The Response Times of Priority Classes under Preemptive Resume in $M/M/m$ Queues

J.P. Buzen

BGS Systems, Inc., Waltham, Massachusetts

A.B. Bondi

Department of Computer Sciences,
Purdue University, West Lafayette, Indiana

CSD-TR No. 307

ABSTRACT

Expressions are derived for the mean response times of each priority level in a multi-server $M/M/m$ queue operating under preemptive resume scheduling. Exact results are obtained for cases where all priorities have the same mean service times; approximate results are obtained for the more general case where mean service times may differ. The results hold for any number of servers and any number of classes. For each priority level, it is assumed that arrivals are Poisson and service times are exponentially distributed.

*This work was supported by NSF grant number MCS78-01720.
Several authors have analyzed the behavior of M/M/1 multiple class queues operating under preemptive resume priority. For the two class case, White and Christie (1958) have produced generating functions for the steady state queue length distributions. Marks (1973) has also produced generating functions for this problem, but in a form which lends itself to an efficient algorithm for computing the joint steady state queue length probabilities.

The methods used in these papers are not easily generalized to the multiple server case, particularly when the classes have different mean service times. However, while it is not easy to extract the joint queue length distributions for this problem, it is possible to give expressions for certain performance measures. For the multiple server case in which all priority classes have the same mean service time, Brosh (1969) gives an expression for the expected time from arrival to inception of service and establishes bounds for the expected response times and queue lengths of the different customer

*This work was supported by NSF grant number MCS78-01720.*
classes. For lower priority customers, the response time cannot be obtained directly from Brosh's result as the service period will be subject to interruptions.

Other results on multiple servers and priorities have been published by Taylor and Templeton (1980) and Abolnikov and Yasnogorodsky (1974). Their papers deal with a discipline in which priority is given to especially urgent jobs (e.g., in an ambulance service) so long as the number of busy servers exceeds a set threshold. Preemption is not used here and unanswered urgent requests are lost.

Obtaining the response times of multiple server preemptive priority queues when the levels have different mean service times has been described by Heyman (1977) as a particularly important unsolved problem. Recently, Mitrani and King (1981) have obtained the mean response times for \( M/\bar{M}/m \) queues with two priority levels and an arbitrary number of servers. The service times for each priority level are exponentially distributed and their means need not be the same. The solution obtained by Mitrani and King is expressed in terms of generating functions for the joint and marginal queue length distributions that must be evaluated numerically for each set of model parameters. The solution of the equations is difficult, and, as the authors note, subject to numerical instability. For more than two levels of priority, Mitrani and King suggest a heuristic in which all levels of priority save the lowest are amalgamated into one high priority class and are assumed to have the same service rate within that class. The system is then evaluated as a two-priority system with the lowest priority class at the second level. This heuristic will be discussed later.

This paper presents exact results for the mean queue length and response time at each priority level in an \( M/\bar{M}/m \) queue in which all customer classes have the same mean service time. These results, which are presented in Sections 2 and 3, are applicable to systems with an arbitrary number of priority classes. The derivation of these results is reasonably straightforward.

In Section 4, the case where mean service times differ is considered. A new approximate algorithm for computing the mean response times at each priority level is
derived. This algorithm which is based on the exact solution obtained in Section 3, is shown to yield good results in a series of examples.

Both the exact and the approximate solutions are computationally stable and involve only the direct evaluation of simple algebraic expressions. These results are applicable to the study of computer systems with multiple processors whose jobs are subject to preemptive resume priority scheduling.

1. Notation

Consider a preemptive resume M/M/m queueing system with customers at r levels of priority such that

class i customers have priority over class j customers if \( i \leq j \leq r \).

customers within the same priority class follow the FCFS discipline.

customers of class i arrive in a Poisson stream with rate \( \lambda_i \), and have i.i.d. exponential service times with mean \( 1/\mu_i \).

