A THRESHOLD STRATEGY FOR A FIRST IN FIRST OUT HETEROGENEOUS TWO SERVER QUEUEING SYSTEM

by

Rajesh Kumar Pankaj

B.Tech. Indian Institute of Technology, Kanpur
(1986)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF

MASTER OF SCIENCE

IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

March 1988

Signature of Author _____
Department of Electrical Engineering and Computer Science
March 9, 1988

Certified by _____
Robert G. Gallager
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

A THRESHOLD STRATEGY FOR A FIRST IN FIRST OUT HETEROGENEOUS TWO SERVER QUEUEING SYSTEM

by

Rajesh Kumar Pankaj

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
March 1988

ABSTRACT

In data networks, a packet has to wait for other packets in a variety of situations. The aim of this work is to model and minimize the delay for the special case in which the packets are transmitted over two different links of different speeds. This system is modelled as a First In First Out queue and its delay is analyzed.

A single waiting queue is used to feed customers to both the servers. To determine how to use the servers to serve the customers, a threshold strategy is proposed. The threshold that minimizes the delay is computed as a function of the arrival and service rates. The threshold strategy is compared with a similar strategy for the two server non-FIFO queue. The optimal threshold is compared with the threshold determined by the intuitive *greedy algorithm*. The performance of the system with the optimal threshold is compared with its performance with the greedy threshold.

Thesis Supervisor: Dr. Robert G. Gallager
Title: Professor of Electrical Engineering and Computer Science

ACKNOWLEDGEMENTS

I thank my parents, Asha Maldahiyar and Binod Kumar Maldahiyar, for their love and encouragement.

I thank my advisor, Robert Gallager for his guidance.

I thank my friends Emre Telatar, John Spinelli, Ying Li, Whay Lee and Louise Alterman for their help and support.

I thank Arthur Giordani for the fine art work.

This work was in part supported by Vinton Hayes Fellowship. The funding was greatly appreciated.

Contents

List of Figures

List of Tables

1 Introduction

In many queueing systems we come across situations in which two or more servers are available but they do not work at the same speed. Examples of such systems range from the check out counter of a supermarket to a computer with different peripherals working at different speeds.

Our primary interest lies in packet-switched computer networks in which two nodes might be connected by two or more links of different capacities. In such cases it is important to know how to use the links to achieve optimum performance. We use mean system delay as the performance measure and make the following assumptions to model such a system.

1. Packets arrive at a node for transmission according to a Poisson process with arrival rate λ.

2. The transmission time for a packet over link i is an exponentially distributed random variable with mean $\frac{1}{\mu_i}$. We will include the propagation delay in the transmission time. The exponential transmission time assumption will not be good if the propagation delay is large. We will restrict ourselves to the situation in which there are exactly two links, i.e. i=1,2.

3. A packet being transmitted over a link has to be completely transmitted over that link. In other words, switching links is not allowed in the middle of a transmission. This assumption comes from the observation that an ARQ protocol operating on these transmissions

will not be able to recognize a packet as a packet if it is transmitted in two parts over two different links.

4. There is infinite storage space at each node.

The above assumptions do not model the delay incurred in re-ordering the packets at the destination node. This delay is important in a number of cases. A multi-packet message can not be used until all its packets are re-ordered. In a voice communication system, the delay is important and the total delay includes the re-ordering delay as well. Another example in which re-ordering is important is when a network wants to use another network as a link to communicate between two of its parts. The packets should stay in order in the network being used as a link, so that the link level protocols of the other network will work.

To model the re-ordering delay we will add an additional constraint to the system.

5. **FIFO constraint:** Let us call a packet a predecessor of another packet if the former arrived in the system before the latter. We will assume that a packet that is transmitted to the destination before one or more of its predecessors must wait for the predecessors to finish transmission. The packet exits the system as soon as all its predecessors complete transmission. This constraint takes the re-ordering delay into account.

2 Background

This problem, without the FIFO constraint, was analyzed by Larsen and Agrawala [3]. They proposed and analyzed a threshold queueing strategy to minimize the mean system delay. This system works with the following assumptions.

