Processing Times for Transaction Servers with Quality of Service Differentiation.

Processing Times for Transaction Servers with Quality of Service Differentiation

J.L. van den Berg, R.D. van der Mei, B.M.M. Gijsen, M.J. Pikaart, R. Vranken
KPN Research, P.O. Box 421, 2260 AK Leidschendam, The Netherlands
{j.l.vandenberg | r.d.vandermei }@kpn.com

Abstract

We study the processing times of jobs executed by transaction servers supporting Quality of Service differentiation. To this end, we model the job processing times as the sojourn times in a multiple-server processor-sharing system with two priority classes. We present a closed-form expression for the mean sojourn time of high priority customers, based on a classical result on Generalized Processor Sharing models. For the low priority customers we develop an exact expression for the mean sojourn time in the single-server case, and for the multiple-server case we propose and validate simple and fast-to-evaluate approximations. Numerical results demonstrate that the approximations are highly accurate for a broad range of parameter settings.

1 Introduction

The explosive growth of the Internet, the increasing popularity of PCs and the advances in high-speed networking have boosted the development of Information and Communication Technology (ICT), which is becoming an integrated part of our modern society. ICT enables applications to be supported by geographically distributed server and database systems that can communicate with each other; these applications are commonly referred to as distributed applications. A typical feature of distributed applications is that an end-user transaction generates a sequence of sub-transactions to be performed in parallel and/or in series at different systems. Typical examples of distributed applications are the World Wide Web and other on-line services, billing chains, cash dispensers and credit card payments. With the growth of Internet popularity the performance of Internet services is gradually degrading, sometimes to unacceptable levels. Moreover, a growing number of new real-time services (e.g., Internet telephony, video conferencing) require strict guarantees on the Quality of Service (QoS) observed by the end users. This has raised the need for Internet Service Providers (ISPs) to migrate from best-effort type of Internet service offerings to services with QoS guarantees and QoS differentiation. These observations raise the need to investigate ways to control and predict the QoS of distributed services under realistic or expected load scenarios.

Quality of Service is a rather general term referring to measures such as service availability, reliability, transaction throughput and response times. In this paper we focus on response times. End-to-end transaction response time can be subdivided into the processing times of the sub-transactions at components in the distributed system (e.g. Web servers, database systems, network interfaces). In this paper we focus on the response times of a single bottleneck component, in particular, a host computer on which a multi-threaded server is installed. It is assumed here that the thread spawning algorithm is implemented such that there is always at least one thread available to serve newly incoming transaction requests, and that the performance bottleneck is the limited amount of CPU capacity, which is a realistic assumption for transaction servers with a significant amount of server-side processing. The reader is referred to [12] (and reference therein) for results on the performance modeling of multithreaded servers, and to [8, 13, 18] for extensions of the model in [12] to transaction servers with significant server-side scripting; other references on transaction server performance are [17, 9, 7, 1, 5], amongst others. In this paper, the transaction server is modeled as a multiple-server queueing model with processor sharing service discipline with two priority classes. In this model the servers represent the processors, the customers represent transactions, the customer classes represent QoS classes, the processor sharing service discipline represents the fact that the threads handling the transaction requests in parallel share the processing power of the processors, and the sojourn times represent the processing times of the transactions.

In the literature, a variety of papers focus on processor sharing models. For the M/G/1-PS system Yashkov [21], Ott [16] and Van den Berg and Boxma [2] derive (implicit) expressions for the Laplace Stieltjes Transform (LST) of the sojourn time distribution. Van den Berg [3] obtains a simple and fast approximation for the second moment of the sojourn time in the M/G/1-PS queue. Cohen [6] considers so-called Generalized Processor Sharing (GPS) systems, in which the service rate of the customers in the system is an arbitrary function of the number of customers present.[1] He derives explicit expressions for the distribution of the number of customers in the system, see Section 3 for more details. The reader is referred to Yashkov [22, 23] for overviews of the available results on processor sharing systems. A specific feature of the model studied in the present paper is that the service rate available to low priority customers varies in time due to the fluctuations in the number of high priority customers. In the literature several papers are devoted to processor sharing models with fluctuating service rates. Nunez-Queija [15] considers an M/M/1-PS model with an ON/OFF server, and derives closed-form expressions for the expected conditional sojourn time. He also derives closed-form expressions for the limiting sojourn-time distribution under heavy traffic assumptions. Nunez-Queija et al. [14] consider a multiple-server model with two priority classes, where the high priority customers may be blocked when all servers are busy, whereas the low priority customers utilize the remaining service capacity in a processor-sharing fashion. For this model, expressions for the blocking probabilities for the high priority customers, and the (conditional) sojourn times for the low priority customers are given, requiring the solution of a set of linear equations. Litjens and Boucherie [11] consider an extension of the model in Nunez-Queija et al. [14] assuming that the high priority customers can be buffered, and propose a numerical approach to calculate the performance parameters of interest.

