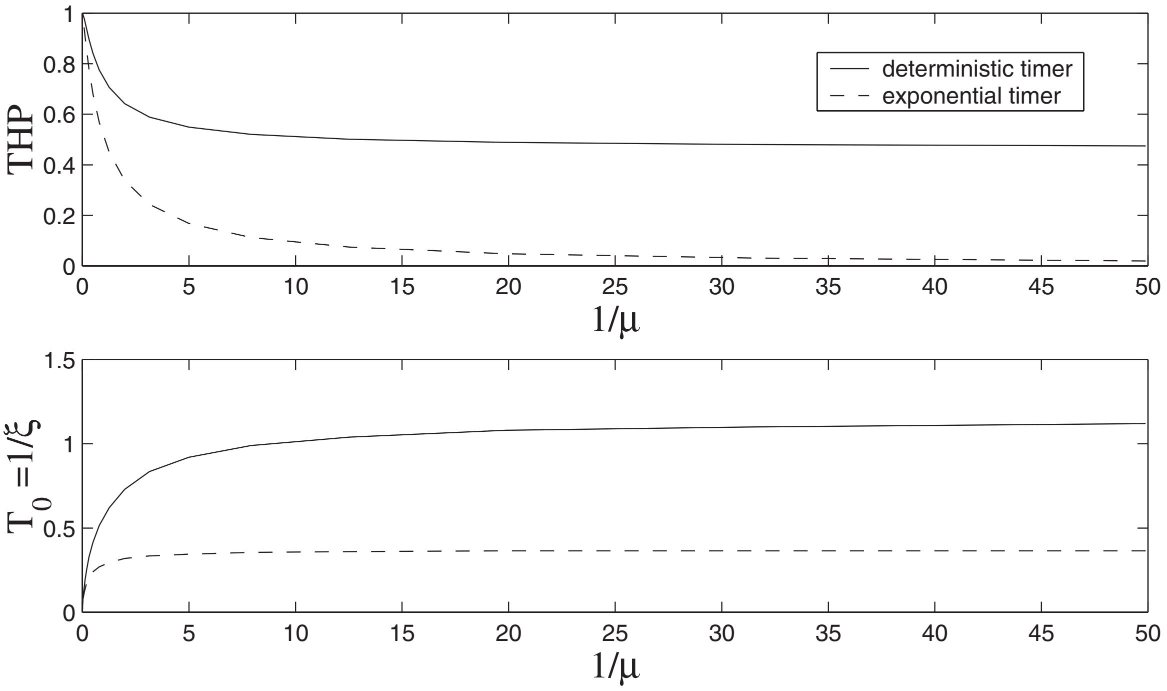

Figure 6. Optimal threshold value and throughput of the deterministic and exponential timers as a function of μ, $C = 1$.