Denote the mean queue length of class i customers by \( \bar{n}_i \) and their mean response time by \( R_i \). We shall also use \( \bar{n}(p) \) to denote the sum of the first \( p \) values of \( \bar{n}_i \), and \( \lambda(p) \) to denote the sum of the first \( p \) values of \( \lambda_i \), and \( \bar{R}(p) \) to denote the overall average of the mean response times of the \( p \) highest priorities. Thus,

\[
\bar{n}(p) = \sum_{i=1}^{p} \bar{n}_i, \quad p = 1, 2, ..., r. \tag{1}
\]

\[
\lambda(p) = \sum_{i=1}^{p} \lambda_i, \quad p = 1, 2, ..., r. \tag{2}
\]

By Little's Law,

\[
\bar{R}(p) = \sum_{i=1}^{p} \frac{\lambda_i R_i}{\lambda(p)} \quad p = 1, 2, ..., r. \tag{3}
\]

To ensure the existence of finite waiting times for the \( p \) highest priority classes (Cobham [1955]), also assume that the total traffic intensity satisfies

\[
\rho(p) = \sum_{i=1}^{p} \left( \lambda_i / m \mu_i \right) < 1
\]

Our analysis makes use of the standard expressions for the mean response time \( K \)
of a simple FCFS $M/M/m$ queue with arrival rate $\lambda$ and service rate $\mu$. By Kleinrock (1976)

$$R = \frac{1}{u} + \frac{\rho P_m(\rho)}{u(1-\rho)}$$

(4)

where

$$\rho = \frac{\lambda}{m \mu}$$

(5)

$$P_m(\rho) = \frac{\lambda^m (m \rho)^m}{m! (1-\rho)}$$

(6)

and

$$P_0(\rho) = \left( \sum_{i=0}^{m-1} \frac{(m \rho)^i}{i!} \frac{(m \rho)^m}{m! (1-\rho)} \right)^{-1}$$

(7)

$P_m(\rho)$ is the probability of having $m$ or more customers in an $M/M/m$ queue with traffic intensity $\rho$, while $P_0(\rho)$ is the probability of the queue being empty.

2. Two Priority Classes with Equal Mean Service Times

For the special case where $\tau = 2$ and $\mu_1 = \mu_2 = \mu$, expressions for the response times of the individual classes can be easily derived using the following two intuitive assumptions, which are proved in the appendices.

**Assumption A**

The mean response time of the highest priority jobs, $R_1$, is equal to the mean response time of an $M/M/m$ queue with arrival rate $\lambda_1$ and mean service rate $\mu$ under FCFS scheduling. Thus, the low priority jobs may be disregarded when computing $R_1$, the response time of the high priority jobs. In general, lower priority customers have no impact on the mean response time of higher priority customers. The proof is given in Appendix A.

**Assumption B**

The average response time taken over all customers, $\bar{R}_{(2)}$, is equal to the mean response time of an $M/M/m$ queue with arrival rate $\lambda_1 + \lambda_2$ and mean service rate $\mu$ under FCFS scheduling. Thus, while preemptive resume scheduling will affect the mean response time of the individual classes, the average response time of the combined
classes will be the same as under FCFS scheduling. The proof is given in Appendix B.

Note that Assumption B depends strongly on the condition that \( \mu_1 = \mu_2 = \mu \) and that all service times are memoryless (exponential). In this case changing the service discipline from FCFS to preemptive resume leaves the departure process at the server exactly the same.

By Assumption A, the value of \( R_1 \) may be obtained by evaluating equation (4) with \( a = \lambda_1 \) and \( u = \mu \). Similarly, Assumption B implies that \( \bar{R}_{(2)} \) may be obtained by evaluating equation (4) with \( a = \lambda_{(2)} \) and \( u = \mu \).

Note that equation (3) implies that

\[
R_2 = \frac{\lambda_{(2)} \bar{R}_{(2)} - \lambda_1 R_1}{\lambda_2} \tag{8}
\]

Substituting the values of \( R_1 \) and \( \bar{R}_{(2)} \) obtained from equation (4) above, we have

\[
R_2 = \frac{1}{\mu} + \frac{\mu_2 \rho_{(2)}(\rho_{(2)} - \rho_{(1)})}{\lambda_2 (1 - \rho_{(2)})} - \frac{\rho_{(1)} \mu_2 (\rho_{(1)})}{\lambda_2 (1 - \rho_{(1)})} \tag{9}
\]

Equation (9) describes the mean response time of the low priority jobs in an \( M/M/1 \) queue with two priority classes when both have the same mean service time. As already noted, Xitrani and King (1981) present a considerably more complex expression for \( R_2 \), but without requiring that \( \mu_1 = \mu_2 \).

