SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NY 14853-3801

TECHNICAL REPORT NO. 1138

November 1995

On the Computation of Queue Length Probabilities in a Two-Priority Class M/G/1 Queue[1]

by

Refik Güllü, Peter Jackson
and Jack Muckstadt

On the Computation of Queue Length Probabilities in a Two-Priority Class M/G/1 Queue

Refik Güllü

Peter Jackson

Jack Muckstadt

1 Introduction

In this article we present a recursive computational procedure for obtaining the stationary queue length probabilities in a two-priority class M/G/1 queueing system. We assume that there are two classes of jobs arriving to a single server queue, following Poisson processes with arrival rates λ_H (for high priority jobs), and λ_L (for low priority jobs). We suppose that the service times of the jobs have general distributions with distribution functions $F_H(x)$ (for the high priority jobs), and $F_L(x)$ (for the low priority jobs). We specifically analyze the preemptive-resume priority discipline, where a high priority job will interrupt the service of a low priority job, and the low priority job returns to service from the point it left off once the service for all high priority jobs in the system have been completed. First-Come First-Served(FCFS) discipline is used within a priority class. Such queueing systems have a wide variaty of applications in operations research and computer science. For example, a single-machine, multi-product production environment can be modelled using the particular queueing discipline described, where the high priority jobs correspond to make-to-order, and the low priority jobs correspond to make-to-stock products (Carr et al.[1993]). In various such applications one often needs to compute the stationary queue length or the number in the system probababilities corresponding to high and low priority classes. Because of the preemptive discipline, the queue length process for the jobs belonging to the high priority class behaves exactly as in a single-priority M/G/1 queue. However, the queue length process for the jobs corresponding to the low priority class behave considerably different than its single priority M/G/1 correspondance, since the queue length process for those type of jobs is affected by the high priority type of arrivals. Priority queueing systems have attracted considerable attention of both practitionars and researchers. Jaiswal [1968] is solely devoted to various aspects of such queueing systems. Other texts, such as Cohen [1969], also allocate a substantial material to priority queues and its related matters. The steady state probabilities for the number of low priority jobs in the system are characterized in Jaiswal [1968] by means of their generating function. Carr et al. [1993] utilized this generating function to obtain the number in the system probabilities for a system where both low and high priority jobs have the same unit service times. Here, we extend their results and develop procedures that yield the number in the system probabilities under general and possibly non-identical service time distributions for low and high priority jobs. Our main contribution is to develope an algorithm which produces exact queue length probabilities for a wide range of service time distribution functions.

The rest of the paper is organized as follows. In Section 2 we present the notation, and discuss some preliminary definitions for the preemtive-resume priority queueing system. In Section 3, main analytical results of the article are presented. In Section 4, an algorithm is developed for computing the queue length probabilities for the number of low priority jobs in the system. We also demonstrate the algorithm on an example. The conclusions are presented in Section 5.

2 Preliminaries

In this section we define some notation and preliminary definitions. Let L and H denote the indices for the low and high priority jobs respectively.

λ_r: arrival rate for r priority class jobs, $r \in \{L, H\}$,

μ_r: service rate for r priority class jobs, $r \in \{L, H\}$,

$\rho_r = \lambda_r / \mu_r$, $r \in \{L, H\}$,

$\rho = \rho_L + \rho_H$,

Q_r: steady state number in the system for r priority class jobs, $r \in \{L, H\}$,

$F_r(x)$: service time distribution for r priorority class jobs, $r \in \{L, H\}$.

