

LABOR REQUIREMENTS FOR MULTI-SERVER MULTI-CLASS FINITE QUEUES*

by

Fred F. Easton
Robert H. Brethen Operations Management Institute
School of Management, Syracuse University
Syracuse, NY 13244-2130
(315) 443-3463
ffeaston@syr.edu

File name: H₂/H₂/C/N_062902.doc
Revised: June 29, 2002

Working paper: please do not quote, cite, or reproduce without permission of author.

LABOR REQUIREMENTS FOR MULTI-SERVER MULTI-CLASS FINITE QUEUES

ABSTRACT

Most of North America's 70,000 call centers use Erlang C (M/M/C) or Erlang B (M/M/C/N) queueing models to determine the minimum staffing levels needed to meet quality of service (QOS) objectives. However, many of these centers serve two or more populations, each with distinct arrival and service time distributions. While some characteristics of multi-class queues can be accurately approximated by single class models such as Erlang C or B, this approach understates the service time variance (a mixture of exponential distributions) for multi-class systems and consequently, the probability of extreme delays. As a result, these popular approximation procedures often underestimate minimum staffing levels for a given QOS goal.

To help address this issue, we develop a finite multi-server queueing model for a system that provides two types of service, classified $H_2/H_2/C/N$. We show the system is reversible, allowing efficient computation of its two-dimensional state probabilities and other common queueing statistics. We then compare $H_2/H_2/C/N$'s estimates of the behavior of two-class finite queues with those obtained from M/M/C/N. In general, we find that the accuracy of M/M/C/N's estimates for average queue time, QOS, and minimum staffing levels diminishes as the ratio of the two service rates departs from unity. For example, with service rate ratios of 24, M/M/C/N underestimates minimum staffing requirements for QOS goals by 11 - 33 percent, depending on system size and traffic intensity.

LABOR REQUIREMENTS FOR MULTI-SERVER MULTI-CLASS FINITE QUEUES

1. INTRODUCTION

For most businesses, including North America's 70,000 call centers (Robinson, 1999) extreme queue times lead to customer dissatisfaction, renegeing, and lost sales (Davis, 1991). To determine the minimum staffing levels needed to assure acceptable customer waiting times, managers often rely on queueing models like the Erlang C (M/M/C/ ∞) delay model and the Erlang B (M/M/C/N) loss model (Robertazzi, 1990; Mehrotra, Profozich, & Bapat, 1997; Reynolds, 1998; Leamon, 1999). In fact, Erlang C is built into virtually all commercial workforce management software (Cleveland, 1999).

Erlang B and C models assume equally proficient calltakers that handle a single type of call with FCFS priority and patient customers. Erlang B also assumes that maximum system occupancy is constrained to at most N , the number of available telephone trunks. Interarrival times and service times are assumed to be exponential i.i.d. random variables with means $1/\lambda$ and $1/\mu$, respectively. However, many familiar queueing systems serve more than one distinct population or class of customers, each with its own arrival and service time distributions.

For example, catalog merchandisers handle calls for both new orders and more lengthy RMA/service inquiries (Andrews & Parsons, 1989). Residential energy distributors handle calls for both simple account changes (turn-ons, turn-offs) and more complex customer billing and payment problems. Forty percent of the "911 calls" received by some emergency call centers are actually inquiries about road conditions, the weather, or other inappropriate requests for service that dispatchers terminate as quickly as possible (Wolcott, 1999). Nationwide, 911 call centers

often receive multiple reports of the same incident (Chen, 1999). Emergency dispatchers usually have enough information to respond to the incident after answering the first call or two, and tend to process subsequent reports much faster than the initial calls.

Erlang B and C models assume exponential inter-arrival and service time. Thus, they can only approximate the behavior of multi-class systems with inter-arrival and service time distributions that are mixtures of exponentials. According to Leamon (1999), there are two common strategies for approximating the minimum staffing levels needed to meet Quality of Service (QOS) objectives with single-class queueing models. One option is to average the inter-arrival and service times over all classes to obtain the parameters Erlang C needs to determine ideal staffing levels. Leamon (1999) warns this approach understates the true staffing levels needed to meet quality of service objectives. Alternatively, managers can separate the arrival streams, compute the minimum labor requirements for each type of service independently, then sum the results. Whitt (1999b) notes this approach tends to overstate minimum staffing requirements. Both types of errors may have significant economic consequences for service operations. However, the continued reliance on Erlang model approximations of multi-class queues implies that managers may not be aware of the potential magnitude of these errors.

A custom simulation model of a multi-class queue could yield more accurate estimates of the relationship between staffing levels and queue times. However, such models are expensive to develop and maintain and generally have to be run repeatedly to return a reliable estimate of system performance. They are rarely used to support day-to-day operating decisions like calculating hourly staffing levels (Mehrotra et al. 1997). By contrast, closed-form queueing models like Erlang C require little computational effort and allow generic labor management systems to estimate ideal staffing levels for a variety of service businesses. These rapid

calculations also permit labor management programs to evaluate the economics of alternative scheduling solutions (Easton & Rossin, 1996).

Multi-class queueing models for both single and multiple servers have previously been addressed in the literature. Polling models (Takagi, 1990) are single server queueing systems that process more than one type of job. However, call centers typically employ multiple servers, so polling models are unlikely to improve on the estimates obtained with Erlang C. Multiple customer classes are also considered in priority queueing systems (Gross & Harris, 1998). However, these systems assume priorities other than FCFS.

Fayolle, King, & Mitrani (1982) and Kao & Wilson (1999) investigated multi-server infinite queues with two customer classes. Fayolle et al (1982) developed steady-state conditions for limited state dependencies and specialized servers, acknowledging that closed-form solutions for general state probabilities remained an open question. Kao & Wilson (1999) evaluated alternative approximation procedures for various performance measures such as average queue length. Because call centers have finite queues, however, approximations for infinite series may be unnecessary. Furthermore, while metrics such as average time in queue are useful, today's call center managers often staff to achieve Quality of Service objectives (such as the percentage α of arriving callers who are in queue for β or fewer seconds). Thus, staffing decisions usually require information about the entire queue time distribution (Segal, 1974).

Whitt (1999a) considered multi-server queueing systems that provide two different types of service and recursively estimated the mean and variance of the queue time distribution for a newly arrived caller. Whitt's goal was to provide a new arrival with an accurate waiting time estimate by exploiting information about the number of customers in each class who are *in service* when the new caller arrives. However, in call centers where new arrivals present as

members of a single class (say, callers who dial 911), it may be difficult to classify callers until the service is completed, even with technology such as ANI or DNIS (Crisafulli, 1998).