3. The Solution for \( p \geq 2 \) Priority Levels with Equal Mean Service Times

If \( p \geq 2 \), equation (3) implies that

\[
\bar{R}_{(p)} = \frac{\sum_{i=1}^{p} \lambda_i R_i}{\lambda_{(p)}} = \frac{\sum_{i=1}^{p-1} \lambda_i R_i}{\lambda_{(p)}} + \lambda_p \frac{R_p}{\lambda_{(p)}} = \frac{\lambda_{(p-1)} R_{(p-1)} + \lambda_p R_p}{\lambda_{(p)}}, \quad p = 2, 3, \ldots, r \tag{10}
\]

Thus,

\[
R_p = \frac{\lambda_{(p-1)} R_{(p-1)} - \lambda_{(p-1)} R_{(p-1)}}{\lambda_p}, \quad p = 2, 3, \ldots, r \tag{11}
\]
To apply equation (11), it is necessary to know the values of $\bar{R}_p(p)$. The following assumption can be used to derive these quantities. Like Assumptions A and B, this assumption is intuitively appealing. It is proved rigorously in Appendix C.

**Assumption C**

The mean response time of the aggregate of the first $p$ priority classes, $\bar{R}_p$, is equal to the mean response time of an $K/\lambda/m$ queue with arrival rate $\lambda_p$ and service rate $\mu$ under FCFS scheduling, for $p=1,2,\ldots,r$. In other words, the overall mean response time for the first $p$ priority classes would remain the same if the scheduling discipline were changed to FCFS and jobs in classes $p+1,r+2,\ldots,r$ were eliminated. This is proved rigorously in Appendix C. By setting $r=p=2$, Assumption C follows trivially from Assumption C. Similarly, Assumption A follows trivially from Assumption C by setting $p=1$.

Assumption C implies that the values of $\bar{R}_{(p-1)}$ and $\bar{R}_p$ may be obtained directly from equation (4) by setting $u=\lambda_{(p-1)}$ and $u=\lambda_p$ respectively, and $u=\mu$. Substituting these values into equation (11) implies

$$R_p = \frac{1}{\mu} + \frac{\rho_p \rho_m \rho_{(n)}}{\lambda_p (1-\rho_p)} - \frac{\rho_{(p-1)} \rho_m \rho_{(n)}}{\lambda_p (1-\rho_{(p-1)})}, p=2,3,\ldots,r$$

where

$$\rho_p = \frac{\lambda_p}{m \mu}, p=1,2,3,\ldots,r$$

For $p=1$, $R_1$ is given as in equation (4) with $u=\lambda_1$ and $u=\mu$ since Assumption A still applies.

This expression for $R_p$ is exact for any number of classes $r$ and any number of servers $m$, provided that $\mu_1=\mu_2=\ldots=\mu_r=\mu$. The evaluation of equation (12) is straightforward and poses no special numerical difficulties.

4. **Approximate Solution For Unequal Mean Service Times**

When the requirement that all mean service times must be equal is relaxed, the same basic strategy can still be used to obtain the mean response times for each prior-
ility level; however, the exact values of $R_{(p)}$ are not readily available, since Assumption C no longer holds. Therefore, approximate expressions for $R_{(p)}$ will be derived for $p = 2, 3, \ldots, r$. Substitution of these expressions into equation (11) will then yield approximate values of $R_p$.

Some additional notation is required to facilitate discussion of this problem. Let $\bar{\mu}(p)$ denote the mean service rate, weighted by arrival rate, of the $p$ highest priority classes.

$$\bar{\mu}(p) = \frac{1}{1} \sum_{i=1}^{p} \frac{\lambda_i}{\mu_i}$$

Let $R(d, \mu(p), \lambda(p), m)$ denote the mean response time of the $p$ highest classes in an $\infty / \infty / m$ queue operating under discipline $d$, where $\mu(p)$ and $\lambda(p)$ are $p$-vectors of service and arrival rates respectively and where $d$ is either FCFS or PRI.

When the $\lambda_i$'s are all equal to $\bar{\mu}(p)$ in an $\infty / \infty / m$ system, Assumption C may be restated as

$$R(PRI, \mu(p), \lambda(p), m) = R(FCFS, \mu(p), \lambda(p), m)$$

Now, consider the case when the individual values of the $\lambda_i$'s are not necessarily equal. In this case, equation (13) does not hold because the mean time spent waiting in the queue is not the same under FCFS and PRI scheduling. That is, the ratio

$$\eta = \frac{R(PRI, \mu(p), \lambda(p), m) - 1/\bar{\mu}(p)}{R(FCFS, \mu(p), \lambda(p), m) - 1/\bar{\mu}(p)}$$

is not equal to unity in general unless $\mu_1 = \mu_2 = \cdots = \mu_p = \bar{\mu}(p)$.