We need to define some terminology used in the context of priority queues. The busy period is defined as the length of time that begins with a customer arriving into an empty system and ends the next time the system becomes empty. The completion time of a unit is defined as the period that begins the instant service begins on a unit and ends the instant the server becomes free to take the next unit of that class. In a preemptive queueing discipline, the completion time of a lower priority job can be much higher than the job's service time. Let $\Pi_L(\theta)$ denote the probability generating function of the stationary number of low priority jobs in the system. Then by Jaiswal, we have, for $\rho < 1$,

$$\Pi_L(\theta) = (1-\rho)\{1 + \frac{\lambda_H}{\lambda_L}\frac{1 - b(\lambda_L(1-\theta))}{1-\theta}\}\{\frac{(1-\theta)c(\lambda_L(1-\theta))}{c(\lambda_L(1-\theta)) - \theta}\}, \tag{1}$$

where $c(\theta)$ and $b(\theta)$ are the Laplace-Stieltjes transforms (LSTs) for the completion time of a low priority item, and the busy period if the low priority items are ignored, respectively. It turns out that $c(\theta)$ satisfies (see Jaiswal)

$$c(\theta) = U_L(\lambda_H(1 - b(\theta)) + \theta),$$

and $b(\theta)$ satisfies

$$b(\theta) = U_H(\lambda_H(1 - b(\theta)) + \theta),$$

where

$$U_r(s) = \int_0^\infty e^{-sx} dF_r(x),\ r \in \{L, H\}.$$

We derive the probability mass function of the low priority jobs in the system from its generating function through differentiation. That is to say, if

$$\Pi_L(\theta) := \sum_{i=0}^{\infty} \Pr\{Q_L = i\}\theta^i,$$

then, for $n = 0, 1, \ldots,$

$$\Pr\{Q_L = n\} = \frac{1}{n!}\frac{d^n}{d\theta^n}\Pi(\theta)|_{\theta=0}. \tag{2}$$

For any function $f(x)$, define $f^{(k)}(a) := d^k f(x)/dx^k|_{x=a}$. Also let $f^{(0)}(x) \equiv f(x)$.

3 Results for Low Priority Jobs

In this section we provide various results that lead to an algorithm for computing the queue length probabilities of the low priority items in the two-priority class queueing system described above. In Proposition 1 we provide a recursive formula for computing the derivatives of the generating function for the number of low priority jobs in the system. The recursion is in terms of the derivatives of LSTs for busy period and completion time random variables. By using the transformation defined in equation (2), we obtain the queue length probabilities (Corollary (1)). Proposition 2, and Corollary 2 provides a method for computing the derivatives of the LSTs of the busy period and the completion time.

Proposition 1 *For all* $k = 1, 2, \ldots,$

$$\begin{aligned}\frac{\Pi_L^{(k)}(\theta)(c(\lambda_L(1-\theta)) - \theta)}{1-\rho} &= (1-\rho)^{-1}\{k\Pi_L^{(k-1)}(\theta) \\ &- \sum_{j=0}^{k-1}\binom{k}{j}(-\lambda_L)^{k-j}c^{(k-j)}(\lambda_L(1-\theta))\Pi_L^{(j)}(\theta)\}\end{aligned}$$

$$
\begin{aligned}
+(-\lambda_L)^{k-1} \quad & \lambda_H \sum_{j=0}^{k} \binom{k}{j} b^{(j)}(\lambda_L(1-\theta))c^{(k-j)}(\lambda_L(1-\theta)) \\
- \quad & k(-\lambda_L)^{k-1}c^{(k-1)}(\lambda_L(1-\theta)) \\
+ \quad & (-\lambda_L)^k(1-\theta+\frac{\lambda_H}{\lambda_L})c^{(k)}(\lambda_L(1-\theta)) \qquad (3)
\end{aligned}
$$

Proof: By induction. Details are provided in the Appendix.□

By setting $\theta = 0$ in (1) we obtain $\Pr\{Q_L = 0\}$. By setting $\theta = 0$ in (3) and dividing by $k!$ yields the following.

Corollary 1

$$
\begin{aligned}
\Pr\{Q_L = 0\} &= (1-\rho)\{1 + \frac{\lambda_H}{\lambda_L}(1 - b(\lambda_L))\}, \\
\Pr\{Q_L = k\} &= \frac{(1-\rho)}{c(\lambda_L)}\{(1-\rho)^{-1}\{k \Pr\{Q_L = k-1\} \\
&- \sum_{j=0}^{k-1} \frac{1}{(k-j)!}(-\lambda_L)^{k-j}c^{(k-j)}(\lambda_L)\Pr\{Q_L = j\}\} \\
\lambda_H \quad & (-\lambda_L)^{k-1} \sum_{j=0}^{k} \frac{1}{k!(k-j)!} b^{(j)}(\lambda_L)c^{(k-j)}(\lambda_L) \\
&- \frac{1}{(k-1)!}(-\lambda_L)^{k-1}c^{(k-1)}(\lambda_L) + \frac{1}{k!}(-\lambda_L)^k c^{(k)}(\lambda_L)(1 + \frac{\lambda_H}{\lambda_L})\}
\end{aligned}
$$

for $k = 1, 2, \ldots$.