In this paper, our goal is to characterize the steady-state behavior of a two-class multi-server finite queue and demonstrate the potential for staffing errors that managers can expect when they approximate queue behavior with standard (Erlang) queueing formulae. In section 2, we present a 2-dimensional birth-death model for the proposed system, show that it is reversible, and derive its state probability equations and its queue time distribution. In Section 3, we compare our model results to those for a single class M/M/C/N model with both averaged and separated multi-class arrival and service time data, and demonstrate that estimation errors increase to disturbing levels as the distance between the means of the two service time distributions increase. We present our conclusions and suggest possible extensions to this work in Section 4.

2. Model $H_2/H_2/C/N$

The system we wish to study operates as follows. New arrivals originate from either of two large populations, class 1 or class 2. The arrival rates for customers from class 1 and class 2 are Poisson-distributed random variables with mean arrival rates λ_1 and λ_2 , respectively. Both types of arrivals present at the same entry point, but new callers are denied entry (i.e. blocked) whenever system occupancy = N. Customers admitted to the system advance through the queue FCFS. Consistent with the assumptions of Erlang B & C, we assume patient customers who never renege before their service is initiated. Upon reaching one of the C identical servers, the customer's needs are established and the service is completed at an average rate of μ_1 or μ_2 , depending on the type of service required. The customer then exits the system. A schematic of this system is shown in Figure 1.

(please insert Figure 1 about here)

In most queueing situations, waiting time is an important performance metric. Call centers can avoid long customer waits, and the subsequent degradation of perceived quality (Taylor, 1994), by matching their service capacity to call volume (Sasser, 1976). Generally, capacity decisions in call centers are driven by the organization's QOS goals (Segal, 1974; Andrews and Parsons, 1993). QOS can be expressed as the fraction α of arriving customers who are queued less than some pre-determined time β , or α/β . The economic implications of alternative QOS levels are discussed in Andrews & Parsons (1993), Davis (1991), and Easton & Rossin (1996).

Virtually all call center workforce management systems utilize some variation of Erlang C (M/M/C/ ∞) to determine minimum staffing levels (Durr, 1994; Cleveland, 1999). Suppose, however, that an organization provides two distinct services with average processing times of $1/\mu_1$ and $1/\mu_2$, respectively, and that arrivals from the two independent classes occur at rates λ_1 and λ_2 , respectively. If we assume that both the arrival and service processes are mixtures of exponential distributions, this system may be classified as H₂/H₂/C/N. The aggregate arrival rate for the system is $\lambda = \lambda_1 + \lambda_2$. The mean service time for this mixture of classes is:

$$1/\mu = \left(\frac{1}{\lambda_1 + \lambda_2} \right) \left(\frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} \right). \quad (1)$$

Single class queueing models assume queued customers advance in line every $1/C\mu$ time units, on average, until reaching a server. In multi-class queueing systems, however, the expected time to the next completion depends on the mix of customers currently in service. Therefore, the mean queue time estimates produced by M/M/C/N models for multi-class systems may be unreliable. Furthermore, the service time distribution for M/M/C/N is exponential, with variance $1/\mu^2$. In H₂/H₂/C/N, the service times for two customer classes are independent and

exponentially distributed, so the variance for this mixture of service times is the weighted sum of their variances, or:

$$\sigma^2 = \left(\frac{1}{\lambda_1 + \lambda_2} \right) \left(\frac{\lambda_1}{\mu_1^2} + \frac{\lambda_2}{\mu_2^2} \right) \quad (2)$$

Service time variance $\sigma^2 > 1/\mu^2$ whenever $\mu_1 \neq \mu_2$, so Leamon's (1999) contention that Erlang C underestimates the minimum staffing level necessary to achieve a desired QOS seems valid.

To better understand the queue time distribution and other aspects of system behavior for $H_2/H_2/C/N$, we require the following notation:

NOTATION:

- C The number of servers available to process transactions
- N The maximum occupancy of the system, including customers in service and in queue.
- P_{ij} The probability that exactly i type 1 customers and j type 2 customers are in the system.
- λ_1, λ_2 The mean arrival rates for customer classes 1 and 2, assumed to be Poisson distributed.
- μ_1, μ_2 The mean service rate for an agent processing type 1 and type 2 transactions, respectively, assumed to be Poisson distributed.
- ρ_i λ_i/μ_i
- $E_1[i,j,C]$ ($E_2[i,j,C]$) the expected number of agents deployed to Type 1 (Type 2) customers when there are C agents available and there are i type 1 customers and j type 2 customers in the system. Equivalently, the expected number of Type 1 (Type 2) customers in service when the system is at state (i,j) .
- $L_{ijC}[U_{ijC}]$ The probability of one Type 1 [Type 2] service completion in the next instant, given the system is currently in state (i,j) and staffed with C servers, defined $U_{ijC} = E_1(i,j,C)\mu_1$ and $L_{ijC} = E_2(i,j,C)\mu_2$.

2.1 Birth-Death Diagram

In a finite queue, system occupancy is limited to a maximum of N simultaneous connections, which could be a mix of Type 1 and Type 2 customers. In Figure 2, the nodes represent possible system states. We assume that the probability of occupying a particular state is stationary over a relevant planning horizon (say, the next 30 minutes) and unlikely to change from one instant to the next. The arcs represent possible transitions between adjacent states (due to a new arrival or a service completion) that could take place during a small interval of time. The values adjacent to each arc represent the probability of exactly one new arrival or one service completion during an infinitesimally small time interval.

(Please insert Figure 2 about here)

2.2 Expected Server Deployments (Number Of Callers In Service)

If there are more than C customers in the system, some of them will be queued until a server is available. If a large proportion of the queued customers belong to the class with longer service times, their waits could be much greater than average. Since we can't always classify arrivals until after the service is completed, managers may not know with certainty how many customers of each class are in service at a particular instant. However, for a given system state (i,j) , we can estimate the expected number of type 1 and type 2 callers in service at any instant with the hyper-geometric distribution.