Equation (14) defines $\eta$ as the ratio of two waiting times. In the discussion which follows, $W(\ldots, \ldots)$ will be used to denote the waiting time of a system with response time $R(\ldots, \ldots)$. Thus, equation (14) may be rewritten as

$$\eta = \frac{W(PRI, \mu(p), \lambda(p), m)}{W(FCFS, \mu(p), \lambda(p), m)}$$

The ratio $\eta$ is primarily influenced by the magnitude of the differences between the
mean service times of the individual classes. If the \( \mu_i \)'s are approximately equal, \( \eta \) should be very close to unity since converting from PRI to FCFS will only have a small effect on the departure process. Conversely, if the \( \mu_i \)'s are very dissimilar, converting from PRI to FCFS could have a substantial effect on the departure process and \( \eta \) could differ significantly from unity. Hence, any modification of the system which preserves the ratios between the \( \mu_i \)'s and also preserves the traffic intensity should not affect the value of \( \eta \) significantly.

One such modification is to replace the \( m \) servers by a single server \( m \) times as fast. This preserves the ratios between the \( \mu_i \)'s, while leaving the traffic intensities the same, and yields the following approximation:

\[
\frac{W(PRI, m \mu(p) \lambda(p), m)}{W(FCFS, m \mu(p) \lambda(p), m)} \approx \frac{W(PRI, m \mu(p) \lambda(p), 1)}{W(FCFS, m \mu(p) \lambda(p), 1)}
\]

(15)

Rearranging to obtain approximations to the quantities needed in equation (11).

\[
W(PRI, m \mu(p) \lambda(p), m) \approx W(PRI, m \mu(p) \lambda(p), 1) \cdot \frac{W(FCFS, m \mu(p) \lambda(p), m)}{W(FCFS, m \mu(p) \lambda(p), 1)}
\]

(16)

To evaluate the right hand side of (16), note first that \( R(PRI, m \mu(p) \lambda(p), 1) \) is derivable from the response time of the individual classes, \( r_k \), in an \( M/M/1 \) preemptive-resume priority system (Kleinrock [1976]). In this case, \( \tau_1 \) is given by

\[
\tau_1 = \frac{1}{m \mu_1 - \lambda_1}
\]

and \( \tau_k \) for \( k \geq 2 \) is given in Kleinrock (1976) as

\[
\tau_k = \frac{(1/m \mu_k)(1-\rho_k) + \sum_{i=1}^{k} \lambda_i / \mu_i}{{(1-\rho_k) - \sum_{i=1}^{k} \lambda_i / \mu_i}}
\]

(17)

Finally, the average response time over the \( p \) highest priority classes is given by

\[
R(PRI, m \mu(p) \lambda(p), 1) = \frac{1}{\lambda(p)} \sum_{i=1}^{p} \lambda_i \tau_i
\]

(18)

Now consider the remaining terms in equation (16). Let

\[
T = \frac{W(FCFS, m \mu(p) \lambda(p), m)}{W(FCFS, m \mu(p) \lambda(p), 1)}
\]

(19)

The numerator and denominator in this expression could, in principle, be derived by
analyzing the appropriate $M/M_p/m$ and $M/M_p/1$ queues respectively, but this approach is numerically complex. Instead, consider the factors that influence $\gamma$.

Note that the $m$ server queue in the numerator and the single server queue in the denominator have the same arrival process parameters $\lambda(p)$. Also, the service completion processes are identical whenever all servers in the $m$ server queue are active. On the other hand, as the number of active servers in the $m$ server queue decreases from $m$ to 1, the service completion processes in the two queues become increasingly dissimilar.

Since the number of active servers is primarily a function of the overall traffic intensity, this suggests that the traffic intensity is the most important factor influencing $\gamma$, and that the relative differences between the values of the $\mu_i$'s have only a minor impact. Thus, transforming the numerator and denominator of equation (19) in a way that preserves the traffic intensities but eliminates the differences between the $\mu_i$'s should yield a satisfactory approximation for $\gamma$.