In this paper, we study the mean sojourn times in a multiple-server queueing model with processor sharing service discipline and two priority classes. For the high priority class, we present closed-form expressions for the mean sojourn times in a general parameter setting, which is a special case of the results obtained by Cohen [6] for the GPS-model (addressed above). For low priority customers, closed-form expressions are derived for the single-server

[1] Note, that Cohen's GPS system is not the same as the in the context of ATM and IP networks well known Generalized Processor Sharing cell/packet scheduling mechanism (also called Weighted Fair Queueing).

case. For the multiple-server case, exact results seem to be attainable only via the numerical solution of a (large) set of linear equations, as e.g. in the similar model analyzed in [14]. We propose and test a *fast-to-evaluate* approximation for the mean sojourn time of the low priority customers. This approximation can be expressed in a closed form formula providing much insight into the impact of the different system parameters. Moreover, the approximation approach can easily be extended to a more general class of models (see Section 5 for details). Numerical results demonstrate that the approximations are highly accurate for a broad range of parameter settings.

The remainder of this paper is organized as follows. In section 2, the precise description of the examined model and the performance measures are discussed. In section 3, exact and approximate evaluation methods are presented. The numerical results used for comparison and validation of the approximation methods are presented in section 4. Finally, section 5 contains concluding remarks and addresses a number of topics for further research.

2 Model

Consider an M/M/C processor sharing model with two priority classes. The C servers are identical and process requests at unit rate. High priority customers have strict priority (preemptive resume) over low priority customers. High and low priority customers arrive according to independent Poisson processes with rates λ_H and λ_L, respectively. The service-time requirements of the high and low priority customers are exponentially distributed with means β_H, and β_L, respectively. The average load of the high and low priority classes is denoted by $\rho_H := \lambda_H\beta_H$ and $\rho_L := \lambda_L\beta_L$, and the total load of the system is denoted by $\rho_{H+L} := \rho_H + \rho_L$. The service process of high priority customers alternates between two modes: a *normal mode* and a *processor sharing mode*. The process is in normal mode when the number of high priority customers does not exceed C. In that case, each high priority customer occupies a single server and is served at unit rate. When the number of high-priority customers is larger than C, the system switches to a processor sharing mode. In that case, the total service capacity C is equally shared among the high priority customers: when there are $n_H \geq C$ high-priority customers in the system, each of these customers is served in a processor sharing fashion with rate C/n_H. Notice that customers are not buffered, and there is no customer blocking. The servers not used by the high priority customers are available for service of the low priority customers. The service process of low priority customers also switches between a normal model and a processor sharing mode. The low priority service process is in normal mode if the total number of customers in the system does not exceed C; in that case, each customer is served by a single server at unit rate. The low priority service process switches to processor sharing mode when the total number of customers exceeds C: when there are C_L servers available for serving low priority customers and there are $n_L \geq C_L$ low priority customers in the system, then each of these customers is served at rate C_L/n_L. When all servers are occupied by the high priority customers (i.e., $n_H \geq C$), the service of the low priority customers is stopped; their service is continued as soon as the number of high priority customers becomes less than C. Recall that the priority rule is pre-emptive resume. The stability condition of the system is $\rho_{H+L} < C$. Throughout, it is assumed the system is stable and in steady state. Denote by S_H and S_L the steady state sojourn time of an arbitrary high priority and low priority customer respectively. In this paper, our main focus is on $E[S_H]$ and $E[S_L]$, i.e.

the mean sojourn times of both high and low priority customers.

