Algorithmic Approach to Bounding the Mean Response Time of a Minimum Expected Delay Routing System*

John C.S. Lui cslui@cs.ucla.edu
Richard R. Muntz muntz@cs.ucla.edu

UCLA Computer Science Department,Los Angeles, CA 90024-1596, USA

Abstract

In this paper we present an algorithmic approach to bounding the mean response time of a multi-server system in which the minimum expected delay routing policy is used, i.e., an arriving job will join the queue which has the minimal expected value of unfinished work. We assume the queueing system to have K servers, each with an infinite capacity queue. The arrival process is Poisson with parameter λ, and the service time of server i is exponentially distributed with mean $1/\mu_i$, $1 \leq i \leq K$. The computation algorithm we present allows one to tradeoff accuracy and computational cost. Upper and lower bounds on the expected response time and expected number of customers are computed; the spread between the bounds can be reduced with additional space and time complexity. Examples are presented which illustrate the excellent relative accuracy attainable with relatively little computation.

1 Introduction

In this paper, we are concerned with bounding the mean response time (and thereby the mean number of customers in the system) of the *minimum expected delay* routing policy (a natural generalization of the shortest queue routing policy). The system under study has K servers, where $K \geq 2$. Each server has an infinite capacity queue and service rates are exponentially distributed with rate μ_i, $i = 1, 2, \ldots, K$. Without loss of generality, we assume $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. The job arrival process is Poisson with rate λ. Upon arrival, the job joins the queue with minimal expected unfinished work (the formal definition of the routing discipline is given later). In case of a tie, the job joins the server with the lowest index. If all the service rates are the same, then the scheduling policy reduces to the classic shortest queue routing policy.

Joining the shortest queue is a natural way to balance the load in a multi-server system and thereby achieve better system performance, i.e., mean response time. One of the major difficulties in analyzing this kind of a routing discipline is the lack of a closed form solution since the queues in the system are not independent because the arrival to each server depends on the state of the entire system. The state space of the system is multidimensional in nature and infinite in each of the K dimensions. Most of the published results are limited to the case where $K = 2$ with exponential interarrival and service times.

We start with a brief review of the published literature on the shortest queue routing problem. Winston [23] showed that shortest queue routing is optimal in the sense that it maximizes the discounted number of customers to complete service in any specified interval of time. It is important to note that shortest queue policy in a homogeneous system is both *socially* and *individually* optimal [22]. Kingman [11], and later Flatto and McKean [8] studied this problem with $K = 2$ via transform methods. They obtained an expression for the mean number of jobs in the system expressed as an infinite sum which can be simplified under a heavy traffic assumption. Cohen and Boxma [4] treated a similar problem as a Reimann-Hilbert boundary problem and obtained a functional representation for the mean number of customers in the system. Conolly [3] studied the same model as in [8, 11] and proposed an approximation algorithm for evaluating equilibrium state probabilities via state truncation. Rao and Posner [20] proposed an approximation algorithm to analyze a system with $K = 2$ and in which each server has a different service rate (heterogeneous servers). An arriving job joined the server with smaller number of jobs

*This research was supported in part by a grant from NSF INT-8514377, CNPq-Brazil.

1992 ACM SIGMETRICS & PERFORMANCE '92-6/92/R.I., USA
© 1992 ACM 0-89791-508-9/92/0005/0140...$1.50

(rather than joining the server with minimum expected delay). The analysis approach involved treating one of the queues as having a bounded capacity so that the transition rate matrix for the modified system could be expressed in a matrix-geometric form [19]. Grassman [10] studied the same problem with $K = 2$ and solved for transient and steady state behavior. Halfin [9] studied the two servers problem and used a linear programming technique to compute bounds on the mean number of customers in the system. Blanc [2] studied the join the shortest queue problem with an arbitrary number of heterogeneous servers. He proposed an approximation method which was based on power series expansions and recursion which required a substantial computational effort. Nelson and Philips [17, 18] proposed an approximation for the mean response time with K homogeneous servers. More importantly, the approximation allowed general interarrival and service time distributions. Avritzer [1] studied a dynamic load balancing algorithm which used 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 servers and simultaneously provided error bounds.

The major contribution of this paper is a computation algorithm that (1) allows more than $K \geq 2$ servers, (2) allows heterogeneous servers, (3) includes scheduling based on queue lengths and service rates (thus, a generalization of joining the shortest queue) and (4) provides error bounds. The bounding methodology also allows one to tradeoff accuracy and computational cost, as will be demonstrated.

In Section 2, we define formally the queueing system we are analyzing. In Sections 3 and 4, we present Markov models which provide upper and lower bounds on the mean response time, and we formally prove that these modified models do provide bounds. In this paper, we show how we can further reduce the state space by lumping similar states. In Section 5, we present two numerical examples and show that the bounds are indeed tight. Conclusions are presented in Section 6.

2 Minimum Expected Delay Routing Model

We consider a system with $K \geq 2$ servers, each with its own infinite capacity queue and exponential service rate μ_i, $i = 1, 2, \ldots, K$, where $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. The job arrival process is Poisson with rate λ. Let $n_i(t)$ be the number of customers at the i^{th} server at time t. Let $U_i(t) = (1 + n_i(t))/\mu_i$, which is the expected unfinished work at the i^{th} server if the new customer joins queue i. Define $U^*(t) = \min\{U_i(t), i = 1, \ldots, K\}$. Upon arrival of a job at time t, the job will join a server j where $U_j(t) = U^*(t)$. If a tie occurs, the job will join the lowest index server in the set $\{j \mid U_j(t) = U^*(t)\}$. A special case of this routing discipline is when all service rates are equal, and in this case it reduces to the classic shortest queue routing problem. We can construct a Markov model, M, for this queueing system with state space:

$$\{s = [n_1, n_2, \ldots, n_K] \mid n_i \geq 0, \quad i = 1, \ldots, K\}$$

Assume the system is stable; that is $\rho = \lambda / \sum_{i=1}^{K} \mu_i < 1$. The steady state probability vector for this continuous-time Markov model is the solution to:

$$\vec{\pi} Q = \vec{0} \quad \text{and} \quad \vec{\pi} \underline{e} = 1 \tag{1}$$

where $\vec{\pi}$ is the steady state probability vector, Q is the transition rate matrix, and $\underline{e}$ denotes an appropriately dimensioned column vector of 1's.