To illustrate, suppose the system is staffed by $C = 3$ agents and consider state (i,j) , where the system is occupied by $i = 2$ type 1 callers and $j = 2$ type 2 callers. The first server will be handling a type 1 call with probability $i/(i+j)$ and a type 2 call with probability $j/(i+j)$. The deployment of the first server influences the probabilities for the type of calls the remaining servers can receive. For example, if server 1 is handling a type 1 call, the probability that the

second server also has a type 1 call is reduced to $(i-1)/(i+j-1)$; for a type 2 call the probability increases to $j/(i+j-1)$. For this form of sampling without replacement, the probability that exactly X callers of a particular type are in service when there are C servers, i type 1 callers and j type 2 callers is given by the PMF for the hyper-geometric distribution. The expected number of type 1 and type 2 callers in service when C agents are on duty and the system is occupied by i type 1 callers and j type 2 callers is obtained from the expectations for this distribution. That is, for i and j integers ≥ 0 ,

$$E_1(i, j, C) = \frac{\min\{C, i+j\}i}{i+j} \quad (3)$$

and

$$E_2(i, j, C) = \frac{\min\{C, i+j\}j}{i+j} \quad (4)$$

2.3 Flow Balance Equations and Reversibility

If the system is stationary, the probability of leaving a particular state during any one instant must equal the probability of returning from any of its up to four neighboring states during that same instant. In general, this gives rise to a set of two-dimensional flow balance equations, one for each state, with the form:

$$\begin{aligned} \text{State } (i, j) \quad & P_{ij}(\lambda_1 + \lambda_2 + E_1(i, j, C)\mu_1 + E_2(i, j, C)\mu_2) = \\ & P_{i-1, j}\lambda_1 + P_{i, j-1}\lambda_2 + P_{i, j+1}E_2(i, j+1, C)\mu_2 + P_{i+1, j}E_1(i+1, j, C)\mu_1. \end{aligned} \quad (5)$$

For a finite queue with maximum occupancy of N , there are $(N+1)(N+2)/2$ such linear balance equations with an equal number of state probabilities whose values we wish to determine. These state probabilities can be computed with numerical methods (Fayolle et al., 1982). However, the analysis of stationary queueing systems is greatly simplified if the system is reversible in time, since we can estimate state probabilities from the probability flux between

any pair of adjacent states (Nelson, 1993). For example, if the two-class multi-server finite queue we are considering is reversible, the probability of occupying a given state simplifies to:

$$P_{ij} = P_{i-1,j} \frac{\lambda_1}{E_1(i,j,C)\mu_1} = P_{i,j-1} \frac{\lambda_2}{E_2(i,j,C)\mu_2} \quad \forall (i,j) \mid i+j \geq 1 \quad (6)$$

A reversible Markov process is statistically identical when viewed in either the forward or backward direction from an arbitrary point in time (Kolmogorov, 1936). Kelly (1979) proves a simple, sufficient condition for reversibility in a stationary Markov process: if its graph forms a tree (e.g., a connected graph with one fewer reversible arcs than nodes). However, the birth-death process for the two-class multi-server queue is a tree only when $N = 1$ (see Figure 2).

When maximum system occupancy is greater than one, it is easy to find non-backtracking pathways in Figure 2 that begin and end at the same node. Fortunately, a tree structure is not a necessary condition for reversibility. It can also be established by algebraic methods or from Kolmogorov's criterion (Kelly, 1979).

Theorem 1: the 2-class multi-server finite queue is a reversible Markov process, and thus equation (6) holds for all states $(i,j) \mid i+j \geq 1$.

Proof: Kolmogorov's criterion (Kelly, 1979) asserts that a stationary Markov process is reversible if and only if the product of the transition probabilities $q(i,j)$ between any sequence of states that begins at state 1, visits state m , then returns to state 1 satisfies the relationship:

$$q(1,2)q(2,3)\dots q(m-1,m)q(m,1) = q(1,m)q(m,m-1)\dots q(2,1) \quad (7.1)$$

For two-dimensional Markov processes, a path $1 \rightarrow m \rightarrow 1$ could include transition probability terms that are unique to either the forward or the reverse pathways. Referring again to Figure 2, consider an arbitrary path from state $(0,0)$ to state $(2,1)$ and back, say $(0,0)$, $(0,1)$, $(1,1)$, $(2,1)$, $(2,0)$, $(1,0)$, $(0,0)$. Kolmogorov's reversibility criterion requires that the product of the transition rates along this cycle are the same whether we traverse the path in a clockwise or counter-clockwise direction. That is:

Clockwise transition probability chain = Counter-clockwise transition probability chain,
or

$$\lambda_2 \lambda_1 \lambda_1 E_2(2,1,C) \mu_2 E_1(2,0,C) \mu_1 E_1(1,0,C) \mu_1 = \lambda_1 \lambda_1 \lambda_2 E_1(2,1,C) \mu_1 E_1(1,1,C) \mu_1 E_2(0,1,C) \mu_2 \quad (7.2)$$

Since movement on this graph is rectilinear, the mean arrival rates (λ_1 and λ_2) and service rates (μ_1 and μ_2) on each side of the equality will always cancel. This leaves the clockwise and counter-clockwise products of the expected server deployments along the path from state (2,1) to the origin, or:

$$E_2(2,1,C) E_1(2,0,C) \times E_1(1,0,C) = E_1(2,1,C) E_1(1,1,C) E_2(0,1,C). \quad (7.3)$$

The plan of our proof is to show that equations like (7.3) hold for any destination state (i,j).

If $(i+j) \leq C$, then from equations (3) and (4) each side of equation 7.3 will evaluate to $i! \times j!$, thereby satisfying Kolmogorov's criterion. For $(i+j) \geq C$, assume any path from state (i,j) back to the origin can be rearranged to complete all transactions of one class before beginning the other. From equations (3) and (4), the product chains of server deployments like (7.3) can be expressed for the counterclockwise and clockwise cycles leading to state (i,j) ($i+j \geq C$) are:

$$\left[\prod_{i'=1}^i \frac{\min\{C, i'+j\} i'}{(i'+j)} \right] \left[\prod_{j'=1}^j \frac{\min\{C, j'\} j'}{j'} \right] = \left[\prod_{j'=1}^j \frac{\min\{C, i+j'\} j'}{(i+j')} \right] \left[\prod_{i'=1}^i \frac{\min\{C, i'\} i'}{i'} \right] \quad (7.4)$$

To establish reversibility, it is sufficient to show that the above relationship holds for all destination states (i,j). Expanding and simplifying the denominators on each side, we have:

$$\frac{\left[\prod_{i'=1}^i \min\{C, i'+j\} i' \right] \left[\prod_{j'=1}^j \min\{C, j'\} j' \right]}{[(i+j)! / j!][j!]} = \frac{\left[\prod_{j'=1}^j \min\{C, i+j'\} j' \right] \left[\prod_{i'=1}^i \min\{C, i'\} i' \right]}{[(i+j)! / i][i!]} \quad (7.5)$$

The equivalent denominators on each side of the equality cancel one another. However, the min operator suggests we could encounter four distinct cases, depending on the relationship between destination state (i,j) and C. To complete the proof, we now evaluate each of 4 possible cases by splitting each minima into two terms, those which evaluate to C and those which evaluate to a sum less than C.