1. The customer arrival process is assumed to be a Poisson process with mean arrival rate λ.

2. The service times for the two servers are exponentially distributed with mean service rates μ_1 and μ_2 respectively. Server 1 is assumed to be faster than server 2, i.e. $\mu_1 \geq \mu_2$.

3. The queue is stable, i.e. $\mu_1 + \mu_2 > \lambda$.

4. The queue has infinite capacity.

5. Whenever a customer is available to receive service, server 1 remains busy.

6. Server 2 stays idle if the number of customers in the system is below a certain threshold. If the number increases beyond the threshold, server 2 starts serving a customer. Once it starts to serve, the server does not stop until that customer is completed. After that, if the number of customers in the system is still beyond the threshold, the server takes another customer. Otherwise, it becomes idle. In particular if the threshold is m, the second server will stay active if there

are more than m customers in the system including the one being served by server 1.

Larsen and Agrawala [3] have taken the mean number of customers in the system $\overline{N}$ as their performance measure; $\overline{N}$ is proportional to the mean system delay $\overline{D}$ according to Little's formula $\lambda\overline{D} = \overline{N}$.

This system can be described as a continuous time Markov chain as shown in figure 1. The state description has two components. The first, a number, represents the total number of customers in the system including those being served. The second, either I or B, denotes whether server 2 is idle or busy, respectively.

The following expression [3] gives the mean number of customers in the system

$$\overline{N}_m(\nu_1,\nu_2) = \frac{\nu_1(\nu_1+\nu_2) + \sum_{i=0}^{m}\left[(\nu_1+\nu_2-1)^2\sum_{j=0}^{i} j\nu_1^{i-j} + (i+1)(\nu_1+\nu_2-1)+1\right]\nu_2 .}{(\nu_1+\nu_2-1)\left\{\nu_1 + \sum_{i=0}^{m}\left[(\nu_1+\nu_2-1)\sum_{j=0}^{i}\nu_1^j + 1\right]\nu_2 f_i\right\}} \tag{1}$$

where

$$\nu_1 = \mu_1/\lambda \tag{2}$$

$$\nu_2 = \mu_2/\lambda \tag{3}$$

$$f_i = \sum_{j=0}^{\lfloor \frac{i}{2} \rfloor} (-1)^j \begin{pmatrix} i-j \\ j \end{pmatrix} \left(\frac{\nu_1+\nu_2+1}{\nu_1}\right)^{i-2j} \left(\frac{1}{\nu_1}\right)^j \tag{4}$$

and m is the threshold.

The boundary between the region in the $\nu_1\nu_2-$plane where m is the optimal threshold and the region where $m+1$ is the optimal threshold is

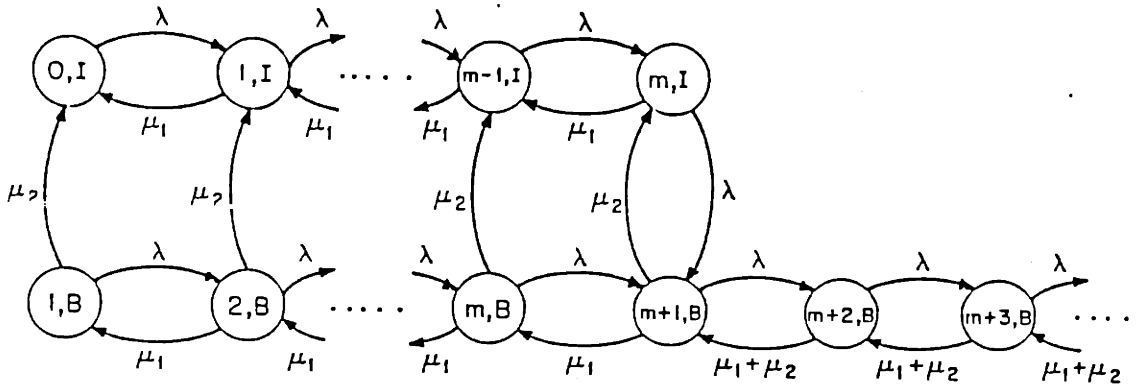

Figure 1: Transition diagram for the Markov chain of a heterogeneous two-server queue operating at threshold m.