We can transform this continuous-time Markov model into a discrete-time Markov model via uniformization [21] (the rationale behind this transformation is to facilitate the comparison of the original model and the modified models we introduce later). To express the one-step transition probabilities for this discrete-time Markov chain, we need the following notation:

$\mathcal{S}$	=	total state space of the original model, M.
h	=	$[\lambda + \sum_{i=1}^{K} \mu_i]^{-1}$
$U^*(s)$	=	$\min\{U_i \mid U_i = (1 + n_i)/\mu_i, i = 1 \ldots, K, s = [n_1, n_2, \ldots, n_K]\}$.
$n_a(s)$	=	set of servers in state s for which $U_i = U^*(s)$.
$n^*(s)$	=	the lowest index for servers in set $n_a(s)$.
$1\{c\}$	=	indicator function for condition c.

The one-step transition probabilities for a given state $s = [n_1, \ldots, n_i, \ldots, n_k]$ are:

$$s \rightarrow s + \epsilon_i \qquad 1\{i = n^*(s)\} h\lambda \tag{2}$$

$$s \rightarrow s - \epsilon_i \qquad 1\{n_i > 0\} h\mu_i \tag{3}$$

$$s \rightarrow s \qquad 1 - h[\lambda + \sum_{i=1}^{K} 1\{n_i > 0\}\mu_i] \tag{4}$$

where $s + \epsilon_i$ is the state s with one additional customer in the i^{th} queue. Also note that:

$$1 - h[\lambda + \sum_{i=1}^{K} 1\{n_i > 0\}\mu_i] = \sum_{i=1}^{K} 1\{n_i = 0\}\mu_i h \tag{5}$$

Let P be the transition probability matrix for the transformed discrete-time Markov chain; we can obtain the

steady state probability, at least theoretically, by solving the following system of linear equations:

$$\vec{\pi}P = \vec{\pi} \quad and \quad \vec{\pi}\underline{e} = 1 \tag{6}$$

Of course, based on the state description, $\vec{\pi}$ is a K-dimensional vector which is infinite in each dimension. The exact solution to this problem has been found thus far to be intractable.

In general, the original problem does not possess a closed form solution, and it is impossible to solve the problem numerically due to its state space cardinality. Since the Markov chain lacks special structure, techniques such as the matrix-geometric methods do not apply in general. One natural way to approach this problem is to *construct* another model that closely bounds the performance of the original problem and at the same time, has either a known closed form solution or at least is efficiently evaluable by numerical methods.

An important observation is that the motivation for using minimum expected delay policy is to balance the workload among all servers in the systems. Consider a system of two servers with equal service rates in which the current state is [5, 1]. The purpose of using the routing policy is to balance the system as much as possible; therefore it is reasonable to assume that a highly unbalanced state(e.g., [5, 1]) has a much smaller probability mass than a balanced state (i.e., [3, 3]). This crucial insight provides the rational 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. The number of states in the most popular subset is a function of the accuracy demanded and 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 behavior can be shown to bound the behavior of the original model.

In the following two sections, we present two Markov models which can provide an upper bound and a lower bound mean response time. We also present numerical procedures for efficiently solving these two modified models.

3 Upper Bound Model

In this section, we construct a modified Markov model, M_u, which provides an upper bound for the mean response time and the mean number of customers of the original model, M. For the upper bound model M_u, we assume that we have the same system configuration, namely that the job arrival process is Poisson with rate λ and the system has K servers with service rates $\mu_i, i = 1, 2, \ldots, K$, where $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$.

The upper bound model can be described as follows. There are two additional model parameters for M_u. First, we have a threshold parameter d which indicates the degree of *imbalance* permitted between different servers' queues (a formal definition for d will be given later.) A job may depart from the system only if its departure does not violate the maximum degree of imbalance permitted. If the job departure violates the threshold setting, the job restarts itself within the same server. Intuitively, this mechanism forces a job to stay in the system at least as long as in the original model and thereby increases the mean number of jobs in the system. The rationale behind the threshold parameter is to generate a model that has a state space which is a small subset of the state space of the original model. The second parameter is the *artificial capacity*, C_i, where $i = 1, 2, ..., K$ (again, C_i will be precisely defined later) for each server. Whenever a job arrives to the system and finds that each server has an integer multiple of C_i jobs, each server will put all jobs in its queue (except for the arriving job) into a suspended state, and a new busy cycle is started. This busy cycle will end when all servers complete all jobs except for the suspended jobs. The suspended jobs are then released and can be served. Note that the definition here is recursive. During the busy period following suspension of a set of jobs, the capacities C_i can again be exceeded, causing another set of jobs to be suspended. When a busy period ends, only the set of jobs suspended at the initiation of that busy period are released for service. The purpose of the C_i, $1 \leq i \leq K$, is to create a matrix with repetitive structure; based on that structure, we will be able to derive an efficient numerical solution algorithm. The computation algorithm is based on a partitioning of the state space of M_u into $\{\mathcal{S}_0 \cup \mathcal{S}_1 \cdots\}$ such that all states in $\mathcal{S}_i$, $i \geq 0$ satisfy the condition $iC_j \leq n_j \leq (i+1)C_j$ for $j = 1, \ldots, K$. Due to the routing of arrivals and the constraint on departures, we can show that there is only one transition from $\mathcal{S}_i$ to $\mathcal{S}_{i+1}$ 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 easily form a reduced model in which each $\mathcal{S}_i$ is exactly aggregated. This modification to the model should also increase the mean number of jobs in the system compared to the original model since service of a suspended job can only be resumed when all the active jobs 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 $\mathcal{S}_0$ consists of all states for which each queue has be-

tween 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 through state $[10,10,10,10]$. This is due to the 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 $[10,10,10,10]$. This is due to the rule introduced in M_u that suspended customers are only served when the busy period (corresponding to states in $\mathcal{S}_1$) has completed. For a heterogeneous server system, the values of C_i' have to be chosen to be proportional to the relative service rate in the system to maintain the same structure for the transition rate matrix.

An important point is that the parameters d and C_i' can be chosen to control the extent to which M_u behaves like the original model M, i.e., the larger d and C_i' are, the larger the portion of the state space that has behavior identical to the original model. Now, let us define the following variables for M_u.

