Bounding the Mean Response Time of the Minimum Expected Delay Routing Policy: An Algorithmic Approach

John C.S. Lui[1]
Computer Science Department
The Chinese University of Hong Kong

Richard R. Muntz[2]
UCLA
Computer Science Department

Don Towsley[3]
University of Massachusetts
Computer Science Department

[1]The work of this author was supported in part by the Direct Research Grant from CUHK

[2]The work of this author was supported in part by the National Science Foundation under grant CCR-9215064.

[3]The work of this author was supported in part by the National Science Foundation under grant NCR-9116183.

Abstract

*Balancing loads in a multiprocessor or multicomputer system can have a significant impact on performance. In this paper, we model such a system as a heterogeneous multi-server queueing system. We study the behavior of such a system operating under the Minimum Expected Delay (MED) routing policy, i.e., an arriving customer is assigned to the queue which has the minimal expected value of unfinished work. This routing discipline can be viewed as a generalization of the join-the-shortest queue (*SQ*) discipline for homogeneous servers. There is no closed-form solution for this class of queueing problem. In this paper, we provide a methodology to compute upper and lower bounds on the mean response time of the system. This methodology allows one to tradeoff the tightness of the bounds and computational cost. Applications and numerical examples are presented which show how to use this methodology for deriving performance measures and also illustrating that the excellent accuracy of the computational algorithm which is achievable with modest computational cost.*

1 Introduction

With the advent of multiprocessors and multicomputer systems, there has been considerable interest in the problem of balancing the load among processors or computers. In this paper, we model such systems as heterogeneous multi-server queueing systems. Using such models, we investigate the performance of such systems operating under the minimum expected delay (MED) routing policy. Although not optimal, this policy can provide excellent performance in these systems. Some major difficulties in analyzing this kind of a routing policy, even under Markovian assumptions, are (1) each queue in the system is correlated because the arrival process to each server depends on the state of the entire system and, (2) since each queue has infinite capacity, the state space of the system is multi-dimensional in nature and is infinite in each of the dimensions. In its general form, there is no known closed-form solution, and it is impossible to exactly solve the problem numerically due to the infinite state space. One way to approach this problem is to *construct* a modified model which provably bounds the performance of the original policy and for which the performance measures of the modified model can be easily computed.

The goal of this paper is to analyze the minimum expected delay (MED) routing algorithm (a natural generalization of the join-the-shortest-queue (SQ) policy for homogeneous servers) in a multiprocessor or multicomputer systems, which is modeled as a heterogeneous multi-server queueing system. Let K be the number of servers, where $K \geq 2$. Each server has an infinite capacity queue, and service rates are exponentially distributed with rates μ_i, $i = 1, 2, \ldots, K$. Without loss of generality, we assume $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. The customer arrival process is a Poisson process with a mean rate of λ. We propose a methodology which provides upper and lower bounds on the mean number of customers (and thereby the mean response time) in the system and which can be used to trade off the tightness of the bounds with the computational cost. By virtue of providing bounds, rather than simply an approximation, our results are distinguished from previous work on this problem.

We begin with a brief review of the published literature on the join-the-shortest-queue routing problem. The optimality of the SQ policy for homogeneous multi-server systems has been established in numerous papers [11, 31, 33]. In its most general form, SQ has been shown to minimize the queue length vector in the sense of Schur-convex ordering, [31]. This carries the implication that the total number of customers in the system is stochastically minimized by the SQ policy as is the mean stationary response time (when it exists).

Of more interest to us is the literature dealing with the performance evaluation of the SQ and MED policies. In [5], MED policy was mentioned in the user search sequences. In the case of the SQ policy for two identical servers, numerous authors have provided exact, though not necessarily computable solutions (Kingman [15], Flatto and McKean [12], Zhao and Grassmann [35], and Cohen and Boxma [8], Adan [1], et al). Several authors have provided similar solutions to the heterogeneous server problem; e.g., Knessl, et al. [16] and Adan, et al. [2]. The last paper (also [1]) is interesting because it can generate a sequence of increasingly more accurate approximations with error bounds that decrease exponentially. Recently, Adan, et al [3] has provide an error bound for a homogeneous server system.

Numerous authors have proposed approximations for the SQ and MED policies. These include Conolly [7], Rao and Posner [28], and Towsley and Chen [30] in the case of the SQ policy. The first of these treats both queues as having bounded capacity whereas the last two treat only one queue as having bounded capacity. The last two papers produce solutions that can be expressed in a matrix-geometric form [25]. The last paper, [30], is also noteworthy in that it provides upper and lower bounds on various performance statistics that are established using less sophisticated sample path techniques than are used in this paper. Grassmann [14] studied the same problem with $K = 2$ and solved for transient and steady state behavior. Halfin [13] studied the two servers problem and used a linear programming technique to compute bounds on the mean number of customers in the system. Blanc [6] studied the SQ routing policy with an arbitrary number of heterogeneous servers. He proposed an approximation method which was based on power series expansions and a recursion which required a substantial computational effort. Various approximations for computing the mean response time of K homogeneous servers have been proposed by Lin and Raghavendra [20], Nelson and Philips [23, 24], and Wang and Morris [32]. Zhao and Grassmann [34] studied the shortest queue model with jockeying. This problem has the matrix-geometric form and an explicit solution can be obtained. Avritzer [4] studied a dynamic load balancing algorithm which used a threshold policy in an asymmetric distributed system. The result was only applicable to two distinct types of servers and a small class of threshold sizes. No formal proof was given on how to obtain performance bounds. None of the work cited above treated more than two heterogeneous servers and simultaneously provided error bounds. Lui and Muntz [21] were the first to propose a methodology to bound the mean response time of a minimum expected delay routing system. This paper differs from [21] in several ways. First, we derive improved bounds for the homogeneous servers case, and secondly, we use sample path analysis to prove the bounds, yielding more elegant and intuitive proofs.

This work distinguishes itself from previous published results in that it simultaneously (1) allows more than $K \geq 2$ servers, (2) allows heterogeneous servers, (3) includes a scheduling policy based on queue lengths and service rates (thus, we treat a generalization of the join-the-shortest queue for homogeneous systems) and (4) provides error bounds on the mean number of customers (and thereby mean response time) in the system. The bounding methodology has the desirable property that it allows one to tradeoff accuracy and computational cost, as will be demonstrated.

The organization of the paper is as follows. In Section 2, we formally define the queueing model we are analyzing. In Sections 3 and 4, we present the modified models and prove that these models do provide bounds. In Section 5, we provide a methodology for obtaining tighter bounds in the special case of homogeneous servers. Section 6 shows that using lumpability, we can reduce the state space further and thereby obtain better bounds at the same cost. In Section 7, we present two applications with numerical examples which show the excellent accuracy of this methodology. The first application will be to datagram routing within a communication node and the second example will be to model the performance of a parallel job in a workstation cluster environment. Conclusions are given in Section 8.

2 Model

We model a multiprocessor or multicomputer system as a queueing system, as depicted in Figure 1, with K heterogeneous servers with associated queues being fed by a Poisson process[1] with mean rate λ. The service times at servers form mutually independent sequences of exponential random variables that are also independent of arrival times with rates $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. Let Υ denote a policy that routes an arriving customer to a server on the basis of the server queue lengths and mean service time and service rate of the servers. Let $l^*_\Upsilon(\boldsymbol{N})$ denote the identity of the queue to which the customer is routed under policy Υ when the queue length vector is $\boldsymbol{N}$. When we are interested in the joint queue length at time t under a specific policy, we will denote it as $\boldsymbol{N}^\Upsilon(t)$. We assume that p is stationary.

We define the minimum expected delay (MED) routing policy as follow. Let $N_i(t)$ be the number of customers at server i (on that server or in the server's queue) at time t. We define $u_i(t) = (1 + N_i(t))/\mu_i$, which is the mean unfinished work at the i-th server if a customer arrives at time t and is assigned to the i-th server. Let us define $u^*(t) = \min\{u_i(t), i = 1, \ldots, K\}$. Upon arrival of a customer at time t, the customer joins a server j where $u_j(t) = u^*(t)$. If a tie occurs, the customer chooses the server with the lowest index. When all service rates are equal, this MED routing policy reduces to the classic join-the-shortest queue (SQ) routing algorithm.

Assume the system is stable, that is $\lambda < \sum_{i=1}^{K} \mu_i$. Then $\lim_{t\to\infty} N_i(t) = N_i$. We can

[1] Although in the computational algorithm we assume the arrival process to be Poisson, the proofs for the bounds can accommodate a general arrival process

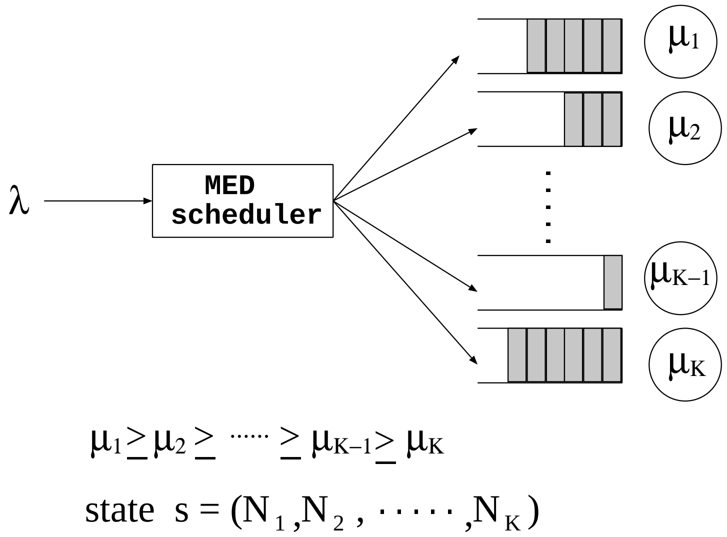

Figure 1: Minimum Expected Delay Routing Policy Queueing Model

construct a Markov model, M, for this queueing system with state space:

$$\{ \quad (N_1, N_2, \ldots, N_K) \,|\, N_i \geq 0, \quad i = 1, \ldots, K\} \quad \}$$

The unique steady state probability vector for this continuous-time Markov model satisfies the following system of linear equations:

$$\vec{\pi} G = \vec{0} \qquad and \qquad \vec{\pi}\underline{e} = 1 \tag{1}$$

where $\vec{\pi}$ is the K-dimensional steady state probability vector, $\underline{e}$ denotes an appropriately dimensioned column vector of $1's$ and G is the transition rate matrix having the following structure:

$$\begin{array}{lcll} (N_1, \ldots, N_i, \ldots, N_K) & \rightarrow & (N_1, \ldots, N_i + 1, \ldots, N_K) & \quad 1\{i = min\{k|u_k = u^*\}\}\lambda \\ (N_1, \ldots, N_i, \ldots, N_K) & \rightarrow & (N_1, \ldots, N_i - 1, \ldots, N_K) & \quad 1\{N_i > 0\}\mu_i \end{array}$$

and the balance equation can be expressed as:

$$\begin{aligned} \lambda + \sum_{i=1}^{K} 1\{N_i > 0\}\mu_i \pi(N_1, \ldots, N_K) \quad &= \quad \sum_{i=1}^{K} \pi(N_1, \ldots, N_i + 1, \ldots, N_K) + \\ &\lambda \sum_{i=1}^{K} 1\{i = l^*_{MED}((N_1, \ldots, N_i - 1, \ldots, N_K))\pi(N_1, \ldots, N_i - 1, \ldots, N_K) \end{aligned}$$

The above model does not possess a known closed form solution, and it is not possible to solve the problem numerically due to its infinite state space cardinality. Since the Markov process lacks the appropriate special structure, techniques such as matrix-geometric methods do not apply. One natural way to approach this problem is to *construct* other models that closely bound the performance of the original problem and which, at the same time, have either known closed form solutions or at least can be efficiently evaluated by numerical methods.

