Volume 35, Issue 5, September 2007

ISSN 0167-6377

ELSEVIER

Operations Research Letters

Available online at www.sciencedirect.com
ScienceDirect

Operations Research Letters 35 (2007) 591–594

Operations Research Letters
www.elsevier.com/locate/orl

Waiting times in queues with relative priorities

Moshe Haviv[a],*, Jan van der Wal[b],[c]

[a]Department of Statistics, The Hebrew University of Jerusalem, 91905 Jerusalem, Israel
[b]Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands
[c]Faculty of Economics and Business, University of Amsterdam, Amsterdam, The Netherlands

Received 3 March 2006; accepted 6 October 2006
Available online 1 December 2006

Abstract

This paper determines the mean waiting times for a single server multi-class queueing model with Poisson arrivals and relative priorities. If the server becomes idle, the probability that the next job is from class-i is proportional to the product between the number of class-i jobs present and their priority parameter.

Keywords: Non-preemptive relative priorities; $M|G|1$ queues

1. Introduction

In this paper we consider the following single server multi-class queueing model with *relative* priorities, first suggested in [4]. There are N classes of jobs generated by independent Poisson arrival processes with arrival rate λ_i for class i. Associated with class-i is a positive priority parameter p_i. If upon service completion there are n_j jobs of class-j, $1 \leqslant j \leqslant N$, then the next job to commence service is from class-i with probability

$$\frac{n_i p_i}{\Sigma_{j=1}^{N} n_j p_j}, \quad 1 \leqslant i \leqslant N. \tag{1}$$

Once a job has started service, it is served without interruption until completion. Since we look only at the first moment of queueing times, the entrance discipline among jobs belonging to the same class can be any *non-anticipating* discipline such as, FCFS, LCFS without preemption, or random order. (Recall that a queueing discipline is said to be non-anticipating if decisions regarding which job commences service next, are taken independently of actual service requirements.) For technical reasons we assume random entrance within each class.

In Section 2 we derive the expected waiting time of a job given its class. These expected values are functions of the following parameters: (1) the arrival rates of all classes, (2) the first and second moments of the service requirements of all classes, denoted by $\overline{x}_i$ and $\overline{x^2}_i$, respectively, $1 \leqslant i \leqslant N$, and (3) the priority parameters. As this model is within the framework of

* Corresponding author.
E-mail addresses: haviv@mscc.huji.ac.il (M. Haviv), jan.v.d.wal@tue.nl (J. van der Wal).

0167-6377/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2006.10.003

the parameterized families of priority regimes defined in [6], any vector of mean waiting times which is in the interior of the convex-hull defined by the $N!$ extreme points corresponding to some order of *absolute* priorities, is achievable by an appropriate selection of *relative* priority parameters. The model in [6] assumes preemptive regimes but the analysis follows verbatim to the set of non-preemptive regimes.

In Section 3 we discuss some closely related relative priority models. In these models lotteries deciding who receives the completed service (or the class-independent product produced by the server) are performed at instants of service completions.

Our model is related to the well-known model of *discriminatory processor sharing* (DPS), see the seminal paper [3] or the recent survey [1]. An essential difference with DPS is that for DPS all jobs in the system have already received some service, whereas in our relative priority model, apart from the job in service, all jobs are 'fresh'. This explains why for our model closed-form expressions for the mean waiting times can be obtained for general service distributions, whereas for DPS service times have to be exponential to achieve this goal.

2. The non-preemptive relative priority model

We first derive the mean remaining queueing time for a job given its class, starting from the service commencement of an arbitrary job and conditioning on the numbers of other jobs from all classes. From this we derive the mean queueing time upon arrival. Recall that within a class, we assume random order of entrance to service. The results below, for these conditional mean waiting times hold only under this service regime. The unconditional mean waiting times, however, are discipline independent, provided it is non-anticipative. (Other quantities such as the variance of the waiting time or the mean waiting times conditional on the system's state, do depend on the discipline.)

Let $\rho_i = \lambda_i \overline{x}_i$ be the traffic intensity associated with class i, $1 \leqslant i \leqslant N$, and define

$$\tau_i = \sum_{j=1}^{N} \rho_j \frac{p_j}{p_i + p_j}, \quad 1 \leqslant i \leqslant N,$$

where $p_i/(p_i + p_j)$ is the probability that if a class-i job and a class-j job happen to be in the queue at the same time, then the class-i job is the first among the two to enter service. Then τ_i is the traffic intensity of jobs that turn out to have priority over a given class-i job: $\lambda_j p_j/(p_i+p_j)$ is the effective arrival rate of class-j jobs that overtake the class-i job, $\rho_j p_j/(p_i+p_j)$ their traffic intensity and τ_i the sum over all classes.

From this we get the following result that is well known for queueing models with absolute priorities. Tag a class-i job at any point in time in the system. Let x be the (expected) amount of work due to other jobs present in the system at that instant and that has to be executed before the tagged job can go into service. Then the total expected waiting time $W_i(x)$ for the tagged job is

$$W_i(x) = \frac{x}{1 - \tau_i}. \tag{2}$$

A formal proof for this can be given along the lines of Cobham [2]. The more intuitive but somehow heuristic argument is that the mean waiting time is the sum of the work found on arrival and all work that arrives while the job is waiting and overtakes it, resulting in the identity $W_i(x) = x + W_i(x)\tau_i$.