$\mathcal{S}_u$ = total state space of M_u, where $\mathcal{S}_u \subset \mathcal{S}$.
h = $[\lambda + \sum_{i=1}^{K} \mu_i]^{-1}$
C_i' = $\lfloor \frac{\mu_i}{\mu_1} C' \rfloor$, $i = 1, 2, \ldots, K$, where C' is some positive integer such that $\lfloor \frac{\mu_i}{\mu_1} C' \rfloor \geq 1$.
d = threshold setting where $(C_1' - C' + 1) \leq d \leq C_1'$.
$n_{max}(s)$ = $\max\{n_i | s = [n_1, \ldots, n_i, \ldots, n_K]\}$.
$l(s)$ = smallest integer l such that $lC_i' - n_i \geq 0$ for all servers i, $i = 0, 1, \ldots, K$ in state s. Note that $l(s)$ is the depth of recursion of job suspensions in state s.

We transform this continuous-time Markov model into a discrete-time Markov chain with the same uniformization parameter h which we used in the original model M. The one-step transition probabilities of the discrete-time Markov chain for a given state $s = [n_1, .., n_i, .., n_k]$ are:

$$s \rightarrow s + e_i \quad 1\{i = n^*(s)\} h\lambda \tag{7}$$

$$s \rightarrow s - e_i \quad 1\{n_i > 0\} 1\{n_{max}(s) - n_i < d\} 1\{n_i - (l(s) - 1)C_i' > 0\} h\mu_i \tag{8}$$

$$s \rightarrow s \quad 1 - h[\lambda + \sum_{i=1}^{K} 1\{n_i > 0\} 1\{n_{max}(s) - n_i < d\} 1\{n_i - (l(s) - 1)C_i' > 0\} \mu_i] \tag{9}$$

Note that for transition $s \rightarrow s - e_i$, the second indicator function reflects that a job cannot depart if it violates the maximum degree of imbalance permitted. The third indicator function reflects that a job cannot depart if it is in a suspended state. We are now in a position to formally compare the original (M) and the modified Markov chain (M_u) and prove that the mean response time of M_u is an upper bound on the mean response time of M.

3.1 Proof of upper bound mean response time

Our proof that the mean response time of the modified model is an upper bound on the mean response time of the original model follows the approach in [7]. Let T and T_u be the one-step expectation operators of the original model M and the upper bound Markov model M_u. That is for any non-decreasing function f, we define T in terms of the one-step transition probabilities to be:

$$Tf(s) = \sum_{s' \in \mathcal{S}} p[s \rightarrow s'] f(s')$$

$$T_u f(s) = \sum_{s' \in \mathcal{S}_u} p_u[s \rightarrow s'] f(s')$$

where $p[s \rightarrow s']$ ($p_u[s \rightarrow s']$) is the transition probability from state s to state s' in M (M_u), $\forall s, s' \in \mathcal{S}_u$.

Let R and R_u be the mean response time of M and M_u respectively. And let N and N_u be the mean number of customers in the system for M and M_u. To show $R \leq R_u$, all we need to show is $N \leq N_u$ since the average arrival rate for both models is λ. Define the reward for state s as $r(s) = \sum_{i=1}^{K} n_i$ for both models. The mean number of customers in the system can be expressed in term of the expected reward function:

$$N = \sum_{s \in \mathcal{S}} r(s)\pi(s) \tag{10}$$

Let $V^t(s)$ be the total expected reward over t periods with the one-step reward function r when starting in state s. We have:

$$V^t[s] = \sum_{k=0}^{t-1} T^k[r(s)]$$

with T^0 being the identity function. By the Markovian property, we have:

$$V^t[s] = r(s) + TV^{t-1}[s]$$

Since both Markov models are irreducible (easily seen from their definitions), steady state performance measures are independent of the initial state s', and we have:

$$N = \sum_{s \in \mathcal{S}} r(s)\pi(s) = \lim_{t \rightarrow \infty} \frac{1}{t} V^t[s'] \tag{11}$$

and (12)

$$N_u = \sum_{s \in \mathcal{S}_u} r(s)\pi_u(s) = \lim_{t \rightarrow \infty} \frac{1}{t} V_u^t[s'] \tag{13}$$

To show $N_u \geq N$, we have to show $V_u^t(s) \geq V^t(s)$ for all t and all $s \in \mathcal{S}_u$. As illustrated in [7], to show that $N_u \geq N$, it is sufficient to show:

$$(T_u - T)V^t[s] \quad \geq \quad 0 \tag{14}$$

for all $t \geq 0$ and for all $s \in \mathcal{S}_u$.

Based on the definition of the one-step expectation operator on the original model M and the upper bound model M_u, we have the following relationship for any state $s \in \mathcal{S}_u$:

$$\begin{aligned}(T_u - T)f(s) \; = \; \sum_{i=1}^{K} 1\{((n_i > 0) \wedge (n_{max}(s) - n_i = d)) \; | \\ ((n_i > 0) \wedge (n_i - (l(s)-1)C_i' = 0))\} \\ \mu_i h[f(s) - f(s - e_i)]\end{aligned} \tag{15}$$

where the symbol "$\wedge$" is the logical AND and "$|$" is the logical OR operator. Substituting $V^t(s)$ for $f(s)$, it follows easily that Equation (14) is satisfied if the following conditions are satisfied:

$$\begin{aligned}&V^t[s] - V^t[s - e_i] \geq 0 \\ &\quad for \; i = 1, \ldots, K; t \geq 0; n_i > 0; \; s \in \mathcal{S}_u\end{aligned} \tag{16}$$

Theorem 1

$$\begin{aligned}&V^t[s] - V^t[s - e_i] \geq 0 \\ &\quad for \; i = 1, \ldots, K; t \geq 0; n_i > 0; \; s \in \mathcal{S}_u\end{aligned}$$

Proof: The proof is given in the Appendix. ∎

3.2 Computational algorithm for solving the upper bound model

In this section, we describe an algorithm for computing the mean response time of the upper bound model. We define a partition of 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:

$\mathcal{S}_0$ = set of states with $n_j \leq C_j'$, $j = 1, 2, \ldots, K$.
$\mathcal{S}_i$ = set of states with $iC_j' \leq n_j \leq (i+1)C_j'$, $j = 1, 2, \ldots, K$ and for $i \geq 1$.
$P_{\mathcal{S}_i, \mathcal{S}_j}$ = transition probability matrix from states in $\mathcal{S}_i$ to states in $\mathcal{S}_j$.