It is intuitive that the stationary state probabilities for the model M are highly *skewed* or, in other words, the probability mass of the system is concentrated in some relatively small subset of the state space rather than distributed nearly uniformly over the entire state space. For example, consider a system of four homogeneous servers. The purpose of using the routing policy discussed above is to balance the load of the system as much as possible; therefore it is reasonable to assume that a highly unbalanced state(e.g., $(8,4,3,1)$) has a much smaller probability mass than a balanced state (e.g., $(4,4,4,4)$). This crucial insight provides the rationale for constructing two modified versions of the original model which can be shown to bound the mean response time of the original system. In both cases we represent the exact behavior (transition rates) for the most "popular" states (where most of the probability mass resides). The number of states in the most popular subset is a function of the accuracy demanded and the computational cost one is willing to pay. When the system leaves this subset we modify the behavior of the system in such a way that (a) the modified system has an efficient solution and (b) the modified model's behavior can be shown to bound the behavior of the original model from above or from below. Therefore, one modified model provides an upper bound on the mean response time while another provides a lower bound on the mean response time. In the next section, we discuss the upper bound model and then, in the following section, we cover the lower bound model.

3 Upper Bound

In this section, we present a modified model, M^u, which provides an upper bound for the mean response time and the mean number of customers in the system for the original model, M. The upper bound model has the same system configuration, namely that the customer arrival process is a Poisson process. and K servers with service rates μ_i, $i = 1, 2, \ldots, K$, where $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$.[2] The upper bound model M^u has two additional parameters. The first parameter we term the *artificial capacity vector* $\overline{\mathbf{C}} = (C_1, \ldots, C_K)$. The second parameter is a threshold setting d, which is the maximum allowable difference between the longest queue and the shortest queue in M^u. We first give the formal definition of these two parameters and, in the following paragraph, we

[2]We require that the service rates be rational numbers such that they can be expressed as integers after normalization, i.e., the service rates are mutually commensurable.

give the intuitive idea of how these two parameters can be used construct the upper bound model M^u.

Definition 1 *Let $\overline{\mathbf{C}^*} = (C_1^*, \ldots, C_K^*)$ be a vector where $C_i^* = \frac{\mu_i}{\mu_1} C$ and C is chosen to be the minimum positive integer such that C_i^* is a positive integer for $i = 1, \ldots, K$. Then the artificial capacity vector $\overline{\mathbf{C}}$ is an integer multiple of $\overline{\mathbf{C}^*}$, i.e, $\overline{\mathbf{C}} = j\overline{\mathbf{C}^*}$ for some $j \geq 1$.*

For example, if $\mu_1 = \frac{3}{2}$ and $\mu_2 = 1$, then $C = 3$ and $\overline{\mathbf{C}^*} = (3, 2)$. So the artificial capacity vector can be $\overline{\mathbf{C}} = j\overline{\mathbf{C}^*}$ for any $j \geq 1$.

Definition 2 *Let a state of the model be $s = (N_1, \ldots, N_K)$. Let N_i^* be the number of active customers*[3] *in the i^{th} server. Define $q(s)$ be the* degree of imbalance *for state s, as:*

$$q(s) = max\{N_i^* - N_j^* \;\Big|\; where\ i, j \in \{1, \ldots, K\}\}$$

Definition 3 *Let d be the threshold setting in the modified model. We require that $q(s) \leq d$ for each state s in model M^u.*

We first give an intuitive idea of the construction of model M^u. In M^u, the degree of imbalance is required to be less than or equal to the parameter d. A customer may depart from the system only if its departure does not violate the maximum degree of imbalance permitted. If the customer departure would violate the threshold setting, the customer restarts its service within the same server. Intuitively, this mechanism forces a customer to stay in the system at least as long as in the original model and thereby increases the number of customers in the system. Note that, due to the routing policy, an arrival never causes the degree of imbalance to exceed d. The rationale behind the threshold parameter is to generate a model with a state space which is a subset of the state space of the original model.

The second parameter is the *artificial capacity*, C_i, $i = 1, 2, ..., K$ for each server. In model M^u, there are two classes of customers, active customers and suspended customers. At any point in time, there are never more than C_i active customers in queue i; all of the remaining customers are suspended. Whenever a customer arrives to the system and finds that each server i, $i = 1, \ldots, K$, has *exactly* an integer multiple of C_i customers, all active customers in the system (except for the arriving customer) are put into a *suspended mode* and a new "busy cycle" is started. This busy cycle ends when all servers complete all active customers. C_i suspended customers are then released from queue $i = 1, \ldots, K$ and

[3]definition of active customers will be define in later paragraph.

can be served. Note that the definition here is recursive; during the busy period following suspension of a set of customers, the capacities C_i can *again* be exceeded, causing *another* set of customers to be suspended. When a busy cycle ends, only the set of customers suspended at the initiation of that busy cycle is released for service. The purpose of the C_i, $1 \leq i \leq K$, is to create a matrix with a *repetitive structure*; based on that structure, we will be able to derive an efficient numerical solution algorithm. The computation algorithm is based on partitioning the state space of M^u into $\cup_{i=0}^{\infty} \mathcal{S}_i \cdots$ where:

$$
\begin{aligned}
\mathcal{S}_0 &= \{(N_1, \ldots, N_K) | 0 \leq N_j \leq C_j \text{ for } j = 1, \ldots, K\} \\
\mathcal{S}_i &= \{(N_1, \ldots, N_K) | iC_j \leq N_j \leq (i+1)C_j \text{ for } j = 1, \ldots, K\} - \{(iC_1, \ldots, iC_K)\} \quad i \geq 1
\end{aligned}
$$

Due to the routing of arrivals and the constraint on departures, we can show that all transitions from $\mathcal{S}_i$ to $\mathcal{S}_{i+1}$ occur through one state in $\mathcal{S}_i$ and the transitions from $\mathcal{S}_{i+1}$ to $\mathcal{S}_i$ can only go to one state in $\mathcal{S}_i$. As will be shown later, this property allows us to efficiently solve the model via *exact* decomposition based on the partition $\{\mathcal{S}_0 \cup \mathcal{S}_1 \cup \cdots\}$. Intuitively, this second modification to the model should also increase the mean number of customers in the system compared to the original model since additional server idle time is introduced and service of a suspended customers can only be resumed when all active customers depart from the system.

As an example, assume that we have a system with four homogeneous servers, and we let $C_i = 10$, for $i = 1, 2, 3, 4$. It is easy to see that $\mathcal{S}_0$ includes all states for which each queue has between 0 to 10 customers; $\mathcal{S}_1$ consists of all states for which each queue has 10 suspended customers, and has between 0 to 10 active customers and at least one queue has an active customer. Observe that the only transition from $\mathcal{S}_0$ to $\mathcal{S}_1$ is from state $(10, 10, 10, 10)$. This is due to the shortest expected delay routing of arrivals. The only non-zero transitions from $\mathcal{S}_1$ to $\mathcal{S}_0$ are from states $(11, 10, 10, 10)$, $(10, 11, 10, 10)$, $(10, 10, 11, 10)$ and $(10, 10, 10, 11)$ to state $(10, 10, 10, 10)$. This is due to the rule introduced in M^u to the effect that suspended customers are only served when the busy period (corresponding to states in $\mathcal{S}_1$) has completed. An important point is that the parameters d and C_i, for $i = 1, \ldots, K$, can be chosen to control the extent to which M^u behaves like the original model M, i.e., the larger d and the C_i's are, the larger the portion of the state space that has behavior identical to the original model.

3.1 Proof of Upper Bound

In this section, we prove that the model M^u provides an upper bound on the number of customers in the system at any point in time. In the case that the model exhibits stationary behavior, Little's result can be invoked to show that M^u provides an upper bound for the mean response time. We therefore concentrate on the mean number in the system in the remainder of this section. It is important to point out that the proofs can accommodate a general arrival process. We start by defining an auxiliary concept that will be useful in the proof.

Definition 4 *A policy Υ is a proper policy if $\boldsymbol{N} \leq \boldsymbol{N}'$ (here " $\leq$ " is taken to mean componentwise) implies that $\boldsymbol{N} + \underline{\boldsymbol{e}}^T_{l^*_\Upsilon(\boldsymbol{N})} \leq \boldsymbol{N}' + \underline{\boldsymbol{e}}^T_{l^*_\Upsilon(\boldsymbol{N}')}$ where $\underline{\boldsymbol{e}}_k$ is the column vector of all 0's except for a 1 in position k.*

It is easy to see that the minimum expected delay routing policy is a proper routing policy.

In establishing an upper bound, it is useful to look at the times when events such as arrivals and departures occur. In the latter case, it is useful to think of each server as continuously serving customers. If the queue is empty, then the server serves a *fictitious* customer. Hence *service events* at server k occur as a Poisson process with parameter μ_k. (Note that a service event is a departure event only when there is a customer in the queue.) Furthermore, if a customer is routed to an empty queue, then it is assigned the remaining service time of the fictitious customer on the server. The exponential assumption guarantees that the time to the next service event is an exponential random variable with the same parameter. It follows that, under this interpretation, the service times are still i.i.d. exponential with the same mean.