2.1. Mean queueing times upon service commencement

Before looking at the total queueing time of a class-i job, let us first look at the part of the queueing time caused by a specific other job, and its offspring, i.e., the jobs that arrive while this job or anyone of its children, ad infinitum, is being served. Denote the delay the class-j job causes the class-i job by D_{ij} and its Laplace Stieltjes transform (LST) by D^*_{ij}. Let further G_j be the service time distribution function for a class-j job and G^*_j its LST. Then, one may verify that

$$\begin{aligned} D^*_{ij}(s) = {} & \frac{p_i}{p_i + p_j} + \frac{p_j}{p_i + p_j} \int_0^\infty \mathrm{e}^{-st} \\ & \times \prod_{l=1}^{N} \sum_{m_l=0}^{\infty} \frac{(\lambda_l t D^*_{il}(s))^{m_l}}{m_l!} \mathrm{e}^{-\lambda_l t}\, \mathrm{d}G_j(t) \\ = {} & \frac{p_i}{p_i + p_j} + \frac{p_j}{p_i + p_j} \\ & \times \int_0^\infty \mathrm{e}^{-st - \sum_{l=1}^{N} \lambda_l t (1 - D^*_{il}(s))}\, \mathrm{d}G_j(t) \end{aligned}$$

$$= \frac{p_i}{p_i + p_j} + \frac{p_j}{p_i + p_j} \times G_j^*\left(s + \sum\nolimits_{l=1}^{N} \lambda_l (1 - D_{il}^*(s))\right). \quad (3)$$

Differentiating this with respect to s and inserting $s=0$, shows that the mean delay caused by the class-j job on a class-i job, to be denoted by A_{ij}, satisfies

$$A_{ij} = \frac{p_j}{p_i + p_j} \bar{x_j} \left(1 + \sum_{l=1}^{N} \lambda_l A_{il}\right), \quad 1 \leqslant j \leqslant N. \quad (4)$$

Multiplying both sides by λ_j and summing over j shows that $\sum_{l=1}^{N} \lambda_l A_{il} = \tau_i/(1-\tau_i)$, so that we can write this in the equally intuitive form

$$A_{ij} = \frac{p_j}{p_i + p_j} \frac{1}{1 - \tau_i} \bar{x}_j. \quad (5)$$

Note that higher moments for the delay a class-j job inflicts on a class-i job can be found by taking, recursively, derivatives of (3), inserting $s = 0$ and solving the resulting systems of linear equations.

Define $W(i, \underline{n})$ to be the expected remaining queueing time (service time exclusive) for a class-i job given that it is in the system together with $\underline{n} = (n_1, \ldots, n_N)$ *other* jobs, at the moment the lottery deciding who enters next is performed. Then adding up the delays caused by all other jobs gives

$$W(i, \underline{n}) = \sum_{j=1}^{N} n_j A_{ij} = \frac{1}{1 - \tau_i} \sum_{j=1}^{N} \frac{p_j}{p_i + p_j} n_j \bar{x}_j. \quad (6)$$

2.2. *Mean queueing times upon arrival*

Using PASTA and Little's law, we use the solution given in (6) to determine the unconditional mean queueing time at arrival instants. Let W_i be the unconditional mean queueing time and Q_i the unconditional mean number of class-i jobs queueing up for service. Let W_0 be the mean residual amount of work in service, so (by standard renewal arguments) $W_0 = \Sigma_{i=1}^{N} \lambda_i \overline{x^2}_i/2$. Since the conditional queueing time is an affine function of the numbers of jobs found in the queue, the mean queueing time of a job depends on the number of jobs from various classes only through their mean values. In the time from an arrival instant to the moment the server is ready for the next job (this time might be 0 but has expected value W_0), we get a mean number of $\lambda_j W_0$ of class-j arrivals, each causing an additional mean delay of A_{ij}. Thus,

$$W_i = W_0 + \sum_j (Q_j + \lambda_j W_0) A_{ij}, \quad 1 \leqslant i \leqslant N. \quad (7)$$

Multiplying (7) with $1 - \tau_i$ and using the identity $(1 - \tau_i) Q_j A_{ij} = W_j \rho_j p_j/(p_i + p_j)$ (which follows from (5)), we get

$$(1 - \tau_i) W_i = W_0 + \sum_j W_j \rho_j \frac{p_j}{p_i + p_j}, \quad 1 \leqslant i \leqslant N \quad (8)$$

or, equivalently,

$$W_i = W_0 + \sum_j W_j \rho_j \frac{p_j}{p_i + p_j} + \tau_i W_i, \quad 1 \leqslant i \leqslant N. \quad (9)$$

So the mean queueing times can be obtained by solving a linear system:

Theorem 2.1. *Let the $N \times N$ matrix M be defined by*

$$M_{ij} = \begin{cases} -\rho_j \dfrac{p_j}{p_i + p_j}, & 1 \leqslant j \neq i \leqslant N, \\ 1 - \tau_i - \dfrac{\rho_i}{2}, & 1 \leqslant j = i \leqslant N. \end{cases}$$