It should be noted that the recursive form provided in Corollary 1 for $\Pr\{Q_L = k\}$ is not only a function of the probability terms $\Pr\{Q_L = i\}$ for $i \leq k-1$, but also a function of the derivatives of $b(\theta)$ and $c(\theta)$ evaluated at $\theta = \lambda_L$. In the remainder of this section, we develop recursive formulas for the derivatives of these functions. Define $Z(\theta, x) = Exp\{-[\lambda_H(1-b(\theta))+\theta]x\}$, and $Z^{(k)}(a, x) = \partial^k Z(\theta, x)/\partial\theta^k|_{\theta=a}$. We can rewrite $b(\theta)$, and $c(\theta)$ as

$$
\begin{aligned}
b(\theta) &= \int_0^\infty Z(\theta, x)dF_H(x), \qquad (4) \\
c(\theta) &= \int_0^\infty Z(\theta, x)dF_L(x). \qquad (5)
\end{aligned}
$$

By taking derivatives of equations (4) and (5) for all $k \geq 1$, we obtain

$$b^{(k)}(\theta) = \int_0^\infty Z^{(k)}(\theta, x) dF_H(x), \qquad (6)$$

$$c^{(k)}(\theta) = \int_0^\infty Z^{(k)}(\theta, x) dF_L(x). \qquad (7)$$

Let,

$$\mathcal{Z}_r(\theta, j, i) := \int_0^\infty x^j Z^{(i)}(\theta, x) dF_r(x), r \in \{L, H\}, j \geq 0, i \geq 0.$$

In particular the integrals in (6) and (7) are equal to $\mathcal{Z}_H(\theta, 0, k)$, and $\mathcal{Z}_L(\theta, 0, k)$ respectively. It turns out that it is possible to write the kth derivative of $Z(\theta, x)$ in terms of combination of Zs with lower order derivatives.

Proposition 2 *For* $k = 1, 2, \ldots,$

$$Z^{(k)}(\theta, x) = -xZ^{(k-1)}(\theta, x) + \lambda_H \sum_{j=0}^{k-1} \binom{k-1}{j} x Z^{(j)}(\theta, x) b^{(k-j)}(\theta). \qquad (8)$$

$$\mathcal{Z}_r(\theta, j, k) = -\mathcal{Z}_r(\theta, j+1, k-1) + \lambda_H \sum_{i=0}^{k-1} \binom{k-1}{i} b^{(k-i)}(\theta) \mathcal{Z}_r(\theta, j+1, i), \qquad (9)$$

for all $j \geq 0$, $k \geq 1$, $r \in \{L, H\}$.

Proof: Proof of equation (8) can be found in the Appendix. Equation (9) follows by multiplying both sides of equation (8) with x^j and integrating with respect to $dF_r(x)$. □
By letting $j = 0$ in equation (9), we obtain the following corollary.

Corollary 2 *For all* $k = 1, 2, \ldots,$

$$b^{(k)}(\theta) = \frac{-\mathcal{Z}_H(\theta, 1, k-1) + \lambda_H \sum_{i=1}^{k-1} b^{(k-i)}(\theta) \mathcal{Z}_H(\theta, 1, i)}{1 - \lambda_H \mathcal{Z}_H(\theta, 1, 0)}$$

$$c^{(k)}(\theta) = -\mathcal{Z}_L(\theta, 1, k-1) + \lambda_H \sum_{i=0}^{k-1} b^{(k-i)}(\theta) \mathcal{Z}_L(\theta, 1, i)$$