Case 1: $i \leq C, j \leq C$: Assuming an empty product term is unity by definition, (7.5) can be expressed:

$$\left[\prod_{i'=1}^{C-j} (i'+j) i' \right] \left[\prod_{i'=C-j+1}^i C i' \right] \left[\prod_{j'=1}^j j' j' \right] = \left[\prod_{j'=1}^{C-i} (i+j') j' \right] \left[\prod_{j'=C-i+1}^j C j' \right] \left[\prod_{i'=1}^i i' i' \right]. \quad (7.5.1)$$

Case 2: $i \leq C, j > C$: Here, separating the minima functions yields:

$$\left[\prod_{i'=1}^i C i' \right] \left[\prod_{j'=1}^C j' j' \right] \left[\prod_{j'=C+1}^j C j' \right] = \left[\prod_{j'=1}^{C-i} (i+j') j' \right] \left[\prod_{j'=C-i+1}^j C j' \right] \left[\prod_{i'=1}^i i' i' \right]. \quad (7.5.2)$$

Case 3: $i > C, j \leq C$: For this case, (7.4) is restated:

$$\left[\prod_{i'=1}^{C-j} (i'+j) i' \right] \left[\prod_{i'=C-j+1}^i C i' \right] \left[\prod_{j'=1}^j j' j' \right] = \left[\prod_{j'=1}^j C j' \right] \left[\prod_{i'=1}^C i' i' \right] \left[\prod_{i'=C+1}^i C i' \right], \quad (7.5.3)$$

Case 4: $i > C, j > C$: For the final case, we rewrite (7.4) as:

$$\left[\prod_{i'=1}^i C i' \right] \left[\prod_{j'=1}^C C j' \right] \left[\prod_{j'=C+1}^j j' j' \right] = \left[\prod_{j'=1}^j C j' \right] \left[\prod_{i'=1}^C i' i' \right] \left[\prod_{i'=C+1}^i C i' \right] \quad (7.5.4)$$

Expanding and simplifying the equalities in (7.5) - (7.8), we obtain for each case:

$$C! C^{i+j-C} i! j! = C! C^{i+j-C} j! i!. \quad (7.6)$$

Since the equality holds for all four cases, Kolmogorov's criterion is satisfied for the clockwise and counterclockwise pathways leading from the origin of the birth-death diagram to any arbitrary state (i, j) where $i+j > C$. Thus the $H_2/H_2/C/N$ Markov chain is reversible, and equation (6) may be used to estimate all state probabilities.

While state probabilities (6) can be expressed by reference to an adjacent state, it is often more convenient to express P_{ij} in terms of the empty state probability. After substitution, equation (6) can be rewritten as:

$$P(i, j) = P(0,0) \left(\frac{\lambda_1}{\mu_1} \right)^i \left(\frac{\lambda_2}{\mu_2} \right)^j \left[\left(\prod_{j'=1}^j [E_2(i, j', C)] \right) \left(\prod_{i'=1}^i E_1(i', 0, C) \right) \right]^{-1}, \quad \text{for } (i+j) > 0, \quad (8)$$

where an empty product is unity by definition (Kleinrock, 1975, p. 92). Finally, since the state probabilities must sum to one, we have:

$$P(0,0) = \left[\sum_{i=0}^N \sum_{j=0}^{N-i} \frac{\left(\frac{\lambda_1}{\mu_1} \right)^i \left(\frac{\lambda_2}{\mu_2} \right)^j}{\left(\prod_{j'=1}^j [E_2(i, j', C)] \right) \left(\prod_{i'=1}^i E_1(i', 0, C) \right)} \right]^{-1} \quad (9)$$

In some respects, $M/M/C/N$ is a useful approximation for the behavior of $H_2/H_2/C/N$. For example, we can easily show that both models return the same probability of a particular

occupancy level for a two-class system. To illustrate this point, and identify some of the key differences between the two models, consider the state probabilities for a representative case that are shown in Table 1A.

(please insert Table 1 about here)

State probabilities $P(k)$ for M/M/C/N appear in the top two rows of Part A of the table. These state probabilities equal the sum of the H₂/H₂/C/N state probabilities $P(i,j)$ along each diagonal, which contain all (i,j) combinations that sum to k . This property assures that the expected number in the system for both models is identical, which in this example is $E(k) = E(i+j) = 3.10$. However, H₂/H₂/C/N allows us to disaggregate expected occupancy by class of caller, revealing the expected number of type 1 and type 2 callers in the system to be $E(i) = 0.52$ and $E(j) = 2.58$, respectively. In general, we expect the ratio of type 1 to type 2 callers to vary with the ratio ρ_1/ρ_2 . As this ratio decreases, we expect a higher proportion of customers with long transaction times in the system. This imbalance helps explain H₂/H₂/C/N's greater service time variance and motivates the characterization of the queue time distribution for this system.

2.4 Queue time distribution

Call center managers control their capacity through employee scheduling decisions. The number of employees on duty at a particular time, along with the expected arrival and service time distributions for that time, determine the system's expected QOS (the fraction of arriving customers α whose wait is β or less). Therefore, most staffing and scheduling decisions are based on the distribution of caller delays and queue times for alternate staffing levels (Matan & Nourbakhsh, 1998).

To determine the QOS for a given staffing level in an H₂/H₂/C/N system, we must first consider the mix of type 1 and type 2 callers who were in the system when a new caller arrives.