Consider the i-th event. Let $\boldsymbol{N}_i = (N_{i,1}, \cdots, N_{i,K})$ be the joint queue lengths immediately after the i-th event. Let $\boldsymbol{N}_0$ denote the initial queue lengths. We have the following evolution equations. If the $(i+1)$-st event corresponds to an arrival,

$$N_{i+1,k} = N_{i,k} + 1\{l^*_p(\boldsymbol{N}_i) = k\}. \qquad 1 \leq k \leq K \tag{2}$$

If the $(i+1)$-st event corresponds to a service event at server j,

$$N_{i+1,k} = \begin{cases} n_{i,k}, & k \neq j, \\ (N_{i,j} - 1)^+, & k = j. \end{cases} \tag{3}$$

Now suppose that we have a modified system for which we define a new binary valued random variable Y_i that takes on the value 0 if no customer is allowed to depart and the value 1 if a customer is allowed to depart at the i-th event (provided that it is a service event). In the original model M, the random variable Y_i is always equal to 1. On the other hand, in the upper bound model M^u presented above, Y_i can be 0 or 1 depending on the model state. Let $\boldsymbol{N}^u(t)$ be the joint queue lengths for the model M^u. We have the following evolution equations at the time of arrival and service events. If the $(i+1)$-st event corresponds to an arrival,

$$N^u_{i+1,k} = N^u_{i,k} + 1\{l^*_p(\boldsymbol{N}^u_i) = k\}, \qquad 1 \leq k \leq K \tag{4}$$

If it is a service event at server j,

$$N^u_{i+1,k} = \begin{cases} N^u_{i,k}, & k \neq j, \\ (N^u_{i,j} - Y_{i+1})^+, & k = j. \end{cases} \tag{5}$$

Lemma 1 *If $\boldsymbol{N}(0) \leq_{st} \boldsymbol{N}^u(0)$ and p is a proper routing policy, then*

$$\boldsymbol{N}(t) \leq_{st} \boldsymbol{N}^u(t), \qquad t \geq 0.$$

Proof. Couple the initial queue lengths so that $N(0) \leq N^u(0)$. Condition on the initial queue lengths, arrival times, and service event times. The proof is by induction on the event times to establish the deterministic relation:

$$\boldsymbol{N}_i \leq \boldsymbol{N}^u_i, \quad i \geq 0.$$

For $i = 0$, $\boldsymbol{N}(0) \leq \boldsymbol{N}^u(0)$.

Assume $\boldsymbol{N}_i \leq \boldsymbol{N}^u_i$ holds for $i = k$. For $i = k+1$, if the i-th event is an arrival event, then by the definition of a proper policy the relationship holds. If the i-th event is a service event, then due to $Y_{k+1} \leq 1$, the relationship holds. Therefore, the upper bound model M^u satisfies the assumptions described above, and we have $\boldsymbol{N}(t) \leq \boldsymbol{N}^u(t)$. By removing the conditions on initial queue lengths, arrival times, and service event times, we have:

$$\boldsymbol{N}(t) \quad \leq_{st} \quad \boldsymbol{N}^u(t), \qquad for \quad t \geq 0$$

∎

Let $N_i = \lim_{t\to\infty} N_i(t)$ when it exists, $1 \leq i \leq K$ and $N = \sum_{i=1^K} N_i$. Based on this lemma, we have:

$$E[N] \quad \leq \quad E[N_u].$$

If R and R_u denote the stationary customer response times, when they exist, then by Little's Result,

$$E[R] \quad \leq \quad E[R_u]$$

3.2 Computational algorithm for solving the Model M^u

In this section, we provide an algorithm for computing the mean response time of the upper bound model when the arrival process is Poisson with mean rate λ. We partition the state space of M^u, $\mathcal{S}^u = \cup_{i=0}^{\infty} \mathcal{S}_i$ and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$, $\forall i \neq j$, where:

$$
\begin{array}{lcl}
\mathcal{S}_0 & = & \{(N_1,\ldots,N_K)|0 \le N_j \le C_j \; for \; j = 1,\ldots,K\} \\
\mathcal{S}_i & = & \{(N_1,\ldots,N_K)|iC_j \le N_j \le (i+1)C_j \; for \; j = 1,\ldots,K\} - \{(iC_1,\ldots,iC_K)\} \\
Q_{\mathcal{S}_i,\mathcal{S}_j} & = & \text{transition rate matrix from states in } \mathcal{S}_i \text{ to states in } \mathcal{S}_j.
\end{array}
$$

The transition rate matrix Q^u has the form depicted in Figure 2 when the states are ordered in the natural way.

$$
Q^u = \begin{bmatrix}
Q_{\mathcal{S}_0,\mathcal{S}_0} & Q_{\mathcal{S}_0,\mathcal{S}_1} & 0 & 0 & 0 & \cdots \\
Q_{\mathcal{S}_1,\mathcal{S}_0} & Q_{\mathcal{S}_1,\mathcal{S}_1} & Q_{\mathcal{S}_1,\mathcal{S}_2} & 0 & 0 & \cdots \\
0 & Q_{\mathcal{S}_2,\mathcal{S}_1} & Q_{\mathcal{S}_2,\mathcal{S}_2} & Q_{\mathcal{S}_2,\mathcal{S}_3} & 0 & \cdots \\
0 & 0 & Q_{\mathcal{S}_3,\mathcal{S}_2} & Q_{\mathcal{S}_3,\mathcal{S}_3} & Q_{\mathcal{S}_3,\mathcal{S}_4} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}
$$

Figure 2: Transition Rate Matrix for the Upper Bound Model.

This is a block tridiagonal transition rate matrix and therefore represents a quasi-birth-death process. By aggregating each partition $\mathcal{S}_i$, a birth-death process is formed. First, we show how to obtain the exact conditional state probability vector, given that the system is in partition $\mathcal{S}_i$. Once we have this information, it follows easily that we can obtain the exact aggregate transition rates. We can then obtain the exact stationary state probabilities for the aggregate model. The aggregate state probabilities and the conditional state probabilities together are a complete solution for the stationary state probabilities for the upper bound model M^u.

There are several important features of the upper bound model, M^u. First, there is only a single state in $\mathcal{S}_i$ that has a non-zero transition rate into any state in $\mathcal{S}_{i+1}$, $i \ge 0$. Let us call this state $s_i(C_0)$. State $s_i(C_0)$ is:

$$
s_i(C_0) \;\; = \;\; (N_1, N_2, \ldots, N_K) \in \mathcal{S}_i \;\; where \qquad N_j = (i+1)C_j \;\; \forall \;\; j = 1,2,\ldots,K
$$

This follows from the rule used to assign an arriving customer to a server. Also, there are K states from $\mathcal{S}_i$ that have non-zero transition rates to a state in $\mathcal{S}_{i-1}$ where $i \ge 1$. Each corresponds to a state in which a server is the last to complete its "active" (non-suspended) customer. Let us call these states $s_i(l)$, $1 \le l \le K$, $i \ge 1$. These states are:

$$
\begin{array}{lrcl}
 & s_i(l) & = & (N_1, N_2, \ldots, N_K) \in \mathcal{S}_i \;\; l = 1,\ldots,K \\
\text{where} & N_l & = & iC_l + 1 \;\; and \\
 & N_j & = & iC_j \;\; for \;\; j \ne l \;\; and \;\; j = 1,2,\ldots,K
\end{array}
$$

This follows from the restrictions on departures in the upper bound model. The following are easily seen to be the transition rates between $s_i(C_0)$ and $s_{i+1}(l), l = 1, 2, \ldots, K$:

$$\begin{array}{lcll} s_i(C_0) & \rightarrow & s_{i+1}(l_p^*(s_i(C_0)) & \lambda \\ s_{i+1}(l) & \rightarrow & s_i(C_0) & \mu_l \;\; for \;\; l = 1, 2, \ldots, K \end{array}$$

The second important observation is that the submatrices $Q_{\mathcal{S}_i,\mathcal{S}_i}$, for $i \geq 1$, are all identical. The conditional state probabilities $P\{s \in \mathcal{S}_i | \mathcal{S}_i\}$ can now be computed exactly using the following lemma from [10]:

Lemma 2 *Given an irreducible Markov process with state space $S = A \cup B$ and transition rate matrix:*

$$\left[\begin{array}{cc} Q_{A,A} & Q_{A,B} \\ Q_{B,A} & Q_{B,B} \end{array} \right]$$

where $Q_{i,j}$ is the transition rate sub-matrix from partition i to partition j. If $Q_{B,A}$ has all zero entries except for some non-zero entries in the i-th column, the conditional steady state probability vector, given that the system is in partition A, is the solution to the following system of linear equations:

$$\begin{array}{rcl} \vec{\pi}_{|A} \left[Q_{A,A} + Q_{A,B} \; \underline{\underline{\mathbf{e}}} \; \underline{\underline{\mathbf{e}}}_i^T \right] & = & \vec{0} \\ \vec{\pi}_{|A} \; \underline{\underline{\mathbf{e}}} & = & 1 \end{array}$$

where $\underline{\underline{\mathbf{e}}}_i^T$ is a row vector with a 0 in each component, except for the i-th component which has value 1.

We are now in a position to compute the conditional state probabilities for each partition $\mathcal{S}_i$ of M^u exactly. Without loss of generality, let us consider $\mathcal{S}_i$, for some $i \geq 1$.