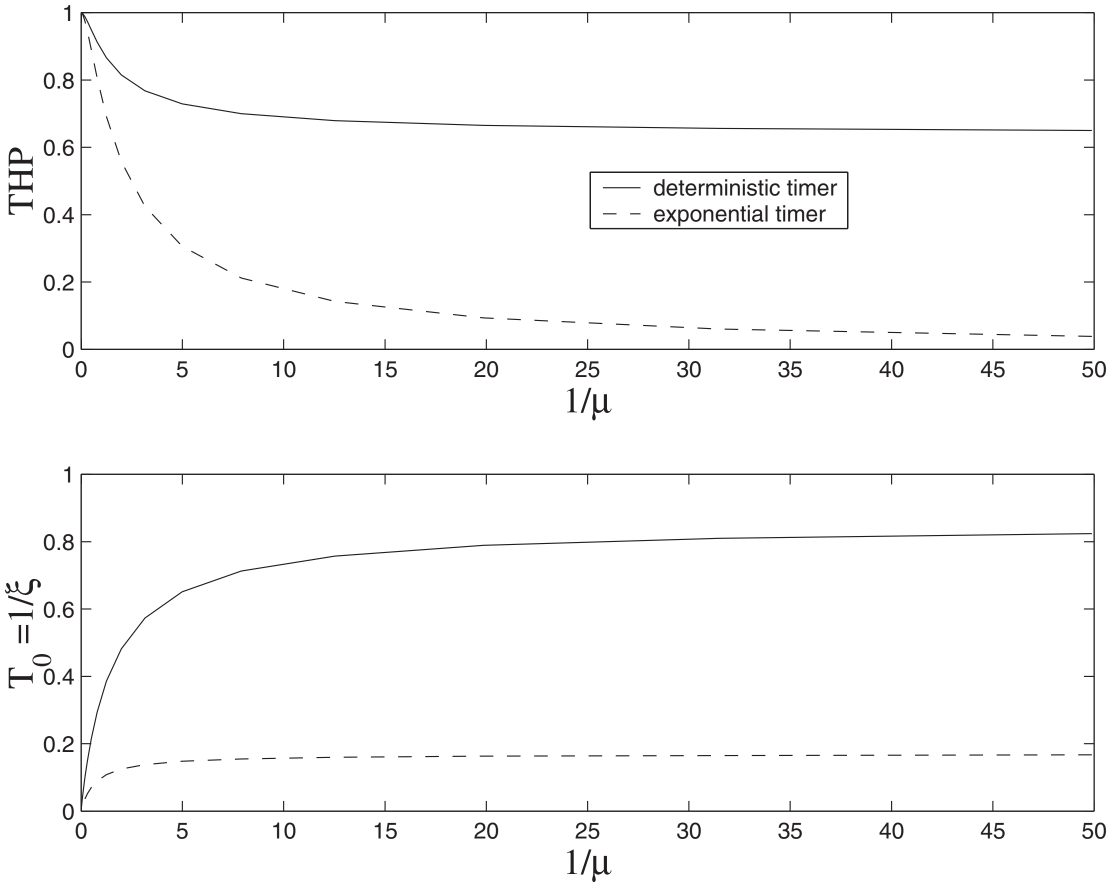

Figure 7. Optimal threshold value and throughput of the deterministic and exponential timers as a function of μ, $C = 2$.

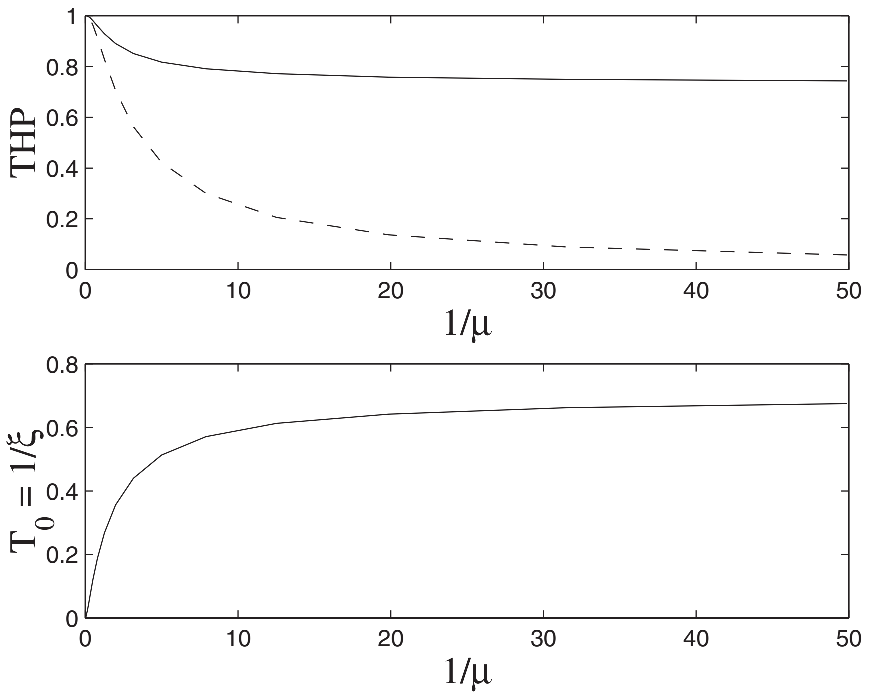

Figure 8. Optimal threshold value and throughput of the deterministic and exponential timers as a function of μ, $C = 3$.