Define d_{ij} as the expected time a caller will wait before reaching a server from state (i,j) . A queued caller whose arrival brings the system to state (i,j) will advance through the queue one position each time a service is completed. Due to the memoryless property of the exponential, the expected time to the next service completion is $[E_1(i,j,C)\mu_1 + E_2(i,j,C)\mu_2]^{-1}$, assuming no queued callers renege. If the first completion after the new caller's arrival was a type 1 service, the remaining expected time to reach a server is $d_{(i-1,j)}$; if the completion was a type 2 service, the expected remaining time to reach a server is $d_{(i,j-1)}$. The probability that the completed service was a type 1 call is proportional to $E_1(i,j,C)\mu_1/[E_1(i,j,C)\mu_1 + E_2(i,j,C)\mu_2]$, or the expected proportion of system capacity allocated to type 1 customers. The expected time to reach a server from state $C < (i+j) \leq N$ can be computed recursively by adding the expected service time for one service completion to the expected times to complete the remaining calls in the queue.

Let d_{ij} = expected time to reach a server from state $C < (i+j) \leq N$. Then

$$d_{ij} = \left[\begin{array}{l} \frac{1}{E_1(i,j,C)\mu_1 + E_2(i,j,C)\mu_2} [1 + E_1(i,j,C)d_{i-1,j} + E_2(i,j,C)d_{i,j-1}] \text{ if } i, j > 0 \\ (d_{i-1,j} + [E_1(i,j,C)\mu_1]^{-1}) \text{ if } j = 0 \\ (d_{i,j-1} + [E_2(i,j,C)\mu_2]^{-1}) \text{ if } i = 0 \end{array} \right] \quad (10)$$

where $d_{ij} = 0 \forall (i+j) \leq C$ and undefined when $(i+j) > N$.

In part B of Table 1, we compare expected waiting times from each state for the M/M/C/N and H₂/H₂/C/N models. For M/M/C/N, expected waiting time increases linearly with queue position. In this example, with $C = 3$ servers, an arrival who brings the system to state 6 would begin service after 3 completions, or an expected wait of 0.200 time units. For the 2-class model, expected waiting time increases with queue position and with the mix of customers

already in the system when an arrival occurs. For example, if a new arrival brings the system to state (3,3), that customer can expect a wait of 0.127. These differences have a significant effect on average queue time. With the parameters assumed in this example, expected waiting time for the two-class model is about 25% greater (0.073 hrs vs 0.058 hrs) than the single-class model.

Furthermore, suppose the waiting time threshold for our QOS (Quality of Service) metric is $\beta = 0.20$. With the parameters assumed here, M/M/C/N predicts $\alpha = 93.3\%$ of all arrivals will wait β or less. However, H₂/H₂/C/N suggests that only 83% of the arrivals will meet the standard. To improve QOS to at least 90%, at least one additional server will be required.

3. Approximation Errors Induced by Applying M/M/C/N to Multi-class Queues

In this portion of the paper, we illustrate the potential for errors of various types when Erlang B (M/M/C/N) is used to approximate the behavior of two-class multi-server finite queues. We first illustrate how key statistics such as average queue time and QOS are distorted by M/M/C/N as the ratio (ρ_1/ρ_2) increases from 1. Then we assess the accuracy of the M/M/C/N's minimum staffing level recommendations to achieve QOS objectives, as the ratio (ρ_1/ρ_2) is varied from 1 to 24.

(please insert Table 2 about here)

In Table 2, we report the sensitivity of various queuing statistics to differences in the mean service rates for the two classes. The top rows of the report include M/M/C/N's estimates for E(Qtime), or expected queue time; Var(Qtime), or queue time variance; E(Q), or expected number in queue; and the probability (α) that queue time will be 0.20 or less. We again assume maximum system occupancy of 7 customer, a staff of 3 servers, and average arrival & service rates of 12 and 5, respectively. In the second part of the table we report H₂/H₂/C/N's estimates of the same statistics when the arrival stream is separated into two parts, with type 1 and type 2

arrivals occurring at rates λ_1 and λ_2 , respectively. To generate different ratios (ρ_1/ρ_2) while maintaining an average service rate of $\mu = 5$ and an average arrival rate of $\lambda = 12$, we increment μ_1 from its initial value of 5 and, using equation (1), reduce μ_2 as necessary to maintain the desired average service rate. For each problem instance, we divide arrivals equally among the two classes. Thus for each of the 26 cases, M/M/C/K would yield the same results as those reported in the first few lines of Table 2. As Table 2 reveals, the difference between H₂/H₂/C/N's estimates of expected queue time, queue time variance, and QOS and those obtained with an M/M/C/N approximation of a two-class queue increases as the ratio (ρ_1/ρ_2) deviates from 1.

In Table 3, we compare the minimum staffing levels recommended by H₂/H₂/C/N with each of the two single-class approximation strategies, to insure that at least 95 percent of all arriving customers wait $\beta = 0.2$ or less. We examine three distinct queueing systems: a small, lightly loaded system ($N=25, \lambda = 20, \mu = 10, P(N) = 0.0002$); a medium-sized system ($N = 40, \lambda = 80, \mu = 5, P(N) = 0.001$) with moderate load, and a heavily-loaded large system ($N = 120, \lambda=240, \mu = 3, P(N) = 0.067$). For each scenario, we vary ρ_1/ρ_2 from 1 to 32, then compute the minimum staffing levels using H₂/H₂/C/N, M/M/C/N with service rate μ based on the average of the two service classes (equation 1), and M/M/C/N with separate arrival streams for each service class.

As expected, we find the two-class model recommends staffing levels that fall between the two single class strategies. Overall, for these examples the staffing recommendations obtained with the aggregate M/M/C/N strategy range from 0 to 33% below those recommended by H₂/H₂/C/N. As with the previous comparison, we find that the magnitude of the staffing errors produced by the single-class approximation tend to increase with the ratio ρ_1/ρ_2 .

Conversely, H₂/H₂/C/N's staffing recommendations are consistently lower than those obtained when the two arrival streams are separated and labor requirements are computed for

each stream with M/M/C/N. For these examples, we found the M/M/C/N approximation based on separate arrival streams overstated minimum labor requirements by 0 - 67%. However, as the ratio ρ_1/ρ_2 increases, the difference between the recommended staffing levels decreases.

(please insert Table 3 about here)

For service operations managers, the magnitude of the errors revealed in this study should be alarming. For industries where fast, consistent response times are a strategic imperative, staffing decisions based on the aggregate M/M/C/N strategy could result in high levels of customer dissatisfaction and lost sales. In addition, this approach appears to understate server workloads, which could exacerbate turnover and absenteeism problems. For low margin or public sector systems, the staffing guidelines obtained by applying M/M/C/N separately to the two arrival streams are likely to unnecessarily squander scarce resources, unnecessarily inflating costs and reducing margins further.