Although the terms $\mathcal{Z}_r(\theta, 1, i)$, $i = 0, 1, \ldots, k-1$ are difficult to evaluate, the term

$$\mathcal{Z}_r(\theta, j, 0) = (-1)^j U_r^{(j)}(\lambda_H(1 - b(\theta)) + \theta), r \in \{L, H\}, j \geq 1, \qquad (10)$$

the jth derivative of U_r evaluated at $\lambda_H(1 - b(\theta)) + \theta$, is easy to compute for a wide range of distribution functions F_r for all $j \geq 0$. In the Appendix we provide some important random variables where the term $U_r^{(j)}(s)$ can explicitly be obtained. This property will be utilized for obtaining an algorithm that computes the necessary Z_r terms in a recursive manner, and finally evaluates $\Pr\{Q_L = k\}$.

4 The Algorithm

In this section an algorithm is presented that computes the queue length probabilities for the low priority class jobs. Before stating the algorithm, we review the dynamics of the computations required. In order to compute $\Pr\{Q_L = k\}$ for any $k \geq 1$, we need $\Pr\{Q_L = i\}$ for $i = 0, 1, \ldots, k-1$, and $c^{(i)}(\lambda_L)$, $b^{(i)}(\lambda_L)$ for $i = 0, 1, \ldots, k$ as stated in Corollary 1. If one progresses computing $\Pr\{Q_L = k\}$ from $k = 0$, it is trivial to verify that once $\Pr\{Q_L = k-1\}$ has been computed, the terms $\Pr\{Q_L = i\}$ for $i = 0, 1, \ldots, k-1$, and $c^{(i)}(\lambda_L)$, $b^{(i)}(\lambda_L)$ for $i = 0, 1, \ldots, k-1$ are already available. For computing $b^{(k)}(\lambda_L)$ we need $b^{(i)}(\lambda_L)$ for $i = 1, 2, \ldots, k-1$ and $\mathcal{Z}_L(\lambda_L, 1, j)$ for $j = 0, 1, 2, \ldots, k-1$ as stated in Corollary 2. Obviously, after the computation of $\Pr\{Q_L = k-1\}$, the terms $c^{(i)}(\lambda_L)$ for $i = 1, 2, \ldots, k-1$ and hence the terms $\mathcal{Z}_L(\lambda_L, 1, j)$ for $j = 0, 1, 2, \ldots, k-2$ are available. Therefore, the only term required for the computation of $\Pr\{Q_L = k\}$ is the term $\mathcal{Z}_L(\lambda_L, 1, k-1)$. Suppose, we obtain the term $\mathcal{Z}_r(\lambda_L, k, 0)$ by using equation (10). Then, using equation (9) successively, we can obtain $\mathcal{Z}_r(\lambda_L, 1, k-1)$ as a combination of terms with lower order derivatives:

$$
\begin{aligned}
\mathcal{Z}_L(\lambda_L, k-1, 1) &= -\mathcal{Z}_L(\lambda_L, k, 0) + \lambda_H \mathcal{Z}_L(\lambda_L, k, 0) \\
\mathcal{Z}_L(\lambda_L, k-2, 2) &= -\mathcal{Z}_L(\lambda_L, k-1, 1) + \lambda_H \sum_{i=0}^{1} \binom{1}{i} b^{(2-i)}(\lambda_L) \mathcal{Z}_L(\lambda_L, k-1, i) \\
&\vdots \\
\mathcal{Z}_L(\lambda_L, 1, k-1) &= -\mathcal{Z}_L(\lambda_L, 2, k-2) + \lambda_L \sum_{i=0}^{k-2} \binom{k-2}{i} b^{(k-1-i)}(\lambda_L) \mathcal{Z}_L(\lambda_L, 2, i).
\end{aligned}
$$

The algorithm provided below efficiently computes the necessary $\mathcal{Z}_r$ terms required, and finally obtains $\Pr\{Q_L = k\}$.