3 Analysis

In section 3.1 we present a closed-form expression for the mean sojourn time of the high priority customers, based on the results obtained by Cohen [6] for the Generalized Processor Sharing (GPS) model. In section 3.2 the results in [6] are further exploited for the development of an approximation for the mean sojourn time of the low priority customers.

3.1 Preliminaries; High Priority Traffic

From the model description it is clear that the behavior of the high priority customers is not influenced by the presence of the low priority customers. In particular, the stochastic behavior of the high priority customer class is easily seen to occur as a special case of the so called Generalized Processor Sharing (GPS) model analyzed in Cohen [6]. In the GPS model, whenever there are i customers present in the system, each customer receives service at a rate $f(i)$, where $f(\cdot)$ is an arbitrary function (under some weak assumptions). Notice that in the present model we have $f(i) = 1$ if $0 \leq i \leq C$, and $f(i) = C/i$ if $i > C$. Cohen derives a general, very useful result for the joint stationary distribution of the number of customers N in the GPS system and their residual service requirements $\underline{T} := (T(1), \ldots, T(N))$, cf. formula (7.19) in [6]:

$$Pr[N = n,\ \underline{T} = \underline{\tau}] = \frac{\frac{\rho^n}{n!}\varphi(n)}{\sum_{k=0}^{\infty} \frac{\rho^k}{k!}\varphi(k)} \prod_{i=1}^{n} \frac{1 - B(\tau(i))}{\beta}, \quad n = 0, 1, ..., \quad \tau(i) \geq 0, \tag{1}$$

where $\varphi(0) := 1$ and $\varphi(n) := \left(\prod_{i=1}^{n} f(i)\right)^{-1}$, for $n = 1, 2, \ldots$, and where $B(\cdot)$ denotes the customers' service requirement distribution, β is the mean service requirement, λ is the customer arrival rate and $\rho := \lambda\beta$. It is readily verified that for the present model we have $\varphi(n) := 1$ if $0 \leq n \leq C$, and $\varphi_H(n) := n! C^{C-n}/C!$ if $n > C$. Inserting this into (1) and integrating over all values of the residual service requirements $\tau(i)$ one can derive the following explicit expression for the mean number of customers $E[N_{GPS}(\rho)]$ in the GPS sytem with load ρ:

$$E[N_{GPS}(\rho)] = \frac{C^C}{C! A(C, \rho)} \left(\frac{C(\rho/C)^{C+1}}{1 - \rho/C} + \frac{(\rho/C)^{C+1}}{(1 - \rho/C)^2} \right) + \frac{1}{A(C, \rho)} \sum_{n=1}^{C} \frac{\rho^n}{(n-1)!}, \tag{2}$$

with

$$A(C, \rho) = \frac{C^C}{C!} \frac{(\rho/C)^{C+1}}{1 - \rho/C} + \sum_{n=0}^{C} \frac{\rho^n}{n!}. \tag{3}$$

Note that for the case $C = 1$ our GPS model reduces to the standard single server processor sharing model. Indeed, for that case the above formula yields the well known result (cf., e.g., Kleinrock [10]): For $C = 1$,

$$E[N_{GPS}(\rho)] = \frac{\rho}{1 - \rho}. \tag{4}$$

We now return to our multi-server processor sharing system with priorities. Our observation at the beginning of this section implies that the mean number of high priority customers in the system, $E[N_H]$, is equal to $E[N_{GPS}(\rho_H)]$. Hence, applying Little's formula, the mean sojourn time $E[S_H]$ is given by:

$$E[S_H] = \frac{E[N_H]}{\lambda_H} = \frac{E[N_{GPS}(\rho_H)]}{\lambda_H}, \tag{5}$$

where $E[N_{GPS}(\cdot)]$ is given by (2) and (3).

Remark 3.1
The special form of formula (1) derived by Cohen [6] implies that the stationary distribution of the number of customers in a GPS system is *insensitive* to the service requirement distribution, apart from its first moment. Thus, the above results also hold for the case of generally distributed service requirements.