4. Conclusions

Queueing systems have been studied for nearly a century. While earlier analytical treatments of the system studied here may exist, most modern call centers continue to base their staffing decisions on standard Erlang models, even when they provide more than one type of service. Using the proposed $H_2/H_2/C/N$ queueing model, we confirmed significant shortcomings in the two M/M/C/N-based strategies for estimating minimum labor requirements for two-class, multi-server finite queues. In general, their accuracy declines as ρ_1/ρ_2 deviates from unity. The proposed model provides operations managers with a convenient, accurate method of estimating minimum labor requirements and other characteristics of such systems. Among the possible extensions to this research, we suggest developing models for three or more classes of service and incorporating customer reneging behavior.

Acknowledgments: The author gratefully acknowledges the valuable assistance and insights provided by former Syracuse University MBA student Leida Kokona, who served as a research assistant for this project.

REFERENCES

Andrews, B. and Parsons, H. 1993. Establishing Telephone Agent Staffing Levels using Economic Analysis. *Interfaces* 23(2), 14-20.

Chen, P. 1999. Calls For Help Need Help Growing Cell Phone Use Includes Inaccurate 911 Requests. *The Post - Standard*, (Oct 8, 1999).

Cleveland, B. 1999. How Do You Calculate Staff? *TeleProfessional*, 12(8), 1999, 98+.

Crisafulli, M. 1998. Implementing Skills-Based Routing in a Service Agency Environment, *Telemarketing & Call Center Solutions*, 16(8), 56,58-60.

Davis, M. 1991. How Long Should a Customer Wait for Service? *Decision Sciences*, 22(2), 421-434.

Durr, B. 1994. Inbound Scheduling Nightmares: Cursing the Darkness or Lighting a Candle. *Telemarketing* 12(7), 44+.

Easton, F. and D. Rossin, 1996. "A Stochastic Goal Program for Employee Scheduling." *Decision Sciences*, 27(3), 541-568.

Fayolle, G., P.J.B. King, and I. Mitrani (1982). The solution of certain two-dimensional Markov models. *Advances in Applied Probability*, 14(2), 295-308.

Gross, D. and C. Harris. 1998. *Fundamentals of Queueing Theory (3rd ed.)*. Wiley: New York.

Kao, E. P. C. and S. Wilson. 1999. Analysis of Nonpreemptive Priority Queues with Multiple Servers and Two Priority Classes. *European Journal of Operational Research*, 118, 181-193.

Kelly, F. 1979. *Reversibility and Stochastic Networks*. Wiley: New York.

Kleinrock, L. 1975. *Queueing Systems Vol 1: Theory*. New York: John Wiley, 92.

Kolmogorov, A. 1936. Zur Theorie der Markoffschen Ketten. *Mathematische Annalen* 112, 155-160.

Leamon, P. 1999. Workforce Management with Skills-based Call Routing: The New Challenge *Call Center Solutions*, 17(9), 88-93.

Leuchter, M. 1999. The New Realities of Staffing. *USBanker*, 109(10), 41-44.

- Matan, O. and Nourbakhsh, I. 1998. Playing the Numbers: Using ACD Statistics for Workforce Management. *Call Center Solutions*, 16(9), 118-123.
- Mehrotra, V., D. Profozich, and V. Bapat. 1997. Simulation: The Best Way to Design Your Call Center. *Call Center Solutions*, 16(5), 28-30.
- Mehrotra, V. 1997. Ringing up Big Business. *OR/MS Today*, 24(4).
- Nelson, R. 1993. The Mathematics of Product Form Queueing Networks. *ACM Computing Surveys*, 25(3), 339-369.
- Reynolds, P. 1998. The Science of Call Center Management. *Communications News*, 35(10), 64-66.
- Robertazzi, T. 1990. *Computer Networks and Systems: Queueing Theory and Performance Evaluation*. Springer-Verlag: New York. 68-72.
- Robinson, D. 1999. Buffalo, N.Y., Attractive to Call Centers Despite Expense, Study Says, *Buffalo News*, November 22.
- Sasser, W. E. 1976. Match Supply and Demand in Service Industries. *Harvard Business Review* (November-December), 133-140.
- Segal, M. 1974. The Operator Scheduling Problem. *Operations Research* 22(4), 808-823.
- Takagi, H. 1986. *Analysis of Polling Systems*. MIT Press, Cambridge, MA.
- Taylor, S. 1994. Waiting for Service: The Relationship Between Delays and Evaluations of Service. *Journal of Marketing* 56(April), 56-69.
- Whitt, W. 1999a. Predicting Queueing Delays. *Management Science*, 45(6), 870-888.
- Whitt, W. 1999b. Partitioning Customers into Service Groups *Management Science*, 45(11), 1579-1592.
- Wolcott, H. 1999. CHP Warns Against Needless 911 Cell Phone Calls. *The Los Angeles Times* (Ventura County Edition), November 22.