The transition rate matrix P_u has the form depicted in Figure 1.

This is a block tridiagonal transition probability matrix and therefore represents a quasi-birth-death process. By aggregating each partition $\mathcal{S}_i$, we can form a birth-death process. Next, we show how to obtain the

$$P_u = \begin{bmatrix} P_{\mathcal{S}_0,\mathcal{S}_0} & P_{\mathcal{S}_0,\mathcal{S}_1} & 0 & 0 & 0 & \cdots \\ P_{\mathcal{S}_1,\mathcal{S}_0} & P_{\mathcal{S}_1,\mathcal{S}_1} & P_{\mathcal{S}_1,\mathcal{S}_2} & 0 & 0 & \cdots \\ 0 & P_{\mathcal{S}_2,\mathcal{S}_1} & P_{\mathcal{S}_2,\mathcal{S}_2} & P_{\mathcal{S}_2,\mathcal{S}_3} & 0 & \cdots \\ 0 & 0 & P_{\mathcal{S}_3,\mathcal{S}_2} & P_{\mathcal{S}_3,\mathcal{S}_3} & P_{\mathcal{S}_3,\mathcal{S}_4} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Figure 1: Transition probability matrix for upper bound model.

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 aggregate transition probabilities exactly.

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

$$\begin{aligned}s_i(C_0') = [n_1, n_2, \ldots, n_K] \in \mathcal{S}_i \;\; where \\ n_j = (i+1)C_j' \;\; \forall \;\; j = 1, 2, \ldots, K\end{aligned}$$

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 probabilities to states in $\mathcal{S}_{i-1}$ where $i \geq 1$. Each corresponds to which server is the last to complete its "active" (non-suspended) customers. Let us call these states $s_i(l)$, $1 \leq l \leq K$, $i \geq 1$. These states are:

$$\begin{aligned}s_i(l) \;\; &= \;\; [n_1, n_2, \ldots, n_K] \in \mathcal{S}_i \;\; where \\ n_l \;\; &= \;\; iC_l' + 1 \;\; and \\ n_j \;\; &= \;\; iC_j' \;\; for \;\; l \neq j \;\; and \;\; l, j = 1, 2, \ldots, K\end{aligned}$$

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

$$\begin{aligned}s_i(C_0') \quad &- \quad s_{i+1}(l) \qquad 1\{l = n^*(s_i(C_0'))\}\lambda h \\ s_{i+1}(l) \quad &- \quad s_i(C_0') \qquad \mu_l h \;\; for \;\; l = 1, 2, \ldots, K\end{aligned}$$

Another important observation is that the submatrices $P_{\mathcal{S}_i,\mathcal{S}_i}$ for $i \geq 1$ are all identical. We now consider how to compute the conditional state probabilities $P\{s \in \mathcal{S}_i | \mathcal{S}_i\}$ exactly. We first need the following result from [6]:

Theorem 2 *Given an irreducible Markov process with state space $S = \{A \cup B\}$ and transition probability matrix:*

$$\begin{bmatrix} P_{A,A} & P_{A,B} \\ P_{B,A} & P_{B,B} \end{bmatrix}$$

where $P_{i,j}$ is the transition probability sub-matrix from partition i to j. If $P_{B,A}$ 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:

$$\begin{aligned} \vec{\pi}_{|A}\left[P_{A,A} + P_{A,B}\,\underline{e}\,\underline{e}_i^T\right] &= \vec{\pi}_{|A} \\ \vec{\pi}_{|A}\,\underline{e} &= 1 \end{aligned}$$

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

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

Lemma 1 *Let $\tilde{P}_{\mathcal{S}_i,\mathcal{S}_i}$ be the transition probability matrix which is similar to $P_{\mathcal{S}_i,\mathcal{S}_i}$ except for the following modification:*

$$\tilde{p}_{s_i(C_0),s_i(C_0)} = p_{s_i(C_0),s_i(C_0)} + \lambda h \qquad (17)$$

$$\tilde{p}_{s_i(l),s_i(l)} = p_{s_i(l),s_i(l)} + \mu_l h \quad \text{where } l = n^*(s_{i-1}(C_0)) \qquad (18)$$

$$\tilde{p}_{s_i(j),s_i(l)} = \mu_l h \quad j = 1,2,\ldots,K; j \neq l \qquad (19)$$

The solution to the following system of linear equations:

$$\vec{\tilde{\pi}}\tilde{P}_{\mathcal{S}_i,\mathcal{S}_i} = \vec{\tilde{\pi}} \quad \text{and} \quad \vec{\tilde{\pi}}\underline{e} = 1$$

provides the conditional steady state probability of state s given that the Markov chain is in some state 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 Theorem 2, the modification of Equation (17) 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 $\mathcal{S}_i^1 = \cup_{j=0}^{j=i-1}\mathcal{S}_j$. Note that there is only one return state in $\mathcal{S}_i$, which is $s_i(n^*(s_{i-1}(C_0)))$. Again, based on Theorem 2, the modification of equation (18) and equation (19) provides the conditional state probability vector, given the system is in state $\mathcal{S}_i$. ∎

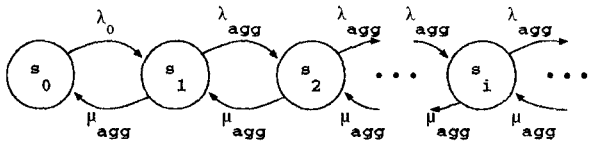

Figure 2: Aggregate Chain for upper bound model

Since we can compute the conditional state probabilities for each partition $\mathcal{S}_i$ exactly, we can exactly aggregate states in each $\mathcal{S}_i$ into a single state $s_i, i \geq 0$. The aggregate chain is depicted in Figure 2.