The arrival rate is 1, and the service rates at the two servers are μ_1 and μ_2.

approximately

$$\nu_2 = \frac{\nu_1}{m+1} - \frac{m}{(m+1)^2} \tag{5}$$

Hence, the optimum threshold m^* can be approximated by

$$m^* = \frac{\nu_1 - 1 + \sqrt{4\nu_2 + (\nu_1 - 1)^2}}{2\nu_2} \tag{6}$$

Lin and Kumar [4] proved that for a two-server heterogeneous queueing system, a threshold strategy is optimal for minimizing the mean number of customers in the system $\overline{N}$ or the mean system delay $\overline{D}$. To prove this, they formulated the queue as a discrete time Markov chain by introducing dummy customers. The Markov decision process was considered with the cost criterion

$$C = E[N(t)\beta^t]$$

where $N(t)$ is the number of customers in the system at time t, and β is some discount factor less than 1.

As the first step, it was shown that to minimize the cost C, the faster server (server 1) should always be kept active. This was done using the value iteration method. Then the policy iteration method [2] was used, starting from threshold 0. It was shown that starting from such a strategy with threshold i, the policy iteration yields another threshold strategy with threshold at most $i+1$. This means that if the iteration converges it will yield a threshold strategy for the discounted cost criterion C.

It was shown that the undiscounted cost criterion, i.e. $\beta = 1$, which is equivalent to minimum mean delay, is a limiting case of the discounted

cost criterion and the above mentioned result holds there.

3 FIFO two-server queue

We will focus our attention on the First-in-First-out(FIFO) heterogeneous two server queueing system. The additional assumptions over the system analyzed by Larsen and Agrawala [3] are as follows.

1. There is a waiting room after the service. If the service of a particular customer is completed before one or more of its predecessors, the customer waits in the waiting room for the predecessors to complete their service. As soon as all the predecessors complete their service, the customer departs the system.

2. The size of the waiting room after service is infinite.

3. The system is stable, i.e. $\mu_1 + \mu_2 > \lambda$.

4. The arrival rate λ is 1. This assumption does not reduce the generality of the system as it only changes the unit of time used.

The first problem is how to use the system. Should there be two separate queues for the two servers or one single queue?

3.1 Why one queue?

Intuitively it is easy to see why a single queue from where customers are fed to both the servers should be superior to a system in which a customer is assigned to either server 1 or server 2 immediately after the arrival. The intuitive reason is that if we defer our decision until the time when it is

absolutely necessary to assign a customer to one server or the other, we get more information about the system compared to the case in which we decide earlier. Therefore the former should be able to perform better.

More precisely, let us suppose we are using a single queue, and the customers are fed to the servers according to some rule. One possible rule is to assign each customer either number 1 or number 2 immediately upon its arrival according to some rule. Customers are served by the correspondingly numbered server. Customers for each server are served in FIFO order.

This situation is exactly the same as a two queue system and therefore it will have the same performance. Therefore, any performance that can be achieved with the two queue system can also be achieved using the single queue system. Hence a single queue system is at least as good as the two queue system.

3.2 The Greedy Algorithm

One intuitive way to feed the customers to the servers from a single queue is by using *the greedy algorithm.* This algorithm works as follows. Whenever a server is free, we calculate the expected delay for each customer currently in the system under each of the following two assumptions:

(i) The customer goes to the free server and starts receiving service immediately.

(ii) The customer and all its predecessors wait in the queue and receive service from the busy server when their turn comes.

This calculation is done according to the arrival time of the customers

in the queue, beginning from the customer that arrived first. The first customer for which the expected delay in case (i) is strictly smaller than that in case (ii) goes to the free server to receive service.

When both the servers are free the customer in front of the queue goes to the server with lower mean service time. After this the second free server will take a customer into service according to the rule described earlier.