Theorem 1 *Let $\tilde{Q}_{\mathcal{S}_i,\mathcal{S}_i}$ be the transition rate matrix which is equal to $Q_{\mathcal{S}_i,\mathcal{S}_i}$ except for the following modifications:*

$$\tilde{q}_{s_i(C_0),s_i(C_0)} = q_{s_i(C_0),s_i(C_0)} + \lambda \tag{6}$$

$$\tilde{q}_{s_i(l),s_i(1)} = q_{s_i(l),s_i(1)} + \mu_l \qquad where\ 1 \leq l \leq K \tag{7}$$

The solution to the following system of linear equations:

$$\vec{\tilde{\pi}} \tilde{Q}_{\mathcal{S}_i,\mathcal{S}_i} = \vec{0} \qquad and \qquad \vec{\tilde{\pi}} \underline{e} = 1$$

is the conditional steady state probability vector for states in $\mathcal{S}_i$, that is:

$$\tilde{\pi}(s) = \frac{\pi(s)}{\sum_{s \in \mathcal{S}_i} \pi(s)} \qquad \forall \; s \in \mathcal{S}_i$$

Proof: Let us partition the state space $\mathcal{S}^u = \{\mathcal{S}_i' \cup \mathcal{S}_i''\}$ where $\mathcal{S}_i' = \cup_{j=0}^{j=i} \mathcal{S}_j$ and $\mathcal{S}_i'' = \{\mathcal{S}^u - \mathcal{S}_i'\}$. There is only a single return state in $\mathcal{S}_i'$, which is $s_i(C_0)$, from the states in $\mathcal{S}_i''$. Based on Lemma 2, the modification of Equation (6) provides the conditional steady state probability, given the system is in $\mathcal{S}_i'$. Now partition the state space $\mathcal{S}_i' = \{\mathcal{S}_i^1 \cup \mathcal{S}_i\}$ where $\mathcal{S}_i^1 = \cup_{j=0}^{i-1} \mathcal{S}_j$. Based on Equation (7) and the definition of the MED routing policy, we obtain the conditional state probability vector, given the system is in state $\mathcal{S}_i$. ∎

Since we can compute the conditional state probabilities for each partition $\mathcal{S}_i$ exactly, we can exactly aggregate each $\mathcal{S}_i$ into a single state $s_i, i \geq 0$. The aggregated process is depicted in Figure 3 where, λ_0, λ_{agg} and μ_{agg} are:

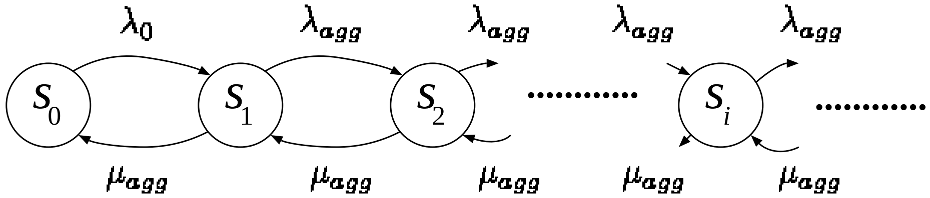

Figure 3: Aggregate Process for the Upper Bound Model.

$$
\begin{aligned}
\lambda_0 &= \tilde{\pi}(s_0(C_0))\ \lambda \\
\lambda_{agg} &= \tilde{\pi}(s_i(C_0))\ \lambda \\
\mu_{agg} &= \sum_{l=1}^{K} \tilde{\pi}(s_i(l))\ \mu_l
\end{aligned}
$$

Solving this chain, we have:

$$
\pi^*(s_0) = \left[1 + \frac{\lambda_0}{\mu_{agg} - \lambda_{agg}}\right]^{-1} \tag{8}
$$

$$
\pi^*(s_i) = \left[1 + \frac{\lambda_0}{\mu_{agg} - \lambda_{agg}}\right]^{-1} \left(\frac{\lambda_0}{\mu_{agg}}\right) \left(\frac{\lambda_{agg}}{\mu_{agg}}\right)^{i-1} \qquad for \quad i = 1, 2, \ldots \tag{9}
$$

To obtain the mean number of customers, N_u, in the upper bound model, let us define the following:

$$
\begin{aligned}
r(s) &= \sum_{i=1}^{K} N_i \qquad\qquad for\ state\ s \in \mathcal{S}^u \\
C_0 &= \sum_{i=1}^{K} C_i
\end{aligned}
$$

$$
\begin{aligned}
\tilde{r}(s) &= r(s) - iC_0 && s \in \mathcal{S}_i \\
\tilde{N}(s_i) &= \sum_{s \in \mathcal{S}_i} \tilde{r}(s)\tilde{\pi}(s)
\end{aligned}
$$

where $\tilde{\pi}(s)$ is the solution of the following Markov chain:

$$\vec{\tilde{\pi}}\tilde{Q}_{\mathcal{S}_i,\mathcal{S}_i} = \vec{0} \qquad and \qquad \vec{\tilde{\pi}}\underline{e} = 1$$

Then we have:

$$N_u \;=\; \tilde{N}(s_0)\pi^*(s_0) + \sum_{i=1}^{\infty}\left[\tilde{N}(s_i) + iC_0\right]\pi^*(s_i) \tag{10}$$

Since $\tilde{N}(s_i) = \tilde{N}(s_j)$ for $i \neq j$ where $i, j \geq 1$, we can simplify the expression above for N_u to:

$$N_u \;=\; \tilde{N}(s_0)\pi^*(s_0) + \tilde{N}(s_i)(1 - \pi^*(s_0)) + C_0\lambda_0 \frac{\mu_{agg}}{(\mu_{agg} - \lambda_{agg})^2}\pi^*(s_0) \tag{11}$$

From Little's result [19], the upper bound mean system response time R_u is:

$$R_u \;=\; \frac{1}{\lambda}\left[\tilde{N}(s_i)\pi^*(s_0) + \tilde{N}(s_1)(1 - \pi^*(s_0)) + C_0\lambda_0 \frac{\mu_{agg}}{(\mu_{agg} - \lambda_{agg})^2}\pi^*(s_0)\right] \tag{12}$$

It is important to note that the upper bound model M^u has a different stability condition compared to the original model M. The original model is stable if:

$$\rho \;=\; \frac{\lambda}{\sum_{i=1}^{K}\mu_i} < 1$$

but the stability condition of the upper bound model is:

$$\rho^u \;=\; \frac{\lambda_{agg}}{\mu_{agg}} < 1$$

In general, $\rho^u > \rho$ but as we increase d and $\overline{C}$, we have $\rho^u \to \rho$ from below.

Lastly, to comment about the computational complexity of the upper bound model M^u. First, we have to obtain the conditional state probabilities for rate matrices $\mathcal{S}_0$ and $\mathcal{S}_1$. This can be accomplish by using numerical methods, such as the power iteration method, as suggested in [29]. Although the theoretical complexity is $O(n^3)$ where n is the dimension of the rate matrix. In practice, the number of operations is much less. After we obtain the conditional state probabilities, we can use Equations (10), (11) and (12) to obtain the expected response time of the upper bound model.

4 Lower Bound

In this section we present a model M^l, which provides a lower bound on the mean response time of the original model. As before, the arrival process is Poisson, the service time are exponentially distributed and that there are K servers with service rates $\mu_i, i = 1, 2, \ldots, K$, where $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. For the lower bound model, in addition to the two parameters introduced for the upper bound model M^u, we define $C_f = \sum_{i=1}^{K} C_i$.

We first give an intuitive idea of the construction of the lower bound model M^l. The modified system alternates between two phases. The *normal service phase* begins when the system is empty and continues until either the maximum degree of imbalance d is exceeded or until the total number of customers exceeds C_f. Once either event occurs, the system enters a *full service* phase where it behaves as a heterogeneous M/M/K system in which, if there are j customers, where $j \leq K$, these j customers are executed on the j fastest servers (i.e., customers are moved to the faster servers instantaneously). The system operates in this mode until the system becomes idle. Once the system empties, it returns to the normal service mode. Intuitively, these modifications yield a lower bound on the mean response time since the modifications are an idealization in which either the model behaves exactly as the original model or the best possible service rate is delivered. While the result is intuitive, we will also formally prove that the modified model M^l yields a lower bound on the mean response time. Of course, it is intended that d and C_i, $i = 1, 2, \ldots, K$, be chosen large enough so that most of the time M^l behaves like the original model. On the other hand, to be able to solve the model efficiently, we would like to keep these parameters small. Numerical examples are given later to illustrate the tradeoffs between the size of a model solved and the spread in the bounds obtained.

4.1 Proof that M^l provides a lower bound for M

In this section, we prove that M^l provides a lower bound on the mean response time of the system for all proper routing policies p. Again, we point out that the proof can accommodate a general arrival process. The proof is based on the following two lemmas. The first is straightforward and requires little explanation. It will be used to establish the bound during the normal service phase.

Lemma 3 *If $\boldsymbol{N}(0) \leq_{st} \boldsymbol{N}'(0)$ and p is a proper routing policy, then*

$$\boldsymbol{N}(t) \leq_{st} \boldsymbol{N}'(t), \quad 0 \leq t.$$

Proof. Without loss of generality, we can couple the systems, so that $\boldsymbol{N}(0) \leq \boldsymbol{N}'(0)$. Now condition on the arrival times and on the service event times at the different servers during the time interval $[0,t]$. A simple induction argument using the fact that p is a proper policy suffices to establish that $\boldsymbol{N}(t) \leq \boldsymbol{N}'(t)$. Removal of the conditioning yields the desired result. ∎

Consider the system operating solely in the full service mode of operation and let $N^f(t)$ denote the *total number of customers* in the system. Let $N^p(t) = \sum_{i=1}^K N_i^p$ denote the total number of customers in the original system under policy p (henceforth referred to as the normal service system).

Lemma 4 *If $N^f(0) \leq_{st} N^p(0)$ and p is proper routing policy, then*

$$N^f(t) \leq_{st} N^p(t), \quad 0 \leq t.$$

Proof. As before, we couple the initial queue lengths so that $N^f(0) \leq N^p(0)$ and condition on the arrival and departure times. Let $\{t_n\}$ be a sequence of times where each t_i corresponds to an arrival or service event. Let $M^p(t_n)$ denote the number of busy servers at time t_n in the system under policy p. Define $\{1,\ldots,K\} \to \{1,\ldots,K\}$ to be a mapping such that $\gamma_n^p(k)$ is the index of the k-th fastest busy server in the system, provided $k \leq M^p(t_n)$. In the case that $K \geq k > M^p(t_n)$, $\gamma_n^p(k)$ is the index of the $(k - M^p(t_n))$-th fastest idle server. (Actually, the idle servers can be mapped in an arbitrary manner.) We introduce the following sequences of random variable's,

- $\{A_n\}$ is a sequence of random variable such that $A_n = 1$ if the n-th event is an arrival and 0 if it is a service event.
- $\{I_n\}$ is an independent and identically distributed sequence of random variable taking values from $\{1,\ldots,K\}$ such that $\Pr[I_n = k] = 1/K$, $k = 1,2,\ldots,K$, and 0 otherwise.
- $\{B_n\}$ is an i.i.d. sequence of uniformly distributed random variable in the interval $[0,1]$.