FIGURE 1
Two-Class Single-Phase Multi-Server Queue with Blocking

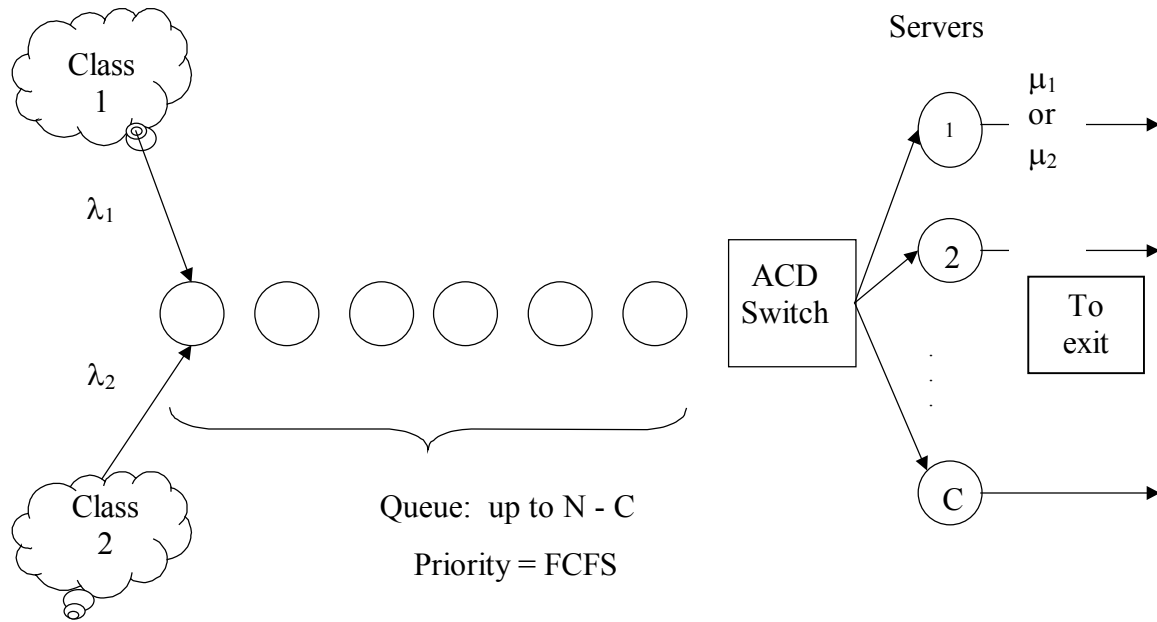
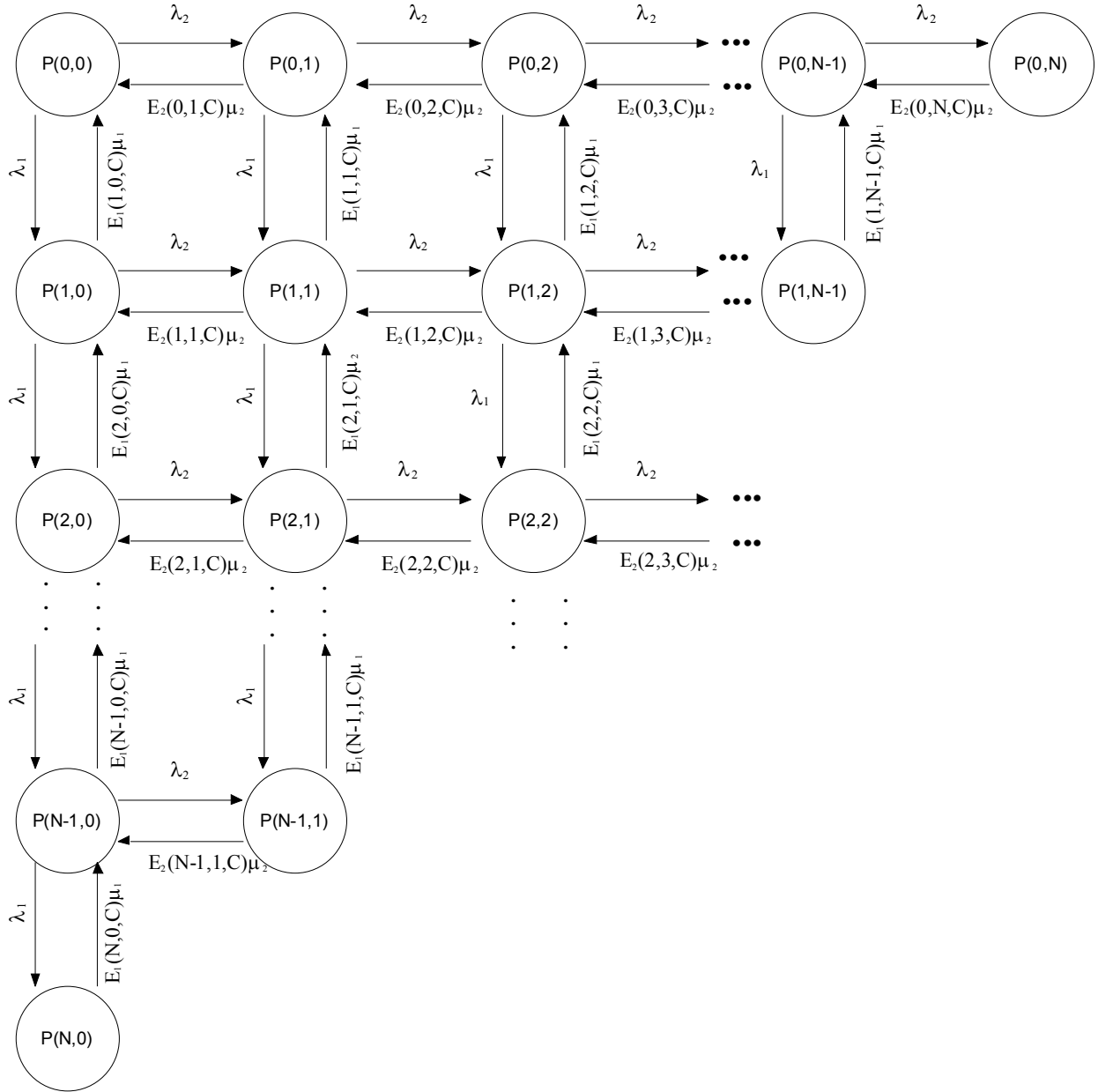


FIGURE 2
Birth-Death Diagram for Two-class, Single-stage Multi-server Queue with Blocking



**TABLE 1: Comparing M/M/C/N and H₂/H₂/C/N for:
 $\lambda_1 = \lambda_2 = 6, \lambda = 12; \mu_1 = 15, \mu_2 = 3, \mu = 5; C = 3$ and $N = 7$**

A. State Probabilities

M/M/C/N	State k:	0	1	2	3	4	5	6	7	E(k) =
	P(k) =	0.0713	0.171	0.205	0.164	0.131	0.105	0.084	0.067	3.10

H ₂ /H ₂ /C/N P(i,j)=	State i/j	j=0	1	2	3	4	5	6	7	$\Sigma P(i,*) =$
	i=0	0.0713	0.1426	0.1426	0.0951	0.0634	0.0423	0.0282	0.0188	0.6041
	1	0.0285	0.0570	0.0570	0.0507	0.0423	0.0338	0.0263		0.2956
	2	0.0057	0.0114	0.0152	0.0169	0.0169	0.0158			0.0819
	3	0.0008	0.0020	0.0034	0.0045	0.0053				0.0159
	4	0.0001	0.0003	0.0007	0.0011					0.0022
	5	0.0000	0.0001	0.0001						0.0002
	6	0.0000	0.0000							0.0000
	7	0.0000								0.0000
$\Sigma P(*,j) =$	0.1064	0.2135	0.2190	0.1682	0.1278	0.0918	0.0545	0.0188	$\frac{E(i)=0.52}{E(j)=2.58}$	

B. Expected Time in Queue by State

M/M/C/N	State k:	0	1	2	3	4	5	6	7	E(q) =
	E(wait k)	0.0	0.0	0.0	0.0	0.0667	0.1333	0.2000	0.2667	0.0575