Algorithm:

Step 0. Solve

$$\begin{aligned} b(\lambda_L) &= \int_0^\infty Z(\lambda_L, x) dF_H(x) \\ &= U_H(\lambda_H(1 - b(\lambda_L)) + \lambda_L) \\ c(\lambda_L) &= \int_0^\infty Z(\lambda_L, x) dF_H(x) \\ &= U_L(\lambda_H(1 - b(\lambda_L)) + \lambda_L) \end{aligned}$$

for $b(\lambda_L)$ and $c(\lambda_L)$ respectively by using a numerical equation solver.

Step 1. Compute $\Pr\{Q_L = 0\}$.

Step 2. For $k = 1, 2, \ldots$

2.1. Compute $\mathcal{Z}_L(\lambda_L, k, 0)$ and $\mathcal{Z}_H(\lambda_L, k, 0)$ by using equation (10).

2.2. Compute $\mathcal{Z}_r(\lambda_L, k - j, j)$ for $j = 1, 2, \ldots, k - 1$ and $r \in \{L, H\}$ by using equation (9).

2.3. Compute $b^{(k)}(\lambda_L)$, and $c^{(k)}(\lambda_L)$.

2.4. Compute $\Pr\{Q_L = k\}$.

As it can be observed from the algorithm, at the kth execution of Step 2.1. and 2.2, it is required to compute $\mathcal{Z}_r(\lambda_L, k, 0)$ for $r \in \{L, H\}$, and $\mathcal{Z}_r(\lambda_L, k - j, j)$ for $r \in \{L, H\}$ and $j = 1, 2, \ldots, k - 1$. This requires the computation of $2k$ terms.

Proposition 3 *The terms $\mathcal{Z}_r(\lambda_L, i, j)$ for $i = 1, 2, \ldots, n$, $j = 0, 1, \ldots, n - i$, $r \in \{L, H\}$ are available at the end of the nth execution of Step 2.2 of the algorithm. In particular, $\mathcal{Z}_r(\lambda_L, 1, n - 1)$ is obtained, which is required for the computation of* $\Pr\{Q_L = n\}$.

Proof. By induction. At the end of the first execution ($n = 1$) we obtain $\mathcal{Z}_r(\lambda_L, 1, 0)$ for $r \in \{L, H\}$ as desired. Suppose the assertion is true for n. At the $n + 1$st execution of Step 2.1, we compute $\mathcal{Z}_r(\lambda_L, n + 1, 0)$ for $r \in \{L, H\}$. In Step 2.2, in order to compute $\mathcal{Z}_r(\lambda_L, n, 1)$ we require $\mathcal{Z}_r(\lambda_L, n + 1, 0)$. For computing $\mathcal{Z}_r(\lambda_L, n - 1, 2)$ we require

$\mathcal{Z}_r(\lambda_L, n, 1)$, $\mathcal{Z}_r(\lambda_L, n, 0)$, where the latter term is available by the induction hypothesis. For any $i = \{1, 2, \ldots, n\}$, in order to compute $\mathcal{Z}(\lambda_L, i, n+1-i)$, equation (9) requires $\mathcal{Z}(\lambda_L, i+1, n-i-j)$ for $j = 0, 1, \ldots, n-i$, where the last $n-i$ terms are available by the induction hypothesis, and the first term is obtained after the execution of Step 2.2. Hence, the terms $\mathcal{Z}_r(\lambda_L, n+1-j, j)$ for $j = 1, 2, \ldots, n$ are computed in the $n+1$st execution of the algorithm. The collection of the terms that are obtained after the $n+1$st execution is $\mathcal{Z}_r(\lambda_L, i, j)$ for $i = 1, 2, \ldots, n+1$, $j = 0, 1, \ldots, n+1-i$ as required. In particular, the term $\mathcal{Z}_r(\lambda_L, 1, n)$ is obtained as desired. □

Example: We compute a few initial probability terms of an example for demonstrating the algorithm. We choose $\lambda_L = 6$, $\lambda_H = 1.5$. Service times are exponentially distributed with $\mu_L = 10$ and $\mu_H = 5$ ($\rho = 09$). In this case (see Appendix), $U_r(s) = \mu_r/(\mu_r + s)$, and $U_r^{(n)}(s) = n!\mu_r(-1)^n/(\mu_r + s)^{n+1}$ for $n \geq 1$. By using Mathematica, we obtain $b(6) = 0.421299$, and $c(6) = 0.592837$. We compute $\Pr\{Q_L = 0\} = 0.1(1 + 0.25(1 - 0.421299)) = 0.114467$.