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

Solving this chain, we have:

$$\pi^*(s_0) = \left[1 + \frac{\lambda_0}{\mu_{agg} - \lambda_{agg}}\right]^{-1} \qquad (20)$$

$$\pi^*(s_i) = \left[1 + \frac{\lambda_0}{\mu_{agg} - \lambda_{agg}}\right]^{-1}\left(\frac{\lambda_0}{\mu_{agg}}\right)\left(\frac{\lambda_{agg}}{\mu_{agg}}\right)^{i-1} \quad \text{for } i = 1,2,\ldots \qquad (21)$$

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

$$\begin{aligned} C_0' &= \sum_{i=1}^{K} C_i \\ \tilde{r}(s) &= r(s) - iC_0' \qquad s \in \mathcal{S}_i \\ \tilde{N}(s_i) &= \sum_{s\in\mathcal{S}_i}\tilde{r}(s)\tilde{\pi}(s) \end{aligned}$$

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

$$\vec{\tilde{\pi}}\tilde{P}_{\mathcal{S}_i,\mathcal{S}_i} = \vec{\tilde{\pi}} \quad \text{and} \quad \vec{\tilde{\pi}}\underline{e} = 1$$

Then we have:

$$N_u = \tilde{N}(s_0)\pi^*(s_0) + \sum_{i=1}^{\infty}\left[\tilde{N}(s_i) + iC_0'\right]\pi^*(s_i) \qquad (22)$$

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

$$N_u = \tilde{N}(s_0)\pi^*(s_0) + \tilde{N}(s_i)(1 - \pi^*(s_0)) + C_0'\lambda_0\frac{\mu_{agg}}{(\mu_{agg} - \lambda_{agg})^2}\pi^*(s_0) \qquad (23)$$

From Little's Result [13], the upper bound mean system response time is:

$$R_u = N_u/\lambda \qquad (21)$$

4 Lower Bound Model

In this section, we construct a modified Markov model, M_l, which provides a lower bound for the mean response time of the original model, M. We first give an informal description and motivation for the lower bound model. As in the upper bound model, two addtional parameters are used to specify model M_l, namely, d and C_i, $i = 1, \ldots, K$. A job may depart normally from the system only if the departure does not violate the maximum degree of imbalance permitted. If a job departure violates this threshold setting, the system goes into a *full service mode*. In this mode, the system behaves like an $M/M/K$ system with a special service discipline; specifically if there are j customers (where $j \leq K$) in the system, these j customers execute on the j fastest servers. If there are more than K customers in the system, then the system behaves like a regular $M/M/K$ system. The system operates in this mode until the next idle time, and then it starts behaving like the original system again. Also, when a job arrives and finds that the system has C_f customers, where $C_f = \sum_{i=1}^{K} C_i$, the system again begins to operate in a full service mode until the system goes idle and then it reverts back to its original behavior. 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 this 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.

4.1 Proof of lower bound mean response time

In order to facilitate the comparison between M and M_l, we organize the state space for model M using the following notation:

$\mathcal{N}_i$ = set of states with exactly i jobs in the system, where $i = 0, 1, \ldots$
$\mathcal{G}$ = $\{\mathcal{N}_1 \cup \mathcal{N}_2 \cup \cdots \mathcal{N}_{C_f}\}$
$Q_{i,j}$ = submatrix containing transition rates from states in $\mathcal{N}_i$ to states in $\mathcal{N}_j$
$Q_{i,\mathcal{G}}$ = submatrix containing transition rates from states in $\mathcal{N}_i$ to states in $\mathcal{G}$

Figure 3 illustrates the form of the transition rate matrix for model M when states are ordered according to the number of customers in the system.

Using the state replication technique from [16], it is easy to show that we can transform the model M into another model, M_1, by duplicating the states in $\mathcal{G}$ without perturbing the expected number of customer in the system. Let us call the duplicated set of states $\mathcal{G}'$. The transition rate matrix M_1, which results from the duplication of states in $\mathcal{G}$, is illustrated in Figure 4. More formally, if[1] $[\underline{\pi}_0, \underline{\pi}_{\mathcal{G}}, \underline{\pi}_{>\mathcal{G}}]$ is the steady state solution for model M, the steady state probability vector for model M_1 is:

$$\left[\underline{\pi}_0, \underline{\pi}'_{\mathcal{G}}, \underline{\pi}'_{\mathcal{G}'}, \underline{\pi}_{>\mathcal{G}}\right] \quad where \ \underline{\pi}_{\mathcal{G}} = \underline{\pi}'_{\mathcal{G}} + \underline{\pi}'_{\mathcal{G}'}$$

Note that there is a one to one mapping between the

$$\begin{bmatrix} Q_{0,0} & Q_{0,\mathcal{G}} & 0 & 0 & \cdots \\ Q_{\mathcal{G},0} & Q_{\mathcal{G},\mathcal{G}} & Q_{\mathcal{G},C_f+1} & 0 & \cdots \\ 0 & Q_{C_f+1,\mathcal{G}} & Q_{C_f+1,C_f+1} & Q_{C_f+1,C_f+2} & \cdots \\ 0 & 0 & Q_{C_f+2,C_f+1} & Q_{C_f+2,C_f+2} & \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Figure 3: Transition rate matrix for M

$$\begin{bmatrix} Q_{0,0} & Q_{0,\mathcal{G}} & 0 & 0 & 0 & \cdots \\ Q_{\mathcal{G},0} & Q_{\mathcal{G},\mathcal{G}} & 0 & Q_{\mathcal{G},C_f+1} & 0 & \cdots \\ Q_{\mathcal{G}',0} & 0 & Q_{\mathcal{G}',\mathcal{G}'} & Q_{\mathcal{G}',C_f+1} & 0 & \cdots \\ 0 & 0 & Q_{C_f+1,\mathcal{G}'} & Q_{C_f+1,C_f+1} & Q_{C_f+1,C_f+2} & \cdots \\ 0 & 0 & 0 & Q_{C_f+2,C_f+1} & Q_{C_f+2,C_f+2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix}$$

Figure 4: Transition rate matrix for M_1.

states in $\mathcal{G}$ and the states in $\mathcal{G}'$, and $Q_{\mathcal{G}',\mathcal{G}'} = Q_{\mathcal{G},\mathcal{G}}$, $Q_{\mathcal{G},0} = Q_{\mathcal{G}',0}$ and $Q_{C_f+1,\mathcal{G}} = Q_{C_f+1,\mathcal{G}'}$. Starting from an empty system, only the states in $\mathcal{G}$ are visited until the number in the system exceeds C_f. When the number in the system falls to C_f again, the states in $\mathcal{G}'$ are visited rather than the states in $\mathcal{G}$ until the system goes idle. At this point the described behavior repeats. Intuitively, the idea is that if C_f is large enough, the number in the system only rarely exceeds C_f and therefore most of the time, M_l behaves exactly as the original model M.