In the case when server 1 is faster than server 2, whenever server 1 becomes free the first customer in the queue is fed to server 1. However, when server 2 is free, a customer will go to server 2 only if there are enough customers in front of it to make the expected delay for server 1 higher than the mean service time of server 2. This suggests a threshold strategy with easily computable thresholds. Figure 10 shows that for a non-FIFO two server queue, the threshold calculated by the greedy algorithm is not always the same as the optimal threshold calculated in [3]. The same can be expected for the FIFO queue as well. In a later section we will try to explain intuitively, why the optimal threshold should be different from the one given by the greedy algorithm.

4 Threshold Strategy for the FIFO Queueing System

We propose a threshold strategy similar to the one given by Larsen and Agrawala [3]. To understand how it works let us look at figure 2.

The overall queueing system has two subsystems: system A where the customers arrive and get served, and system B where a customer waits for its predecessors to complete service at system A. The overall system will be referred to as system C.

System A is exactly like the system discussed earlier in section 2. It has a single stream of arrivals according to a Poisson process with rate 1, and the two servers have exponentially distributed service times with mean $\frac{1}{\mu_1}$ and $\frac{1}{\mu_2}$ respectively. We wish to use the threshold strategy on system A and it will be assumed that $\mu_1 > \mu_2$.

The threshold strategy will work on this system as follows

1. Server 1 will continue to serve customers as long as there are customers waiting for service.

2. Server 2 will start to serve a customer only when the total number of customers in system A, including the customer receiving service from server 1, is greater than a threshold m.

   Server 2 starts serving the $(m+1)$th customer in the system counting from the one being served by server 1 at that instant. This is different from the non-FIFO queue because in a non-FIFO queue the order in

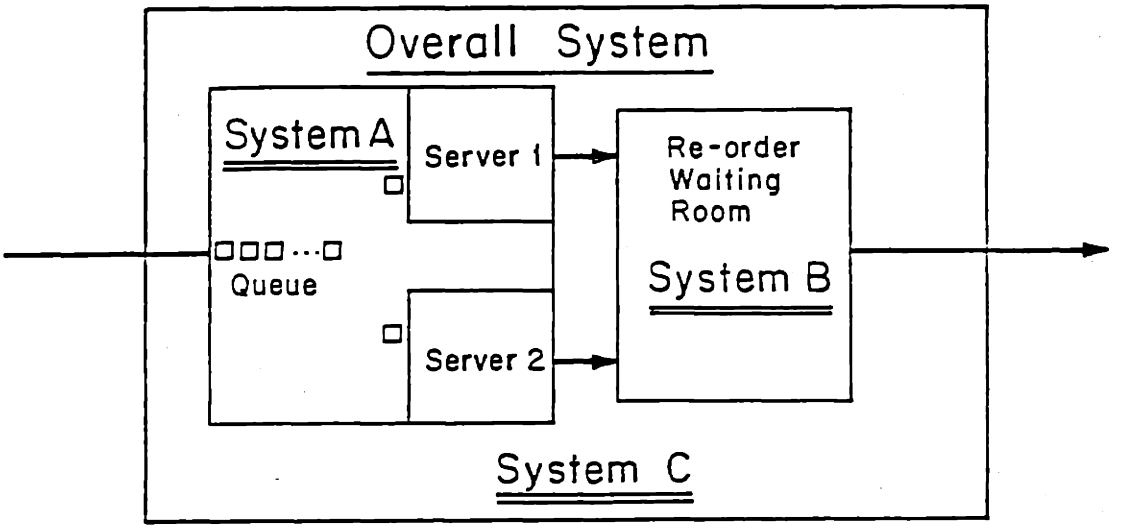

Figure 2: The subsystems of the FIFO, heterogeneous two-server queue

which the customers are served does not change the mean delay as long as preference is not given according to the length of the service times of the customers [1]. In the FIFO queueing system a customer that does not finish service can keep many other customers waiting. Therefore it is important in the FIFO system to specify which particular customer server 2 will take when it has to serve some customer.