Remark 3.2
Equations (1), (2) and (3) imply that the mean number of customers in the GPS system, and hence the mean number $E[N_H]$ of high priority customers in our multi server priority system, depends on the arrival rate and service requirement only through their product $\rho = \lambda\beta$.

Remark 3.3
Formula (1) implies that the distributions of the residual service requirements of the individual customers in a GPS system are all equal and independent of the total number of customers present in the system. In particular, this distribution is the so called 'excess distribution' of the initial service requirements well known from renewal theory (see, e.g., Chapter 1 of Tijms [19]). The mean residual service requirement of each of the customers is given by $\frac{\beta_2}{2\beta}$, where β and β_2 are the first and second moment of the initial service requirement distribution $B(\cdot)$, respectively.

3.2 Low Priority Traffic

The analysis of sojourn times for low priority customers is more complicated. The complication is due to the fact that the service rate at which low priority customers are served fluctuates, depending on the variation in the number of high priority customers in the system. In general, an exact mathematical analysis does not seem to be possible. However, we will show that a simple and accurate approximation for the mean sojourn time of the low priority customers exists.

3.2.1 Equal service requirements: $\beta_H = \beta_L$

Before considering the general case, we note that in the special case where the service requirements of both priority classes are the same, the mean sojourn time of the low priority customers can be derived exactly. To this end, note that the mean number of low-priority customers in the system is given by

$$E[N_L] = E[N_{H+L}] \ - \ E[N_H], \tag{6}$$

where N_{H+L} stands for the *total* number of customers in the system, and N_L and N_H are the number of low and high priority customers in the system, respectively. An exact expression for $E[N_{H+L}]$ can be obtained by observing that the dynamic behavior the behavior of our priority system is stochastically identical to that of the 'corresponding' aggregated GPS system described in Cohen [6], i.e. the GPS system where no distinction is made between the priority classes and where customers with exponentially distributed service requirements (with mean $\beta_H = \beta_L$) arrive according to a Poisson process with rate $\lambda_H + \lambda_L$. To this end, note that the state diagrams of the Markov chains describing the dynamics of $N_{H+L}(t)$ (defined as total the number of customers in the system at time t) are identical. More precisely, in both systems the stochastic process $\{N_{H+L}(t), t > 0\}$ can be described as a continuous-time birth-and-death process with state space $S = \{0, 1, \ldots\}$ and with transition rates $t_{i,i+1} = \lambda_H + \lambda_L$, and $t_{i+1,i} = 1/\beta_H = 1/\beta_L$, for $i = 0, 1, \ldots$. Consequently, $E[N_{H+L}] = E[N_{GPS}(\rho_{H+L})]$. From the analysis for the high priority customers in the previous section we have $E[N_H] = E[N_{GPS}(\rho_H)]$. Finally, from (6) and Little's formula, we obtain the mean sojourn time $E[S_L]$ of the low priority customers:

$$E[S_L] = \frac{E[N_{GPS}(\rho_{H+L})] - E[N_{GPS}(\rho_H)]}{\lambda_L}, \tag{7}$$

where $E[N_{GPS}(\cdot)]$ is given by (2) and (3).

3.2.2 Unequal service requirements: $\beta_H \neq \beta_L$

When the service requirements are unequal, the above derivation based on direct analysis of the total number of customers in the system no longer applies. Now the idea is to consider first the total amount of unfinished work in our priority system and in the corresponding GPS system with the same aggregate input traffic (i.e., Poisson arrivals with rate $\lambda_H + \lambda_L$ and hyper exponentially distributed service requirements). In general, the unfinished work processes in these two systems are not exactly the same. However, based on a more detailed comparison, it may be expected that the total unfinished work W_{H+L} in the priority system can be very well approximated by the unfinished work W_{GPS} in the corresponding GPS system. In particular, it is easily verified that for the single server case ($C = 1$) they are exactly the same; and also for high and low system loads it is readily seen that the approximation is very accurate. So our starting point is:

$$E[W_H] + E[W_L] = E[W_{L+H}] \approx E[W_{GPS}]. \tag{8}$$

The total amount of unfinished work in the GPS system can be obtained from Cohen's general result for the joint distribution of number of customers and their residual service requirements, see (1). In particular, from the special 'product' form of this formula it is easily found that $E[W_{GPS}]$ can be expressed as follows (see also Remark 3.3):

$$E[W_{GPS}] = \frac{\beta_2}{2\beta} E[N_{GPS}(\rho_{H+L})], \tag{9}$$

where β and β_2 are the first and second moment of the hyper-exponentially distributed service requirements of the customers in our corresponding GPS model:

$$\beta = \frac{\lambda_H}{\lambda_H + \lambda_L}\beta_H + \frac{\lambda_L}{\lambda_H + \lambda_L}\beta_L \ , \tag{10}$$

$$\beta_2 = \frac{\lambda_H}{\lambda_H + \lambda_L} 2\beta_H^2 + \frac{\lambda_L}{\lambda_H + \lambda_L} 2\beta_L^2 \ . \tag{11}$$

From the discussion in Section 3.1 about the behavior of the high priority customers and from (1) it is clear that,

$$E[W_H] = \beta_H E[N_{GPS}(\rho_H)] \ . \tag{12}$$

Now, applying the approximation step (8), we obtain the following estimation for the mean amount of unfinished work $E[W_L]$ belonging to the low priority customers in the system:

$$E[W_L] \approx E[W_{GPS}] - E[W_H] = \frac{\beta_2}{2\beta} E[N_{GPS}(\rho_{H+L})] - \beta_H E[N_{GPS}(\rho_H)] \ . \tag{13}$$

Next, using the memoryless property of the exponential distribution, the mean amount of unfinished low priority work $E[W_L]$ can easily be related to the mean number of low priority customers $E[N_L]$ in the system:

$$E[N_L] = \frac{E[W_L]}{\beta_L}. \tag{14}$$

Finally, from Little's formula we obtain

$$E[S_L] = \frac{E[N_L]}{\lambda_L} = \frac{E[W_L]}{\rho_L} \approx \frac{\frac{\beta_2}{2\beta} E[N_{GPS}(\rho_{H+L})] - \beta_H E[N_{GPS}(\rho_H)]}{\rho_L}. \tag{15}$$

Remark 3.4
The above approximation yields exact results for the special case of equal service times for high and low priority customers considered in Section 3.2.1. Indeed, in that case $\frac{\beta_2}{2\beta} = \beta_H = \beta_L$ and, hence, the right hand sides of (15) and (7) are equal.

Remark 3.5
In the single server case (i.e., $C = 1$) the total amount of unfinished work in the system is independent of the priority scheme and service discipline (as long as they are work conserving). That is, the approximation step (8) is exact and we obtain from (15) and (4), after some calculations:

$$E[S_L] = \frac{\beta_L}{(1 - \rho_{H+L})} + \frac{\rho_H \beta_H}{(1 - \rho_{H+L})(1 - \rho_H)}. \tag{16}$$

It is easy to verify that this result coincides with the mean sojourn time result obtained by Nunez-Queija [15] for the M/M/1-PS model with an ON/OFF server. To this end, take the ON and OFF periods in Nunez-Queija's model equal to the idle and busy periods of the high priority customers in our model. More precisely, the idle periods are exponentially distributed with mean $1/\lambda_H$, and the first two moments of a busy period of the M/M/1 model (with first-in-first-out service) are given by $m_1 := \beta_H/(1 - \rho_H)$ and $m_2 := 2\beta_H^2/(1 - \rho_H)^3$, respectively.

4 Numerical Results

To assess the accuracy of the approximations discussed in section 3.2.2, we have performed numerous numerical experiments by comparing the approximations with simulation results. The results are outlined below.