One may make such a transformation by replacing the vector of service rates $\mu(p)$ by a vector of $p$ identical service rates, each equal to $\bar{\mu}(p)$. The numerator and denominator of equation (19) will then correspond to the mean waiting times in simple $M/M/m$ and $M/M/1$ queues with service rates $\bar{\mu}(p)$ and $m\bar{\mu}(p)$ respectively, and arrival rate $\lambda(p)$. This follows from the superposition principle for Poisson processes and the fact that all $p$ service times are exponential with the same mean.

Using equation (15) to evaluate the numerator and denominator, the approximate value of $\gamma$ is given by

$$\gamma \approx \frac{p_m(p)}{\rho(p)}$$

Combining equation (20) with (16) and (18), the values of $\bar{R}(p)$ needed in equation (11) may be approximated as follows:

$$\bar{R}(p) = \frac{1}{\bar{\mu}(p)} + \frac{1}{\bar{\mu}(p)} \left[ \frac{\rho}{m\bar{\mu}(p)} \sum_{k=1}^{m} \frac{\rho}{\lambda(p)} \right]$$

$$\approx \frac{1}{\bar{\mu}(p)} + \left[ \frac{1}{\lambda(p)} \sum_{k=1}^{m} \lambda_k r_k - \frac{1}{m\bar{\mu}(p)} \right] \frac{p_m(p)}{\rho(p)}$$

(21)
where the values of \( \tau_k \) for \( k=1,2,...,r \) are given in equation (17). The approximate values of \( R_k \) (\( k \geq 2 \)) may then be derived directly from equation (11). \( R_1 \) may be evaluated as though the other classes did not exist, using equation (4) with \( a = \lambda_1 \) and \( \mu = \mu_1 \).

This approximate method of obtaining response times may be implemented cheaply. For the case of two priority levels, it produces values for \( R_p \) which are well within 5% of the exact results presented by Khatani and King (1981). Comparisons of their figures and ours for the same input data appear in Tables I and II. The greatest error occurs for a system with a large number of servers (10) in which the high priority class has many times the traffic intensity of the low priority class.

The results of Khatani and King (1981) are exact only for the two-class case. For \( p \) classes (\( p > 2 \)), they suggest amalgamating all the classes save the lowest into one class having mean service rate \( \mu_{(p-1)} \), and treating this as the highest priority class. As they point out, this could be a source of error, as it assumes that service time differences among higher priority classes will have no impact on the lower priority classes. Hence, in the Khatani and King heuristic, \( R_p \) depends on \( \mu_p \) and \( \mu_{(p-1)} \), but not on the individual values of \( \mu_1, \mu_2, ..., \mu_{p-1} \). By contrast, the approximate method presented in this paper is sensitive to these values because they are used explicitly in the computations of the \( \tau_k \)'s in equation (17). It should be noted that both methods of computing the individual response times will be exact when all classes of customers have the same mean service time.

5. Conclusion

A simple method for computing the exact mean response times of individual customer classes in an \( M/M/m \) preemptive resume priority system has been given for the case where all customer classes have equal mean service times. This method has been supported by explicit derivations given in the appendices. For the case of unequal mean service times, an approximate solution, motivated by the method used in the simpler case of equal mean service times, has been presented. This approximation is
applicable to an arbitrary number of customer classes. It attempts to account for the
influence of preemption on each priority level as the parameters of each class are con-
sidered. Comparisons with published results for the two priority case, where an exact
solution exists, show that the approximation is accurate to within 5% in most cases.
Because of its logical consistency and ease of implementation, the approximation
should enjoy a wide range of applications to the modeling of priority systems.

Acknowledgements

This research was conducted during the summer of 1981, while A.B. Bondi was
working at BGS Systems as a research assistant. Support was provided by NSF grant
number MCS79-01729 which also funded Mr. Bondi's travel to Waltham from West Lafay-
ette. Computing facilities were provided by the Department of Computer Sciences at
Purdue University. We would like to thank Subhash C. Agrawal for his valuable com-
ments during this research. We also wish to thank Dr. Daniel Heyman of Bell Labora-
tories for his suggestions on an earlier draft of this paper and for pointing out that the
approximate expression for \( \gamma \) in equation (20) may also be obtained from equation (19)
using the Pollaczek-Khinchin formula in the denominator and an approximation for the
numerator given in equation (11) of Nozaki and Ross (1970).