The evolution of the two systems is described as follows. Let $\boldsymbol{N}_n^p$ denote the joint queue lengths under p immediately after the n-th event and let N_n^f denote the total number of customers in the full service system immediately after the n-th event. Let $N_{n,k}^p$ be the k-th component of $\boldsymbol{N}_n^p$. We have:

$$N_{n,k}^p = (N_{n-1,k}^p - 1\{(A_n = 0)\wedge(I_n = l)\wedge(\gamma_{n-1}^p(l) = k)\wedge(B_n < \mu_k/\mu)\})^+ +$$

$$A_n 1\{l_p^*(\boldsymbol{N}_{n-1}^p) = k\} \tag{13}$$

$$N_n^f \;=\; (N_{n-1}^f - 1\{(A_n = 0) \wedge (I_n = l) \wedge (B_n < \mu_l/\mu)\})^+ + A_n \tag{14}$$

It remains to establish that N_n^f is less than N_n^p (the total number of customers under policy p) immediately after the n-th event for $n = 1, 2, \ldots$. This is easily done by induction.

Basis step. For $t_0 = 0$, the result follows from the coupling of the initial queue lengths.

Inductive step. Assume that the hypothesis holds for the first $n-1$ events. We must distinguish between arrivals and service events. If an arrival occurs at time t_n ($A_n = 1$), then the result follows immediately from the above evolution equations. In the case of a service event, we distinguish between four cases depending on whether I_n corresponds to a busy or idle server in each system.

Case (1): In both systems the server in the chosen position is idle. Then there is no departure from either system and the full service system model continues to have a lower total number of customers, i.e., $N_n^f = N_{n-1}^f \leq N_{n-1}^p = N_n^p$.

Case (2): In the normal service system, the chosen position corresponds to a busy server, but in the full service model it corresponds to an idle server. In the normal service system there can be customers waiting in queues while some servers are idle. This does not occur with the full service system. It follows that the total number of customers in the full service system is strictly less than the total number of customers in the normal service model in the interval $t_{n-1} \leq t < t_n$, i.e., $N_{n-1}^f < N_{n-1}^p$. Hence, $N_n^f \leq N_n^p$ since the normal service system only "catches up" by 1.

Case (3): The server is busy in the full service system, but it is not busy in the normal service system. Clearly $N_n^f \leq N_{n-1}^f \leq N_{n-1}^p = N_n^p$.

Case (4): The servers are busy in both systems. In this case let j be the label of the server in the full service system and let k be the index of the server in the normal service system. Since, in the full service system, the fastest servers are always being utilized it follows that $j \leq k$, i.e., the chosen server in the full service system is at least as fast as the chosen server in the normal mode system. Therefore, if $B_n \leq \mu_k/\mu$, then $B_n \leq \mu_j/\mu$. Hence it follows from the evolution equations, if there is a departure from the normal service system, then there is also a departure from the full service system. So we conclude that $N_n^f \leq N_n^p$

This completes the inductive step. Removal of the conditioning on the initial queue lengths, the arrival times, and the service events completes the proof. ∎

Lastly, let $N^l(t)$ denote the total number of customers in the lower bound system at time t. We have the following result.

Theorem 2 *If $N^l(0) \leq_{st} N^p(0)$, then*

$$N^l(t) \leq_{st} N^p(t), \quad t \geq 0,$$

for any proper routing policy p.

Proof. This follows directly from the above two lemmas by noting that M^l goes through alternating intervals in which it operates in normal mode and full service mode. When the transition is made from the full service phase to the normal phase, $N^l(t) = 0$ which implies that $\boldsymbol{N}^l(t) \leq \boldsymbol{N}^p(t)$ and so the first lemma can be applied during each normal service mode interval. Similarly, when there is a transition from the normal service phase to the full service phase, $\boldsymbol{N}^l(t) \leq \boldsymbol{N}^p(t)$ which implies that $N^f(t) \leq N^p(t)$ and so the second lemma is applicable during every full service mode interval. ∎

It is important to note that the stability conditions for the lower bound model M^l and the original model M are the same.

4.2 Computational algorithm for solving the model M^l

In this section, we describe an algorithm for computing the mean response time of the lower bound model M^l. Let us define the following notation:

$\mathcal{S}_0$	$=$ set of states with $0 \leq N_j \leq C_j$, $j = 1, 2, \ldots, K$ such that the threshold d is respected
$\mathcal{G}_1$	$= \{\mathcal{S}_0 - (0, 0, \ldots, 0)\}$.
a_i	$=$ a state not in the set $\mathcal{S}_0$, in which the system contains exactly i customers.
$Q_{\mathcal{G}_1, a_i}$	$=$ transition rate matrix between $\mathcal{G}_1$ and state a_i.
$g^*_{a_i, a_j}$	$=$ transition rate from state a_i to state a_j.

The transition rate matrix of the model M^l is depicted in Figure 4. (Note that some of the $Q_{\mathcal{G}_1, a_i} = 0$ but this will not effect the development that follows.)

$$\left[\begin{array}{cc|ccccc} Q_{a_0,a_0} & Q_{a_0,\mathcal{G}_1} & 0 & 0 & 0 & 0 & \cdots \\ Q_{\mathcal{G}_1,a_0} & Q_{\mathcal{G}_1,\mathcal{G}_1} & Q_{\mathcal{G}_1,a_1} & Q_{\mathcal{G}_1,a_2} & Q_{\mathcal{G}_1,a_3} & Q_{\mathcal{G}_1,a_4} & \cdots \\ \hline g^*_{a_1,a_0} & 0 & g^*_{a_1,a_1} & \lambda & 0 & 0 & \cdots \\ 0 & 0 & g^*_{a_2,a_1} & g^*_{a_2,a_2} & \lambda & 0 & \cdots \\ 0 & 0 & 0 & g^*_{a_3,a_2} & g^*_{a_3,a_3} & \lambda & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right.$$

Figure 4: Transition rate matrix for lower bound model.

Since $\mathcal{S}_0$ represents all possible states during the normal mode and states $a_i, i \geq 1$, represent all possible states during the full service mode, it is easy to see that the transition rate $g^*_{a_i,a_{i-1}}$ is:

$$g^*_{a_i,a_{i-1}} = \begin{cases} \sum_{j=1}^{i} \mu_j & 1 \leq i \leq K \\ \sum_{j=1}^{K} \mu_j & i > K \end{cases} \tag{15}$$

Observe that if we know the conditional state probabilities for the states in $\mathcal{S}_0$ (where $\mathcal{S}_0 = \{a_0 \cup \mathcal{G}_1\}$), then we can aggregate $\mathcal{S}_0$ as a single state, s_0, and we will have a simple aggregated process from which the mean number of customers in the system can be easily derived. Note that there is only a single entry to $\mathcal{S}_0$ from all states outside $\mathcal{S}_0$ because the system must be idle to switch from full service mode to the normal mode. Based on Lemma 2, the state probabilities conditioned on the system being in $\mathcal{S}_0$ can be obtained by solving the following system of linear equations:

$$\begin{aligned} \vec{\tilde{\pi}}(\mathcal{S}_0) \left[Q_{\mathcal{S}_0,\mathcal{S}_0} + \sum_{i=1}^{C_f+1} Q_{\mathcal{S}_0,a_i} \, \underline{e}_0^T \right] &= \vec{0} \\ \vec{\tilde{\pi}}(\mathcal{S}_0) \, \underline{e} &= 1 \end{aligned}$$

where $\vec{\tilde{\pi}}(\mathcal{S}_0)$ is the steady state probability vector, given that the system is in $\mathcal{S}_0$. We can now apply exact aggregation; the aggregated process is depicted in Figure 5.

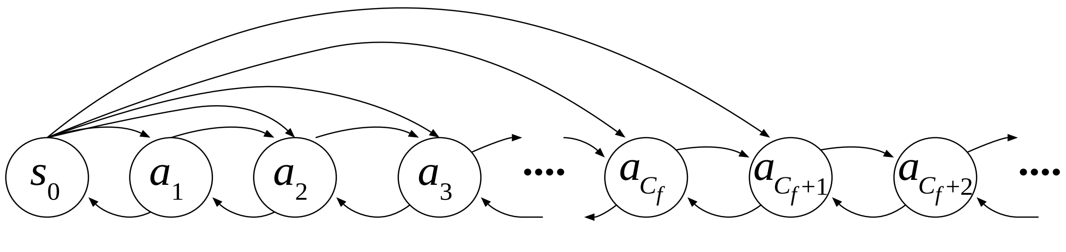

Figure 5: Aggregate Process for the Lower Bound Model.

The transition rates for the aggregated chain are:

$$\begin{aligned} g^*_{s_0,a_i} &= \vec{\tilde{\pi}}(\mathcal{S}_0)Q_{\mathcal{S}_0,a_i} && i=1,\ldots,C_f+1 \\ g^*_{a_i,a_{i+1}} &= \lambda && i \geq 1 \\ g^*_{a_1,s_0} &= \mu_1 \\ g^*_{a_i,a_{i-1}} &= \begin{cases} \sum_{j=1}^{i}\mu_j & i=2,3,\ldots,K \\ \mu^* & \text{otherwise} \end{cases} \end{aligned}$$

where $\mu^* = \sum_{i=1}^{K}\mu_i$.