First, consider a two-server model with $\lambda_L = 1$, $\beta_L = 1$. Define the asymmetry in the service rates of the two priority classes by $a := \beta_H/\beta_L$. We have calculated the exact and approximated values of the mean sojourn times for the low priority customers for different values of the total load per server (i.e., ρ_{H+L}/C) and the asymmetry parameter a (defined above). The "exact" values have been obtained via simulations, and the approximations have been calculated from (15). Denoting the approximated mean sojourn times of the low priority customers by $E_{approx}[S_L]$, and the exact (simulated) results by $E_{exact}[S_L]$, the relative error of the approximations is defined as follows:

$$\Delta := 100 * abs \left(\frac{E_{approx}[S_L] - E_{exact}[S_L]}{E_{exact}[S_L]} \right), \tag{17}$$

where $abs(\cdot)$ stands for the absolute value. Table 1 shows the expected sojourn times for the low priority customers as a function of the total load per server (i.e., ρ_{H+L}/C) and for $a = 1/4$ and 4.

	$a = 1/4$			$a = 4$		
ρ_{H+L}/C	exact	app	Δ	exact	app	Δ
0.60	1.60	1.59	0.3	1.94	2.00	3.3
0.70	2.07	2.05	0.7	3.31	3.43	3.8
0.80	3.06	3.03	0.9	6.61	6.81	3.6
0.90	6.12	6.08	0.7	18.06	18.30	1.3
0.95	12.38	12.28	0.8	42.53	42.67	0.3

Table 1: Expected sojourn times for low priority traffic for different values of the total load per server ($C = 2$).

The results in Table 1 demonstrate that the approximations are extremely accurate in all considered cases. In fact, in all cases considered in Table 1 the relative error (defined above) in the approximations is less than 4%, and even less than 1% in most cases. To assess the validity of the approximations for a larger number of servers, we have also calculated the results for the model with four servers, with $\lambda_L = 2$ and $\beta_L = 1$. Table 2 shows the results.

	$a = 1/4$			$a = 4$		
ρ_{H+L}/C	exact	app	Δ	exact	app	Δ
0.60	1.19	1.19	0.3	1.28	1.32	3.5
0.70	1.41	1.39	0.9	1.81	1.92	6.1
0.80	1.88	1.86	1.4	3.31	3.50	6.0
0.90	3.40	3.36	1.4	8.88	9.15	3.1
0.95	6.51	6.45	1.0	20.98	21.29	1.5

Table 2: Expected sojourn times for low priority traffic for different values of the total load per server ($C = 4$).

The results in Table 2 also show that the approximations are extremely accurate in all cases considered. In fact, the worst-case scenario was found for the case $a = 4$ and $\rho_{H+L}/C = 0.75$ (i.e., where $\lambda_H = 0.25$ and $\beta_H = 4$), and even in that case the relative error was found to be no more than 6.1%.

Next, we consider the impact of the number of servers on the accuracy of the approximations. To this end, Table 3 shows the exact and approximated mean sojourn times for low priority traffic as a function of the number of servers, while the load per server is kept fixed. Table 2 shows the results for $a = 1/4$, $\beta_H = 0.25$, $\beta_L = 1$, $\rho_L/C = 0.5$, and where λ_H is varied such that the load per server (i.e., ρ_{H+L}/C) takes the values 0.70, 0.80 and 0.90.

	$\rho_{H+L}/C = 0.70$			$\rho_{H+L}/C = 0.80$			$\rho_{H+L}/C = 0.90$		
C	exact	app	Δ	exact	app	Δ	exact	app	Δ
2	2.07	2.05	0.8	3.05	3.03	0.8	6.13	6.08	0.8
4	1.41	1.39	0.9	1.88	1.86	1.4	3.40	3.36	1.2
6	1.21	1.21	0.7	1.52	1.50	1.4	2.52	2.48	1.5
8	1.13	1.12	0.6	1.35	1.33	1.2	2.08	2.05	1.6
10	1.09	1.08	0.4	1.25	1.24	1.1	1.83	1.80	1.6

Table 3: Expected sojourn times for low priority traffic for different numbers of servers for given load per server.

Table 3 illustrates that the approximations are strikingly accurate in all considered cases, and does not depend significantly on the number of servers. The maximum relative error was found to be even less than 2%.