Appendix A: The Steady State Distribution of High Priority Customer Queue Length

Notation:

Let \( n = (n_1, n_2, ... n_r) \) be the vector describing the queue lengths of classes
1,2,3,...,r, i.e., the state of the system. Also, let \( \omega_i \) denote the elementary \( r \)-vector with
1 in the \( i \)th position and 0's everywhere else. Define the following indicator functions:

\[
\begin{align*}
  f(n, m) = & 0, n < 0 \\
  = & 0, n \leq 0 \\
  = & \min(n, m) n, m \geq 0
\end{align*}
\]

and

\[
\begin{align*}
  g(n) = & 0, n < 0, \\
  = & 1, n \geq 0.
\end{align*}
\]
For $f$, $n$ denotes the number of customers, and $m$ denotes the number of servers available to the $n$ customers.

Also, let

$$n_0 = 0$$
$$n_j = \sum_{i=1}^{j} n_i, \quad j = 1, 2, \ldots, r$$

$P(n)$ denotes the probability that the system is in state $n$.

With this notation, the steady state equations are given by

$$(\lambda(r) + f(\sum_{i=1}^{r} n_i, m) \mu) P(n) = \sum_{i=1}^{r} g(n_i - 1) \lambda_i P(n - e_i)$$
$$+ \mu \sum_{i=1}^{r} f(n_i + 1, m - n_i - 1) P(n + e_i)$$

(A1)

To obtain the marginal distribution of $n_1$, sum (A1) over all $n$ such that $n_1 = k$ for all $k$.

Define

$$n_k = \sum_{n_1 = k} P(n_1, n_2, \ldots, n_r)$$

Then, we have, on the left hand side of (A1)

$$(\lambda(r) + f(\sum_{i=1}^{r} n_i, m) \mu) P(n) =$$

$$= \lambda(r) \sum_{n_1 = k} P(n) + \mu f(k, m) \sum_{n_1 = k} P(n)$$
$$+ \mu \sum_{n_1 = k} f(\sum_{i=2}^{r} n_i, m - k) P(n)$$
$$= \lambda_1 n_k + (\lambda(r) - \lambda_1) n_k + \mu f(k, m) n_k$$
$$+ \mu \sum_{n_1 = k} \left[ f(\sum_{i=2}^{r} n_i, m - k) P(n) \right]$$

(A2)

On the right hand side we have

$$\sum_{n_1 = k}^{r} g(n_i - 1) \lambda_i P(n - e_i) + \mu \sum_{n_1 = k}^{r} f(n_i + 1, m - n_i - 1) g(m - n_i) P(n + e_i)$$

$$= \lambda_1 g(k - 1) n_k + (\lambda(r) - \lambda_1) n_k + \mu f(k + 1, m) g(m) n_k$$
$$+ \mu \sum_{n_1 = k}^{r} f(n_i + 1, m - n_i - 1) g(m - n_i) P(n + e_i)$$

(A3)

Because the set of $n$ over which we are summing is infinite, and because terms with
\( \pi_i = 0 \) \((i \geq 2)\) make no contribution to the rates of flow between states, the last term in (A3) is equal to the last term in (A2). It therefore follows that

\[
(\lambda_1 + f(k,m)\mu)\pi_k = \gamma(k-1)\lambda_1\pi_{k-1} + f(k+1,m)\mu\pi_{k+1}, \quad k \geq 0
\]  \hspace{1cm} (A4)

Note that (A4) is the steady state equation for an \( M/M/m \) system with FCFS discipline, arrival rate \( \lambda_1 \), and service rate \( \mu \), regardless of how many classes there are.

Thus, the highest priority customers will have the queue length distribution of an \( M/M/m \) system with arrival rate \( \lambda_1 \) and service rate \( \mu \), regardless of the other classes of customers.

Hence, the response time of the highest priority class may be computed as described in Assumption A.

**Appendix B: The Steady State Distribution of the Aggregate Class**

Let \( p_k \) be the probability that there are \( k \) customers in the system in total, \( k = 0,1,2,... \) Regardless of the combination of customer class types, the aggregate customer completion rate of the system will be \( k\mu \) if \( k \leq m \) and \( m\mu \) otherwise. Furthermore, if there are \( k \) customers in the system, the number will be increased to \( k+1 \) at rate \( \lambda_1 + \lambda_2 + ... + \lambda_r = \lambda_{(r)} \), since the arrival process of all classes of customers is Poisson. Therefore, the steady state equations of the aggregate system must be given by

\[
(\lambda_{(r)} + f(k,m)\mu)p_k = \gamma(k-1)\lambda_{(r)}p_{k-1} + f(k+1,m)\mu p_{k+1}
\]  \hspace{1cm} (B1)

where \( f \) and \( \gamma \) are defined as in Appendix A. Hence, the aggregate queue length distribution is the same as that of an \( M/M/m \) system with arrival rate \( \lambda_{(r)} \) and service rate \( \mu \).