Solving the chain, we have:

$$\pi^*(s_0) = \left[1 + \sum_{i=1}^{C_f+1}\sum_{j=1}^{i}[\lambda^{i-j}(\sum_{k=j}^{C_f+1} g^*_{s_0,a_j})(\prod_{k=j}^{i} g^*_{a_k,a_{k-1}})^{-1}] + \frac{\lambda}{\mu^*-\lambda}\sum_{j=1}^{C_f+1}[\lambda^{C_f+1-j}(\sum_{k=j}^{C_f+1} g^*_{s_0,a_j})(\prod_{k=j}^{C_f+1} g^*_{a_k,a_{k-1}})^{-1}]\right]^{-1} \quad (16)$$

$$\pi^*(a_i) = \pi(s_0)\sum_{j=1}^{i}[\lambda^{i-j}(\sum_{k=j}^{C_f+1} g^*_{s_0,a_j})(\prod_{k=j}^{i} g^*_{a_k,a_{k-1}})^{-1}] \qquad i=1,\ldots,C_f+1 \quad (17)$$

$$\pi^*(a_i) = \pi(s_0)(\frac{\lambda}{\mu^*})^{i-C_f-1}\sum_{j=1}^{C_f+1}[\lambda^{C_f+1-j}(\sum_{k=j}^{C_f+1} g^*_{s_0,a_j})(\prod_{k=j}^{C_f+1} g^*_{a_k,a_{k-1}})^{-1}] \qquad i=C_f+2,\ldots \quad (18)$$

To obtain the mean number of customers in the system, N_l, and the mean response time, R_l, let

$$\tilde{N}(\mathcal{S}_0) = \sum_{s\in\mathcal{S}_0} r(s)\tilde{\pi}(s)$$

where $r(s) = \sum N_i$, then we have:

$$\begin{aligned} N_l &= \tilde{N}(\mathcal{S}_0)\pi^*(s_0) + \sum_{i=1}^{C_f} i\pi^*(a_i) + \sum_{i=C_f+1}^{\infty} i\pi^*(a_i) \\ &= \tilde{N}(\mathcal{S}_0)\pi^*(s_0) + \sum_{i=1}^{C_f} i\pi^*(a_i) + \sum_{i=C_f+1}^{\infty} i\,\pi^*(a_{C_f+1})\left(\frac{\lambda}{\mu^*}\right)^{i-C_f-1} \end{aligned}$$

After simplifying, the mean number of customers N_l is:

$$N_l = \tilde{N}(\mathcal{S}_0)\pi^*(s_0) + \sum_{i=1}^{C_f} i\pi^*(a_i) + \frac{C_f\ \pi^*(a_{C_f+1})\mu^*}{\mu^*-\lambda} + \frac{\pi^*(a_{C_f+1})\mu^*}{(\mu^*-\lambda)^2} \quad (19)$$

From Little's result [19], the lower bound mean response time is:

$$R_l \;=\; \frac{1}{\lambda}\left[\tilde{N}(\mathcal{S}_0)\pi^*(s_0) + \sum_{i=1}^{C_f} i\pi^*(a_i) + \frac{C_f\;\pi^*(a_{C_f+1})\mu^*}{\mu^* - \lambda} + \frac{\pi^*(a_{C_f+1})\mu^*}{(\mu^* - \lambda)^2}\right] \qquad (20)$$

Lastly, to comment about the computational complexity of the lower bound model M^l. First, we have to obtain the conditional state probabilities for rate matrix $\mathcal{S}_0$. Again, this can be accomplish by using the power iteration method. After we obtain the conditional state probabilities, we can use Equations (16) to (20) to obtain the expected response time of the lower bound model.

5 Homogeneous Servers

In this section we consider a system with K homogeneous servers having exponential service times with rate μ. In this case, we can improve on the lower bound for the heterogeneous system as well as on the upper bound at high utilization. Here, the minimum expected delay policy becomes the classical *join the shortest queue (SQ)* policy.

We first describe the new upper bound model under very high system utilization. For the upper bound model M^u in the Section 3, we do not have a very tight upper bound under *very high* system utilization, since we put a constraint on the departure events based on the state of the system. Due to this constraint, the upper bound model saturates at a lower traffic intensity; if we can find an upper bound model that saturates at the same point as the original model, we can use the minimum of this model and M^u model as an upper bound. One simple upper bound for the homogeneous case which has the same saturation point as the original model is formed by assigning customers to servers in a cyclic fashion, [11]. In this case, each server in the system behaves as an $E_K/M/1$, and the mean response time of this system is well known [18]. Taking the minimum response time of this model and M^u provides a good upper bound over the entire range of traffic intensity[4].

We now define the new lower bound model under the identical servers assumption. Let $\boldsymbol{N}(t) = (N_1(t), N_2(t), \ldots, N_K(t))$ denote the joint queue lengths at time $t > 0$ under SQ, and let $N(t) = \sum_{k=1}^{K} N_k(t)$. Let $\hat{N}_k(t)$ denote the k-th largest queue length, $k = 1, 2, \ldots, K$ at time $t \geq 0$. The new lower bound system operates as follows:

[4]Note that this approach cannot be applied to the heterogeneous case since a cyclic assignment policy may not provide an upper bound response time.

- Whenever $N(t) < C_f = \sum_{i=1}^{K} C_i$, $\hat{N}_1 - \hat{N}_K = d$, and a departure would normally occur from the smallest queue, it is forced to occur instead from the next largest queue (i.e., if a departure would cause the system to exceed the maximum degree of imbalance d, then the departure is made to occur from the second shortest queue, which is a form of jockeying).
- Whenever $N(t) \geq C_f$ and a departure occurs, it is taken from the largest queue.

Here C and d are parameters that can be tuned to provide a tight bound.

In order to describe in what sense this system is a lower bound, we introduce the concept of *majorization* [22]. Let $\boldsymbol{X}, \boldsymbol{Y} \in I\!N^K$.

Definition 5 *$\boldsymbol{Y}$ is said to majorize $\boldsymbol{X}$ (written $\boldsymbol{X} \prec \boldsymbol{Y}$) iff*

$$\begin{aligned} \sum_{l=1}^{k} \hat{X}_l &\leq \sum_{l=1}^{k} \hat{Y}_l, \quad k = 1, \ldots, K-1, \\ \sum_{l=1}^{K} \hat{X}_l &= \sum_{l=1}^{K} \hat{Y}_l. \end{aligned} \tag{21}$$

where $\hat{X}_l$ ($\hat{Y}_l$) is the $l-$largest component of $\boldsymbol{X}$ ($\boldsymbol{Y}$). If we replace the equality in (21) by

$$\sum_{l=1}^{K} \hat{X}_l \leq \sum_{l=1}^{K} \hat{Y}_l,$$

we obtain a weaker ordering. In this case we say that $\boldsymbol{Y}$ *weakly majorizes* $\boldsymbol{X}$ (written $\boldsymbol{X} \prec_w \boldsymbol{Y}$).

The following lemma states some properties regarding operations that can be performed on $\boldsymbol{X}$ and $\boldsymbol{Y}$ such that weak majorization is preserved.

Lemma 5 *Let $\boldsymbol{X}, \boldsymbol{Y} \in I\!N^K$ such that $\boldsymbol{X} \prec_w \boldsymbol{Y}$, then*

1. $(\hat{X}_1, \ldots, \hat{X}_k, \ldots, \hat{X}_l + 1, \ldots, \hat{X}_K) \prec_w (\hat{Y}_1, \ldots, \hat{Y}_k + 1, \ldots, \hat{Y}_l, \ldots, \hat{Y}_K)$, *for* $1 \leq k \leq l \leq K$
2. $(\hat{X}_1, \ldots, (\hat{X}_k - 1)^+, \ldots, \hat{X}_l, \ldots, \hat{X}_K) \prec_w (\hat{Y}_1, \ldots, \hat{Y}_k, \ldots, (\hat{Y}_l - 1)^+, \ldots, \hat{Y}_K)$, *for* $1 \leq k \leq l \leq K$

Proof. The proof follows in a straightforward manner from the definition of "$\prec_w$". The reader is referred to [22] for a detailed proof. ∎

Before we define a stochastic comparison based on majorization, we introduce the notion of a *Schur-convex function.*

Definition 6 *A function $\phi : I\!N \to I\!R$ is said to be Schur-convex iff*

$$\phi(\boldsymbol{X}) \leq \phi(\boldsymbol{Y}), \quad \forall \boldsymbol{X}, \boldsymbol{Y} \in I\!N^K \qquad \text{such that } \boldsymbol{X} \prec \boldsymbol{Y}.$$

Definition 7 *If $\boldsymbol{X}, \boldsymbol{Y} \in I\!N^K$ are random variables, then we say $\boldsymbol{X}$ is smaller than $\boldsymbol{Y}$ in the sense of Schur-convex order (written $\boldsymbol{X} \leq_{scx} \boldsymbol{Y}$) iff*

$$\phi(\boldsymbol{X}) \leq_{st} \phi(\boldsymbol{Y}), \qquad \forall \text{Schur-convex}\, \phi.$$

If the class of functions is restricted to be increasing Schur-convex, then we say that $\boldsymbol{X}$ is smaller than $\boldsymbol{Y}$ in the sense of increasing Schur-convex order ($\boldsymbol{X} \leq_{iscx} \boldsymbol{Y}$).

Last, if ϕ is a function such that $\phi(\boldsymbol{X}) \leq \phi(\boldsymbol{Y}), \quad \forall \boldsymbol{X}, \boldsymbol{Y} \in I\!N^K$ such that $\boldsymbol{X} \prec_w \boldsymbol{Y}$, then ϕ can be shown to be an increasing Schur-convex function.

Let $\boldsymbol{N}^l(t)$ denote the joint queue length vector for the new lower bound system. We have the following result.

Theorem 3 *If $\boldsymbol{N}^l(0) = \boldsymbol{N}(0)$, then $\boldsymbol{N}^l(t) \leq_{iscx} \boldsymbol{N}(t)$ $\forall t > 0$.*

Proof. Couple the initial queue lengths so that $\boldsymbol{N}^l(0) = \boldsymbol{N}(0)$. Condition on the arrival times of the two systems. For the k-th largest queue, we have an associated *service event process* which is a Poisson process with parameter μ. Whenever a service event occurs associated with the k-th largest queue, a departure occurs if there is one or more customer in the queue at the time of the event. Observe that the coupling of the the service event times at the different servers is only possible if the service times at the servers are all mutually independent sequences of i.i.d exponential random variables with the same parameter.

Let $\{t_i\}_{n=0}^{\infty}$ be the sequence of times at which arrivals or service events occur ($t_0 \equiv 0$). We will establish the relation $\boldsymbol{N}^l(t) \prec_w \boldsymbol{N}(t)$ by induction on the event times. Clearly, if $\boldsymbol{N}^l(t_i) \prec_w \boldsymbol{N}(t_i)$ then $\boldsymbol{N}^l(t) \prec_w \boldsymbol{N}(t)$, $t_i \leq t < t_{i+1}$, $i = 0, 1, \ldots$

Basis step. This follows from the coupling of the initial queue lengths.

Inductive step. Assume that $\boldsymbol{N}^l(t) \prec_w \boldsymbol{N}(t)$ for $t < t_i$. We will establish it now for $t = t_i$. There are two cases depending on whether the event is an arrival or a service event.

Arrival. $\boldsymbol{N}^l(t_i) \prec_w \boldsymbol{N}(t_i)$ follows because arrivals are always to the smallest queue, so property 1 of lemma 5 can be applied.

Service event. There are two cases depending on whether $N^l(t_i^-) < C_f$. In either case, the result follows from an application of property 2 of lemma 5.