Although the states in $\mathcal{G}$ are more popular than the other states in the model, there are still a large number of states in $\mathcal{G}$ which have low steady state probability, for example, those states with large imbalance in queue length. With this in mind, let us partition $\mathcal{G}$ into two sets of states, $\mathcal{G}_1$ and $\mathcal{G}_2$ where $\mathcal{G}_1$ contains all those states that satisfy the threshold setting d, and $\mathcal{G}_2 = \mathcal{G} - \mathcal{G}_1$. Based on the results from [14], transitions from $\mathcal{G}_1$ to $\mathcal{G}_2$ can be transformed into transitions

[1] To simplify notation, we use $\underline{\pi}_{>\mathcal{G}}$ to represent steady state probabilities for states other than state 0 and the states in $\mathcal{G}$

from $\mathcal{G}_1$ to the corresponding states in $\mathcal{G}'$ (since there is a one to one mapping between the states in $\mathcal{G}$ and $\mathcal{G}'$) without perturbing the mean number of customers in the systems. Formally, the steady state probability vector for model M_2 is:

$$\left[\underline{\pi}_0, \underline{\pi}''_{\mathcal{G}_1}, \underline{\pi}''_{\mathcal{G}'}\underline{\pi}_{>\mathcal{G}}\right]$$

$$where \quad \underline{\pi}_{\mathcal{G}} = \left[\underline{\pi}''_{\mathcal{G}_1}, \underline{0}\right] + \underline{\pi}''_{\mathcal{G}'} = \underline{\pi}'_{\mathcal{G}} + \underline{\pi}'_{\mathcal{G}'}$$

The transition rate matrix for this new model M_2 is illustrated in Figure 5.

$$\begin{bmatrix} Q_{0,0} & Q_{0,\mathcal{G}_1} & 0 & 0 & 0 & \cdots \\ Q_{\mathcal{G}_1,0} & Q_{\mathcal{G}_1,\mathcal{G}_1} & Q_{\mathcal{G}_1,\mathcal{G}'} & Q_{\mathcal{G}_1,C_f+1} & 0 & \cdots \\ Q_{\mathcal{G}',0} & 0 & Q_{\mathcal{G}',\mathcal{G}'} & Q_{\mathcal{G}',C_f+1} & 0 & \cdots \\ 0 & 0 & Q_{C_f+1,\mathcal{G}'} & Q_{C_f+1,C_f+1} & Q_{C_f+1,C_f+2} & \cdots \\ 0 & 0 & 0 & Q_{C_f+2,C_f+1} & Q_{C_f+2,C_f+2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Figure 5: Transition rate matrix for M_2.

Now, (conceptually) we apply exact aggregation [5] to the states in $\mathcal{G}'$ and to the states in $\mathcal{N}_i$ for $i > C_f$. That is, we aggregate all states with equal number of customer into a single state. Denote the aggregate state corresponding to i customers in the systems as a_i, and let $g_{i,j}$ be the aggregate rate between aggregate state i and j. The transition rate matrix for this model M_3, is illustrated in Figure 6.

$$\begin{bmatrix} Q_{0,0} & Q_{0,\mathcal{G}_1} & 0 & 0 & 0 & 0 & \cdots \\ Q_{\mathcal{G}_1,0} & Q_{\mathcal{G}_1,\mathcal{G}_1} & Q_{\mathcal{G}_1,a_1} & Q_{\mathcal{G}_1,a_2} & Q_{\mathcal{G}_1,a_3} & Q_{\mathcal{G}_1,a_4} & \cdots \\ g_{a_1,0} & 0 & g_{a_1,a_1} & g_{a_1,a_2} & 0 & 0 & \cdots \\ 0 & 0 & g_{a_2,a_1} & g_{a_2,a_2} & g_{a_2,a_3} & 0 & \cdots \\ 0 & 0 & 0 & g_{a_3,a_2} & g_{a_3,a_3} & g_{a_3,a_4} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Figure 6: Transition rate matrix for M_3.

We are now in a position to compare model M_3 (which has the same expected number of customers as the original model, M) to the lower bound model M_l since they have similar transition structures. Note that in the lower bound model M_l, the system operates in the *full service mode* when it is in states a_i, $i \geq 1$. That is[2]:

$$g^*_{a_i,a_{i-1}} = \begin{cases} \sum_{j=1}^{i} \mu_j & 1 \leq i \leq K \\ \sum_{j=1}^{K} \mu_j & i > K \end{cases} \qquad (25)$$

[2]To simplify notation, we use notation a_0 (a state with no customers in the system) and 0 interchangeably.

It is clear that these aggregate rates $g^*_{a_i,a_{i-1}}$ in M_l are upper bounds on aggregate transition rates $g_{a_i,a_{i-1}}$ in M_3.

Again, to facilitate a formal proof that M_l provides a lower bound, we transform the two continuous time Markov models, M_3 and M_l, into discrete-time Markov chains with the uniformization parameter h. We can then apply the same approach as in Section 3.1 to show that the expected number of customers in the system for model M_l is less than the expected number of customers in model M_3. Based on the difference of the one-step expectation operator T_l (for model M_l) and T, we need the following conditions to hold:

$$V^t(a_{i-1}) - V^t(a_i) \leq 0 \qquad i \geq 1 \text{ and } t \geq 0 \quad (26)$$

Theorem 3

$$V^t(a_{i-1}) - V^t(a_i) \leq 0 \qquad i \geq 1 \text{ and } t \geq 0$$

Proof: The proof is given in [15]. ∎

4.2 Computational algorithm for solving the lower bound model

In this section, we describe an algorithm for computing the mean response time of the lower bound model M_l. Let $\mathcal{S}_0 = \{n_0 \cup \mathcal{G}_1\}$. Again, the transition rate matrix is depicted in Figure 7:

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

Figure 7: Transition rate matrix for lower bound model.

Observe that if we know the conditional state probabilities for states in $\mathcal{S}_0$, we can aggregate $\mathcal{S}_0$ as a single state, s_0, and we will have an efficient algorithm for computing the mean number of customers in the system. Note that there is only a single return state to $\mathcal{S}_0$ from the states outside $\mathcal{S}_0$, and based on Theorem 2, the state probabilities conditioned on the system being in $\mathcal{S}_0$ can be obtained by solving the following system of linear equations:

$$\vec{\pi}(\mathcal{S}_0)\left[Q_{\mathcal{S}_0,\mathcal{S}_0} + \left(\sum_{i=1}^{C_f+1} Q_{\mathcal{S}_0,\mathcal{N}_i}\underline{c}\right)\underline{e}_0^T\right] = 0$$

$$\vec{\pi}(\mathcal{S}_0)\,\underline{e} = 1$$

where $\vec{\pi}(\mathcal{S}_0)$ is the steady state probability vector, given the system is in $\mathcal{S}_0$. We can now apply exact aggregation; the aggregated process is depicted in Figure 8.