Intuitively it is easy to see why server 2 should take the $(m+1)$th customer and not some customer earlier in the queue. The reasoning is that if it is better for server 2 to take the ith customer ($i \leq m$), it should not wait for the arrival of $(m+1)$th customer to start serving the ith customer. A better reasoning is given in the appendix.

For a given value of threshold (say m), system A can be represented as a Markov chain shown in figure 1. We know the mean delay of system A as a function of m, μ_1 and μ_2 as given by equation 1 and Little's formula.

Now, if we can analyze system B itself then we can determine the mean delay for system C.

The arrivals to system B are the departures of system A. Since system A is assumed to be stable, the mean rate of arrivals to system B is λ which is assumed to be 1. However, we do not know the exact statistics of the process. The difficulty in analyzing system B is that the amount of time a customer stays in this system depends upon the future arrivals to system B. This is because a customer in system B waits for the arrival of all the customers that went into system A earlier. Therefore the available information about system A can not be used to analyze system C directly.

We can get around the difficulty of solving system B by the following observations.

Observation 1: The arrivals to system A, which are the same as the arrivals to system C, are independent of the state of system A immediately prior to their arrival. Therefore every arrival finds system A in a typical state.

Observation 2: The statistics of system C delay for a customer is completely determined by the state of system A at the time of the customer's arrival to system A.

This means that if we can find the statistics of the delay for a customer as a function of the state of system A at the time of the customer's arrival, the mean delay for system C can be calculated.

4.1 What are the statistics of the delay of a customer?

Let us take X_1, X_2 and X_{12} as exponentially distributed random variables with mean $\frac{1}{\mu_1}, \frac{1}{\mu_2}$ and $\frac{1}{\mu_1+\mu_2}$ respectively. Also, let Y_i^k denote the random variable which is the sum of k exponential, independent and identically distributed random variables each with mean $\frac{1}{\mu_1}$ for $i = 1$, $\frac{1}{\mu_2}$ for $i = 2$ and $\frac{1}{\mu_1+\mu_2}$ for $i = 12$.

Let us assume that system A is operating with a threshold m. Since the service times are exponentially distributed, the distribution of the residual service time for a customer receiving service at the instant of an arrival is

exponential with the same mean because of the memoryless property of the distribution.

We will look at the following different cases.

Case 1 Upon arrival the customer finds:

(i) $j < m$ customers in system A including those receiving service and,

(ii) server 2 idle.

Immediately after this arrival the number of customers in the system is less than $m + 1$. Therefore, server 2 will stay idle and this new customer will be served by server 1. Also this new customer will not have to wait in system B. Hence the amount of time this customer spends in system C is a sum of $j + 1$ independent, exponentially distributed random variables with mean $\frac{1}{\mu_1}$ which is Y_1^{j+1}.

Case 2 Upon arrival the customer finds:

(i) $j < m$ customers in system A including those being served and,

(ii) server 2 busy.

The customer will be served by server 1. But if server 2 does not complete the service of the customer it is serving, then this new customer will have to wait in system B. This is because the customer being served by server 2 arrived into the system before this new customer. The delay of this customer in system A will be distributed as the sum of j independent, exponentially distributed random variables with mean $\frac{1}{\mu_1}$. The amount of time the customer being served

by server 2 will stay in system A is exponentially distributed with mean $\frac{1}{\mu_2}$. Therefore the overall delay for the new customer in system C will be $\max[Y_i^j, X_2]$.

Case 3 Upon arrival the customer finds:

(i) m customers in system A.

(ii) server 2 idle.

Server 2 is idle and the customer is $(m+1)$th in the system. This means that the new customer will be served by server 2. In this case the amount of time the new customer will stay in system A will be exponentially distributed with mean $\frac{1}{\mu_2}$. But the customer might have to wait for one or more of the m customers waiting for server 1. The amount of time taken by server 1 to complete service of m customers is distributed as the sum of m independent, exponentially distributed random variables with mean $\frac{1}{\mu_1}$. Therefore the overall system C delay for the new customer will be $\max[Y_1^m, X_2]$.