Then,

$$W_i = W_0 \sum_{j=1}^{N} (M^{-1})_{ij}, \quad 1 \leqslant i \leqslant N. \quad (10)$$

In particular, for $N=2$ and (without loss of generality) $p_1 + p_2 = 1$,

$$W_i = \frac{1 - \rho p_i}{(1 - \rho_1 - p_2 \rho_2)(1 - \rho_2 - p_1 \rho_1) - p_1 p_2 \rho_1 \rho_2} W_0, \quad i = 1, 2,$$

where $\rho = \rho_1 + \rho_2$. Moreover,

$$\frac{W_1}{W_2} = \frac{1 - \rho p_1}{1 - \rho p_2}. \quad (11)$$

So, up to the factor of W_0, the mean queueing times across classes are functions only of the traffic intensities ρ_j and of the priority parameters, p_j, $1 \leqslant j \leqslant N$. (In the case where $N = 2$, only $\rho = \rho_1 + \rho_2$ is needed.)

In the DPS model, mean waiting times can also be found by solving a system of linear equations but only in the case of exponential service requirements. See Eq. (4.12) in [3]. With two classes of jobs, and μ_i^{-1} denoting the mean service requirement of class-i jobs, $i=1,2$, it is shown in [3] that for the DPS model

$$\frac{W_1+1/\mu_1}{W_2+1/\mu_2}=\frac{\mu_2}{\mu_1}\frac{D+\mu_1\rho_2(p_2-p_1)}{D+\mu_2\rho_1(p_1-p_2)}, \tag{12}$$

where $D=\mu_1p_1(1-\rho_1)+\mu_2p_2(1-\rho_2)$. Note that the analysis in [3] is for the total time in the system while ours considers queueing time only. This explains the terms μ_i, $i=1,2$, on the left-hand side of (12).

Interestingly enough, the ratios in (11) and in (12) agree in the case where $\mu_1=\mu_2$ (and $p_1+p_2=1$).

3. Variants: production first, lottery later

Suppose some product is produced first and only then a lottery decides which job receives it. Naturally, we here assume only one service time distribution with mean $\overline{x}$ and second moment $\overline{x^2}$. Otherwise, all is as before. Redefine the expected (residual) waiting time $W(i,\underline{n})$ (service inclusive) faced by a tagged class-i job which is in the system with $\underline{n}$ other jobs upon 'the product is ready' moment. Then this value satisfies (6) (with $\overline{x}$ replacing $\overline{x}_j$), and thus is in fact the same affine function. The translation to arrival instants proceeds in a similar way as before, but W_0 depends on the model under consideration. We distinguish three models depending on the server's behavior after serving the last job in a busy period.

Variant 1. The server stops working as soon as the system empties, and waits for the next job to arrive. With probability $\rho=\overline{x}\Sigma_{i=1}^N\lambda_i$ an arrival finds the server busy so the mean residual production time is $\overline{x^2}/2\overline{x}$ and with probability $1-\rho$ the server is idle and the mean residual service time equals $\overline{x}$. So the expected time until the first product is ready is given by $W_0^{(1)}=\rho\overline{x^2}/2\overline{x}+(1-\rho)\overline{x}$.

Variant 2. When the system empties, the server continues production and if the product is ready before the next arrival, it is scrapped and the server immediately restarts production. Now each of the arrivals faces a residual production time, so $W_0^{(2)}=\overline{x^2}/2\overline{x}$.

Variant 3. The server continues until a product is ready, then the server waits (if necessary) for the next arrival before commencing the production of the next product. In this case, with probability ρ an arrival finds the server busy, resulting in a residual production time of $\overline{x^2}/2\overline{x}$. With probability $1-\rho$ the server is idle, thus the product is ready and the system is empty, so the residual time is 0. So, $W_0^{(3)}=\rho\overline{x^2}/2\overline{x}=\lambda\overline{x^2}/2$.

For all three models (7) holds, though with W_0 being replaced with $W_0^{(1)}$, $W_0^{(2)}$ or $W_0^{(3)}$. More importantly, (8)–(10) hold as well.

Acknowledgment

The research of Moshe Haviv was supported in part by the Israel Science Foundation, Grant no. 237/02.

References

[1] E. Altman, A. Avrachenkov, U. Ayesta, A survey on discriminatory processor sharing, Queueing Systems Appl. 53 (2006) 53–63.

[2] A. Cobham, Priority assignment in waiting line problems, Oper. Res. 2 (1954) 70–76.

[3] G. Fayolle, I. Mitrani, R. Iasnogorodski, Sharing a processor among many job classes, J. Assoc. Comput. Mach. 27 (1980) 519–532.

[4] M. Haviv, J. van der Wal, Equilibrium strategies for processor sharing and queues with relative priorities, Probab. Eng. Inform. Sci. 11 (1997) 403–412.

[6] I. Mitrani, J.H. Hine, Complete parameterized families of job scheduling strategies, Acta Inform. 8 (1977) 61–77.