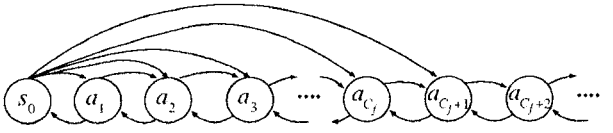

Figure 8: Aggregate Chain for lower bound model

The transition rates for the aggregated chain are:

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

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

Solving the chain, we have:

$$\pi^*(s_0) = \left[1 + \sum_{i=1}^{C_f+1}\sum_{j=1}^{i}[\lambda^{i-j}(\sum_{k=j}^{C_f+1} g^*_{s_0,a_j})(\prod_{k=j}^{i} g^*_{a_k,a_{k-1}})^{-1}] + \frac{\lambda}{\mu^*-\lambda}\sum_{j=1}^{C_f+1}[\lambda^{C_f+1-j}(\sum_{k=j}^{C_f+1} g^*_{s_0,a_j})(\prod_{k=j}^{C_f+1} g^*_{a_k,a_{k-1}})^{-1}]\right]^{-1} \quad (27)$$

$$\pi^*(a_i) = \pi(s_0)\sum_{j=1}^{i}[\lambda^{i-j}(\sum_{k=j}^{C_f+1} g^*_{s_0,a_j})(\prod_{k=j}^{i} g^*_{a_k,a_{k-1}})^{-1}] \quad i = 1,\ldots,C_f+1 \quad (28)$$

$$\pi^*(a_i) = \pi(s_0)(\frac{\lambda}{\mu^*})^{i-C_f-1}\sum_{j=1}^{C_f+1}[\lambda^{C_f+1-j}(\sum_{k=j}^{C_f+1} g^*_{s_0,a_j})(\prod_{k=j}^{C_f+1} g^*_{a_k,a_{k-1}})^{-1}] \quad i = C_f+2,\ldots \quad (29)$$

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

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

Then we have:

$$N_l = \tilde{N}(s0)\pi^*(s_0) + \sum_{i=1}^{\infty} i\pi^*(a_i) \quad (30)$$

$$R_l = N_l/\lambda \quad (31)$$

Remark: Further State Space Reduction

In the previous section, we discussed the methodology of constructing an upper bound model M_u and a lower bound model M_l. The computational costs in solving the models are:

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 [12] studied under what conditions 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 $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$, 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:

$$r_{k,\mathcal{P}_j} = \sum_{l\in\mathcal{P}_j} q_{k,l} \qquad \textit{for } k \in \mathcal{P}_i$$

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:

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

where:

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

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

For example, assume 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 and $J = 3$. 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 greatly reduce the state space of the model that needs to be solved.

5 Numerical Example

In this section, we present two examples to illustrate the bounding algorithm.

The system we consider in our first example consists of four homogeneous servers. To vary the system utilization ρ from 0.1 to 0.9, we fix the input arrival rate at 4.0 and vary the service rates for all servers. For this example, we set $d = 4$. For $\rho = 0.1$ to 0.7, we set $C_i = 7$, for $\rho = 0.8$, we set $C_i = 9$ and for $\rho = 0.9$, we set $C_i = 10$. Table 1 illustrates the upper and lower bound mean response time as a function of system utilization. Percentage error[3] is defined to be $\frac{R_u - R_l}{R_u + R_l}$ x 100%. Note that the bounds are very tight.

The second system we consider has four heterogeneous servers with $\mu_1 = 10, \mu_2 = 9, \mu_3 = 8$ and $\mu_4 = 6$. 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. We set $d = 6$, and for $\rho = 0.1$ to 0.7, we set $\vec{C} =< 9,8,7,5 >$. For $\rho = 0.8$ to 0.9, we set $\vec{C} =< 12,11,10,8 >$. Table 2 illustrates the upper and lower bound mean response time.

To illustrate the tradeoff between computational cost and accuracy of the bounds Let us consider the homogeneous queueing system in the first example. By fixing the system utilization at 0.9 and increasing the number of states generated, we see the improvement of the bounds on the mean response time. The result is illustrated in Table 3.

6 Conclusion

Joining the shortest queue load balancing 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 arrival process is state dependent and therefore no closed form solution exits 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 to generate the entire state space to analyze the Markov model numerically. We have presented an approach to bound the mean response time and the mean number of customers of minimum expected delay routing policy, which is a generalization of the join the shortest queue routing policy. The algorithmic approach provides the flexibility to tradeoff computational resources and tighter bounds. There is ongoing work in how to choose d and C_i such that we can a priori predict the error bounds. Also there is ongoing work in investigating the possibility of bounding the mean response time under more relaxed conditions, e.g., by allowing general interarrival and/or service distributions.

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

Appendix: Proof for Theorem 1

$$V^t[s] - V^t[s-\epsilon_i] \geq 0$$
$$for\ i = 1,\ldots,K; t \geq 0, n_i > 0;\ s \in \mathcal{S}_u$$

Proof: The proof is by induction on t. When $t = 0$, $V^0(s) = 0$ for all s; therefore the condition is satisfied. Now assume the condition is satisfied for $t = m$. For $t = m+1$, we have in general[4]:

$$V^{m+1}(s) - V^{m+1}(s-\epsilon_i) =$$
$$\left\{ r(s) + \sum_{j=1}^{K} \lambda h 1\{j = n^*(s)\} V^m(s+\epsilon_j) + \right.$$
$$\sum_{j=1, j\neq i}^{K} 1\{n_j > 0\}\mu_j h V^m(s-\epsilon_j) +$$
$$\left. \mu_i h V^m(s-\epsilon_i) + \sum_{j=1, j\neq i}^{K} 1\{n_j = 0\}\mu_j h V^m(s) \right\} -$$
$$\left\{ r(s-\epsilon_i) + \sum_{j=1}^{K} \lambda h 1\{j = n^*(s-\epsilon_i)\} V^m(s-\epsilon_i+\epsilon_j) + \right.$$
$$\sum_{j=1, j\neq i}^{K} 1\{n_j > 0\}\mu_j h V^m(s-\epsilon_i-\epsilon_j) +$$
$$1\{n_i - 1 > 0\}\mu_i h V^m(s-\epsilon_i-\epsilon_i) +$$
$$\left(\sum_{j=1, j\neq i}^{K} 1\{n_j = 0\}\mu_j h + 1\{n_i - 1 = 0\}\mu_i h \right)$$
$$\left. V^m(s-\epsilon_i) \right\}$$

Grouping similar terms, we have:

$$V^{m+1}(s) - V^{m+1}(s-\epsilon_i) =$$
$$\left\{ \left[r(s) - r(s-\epsilon_i) \right] + \right.$$

[4] Note that the condition implies that in state s, there is at least one job in the i^{th} queue.