Appendix: Proofs of Propositions

Proof of Proposition 1: We can rewrite $\Pi_L(\theta)$ as

$$\frac{\Pi_L(\theta)(c(\lambda_L(1-\theta)) - \theta)}{1-\rho} = (1-\theta)c(\lambda_L(1-\theta)) + \frac{\lambda_H}{\lambda_L}(1 - b(\lambda_L(1-\theta)))c(\lambda_L(1-\theta)). \quad (11)$$

Taking the first derivative of (11) with respect to θ yields

$$\frac{\Pi_L^{(1)}(\theta)(c(\lambda_L(1-\theta)) - \theta) + (c^{(1)}(\lambda_L(1-\theta))(-\lambda_L) - 1)\Pi_L(\theta)}{1-\rho} =$$

$$\begin{aligned} - \quad & c(\lambda_L(1-\theta)) - \lambda_L c^{(1)}(\lambda_L(1-\theta))(1-\theta) \\ - \quad & \lambda_H c^{(1)}(\lambda_L(1-\theta))(1 - b(\lambda_L(1-\theta)) + \lambda_H b^{(1)}(\lambda_L(1-\theta))c(\lambda_L(1-\theta)), \end{aligned}$$

which conforms to equation (3) for $k = 1$. Assume (3) holds for some $k \geq 1$. Taking the derivative of (3) with respect to θ yields (after straightforward but tedious algebra) equation (3) for $k+1$. The main identity used in the derivation is $\binom{k}{j} + \binom{k}{j-1} = \binom{k+1}{j}$.

Proof of Equation 8: $Z(\theta,x) = Exp[-(\lambda_H(1-b(\theta)) + \lambda_L)]$. The first derivative of $Z(\theta,x)$ with respect to θ is

$$Z^{(1)}(\theta,x) = xZ(\theta,x)(\lambda_H b^{(1)}(\theta) - 1),$$

which is equation (8) for $k=1$. Assume equation (8) holds for any $k \geq 1$. Taking the derivative of (8) with respect to θ again produces

$$\begin{aligned} Z^{(k+1)}(\theta,x) &= -xZ^{(k)}(\theta,x) + \lambda_H \sum_{j=0}^{k-1} \binom{k-1}{j} x\{Z^{(j+1)}(\theta,x)b^{(k-j)}(\theta) \\ &+ b^{(k+1-j)}(\theta)Z^{(j)}(\theta,x)\}. \end{aligned} \quad (12)$$

Arrangeing terms in (12) and using the identity $\binom{k-1}{j} + \binom{k-1}{j-1} = \binom{k}{j}$ gives equation (8) for $k+1$, which completes the proof.

Appendix: $U_r^{(j)}$ for Some Important Service Time Distributions

m-Point Distribution: Service time for class r jobs is equal to μ_i with probability p_i, for $i = 1,2,\ldots,m$. In this case

$$U_r^{(n)}(s) = (-1)^n \sum_{i=1}^{m} \mu_j^n e^{-s\mu_i} p_i, \quad n \geq 0.$$

Erlang Distribution: Service time for class r jobs is Erlang-α distributed with mean α/μ_r. Then, for all $n \geq 0$

$$U_r^{(n)}(s) = \frac{\alpha(\alpha+1)\cdots(\alpha+n)\mu_r^n(-1)^n}{(\mu_r+s)^{n+\alpha}}.$$

Uniform Distribution: Service time for class r jobs is uniformly distributed over $(0, a_r)$. In this case $U_r(s) = (1-e^{-sa_r})/a_r s$, and

$$U_r^{(n)}(s) = \frac{-(-a_r)^{n-1}e^{-sa_r} - nU_r^{(n-1)}(s)}{s}, \quad n \geq 1.$$