Although the quality of the approximations is extremely good in all considered cases, each approximation almost by definition may become less accurate for specific parameter combinations. Extensive numerical experiments have indicated that the worst-case approximations are consistently found in cases where the following two conditions are met: (1) the mean service time for the high priority customers is large compared to the mean service times of the low priority customers (i.e. when $a = \beta_H/\beta_L$ becomes large), and (2) the total load per server is neither high nor low, say between 0.70 and 0.90. To support this observation, we have also performed numerical experiments with highly asymmetric service rates ($a = 10$), and in that case relative errors up to 10-15% were found. This phenomenon is due to the fact that for a large the approximation step in (8) becomes less accurate. An intuitive explanation may be that in the $(H+L)$-system (i.e., the aggregated system without priorities, with arrival rate $\lambda_H + \lambda_L$ and with hyper-exponentially distributed service times with the first two moments given in (10) and (11)) the customers with relatively large service-time requirements receive a lower service rate than in the priority system. Consequently, the total mean amount of work (primarily caused by the high priority customers) in the priority system is expected to be smaller than in the $(H+L)$-system. In this context, it is important to note that this situation is of minor importance because in practice $\beta_H << \beta_L$, i.e., the service requirements of high priority transactions are generally small compared to the service requirements of low priority transactions. Refinement of the approximations presented here is beyond the scope of the present paper, and is an interesting topic for further study.

5 Conclusions and Topics for Further Research

To summarize, in this paper we have studied the average sojourn time in a multi-server processor sharing model with two priority classes. We presented a closed-form expression for the mean sojourn time for the high priority customers. For the low priority customers, an exact expression for the mean sojourn time is presented for the single-server case. For the multi-server case, in the absence of exact detailed results, we have derived a simple and fast-to-evaluate closed-form approximation of the mean sojourn times. Numerical results demonstrate that the approximation is extremely accurate in many cases.

The results presented in this paper lead to a number of challenging topics for further research. First, in the present paper it is assumed that the customers are served in a processor sharing fashion, which seems to be suitable for the modeling of servers that are CPU-bound. Examples are multi-threaded HTTP Web servers with significant server-side scripting (cf. [2]). In many applications however, it is more natural to assume that the customers at the queues are served in a first-in-first-out (FIFO) order. Examples of such applications are file servers requiring a significant amount of disk I/O operations, so that the performance bottleneck is the disk I/O speed, rather than the CPU capacity. This type of applications leads to FIFO-type, rather than PS-type of queueing models. Analysis of multi-server FIFO queuing models with multiple priority classes is a challenging topic for further research. Second, an interesting issue that may have a significant impact on the performance of multi-threaded servers is locking. For example, if a thread of execution needs to access data stored in shared memory locking issues may occur frequently. Analyzing and quantifying the impact of locking mechanisms on server performance is an interesting area for further research. Third, in the present model we consider two priority classes. However, the results in this paper can be extended models with an arbitrary number of priority classes in a straightforward manner. So far, we have not yet considered the accuracy of the approximation methods in models with a multiple priority classes, which addresses an interesting topic for further research. Fourth, in the present paper it is assumed that the arrival processes are homogeneous Poisson arrival processes. However, in many applications the arrival processes are highly correlated, so that the Poisson assumption is not realistic. For instance, the arrival process of transaction requests at HTTP Web servers is characterized by successive burst of arrivals. To capture the impact of correlated arrivals on the processing times of servers with correlated arrivals, the Poisson assumption needs to be relaxed. Analysis of the model with non-Poisson type of arrival processes is topic for further research. Fifth, in this paper it is assumed that the service times are exponentially distributed. It is interesting to investigate whether the results can be generalized to a more general class of service-time distributions. In this context, notice that the results for the high priority class (based on [6]) only depend on the service-time distribution through the mean, so that the results for the high priority class can be extended easily to non-exponential service-time distributions. For the low priority customers, a similar insensitivity property is not generally valid, leaving an interesting challenge for further research. Finally, in this paper we studied the processing times for a single network node, whereas in practice applications require a sequence of transactions. The latter requires extension of the single-node results to a multi-node network environment. In this context, relevant results have been obtained Choudhury and Houck [4] and Whitt [20]. It is a topic for further research to extend the single-node results in the present paper to a network environment with multiple network nodes.

Acknowledgments: The authors are indebted to E.M. Bongenaar, R. Boucherie, S.C. Borst and R. Nunez-Queija for their contributions to this project.