**Appendix C: The Steady State Distribution of the Total of the \( p \) Highest Priority Class Customers in the System**

For Assumption C, it is sufficient to show that the combination of classes \( 1,2,...,p \) will have the same queue length distribution as an \( M/M/m \) queue with arrival rate \( \lambda_{(r)} \) and service rate \( \mu \) for \( p = 1,2,...,r \).

Let \( \omega_k \) be the steady state probability that there are \( k \) jobs of classes \( 1,2,...,p \) in the system. Then
\[ \omega_k = \sum_{n(p)=k} P(n) \]

The steady state equations for the \( \omega_k \)'s are obtained by summing (A1) over all \( n \) such that \( \sum_{i=1}^{r} n_i = k \).

Consider the left hand side of (A1) first. The coefficient of \( P(n) \) is

\[ \lambda(\sigma_i) + f \left( \sum_{i=1}^{r} n_i, m \right) \mu = \lambda(\sigma_i) + \left( \lambda(\sigma_i) - \lambda(\sigma_j) \right) + f \left( \sum_{i=1}^{r} n_i, m \right) \mu + f \left( \sum_{i=1}^{r} n_i, m \right) \mu \]

Therefore, upon summation, the left hand side of (A1) becomes

\[ (\lambda(\sigma_i) + f (k, m) \mu) \omega_k + (\lambda(\sigma_i) - \lambda(\sigma_j)) \omega_k + \sum_{n(p)=k} \sum_{i=2}^{r} f (\sum_{i=1}^{r} n_i, m - k) \mu \omega_k \]

For the arrival terms on the right hand side, we have

\[ \sum_{n(p)=k} \sum_{i=1}^{r} g(n_i) \lambda_i P(n) \]

since the outer sum is infinite and \( g(n) \) is an indicator function taking the value 0 or 1.

This reduces to

\[ \sum_{n(p)=k} \sum_{i=1}^{r} g(n_i) \lambda_i P(n) = (\lambda(\sigma_i) - \lambda(\sigma_j)) \omega_k \]

Thus, the term in \( \lambda(\sigma_i) - \lambda(\sigma_j) \) on the left hand side in (C2) is balanced by an equal term on the right hand side, and they cancel.

The coefficient of \( \mu \) on the right hand side is

\[ \sum_{n(p)=k} \sum_{i=1}^{r} f (n_i+1, m-n_i+1) P(n + a_k) + \sum_{n(p)=k} \sum_{i=1}^{r} f (n_i+1, m-n_i+1) P(n + a_k) \]

\[ = \sum_{n(p)=k} \sum_{i=1}^{r} f (n_i+1, m) P(n + a_k) + \sum_{n(p)=k} \sum_{i=1}^{r} f (n_i+1, m-n_i+1) P(n + a_k) \]

\[ = f (k, m) \omega_k + \sum_{n(p)=k} \sum_{i=1}^{r} f (n_i, m-n_i) P(n + a_k) \]

since the sum over all \( n \) such that \( n(p)=k \) is infinite, as before.
The last term in (C5), when multiplied by $\mu$, cancels with the last term in (C2). It follows that

$$\lambda_{(p)}(k)\omega_k = (\lambda_{(p)}(k-1)\omega_{k-1} + \mu f(k+1,n)\omega_{k+1}, \ k \geq 0 \tag{C6}$$

Hence, the $p$ highest priority classes combined have the same queue length distribution as an $M/M/m$ system with arrival rate $\lambda_{(p)}$ and service rate $\mu$ regardless of the lower priority classes, as required. This is sufficient to prove that Assumption C is correct.

Consequently, one may quickly obtain the result described in Appendix B, merely by setting $p=r$ in equation (C6). Also, the queue length distribution of the highest priority class may be obtained immediately by setting $p=1$.