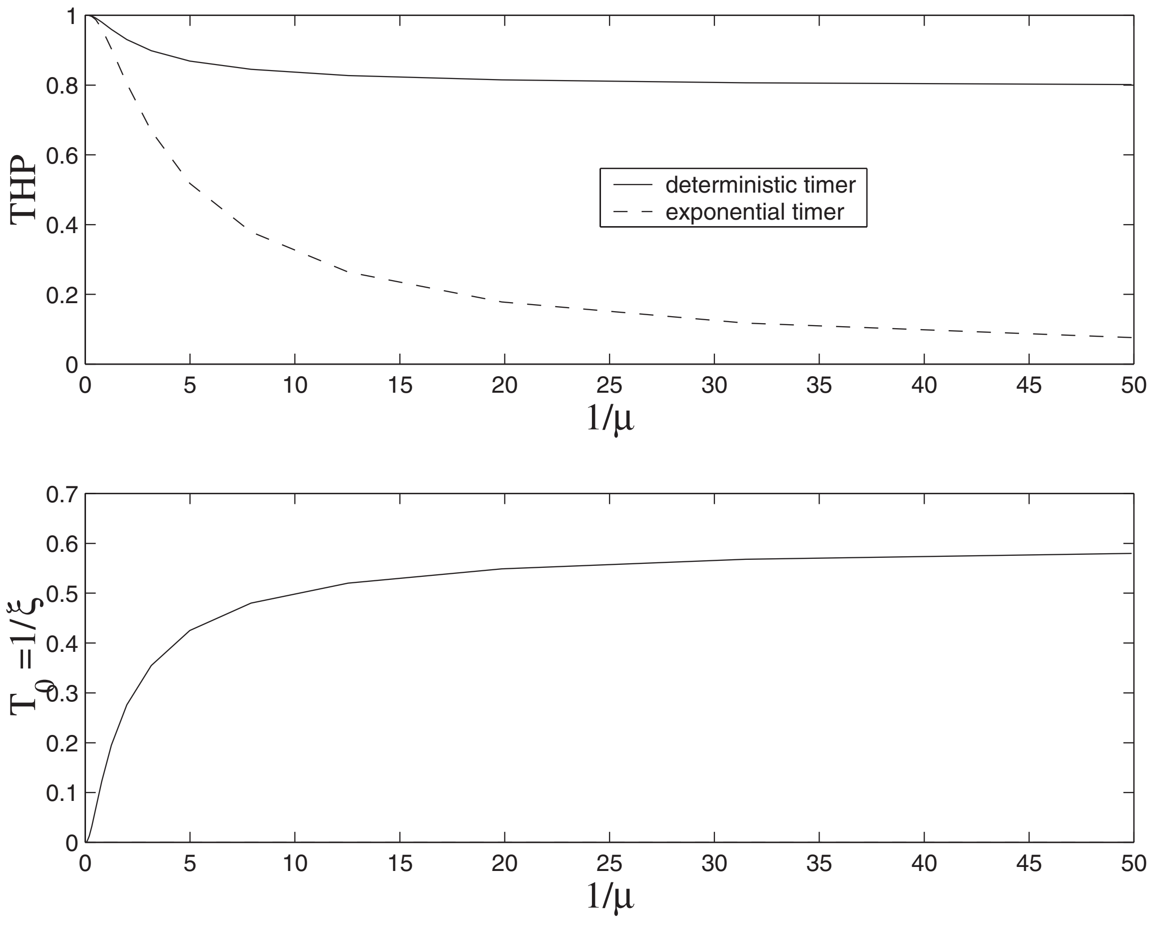

Figure 9. Optimal threshold value and throughput of the deterministic and exponential timers as a function of μ, $C = 4$.

of $C = 2$–4, respectively. We see that it decreases in μ, and becomes almost constant for $\mu \geq \lambda/10$. We also depict the throughputs obtained under the optimal timer. We see again that the deterministic timer always outperforms the exponential one. Without loss of generality, we have considered only the case of $\lambda = 1$.

For $C = 3$ and $C = 4$ it is better not to use a timer in the case of exponentially distributed time: the value of $T_0 = 1/\xi$ is 0 for all tested values of μ!

7. Conclusion

We have studied two main aspects of delay that appear in admission and routing control. The first type is that of the information available to the controller. To study the relevance of the information after a delay, we have studied the performance of the admission policy that waits till the information becomes available in order to take an action (Wait policy) and compared it to the one that does not wait to get that information (RR policy). We obtained a clear threshold on the expected delay above which the RR policy has better performance (lower loss probability) and below which the Wait policy is superior.

We then studied another role of delay, when the delay is itself a control action. In the absence of any information on the system state, we showed that delaying packets at the input buffer before routing them to the network results in better performance of the system (lower losses). We computed the optimal deterministic and exponentially distributed delays which minimize the loss rate and maximize the system's throughput.

References

Altman, E., Başar, T., and Srikant, R., 1999, Congestion control as a stochastic control problem with action delays. *Automatica* (special issue) 1937–1950.

Altman, E., Kofman, D., and Yechiali, U., 1995, Discrete time queues with delayed information. *Queueing Systems*, **19,** 361–376.

Altman, E., and Koole, G., 1995, Control of a random walk with noisy delayed information. *Systems and Control Letters*, **24,** 207–213.

Altman, E., and Nain, P., 1992, Closed-loop control with delayed information. *Performance Evaluation Review*, **20,** 193–204.

Altman, E., and Stidham, S., 1995, Optimality of monotonic policies for two-action Markovian decision processes, with applications to control of queues with delayed information. *Queueing Systems*, **21,** 267–291.

Artiges, D., 1995, Optimal routing into two heterogeneous service stations with delayed information. *IEEE Transactions on Automatic Control*, **40,** 1234–1236.

Bovopoulos, A. D., and Lazar, A. A., 1991, The effect of delayed feedback information on network performance. *Annals of Operations Research*, 581–588.

Grizzle, J. W., Marcus, S. I., and Hsu, K., 1982, A decentralized control strategy for multiaccess broadcast networks. *Large Scale Systems*, **3,** 75–88.

Kuri, J., and Kumar, A., 1992, Optimal control of arrivals to queues with delayed queue length information. In *Proceedings of the 31th IEEE Conference on Decision and Control.*

Kuri, J., and Kumar, A., 1997, On the optimal control of arrivals to a single queue with arbitrary feedback delay. *Queueing Systems*, **27,** 1–16.

Litvak, N., and Yechiali, U., 2003, Routing in queues with delayed information, *Queueing Systems*, **43,** 147–165.

Ross, K. W., and Yao, D. D., 1990, Monotonicity properties for the stochastic knapsack. *IEEE Transactions on Information Theory*, **36,** 1173–1179.

Schoute, F. C., 1998, Decentralized control in packet switched satellite communication. *IEEE Transactions on Automatic Control*, **23,** 362–371.