System Utilization	States Generated	Response Time Upper Bound	Response Time Lower Bound	Spread of Bounds	Percentage Error
0.1	175	0.100074	0.100071		
0.2	175	0.201692	0.201692		
0.3	175	0.309557	0.309557		
0.4	175	0.431429	0.431429		
0.5	175	0.579080	0.579068	0.000012	0.00103 %
0.6	175	0.773178	0.772967	0.000211	0.01364 %
0.7	175	1.061225	1.056777	0.004448	0.21000 %
0.8	245	1.569928	1.554950	0.014978	0.47931 %
0.9	280	2.867803	2.752649	0.115151	2.04883 %

Table 1: Homogeneous servers system

System Utilization	States Generated	Response Time Upper Bound	Response Time Lower Bound	Spread of Bounds	Percentage Error
0.1	3095	0.103573	0.103301	0.000272	0.13148 %
0.2	3095	0.107718	0.107435	0.000283	0.13153 %
0.3	3095	0.113167	0.112859	0.000308	0.13627 %
0.4	3095	0.120737	0.120305	0.000432	0.17922 %
0.5	3095	0.131729	0.131086	0.000643	0.21466 %
0.6	3095	0.148537	0.147701	0.000836	0.28221 %
0.7	3095	0.176870	0.174620	0.002250	0.64013 %
0.8	6410	0.230285	0.225782	0.004503	0.98735 %
0.9	6410	0.391237	0.372385	0.018852	2.46876 %

Table 2: Heterogeneous servers system

$$\left[\sum_{j=1}^{K}\lambda h 1\{j=n^*(s)\}V^m(s+e_j)-\sum_{j=1}^{K}\lambda h 1\{j=n^*(s-e_i)\}V^m(s-e_i+e_j)\right]+\sum_{j=1,j\neq i}^{K}1\{n_j>0\}\mu_j h[V^m(s-e_j)-V^m(s-e_i-e_j)]+\left[\mu_i h V^m(s-e_i)-1\{n_i-1>0\}\mu_i h V^m(s-e_i-e_i)-1\{n_i-1=0\}\mu_i h V^m(s-e_i)\right]+\sum_{j=1,j\neq i}^{K}1\{n_j=0\}\mu_j h\left[V^m(s)-V^m(s-e_i)\right]$$

It is clear that the first [] term is greater than zero. By the induction hypothesis, the third, fourth, and fifth [] terms are greater than zero. It remains to prove that the second [] term is greater than or equal to zero. To answer this question, we break this term into four cases.

Case 1: for state s, $i \neq n^*(s)$ and for state $s-e_i$, $i \neq n^*(s-e_i)$, this implies $n^*(s)=n^*(s-e_i)=j$ where $j \neq i$, the second term is:

$$\sum_{j=1,j\neq}^{K}\lambda h 1\{j=n^*(s)\}\left[(V^m(s+e_j)-V^m(s-e_i+e_i)\right]$$

which is greater than or equal to 0.

Case 2: for state s, $i=n^*(s)$ and for state $s-e_i$, $i=n^*(s-e_i)$, the second term is:

$$\lambda h\left[(V^m(s+e_i)-V^m(s-e_i+e_i)\right] \geq 0$$

Case 3: for state s, $i \neq n^*(s)$ and for state $s-e_i$, $i=n^*(s-e_i)$, the second term is:

$$\left[\sum_{j=1,j\neq i}^{K}\lambda h 1\{j=n^*(s)\}V^m(s+e_j)-V^m(s-e_i+e_i)\right]$$

which is greater than or equal to 0.

d	C	States Generated	Response Time Upper Bound	Response Time Lower Bound	Spread of Bounds	Percentage Errors
4	7	175	3.157382	2.187368	0.670011	11.86968 %
4	9	245	2.927385	2.624671	0.302711	5.15228 %
4	10	280	2.867803	2.752649	0.115154	2.04883 %
5	12	518	2.790852	2.760358	0.030494	0.51932 %

Table 3: Computational Cost vs. Accuracy

Case 4: for state s, $i = n^*(s)$ and for state $s - \epsilon_i$, $i \neq n^*(s - \epsilon_i)$. This case is obviously impossible. ∎

References

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

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

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

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

[5] P. J. Courtois. Decomposability -- queueing and computer system approximation. *Academic Press, New York, 1977.*

[6] 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*

[7] Nico M. van Dijk. The Importance of Bias-terms for Error Bounds and Comparison Results. *First International Conference on Numerical Solution of the Markov Chains, January 1990.*

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

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

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

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

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

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

[14] John C.S. Lui, R.R. Muntz Computing Bounds on the Steady State Availability of Repairable Computer Systems, *Tech. Report UCLA CSD-890066. December, 1989. Submitted for publication.*

[15] John C.S. Lui, R.R. Muntz Algorithmic Approach to Bounding the Mean Response Time of a Minimum Expected Delay Routing System. *Tech. Report UCLA CSD-910064, September, 1989.*

[16] R. R. Muntz, E. de Souza Silva, A. Goyal. Bounding Availability of Repairable Computer Systems, *SIGMETRICS 1989, pp. 29-38, also appeared in the special issue of IEEE-TC on performance evaluation, Dec. 1989.*

[17] 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.*

[18] 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.*

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

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

[21] S.M. Ross. Stochastic Processes. *Wiley Series in Probability and Mathematical Statistics, 1983.*

[22] D. Towsley, P. Sparaggis and C. Cassandras Stochastic Ordering Properties and Optimal Routing Control for a Class of Finite Capacity Queueing Systems. *University of Mass. COINS Technical Report:90-72.*

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