This completes the inductive step and thus we have $\boldsymbol{N}^l(t) \prec_w \boldsymbol{N}(t)$, $t \geq 0$. By the definition of weak majorization ($\prec_w$), this implies that $f(N^l(t)) \leq f(N(t))$ for any increasing Shur-convex function $f(t)$. Removing the conditioning on the arrival times and service times, we have:

$$N^l(t) \leq_{iscx} N(t) \forall t > 0$$

∎

Corollary 1 *If* $\boldsymbol{N}^l(0) \leq_{iscx} \boldsymbol{N}(0)$*, then* $N^l(t) \leq_{st} N(t)$*, for* $t \geq 0$*.*

Proof. This follows from the preceding theorem and the fact that $\phi(\boldsymbol{X}) = \sum_{k=1}^{K} X_k$ is an increasing Schur-convex function. ∎

For the purpose of computing performance measures, let us define the following:

$\mathcal{S}_0$	= set of states with $0 \leq N_j \leq C$ and $\lvert N_i - N_j \rvert \leq d$, $\forall i, j$
$s_o(C_o)$	= this is the only state in $\mathcal{S}_0$ that has a positive transition rate into it from states outside $\mathcal{S}_0$.
$\tilde{\pi}(s_o(C_0))$	= conditional probability of state $s_o(C_0)$, given that the system is in $\mathcal{S}_0$.
s_0	= aggregate state which represents all states of $\mathcal{S}_0$.
s_i	= state which represents the system having $C_f + i$ customers, where $i = 1, 2, \ldots$.
$\pi^*(s_i)$	= steady state probability of state s_i.
$\tilde{N}(\mathcal{S}_0)$	= mean number of customers given that the system is in $\mathcal{S}_0$.

The mean number of customers and mean response time for this lower bound are:

$$
\begin{aligned}
N_l &= \tilde{N}(\mathcal{S}_0)\pi^*(s_0) + \sum_{i=1}^{\infty}[C_f + i]\,\pi^*(s_i) \\
&= \tilde{N}(\mathcal{S}_0)\pi^*(s_0) + C_f(1-\pi^*(s_0)) + \lambda_0\frac{K\mu}{(K\mu-\lambda)^2}\pi^*(s_0) \qquad \mathit{and} \qquad (22)\\
R_l &= \frac{1}{\lambda}\left[\tilde{N}(\mathcal{S}_0)\pi^*(s_0) + C_f(1-\pi^*(s_0)) + \lambda_0\frac{K\mu}{(K\mu-\lambda)^2}\pi^*(s_0)\right] \qquad (23)
\end{aligned}
$$

where:

$$
\begin{aligned}
\lambda_0 &= \tilde{\pi}(s_0(C_0))\lambda \qquad (24)\\
\pi^*(s_0) &= \left[1+\frac{\lambda_0}{K\mu-\lambda}\right]^{-1} \qquad (25)\\
\pi^*(s_i) &= \left[1+\frac{\lambda_0}{K\mu-\lambda}\right]^{-1}\left(\frac{\lambda_0}{K\mu}\right)\left(\frac{\lambda}{K\mu}\right)^{i-1} \qquad \mathit{for} \;\; i=1,2,\ldots \qquad (26)
\end{aligned}
$$

6 State Space Reduction by Lumpability

In the previous section, we discussed a methodology for constructing an upper bound model M^u and a lower bound model M^l. The computational costs in solving the models can be broken down into:

1. obtaining the conditional state probabilities in $\mathcal{S}_0$ and $\mathcal{S}_1$,
2. obtaining the steady state probabilities of the aggregated process and,
3. obtaining the performance measure, e.g., expected response time or expected number of customers.

The larger the state space cardinality of $\mathcal{S}_i$, the more accurate are the results obtained. In this section, we discuss how we can reduce the state space of $\mathcal{S}_i$ by lumping *similar* states.

Kemeny and Snell [17] studied the conditions under which an aggregated process is still Markovian. The condition for a Markov process to be lumpable with respect to a partition $\{\mathcal{P}_0 \cup \mathcal{P}_1 \cup \cdots\}$, where $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$, for $i \neq j$, is that for every pair of sets $\mathcal{P}_i$ and $\mathcal{P}_j$, $r_{k,\mathcal{P}_j}$ has the same value for every state $k \in \mathcal{P}_i$ where

$$
r_{k,\mathcal{P}_j} = \sum_{l\in\mathcal{P}_j} q_{k,l} \qquad \mathit{for}\ k \in \mathcal{P}_i \qquad (27)
$$

We can apply this notion to our minimum expected delay routing problem.

Let J be the number of distinct types of servers in the model where two servers are of the same type if and only if they have the same service rate. For any state s define the following mapping:

$$f : s \rightarrow \{l_i | i = 1, 2, \ldots, J\} \tag{28}$$

where:

l_i = a set of tuples $(\alpha_{ij}, \beta_{ij})$
α_{ij} = a queue length for a server of type i that appears in state s
β_{ij} = the number of servers of type i that have queue length α_{ij} in state s

We define a partition of the state space $\mathcal{S}_u$ ($\mathcal{S}_l$) by specifying that $s_1, s_2 \in \mathcal{S}_u$($\mathcal{S}_l$) are in the same partition if and only if $f(s_1) = f(s_2)$.

For example, assume that we have a four server system with $\mu_1 = \mu_2 = 4$, $\mu_3 = 3$ and $\mu_4 = 2$. There are three distinct types of servers, so $J = 3$. We can group states such as $s_1 = (3, 4, 2, 1)$ and $s_2 = (4, 3, 2, 1)$ into the same partition since the $l_i, i = 1, 2, 3$, for both states are:

$$l_1 = \{(4, 1), (3, 1)\}; l_2 = \{(2, 1)\}; l_3 = \{(1, 1)\}$$

It is not difficult to see that the condition for lumpability is satisfied and we can often significantly reduce the state space of the model.

7 Application and Numerical Examples

In this section, we present two examples to illustrate applications of this methodology and the accuracy of the bounding algorithm.

The first example concerns the transmission policy in a computer network. For a packet switching system, there are basically two modes of data transmission. In one case, the *virtual circuit* transmission mode, a path is first set up from the source node to the destination node. User packets then traverse the network following the path chosen during the initial connection setup. In this mode, user packets can arrive in sequence in which they were transmitted but the user has to pay for the overhead for the initial connection setup (ex: 3-way handshake in TCP/IP). The other transmission mode is the *connectionless* mode, where each individual packet or *datagram* independently traverses the network from source node to destination node. No initial connection is set up in this case and datagrams are forwarded through the network on an individual basis. Routing of each datagram is based on the destination address and the availability of output ports[5] at each intermediate nodes. Usually, *connectionless* transmission can yield better performance since there is no overhead for connection setup, but downside is that the user has to take care of packet resequencing and retransmission in case datagrams are dropped.

In this first example, we assume datagram transmission mode and we want to evaluate the transmission time of datagrams in each communication node (see Figure 6).

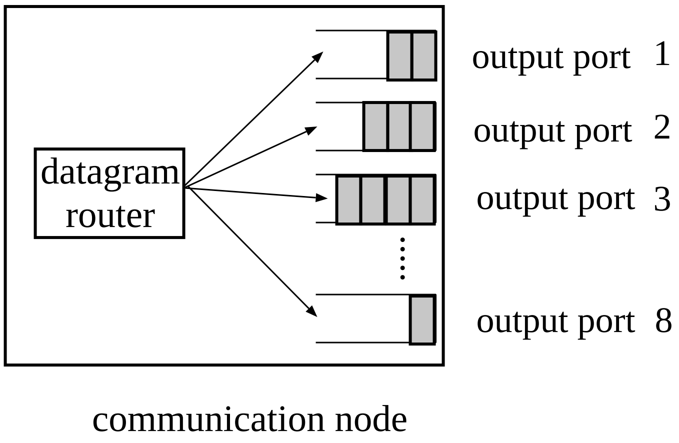

Figure 6: Communication node for sending out datagram.

Assume that the communication node has eight output ports. We model the datagram arrival process of to be a Poisson process with rate λ and each output port as having an exponential service rate of μ. We normalize the service rate to 1.0 and we increase the datagram arrival rate so that the system utilization will vary from 0.1 to 0.9. Table 1 illustrates the upper and lower bound on the mean response time of the datagram(the time to send the datagram across the output link of the communication node) as a function of system utilization. As can be seen by the percentage error[6], the bounding methodology we developed provides a very good accuracy fro the mean response time and at a moderate cost.

To illustrate the tradeoff between computational cost and accuracy of the bounds. We can vary the d and C parameters. By fixing the system utilization at 0.9 and increasing the number of states generated, we see the improvement obtained for the bounds on the mean response time. The results are illustrated in Table 2.

[5]assuming these output ports can reach the destination node

[6]If the spread in the bounds is less than $< 10^{-6}$, we leave the percentage error entry blank.

System Utilization	Response Time Lower Bound	Response Time Upper Bound	Spread of Bounds	Percentage Error
0.1	0.1000252	0.1000252		
0.2	0.2000863	0.2000863		
0.3	0.3008306	0.3008306		
0.4	0.4052623	0.4052623		
0.5	0.5208155	0.5208162	0.0000007	$6.27x10^{-5}$ %
0.6	0.6610700	0.6610820	0.0000120	$9.07x10^{-4}$ %
0.7	0.8521012	0.8522784	0.0001772	0.0103 %
0.8	1.1640786	1.1652135	0.0011349	0.0487 %
0.9	1.9107856	1.9273843	0.0165987	0.4324 %

Table 1: Datagram response time.

d	C	States Generated	Response Time Upper Bound	Response Time Lower Bound	Spread of Bounds	Percentage Errors
4	8	1815	1.8521678	2.1078925	0.2557247	6.4576 %
4	10	2475	1.8973256	2.0013574	0.1040318	2.6684 %
5	12	6831	1.9107856	1.9273843	0.0165987	0.4324 %
6	13	15015	1.9123782	1.9261783	0.0138001	0.3591 %

Table 2: Computational Cost vs. Accuracy.

The second application we consider is a parallel program in a heterogeneous workstation cluster under the *Parallel Virtual Machine* (PVM) environment [26]. We will use the *manager/worker* control decomposition technique [27] where one process on a given workstation will play the role of the manager. The manager farms out the tasks to the workers (other workstations in the network) and the manager keeps track of the progress of the computation as the workers report back with completed tasks. Note that there might be different kinds of workstations (ex: workstations with different processing rates) in this workstation cluster. A simple application of this technique is the construction of a frame buffer in an image processing application. The manager sends sub-portions of the image to the workers, who process the data and return the result. Since some portions of the image may require more computation than others, the manager must based on the processing speed as well as the tasks queue length at each workers to route the next piece of work.