References

[1] P. Barford and M. Crovella (1999). A performance evaluation of hyper text transfer protocols. Proceedings of the *ACM Sigmetrics International Conference on Measurement and Modeling of Computer Systems*, 188-197.

[2] J.L. van den Berg and O.J. Boxma (1991). The M/G/1 queue with processor sharing and its relation to a feedback queue. *Queueing Systems* 9, 365-401.

[3] J.L. van den Berg (1990). *Sojourn times in Feedback and Processor Sharing Queues.* Ph.D. Thesis, University of Utrecht.

[4] G.L. Choudhury and D. Houck (1994). Combined queueing an activity network based modeling of sojourn time distributions in distributed communication systems. In: *Proceedings of ITC-14* (eds. J. Labetoulle and J.W. Roberts), 525-534.

[5] M.E. Crovella, R. Frangioso and M. Harchol-Bacher (1999). Connection scheduling in Web servers. Proceedings of the *USENIX Symposium in Internet Technologies and Systems.*

[6] J.W. Cohen (1979). The multiple phase service network with generalized processor sharing. *Acta Informatica* 12, 245-284.

[7] J. Dilley, R. Friedrich, T. Jin and J. Rolia (1998). Web server performance measurements and modeling techniques. *Performance Evaluation* 33, 5-26.

[8] W.K. Ehrlich, R. Hariharan, P.K. Reeser and R.D. van der Mei (2001). Performance of Web servers in a distributed computing environment. To appear in the proceedings of the *17th International Teletraffic Congress* (Salvador, Brazil).

[9] J. Heidemann, K. Obraczka and J. Touch (1997). Modeling the performance of HTTP over several transport protocols. *IEEE Transaction on Networking* 5, 616-630.

[10] L. Kleinrock (1976). *Queueing Systems, Vol. II.* Wiley, New York.

[11] R.M. Litjens and R. Boucherie (2000). Radio resource sharing in an GSM/GPRS network. In: *Proceedings ITC Specialists Seminar on Mobile Systems and Mobility* (ed. P.J. Emstad), Lillehammer, Norway, 261-274.

[12] R.D. van der Mei, R. Hariharan and P.K. Reeser (2001). Web server performance modeling. *Telecommunications Systems* 16, 361-378.

[13] R.D. van der Mei, W.K. Ehrlich, P.K. Reeser and J.P. Francisco (2000). A decision support system for tuning Web servers in distributed object-oriented network architectures. *ACM Performance Evaluation Review* 27, 57-62.

[14] R. Nunez-Queija, J.L. van den Berg and M. Mandjes (1999). Performance evaluation of strategies of elastic and stream traffic. In: *Teletraffic Engineering in a Competitive World* (eds. P. Key and D. Smith), 1039-1050.

[15] R. Nunez-Queija (2000). Sojourn times in a processor sharing queue with service interruptions. *Queueing Systems* 34, 351-386.

[16] T.J. Ott (1984). The sojourn-time distribution in the M/G/1 queue with processor sharing. *J. Appl. Prob.* 21, 360-378.

[17] L.P. Slothouber. A Model of Web Server Performance. `http://louvx.biap.com/whitepapers/performance/overview/`.

[18] P.K. Reeser, R.D. van der Mei and R. Hariharan (1999). An Analytic Model of a Web Server. In: *Teletraffic Engineering in a Competitive World*, eds. P. Key and D. Smith (Elsevier, Amsterdam), 1199-1208.

[19] H.C. Tijms (1986). *Stochastic Modelling and Analysis.* Wiley, New York.

[20] W. Whitt (1983). The queueing network analyzer. *The Bell Systems Technical Journal* 62, 2779-2812.

[21] Yashkov, S.F. (1983). A derivation of response time distribution for a M/G/1 processor sharing queue, *Problems Contr. & Info. Theory* 12, 133-148.

[22] Yashkov, S.F. (1992). Mathematical problems in the theory of processor-sharing queueing systems. *Journal of Soviet Mathematics* 58, 101-147.

[23] Yashkov, S.F. (1987). Processor-sharing queues: some progress in analysis. *Queueing Systems* 2, 1-17.