H ₂ /H ₂ /C/N D(i,j) =	State i/j	j=0	1	2	3	4	5	6	7	E(q) =
	i=0	0.0	0.0	0.0	0.0	0.1111	0.2222	0.3333	0.4444	0.0731
	1	0.0	0.0	0.0	0.0556	0.1482	0.2519	0.3596		
	2	0.0	0.0	0.0370	0.0940	0.1803	0.2799			
	3	0.0	0.0278	0.0686	0.1268	0.2100				
	4	0.0222	0.0540	0.0976	0.1568					
	5	0.0444	0.0792	0.1250						
	6	0.0667	0.1039							
	7	0.0889								

TABLE 2: Comparison of 2-class and 1-class Multi-server Finite Queue Statistics

M/M/C/N:	N	C	λ		μ			E(Qtime)	Var(Qtime)	E(Number in Q)		P(Qtime \leq 0.2)
	7	3	12		5			0.0576	0.0073	0.8632		0.9327
H ₂ /H ₂ /C/N	N	C	λ_1	λ_2	μ_1	μ_2	(ρ_1/ρ_2)	E(Qtime)	Var(Qtime)	E(NQ) -- Class 1	E(NQ) -- Class 2	P(Qtime \leq 0.2)
	7	3	6	6	5	5.0000	1.0000	0.0576	0.0073	0.4316	0.4316	0.9327
	7	3	6	6	6	4.2857	0.7143	0.0582	0.0075	0.3597	0.5035	0.8898
	7	3	6	6	7	3.8889	0.5556	0.0594	0.0080	0.3083	0.5549	0.8796
	7	3	6	6	8	3.6364	0.4546	0.0610	0.0086	0.2698	0.5934	0.8724
	7	3	6	6	9	3.4615	0.3846	0.0627	0.0093	0.2398	0.6234	0.8681
	7	3	6	6	10	3.3333	0.3333	0.0645	0.0100	0.2158	0.6474	0.8887
	7	3	6	6	11	3.2353	0.2941	0.0663	0.0107	0.1962	0.6670	0.8577
	7	3	6	6	12	3.1579	0.2632	0.0681	0.0114	0.1798	0.6834	0.8492
	7	3	6	6	13	3.0952	0.2381	0.0698	0.0120	0.1660	0.6972	0.8418
	7	3	6	6	14	3.0435	0.2174	0.0715	0.0127	0.1541	0.7091	0.8354
	7	3	6	6	15	3.0000	0.2000	0.0731	0.0133	0.1439	0.7193	0.8297
	7	3	6	6	16	2.9630	0.1852	0.0746	0.0139	0.1349	0.7283	0.8246
	7	3	6	6	17	2.9310	0.1724	0.0760	0.0145	0.1269	0.7363	0.8201
	7	3	6	6	18	2.9032	0.1613	0.0773	0.0150	0.1199	0.7433	0.8195
	7	3	6	6	19	2.8788	0.1515	0.0786	0.0155	0.1136	0.7496	0.8155
	7	3	6	6	20	2.8571	0.1429	0.0798	0.0160	0.1079	0.7553	0.8118
	7	3	6	6	21	2.8378	0.1351	0.0810	0.0164	0.1028	0.7604	0.8085
	7	3	6	6	22	2.8205	0.1282	0.0820	0.0168	0.0981	0.7651	0.8055
	7	3	6	6	23	2.8049	0.1220	0.0831	0.0172	0.0938	0.7694	0.8027
	7	3	6	6	24	2.7907	0.1163	0.0840	0.0176	0.0899	0.7733	0.8002
	7	3	6	6	25	2.7778	0.1111	0.0849	0.0180	0.0863	0.7769	0.7979
	7	3	6	6	26	2.7660	0.1064	0.0858	0.0183	0.0830	0.7802	0.7957
	7	3	6	6	27	2.7551	0.1020	0.0866	0.0186	0.0800	0.7832	0.7937
	7	3	6	6	28	2.7451	0.0980	0.0874	0.0189	0.0771	0.7861	0.7919
	7	3	6	6	29	2.7359	0.0943	0.0881	0.0192	0.0743	0.7888	0.7902
	7	3	6	6	30	2.7273	0.0909	0.0888	0.0195	0.0719	0.7913	0.7886

Table 3: Minimum Labor Requirements to Assure 95% of All Arrivals Wait $\beta = 0.20$ or Less

Queueing System Parameters							Minimum Labor Requirements by Model					
							H ₂ /H ₂ /C/N		M/M/C/N (aggregate)		M/M/C/N (separate)	
N	λ_1	λ_2	μ_1	μ_2	ρ_1/ρ_2	μ	Staffing Level C	P(wait $\leq\beta$)	Staffing Level C	P(wait $\leq\beta$)	C For Type 1	C For Type 2
25	10	10	10	10	1	10	3	0.97403	3	0.97403	2	2
25	10	10	15	7.5	2	10	3	0.96193	3	0.97403	2	3
25	10	10	25	6.25	4	10	4	0.99842	3	0.97403	1	3
25	10	10	45	5.625	8	10	4	0.99684	3	0.97403	1	3
25	10	10	85	5.3125	16	10	4	0.99588	3	0.97403	1	4
25	10	10	125	5.2083	24	10	4	0.99549	3	0.97403	1	4
50	40	40	5	5	1	5	18	0.95372	18	0.95372	10	10
50	40	40	7.5	3.75	2	5	19	0.98653	18	0.95372	7	14
50	40	40	12.5	3.125	4	5	19	0.98014	18	0.95372	5	16
50	40	40	22.5	2.8125	8	5	19	0.96871	18	0.95372	3	18
50	40	40	42.5	2.6563	16	5	19	0.95560	18	0.95372	2	19
50	40	40	62.5	2.6042	24	5	20	0.98324	18	0.95372	1	19
120	120	120	3	3	1	3	75	1.00000	75	1.00000	44	44
120	120	120	4.5	2.25	2	3	76	0.96912	75	1.00000	30	58
120	120	120	7.5	1.875	4	3	78	0.95428	75	1.00000	18	70
120	120	120	13.5	1.6875	8	3	81	0.96688	75	1.00000	10	77
120	120	120	25.5	1.5938	16	3	83	0.96154	75	1.00000	6	81
120	120	120	37.5	1.5625	24	3	84	0.96707	75	1.00000	4	82