Assume that we have nine nodes in the heterogeneous workstation cluster with eight workstations playing the role of the workers. Assume that we have three classes of workstations. The first class has 4 workstations with SPEC92Int rating of 75, the second class has 2 workstations with SPEC92Int rating of 50 and the last class has 2 workstations with SPEC92Int rating of 25. We normalize the MIPS rating into service rate as $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 3, \mu_5 = \mu_6 = 2$, and $\mu_7 = \mu_8 = 1$. We also assume the task arrival process is Poisson process and we are interested in the mean task completion time. The manager will route each task to a worker based on the minimum expected delay (MED) algorithm. This system is depicted in Figure 7.

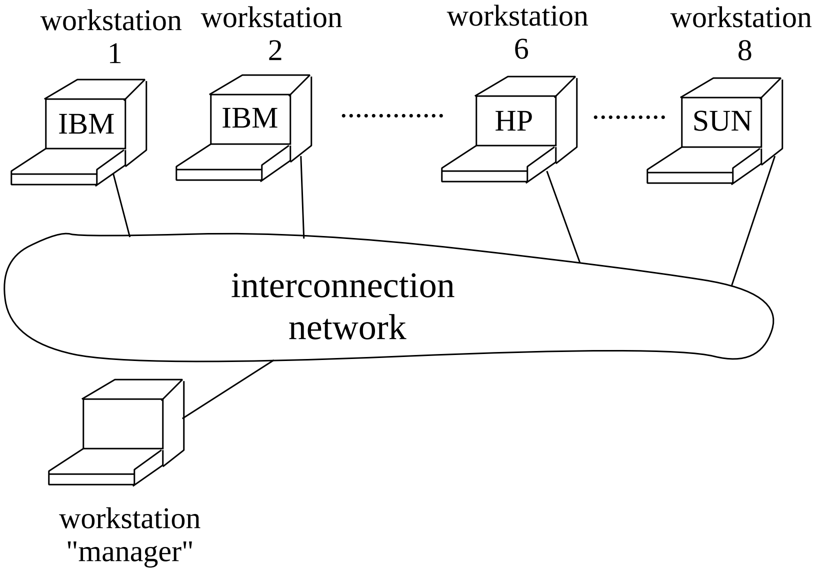

Figure 7: Parallel Program in Heterogeneous Workstation Cluster.

To vary the system utilization from 0.1 to 0.9, we fix the service rates for all servers and vary the input arrival rate. For $\rho = 0.1$ to $\rho = 0.5$, we set $d = 4, \vec{\mathbf{C}} = (6,6,6,6,4,4,2,2)$, for $\rho = 0.6$ to $\rho = 0.8$, we set $d = 5, \vec{\mathbf{C}} = (9,9,9,9,6,6,3,3)$, and for $\rho = 0.8$ to $\rho = 0.9$, we set $d = 6, \vec{\mathbf{C}} = (12,12,12,12,8,8,4,4)$. Table 3 illustrates the upper and lower bound on the mean response time and the tightness of the bounds.

8 Conclusion

The minimum expected delay routing policy is appealing to study not only due to it's simplicity in implementation, but also due to the fact that it is theoretically difficult to analyze because the routing of arrivals is state dependent and no closed form solutions exist in general. Also, due to the fact that each server has an infinite capacity queue, the state space cardinality of the Markov model is infinite, and it becomes impossible

System Utilization	Response Time Lower Bound	Response Time Upper Bound	Spread of Bounds	Percentage Error
0.1	0.3337116	0.3337280	0.0000164	0.00247 %
0.2	0.3380923	0.3381092	0.0000169	0.00250 %
0.3	0.3486201	0.3486987	0.0000786	0.01127 %
0.4	0.3672009	0.3672981	0.0000972	0.01320 %
0.5	0.3978023	0.3979084	0.0001061	0.01333 %
0.6	0.4471098	0.4472873	0.0001775	0.01985 %
0.7	0.5239872	0.5249875	0.0010003	0.09536 %
0.8	0.6506371	0.6609821	0.0103450	0.78872 %
0.9	0.9788093	1.0237158	0.0449065	2.242494%

Table 3: Response time of tasks in Heterogeneous Workstation Cluster.

to generate the entire state space to solve the Markov model numerically. We have presented an approach to bound the mean response time and the mean number of customers in the minimum expected delay routing policy, which is a generalization of the join the shortest queue routing policy. The algorithmic approach provides the flexibility to trade-off computational resources and tighter bounds. There is ongoing work on the subject to determine d and C_i to obtain specified error bounds. We are also investigating the possibility of bounding the mean response time under more relaxed conditions, e.g., by allowing general service distributions.

References

[1] I.J.B.F. Adan, J. Wessels, W.H.M. Zijm. *Analysis of the Symmetric Shortest Queue Problem.* Stochastic Models, Vol.6, 691-713, 1990.

[2] I.J.B.F. Adan, J. Wessels, W.H.M. Zijm. *Analysis of the Asymmetric Shortest Queue Problem.* Queueing Systems, Vol. 8, 1-58, 1991.

[3] Ivo Adan, Geert-Jan van Houtum, Jan van der Wal, *Upper and Lower Bounds for the Waiting Time in the Symmetric Shortest Queue System.* Eindhoven University of Technology, Technical Report COSOR 92 – 09.

[4] Alberto Avritzer. *Dynamic Load Sharing Algorithms in Asymmetric Distributed Systems.* UCLA Computer Science Technical Report, CSD-900023.

[5] Norman R. Baker, *Optimal User Search Sequences and Implications for Information System Operation.* Journal of Applied Probability, Vol.24, 540-546, 1987.

[6] J.P.C. Blanc. *A Note on Waiting Times in Systems with Queues in Parallel.* Journal of Applied Probability, Vol.24, 540-546, 1987.

[7] B.W. Conolly. *The Autostrada Queueing Problem.* Journal of Applied Probability. Vol. 21, 394-403, 1984.

[8] J.W. Cohen, O.J. Boxma. *Boundary Value Problems in Queueing System Analysis.* North Holland, 1983.

[9] P. J. Courtois. *Decomposability — queueing and computer system applications.* Academic Press, New York, 1977.

[10] P. J. Courtois, P. Semal. *Computable Bounds for Conditional Steady-State Probabilities in Large Markov Chains and Queueing Models.* IEEE JSAC, Vol 4, number 6, September, 1986.

[11] A. Ephremides, P. Varaiya, J. Walrand. *A Simple Dynamic Routing Problem.* IEEE Trans. on Auto. Control, Vol. 25, 1980.

[12] L. Flatto, H.P McKean. *Two Queues in Parallel.* Communication on Pure and Applied Mathematics, Vol. 30, 255-263, 1977.

[13] S. Halfin. *The Shortest Queue Problem.* Journal of Applied Probability, Vol. 22, 865-878, 1985.

[14] W.K. Grassmann. *Transient and Steady State Results for Two Parallel Queues.* Omega, 8, 105-112, 1980.

[15] J.F.C Kingman. *Two Similar Queues in Parallel.* Annals of Mathematical Statistics, Vol 32, 1314-1323, 1961.

[16] C. Knessl, B.J. Matkowsky, Z. Schuss, C. Tier. *Two Parallel Queues with Dynamic Routing.* IEEE Trans. on Communications, Vol. 34, 1170-1175, 1986.

[17] J.G. Kemeny and J.L. Snell. *Finite Markov Chains.* Van Nostrand Company, 1960.

[18] L. Kleinrock. *Queueing Systems: Volume I: Theory* Wiley-Interscience Publication . New York. 1975.

[19] J.D.C Little. *A Proof of the Queueing Formula* $L = \lambda W$, Operations Research, Vol 9, 383-387, 1967.

[20] Hwa-Chun and C.S. Raghavendra. *An Analysis of the Join the Shortest Queue Policy* Electrical Engineering Technical Report, University of Southern California, 1991.

[21] J.C.S. Lui, R.R. Muntz. *Algorithmic Approach to Bounding the Response Time of a Minimum Expected Delay Routing System,* Proc. 1992 ACM SIGMETRICS/Performance'92 Conference, 140-152.

[22] Marshall and Olkin. *Inequalities: Theory of Majorization and Applications*, Academic Press, New York, 1979.

[23] R.D. Nelson, T.K. Philips. *An Approximation to the Response Time for Shortest Queue Routing.* ACM SIGMETRICS Vol 17, No 1. 1989, pp 181-189.

[24] R.D. Nelson, T.K. Philips. *An Approximation for the Mean Response Time for Shortest Queue Routing with General Interarrival and Service Times.* IBM T.J. Watson Research Lab, Technical Report RC15429, 1990.

[25] M.F. Neuts. *Matrix-Geometric Solutions in Stochastic Models - An Algorithmic Approach,* The Johns Hopkins University Press, Baltimore, 1981.

[26] A. Giest, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V. Sunderam. *PVM* 3.0 *User's Guide and Reference Manual.* Oak Ridge National Laboratory, 1993.

[27] S. Ragsdale. *Parallel Programming.* McGraw Hill, 1991.

[28] B.M. Rao, M.J.M. Posner. *Algorithmic and Approximate Analysis of the Shorter Queue Model.* Naval Research Logistics, Vol.34, 381-398, 1987.

[29] William J. Stewart, *MARCA: Markov Chain Analyzer, A Software Package for Markov Modeling.* Numerical Solution of Markov Chains, Dekker Press, 1991.

[30] D. Towsley, S. Chen. *Design and Modeling Policies for Two Server Fork/Join Queueing Systems.* University of Mass. COINS Technical Report:91-39.

[31] D. Towsley, P. Sparaggis, C. Cassandras, "Stochastic ordering properties and optimal routing control for a class of finite capacity queueing systems", *IEEE Transactions on Automatic Control*, **37**, 9 1446–1451, September 1992.

[32] Y.T. Wang and Robert J. T. Morris. *Load Sharing in Distributed Systems.* IEEE Transactions on Computers, Vol. C-34, No 3, 204-217. March 1985.

[33] W. Winston. *Optimality of the Shortest Line Discipline,* Journal of Applied Probability. Vol 15, 181-189, 1977.

[34] Y. Zhao and W.K. Grassmann. *The Shortest Queue Model with Jockeying.* Naval Research Logistics. Vol. 37. 773-787, 1990.

[35] Y. Zhao and W.K. Grassmann. *A Numerically Stable Algorithm for Two Server Queue Models.* Queueing Systems, Vol 8, 59-80, 1991.