References


### Table 1: Comparison of Approximate and Exact Response Times for N Servers

<table>
<thead>
<tr>
<th>N</th>
<th>Priority Ordering</th>
<th>Arrival Rates</th>
<th>Service Rates</th>
<th>Traffic Intensity</th>
<th>Class 1</th>
<th>Class 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N*1.000</td>
<td>2.000</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1 2</td>
<td>1.309</td>
<td>1.309</td>
<td>0.000</td>
<td>10.465</td>
<td>10.691</td>
</tr>
<tr>
<td></td>
<td>2 1</td>
<td>1.061</td>
<td>1.064</td>
<td>0.002</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>4</td>
<td>1 2</td>
<td>0.873</td>
<td>0.873</td>
<td>0.000</td>
<td>5.689</td>
<td>5.899</td>
</tr>
<tr>
<td></td>
<td>2 1</td>
<td>1.142</td>
<td>1.146</td>
<td>0.003</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>6</td>
<td>1 2</td>
<td>0.716</td>
<td>0.716</td>
<td>0.000</td>
<td>4.073</td>
<td>4.233</td>
</tr>
<tr>
<td></td>
<td>2 1</td>
<td>0.897</td>
<td>0.899</td>
<td>0.003</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>8</td>
<td>1 2</td>
<td>0.642</td>
<td>0.642</td>
<td>0.000</td>
<td>3.262</td>
<td>3.412</td>
</tr>
<tr>
<td></td>
<td>2 1</td>
<td>0.766</td>
<td>0.766</td>
<td>0.003</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>10</td>
<td>1 2</td>
<td>0.692</td>
<td>0.693</td>
<td>0.000</td>
<td>2.720</td>
<td>2.900</td>
</tr>
<tr>
<td></td>
<td>2 1</td>
<td>0.699</td>
<td>0.699</td>
<td>0.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

The exact response times in this row and the next differ from the corresponding ones in Mitrani and King (1981). We have been informed by Dr. King that the numbers shown here are the correct ones.
Table II: Comparison of Approximate and Exact Response Times for N Servers

Both Classes with Moderate Traffic

<table>
<thead>
<tr>
<th></th>
<th>Class 1</th>
<th></th>
<th>Class 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival Rates</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.450</td>
<td>0.450</td>
<td>0.450</td>
<td>0.000</td>
<td>0.300</td>
</tr>
<tr>
<td>Service Rates</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000/N</td>
<td>1.000/N</td>
<td>1.000/N</td>
<td>0.000</td>
<td>2.000/N</td>
</tr>
<tr>
<td>Traffic Intensity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.45</td>
<td>0.45</td>
<td>0.45</td>
<td>0.000</td>
<td>0.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1 2</td>
<td>2.506</td>
<td>2.508</td>
<td>0.000</td>
<td>3.003</td>
<td>3.008</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>2 1</td>
<td>3.320</td>
<td>3.319</td>
<td>0.000</td>
<td>1.023</td>
<td>1.023</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>1 2</td>
<td>4.234</td>
<td>4.234</td>
<td>0.000</td>
<td>3.574</td>
<td>3.574</td>
<td>0.010</td>
</tr>
<tr>
<td></td>
<td>2 1</td>
<td>4.859</td>
<td>4.850</td>
<td>0.002</td>
<td>2.002</td>
<td>2.002</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>1 2</td>
<td>5.110</td>
<td>5.110</td>
<td>0.000</td>
<td>4.151</td>
<td>4.150</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>2 1</td>
<td>5.994</td>
<td>5.983</td>
<td>0.002</td>
<td>3.000</td>
<td>3.000</td>
<td>0.000</td>
</tr>
<tr>
<td>8</td>
<td>1 2</td>
<td>8.060</td>
<td>8.063</td>
<td>0.000</td>
<td>4.000</td>
<td>4.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>2 1</td>
<td>8.425</td>
<td>8.414</td>
<td>0.001</td>
<td>4.000</td>
<td>4.000</td>
<td>0.000</td>
</tr>
<tr>
<td>10</td>
<td>1 2</td>
<td>10.034</td>
<td>10.034</td>
<td>0.000</td>
<td>5.626</td>
<td>5.626</td>
<td>0.004</td>
</tr>
<tr>
<td></td>
<td>2 1</td>
<td>10.310</td>
<td>10.300</td>
<td>0.002</td>
<td>5.000</td>
<td>5.000</td>
<td>0.005</td>
</tr>
</tbody>
</table>