Case 4 Upon arrival the customer finds:

(i) m customers in system A

(ii) server 2 busy.

This customer will be served by server 1, since after completing the current customer, server 2 will find this new customer at number m or less. In this case the new customer will spend a time distributed as the sum of m independent, exponential random variables with mean

$\frac{1}{\mu_1}$ in system A. The amount of time the customer currently being served by server 2 will spend in system A is an exponential random variable with mean $\frac{1}{\mu_2}$. Therefore, the new customer will have to wait in system C for an amount of time which is $\max[Y_1^m, X_2]$.

The fact that the waiting time for system C for this case and case 3 are the same is important and will help us in finding the distribution of the delay for the next case.

Case 5 Upon arrival the customer finds $j > m$ customers in the system.

Let us divide the wait of this customer into two parts. Let the first part be from the time of arrival until the customer sees exactly m customers ahead in system A including the ones being served and the second part be from that time till the departure of the customer.

Both the servers will be busy during the first part of the wait. Therefore the customer will advance at the rate of $\mu_1 + \mu_2$. Therefore the first part of the delay will be a random variable distributed as the sum of $j-m$ independent, exponentially distributed random variables with mean $\frac{1}{\mu_1+\mu_2}$ which is Y_{12}^{j-m}.

At the beginning of the second part of the delay, the customer will either go to server 2 to receive service or wait for server 1. The former will take place if server 2 is idle at that point, the latter will take place otherwise. These two situations are exactly like case 3 and case 4 respectively. As it was noted earlier, the delay is the same in both the cases which is $\max[Y_1^m, X_2]$.

Therefore system C delay for this new customer is $Y_{12}^{j-m}+\max[Y_1^m, X_2]$.

Now we have the delay statistics of a customer conditioned on the state of system A at the time of arrival. The mean delay of the system can be calculated for a given threshold using these statistics.

5 Calculations

5.1 Mean values of the different random variables

(i) X_1,X_2 and X_{12} are exponentially distributed random variables with mean $\frac{1}{\mu_1}$,$\frac{1}{\mu_2}$ and $\frac{1}{\mu_1+\mu_2}$ respectively.
(ii) Y_1^k is the random variable which is the sum of k independent, exponentially distributed random variables, each with mean $\frac{1}{\mu_1}$. Therefore

$$\mathrm{E}\left[Y_1^k\right] = \frac{k}{\mu_1},$$

Similarly,

$$\mathrm{E}\left[Y_2^k\right] = \frac{k}{\mu_2} \quad \text{and}$$
$$\mathrm{E}\left[Y_{12}^k\right] = \frac{k}{\mu_1+\mu_2}$$

(iii) $Z = \max\left[X_2, Y_1^k\right]$

To determine the mean of the random variable Z, let us look at two independent Poisson arrival processes, process 1 and process 2, with rates μ_1 and μ_2 respectively. The sum of these two processes is another Poisson arrival process with rate $(\mu_1+\mu_2)$. Let us call it process 3. At any point of time, the probability that the next arrival of process 3 will be from process 1 is $\frac{\mu_1}{\mu_1+\mu_2}$ and the probability that it will be from process 2 is $\frac{\mu_2}{\mu_1+\mu_2}$.

Since the probability distributions of all the inter-arrival times are memoryless, let us start from any random time. The mean time taken for k arrivals from process 1 is $\frac{k}{\mu_1}$. The probability that there will be no arrival from process 2 during this time is $\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^k$. The reason for this is the following. Starting from any point, let E_1 be the event that the next arrival

of process 3 is from process 1. Since the interarrival times have memoryless distributions, the instant of the arrival is a renewal point for the processes. Therefore the event of getting k arrivals from process 1 before any arrival from process 2 is k independent repetitions of event E_1 in succession. Hence its probability is $[\mathrm{P}(E_1)]^k$ which is equal to $\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^k$.

Therefore the probability of having at least one arrival from process 2 before k arrivals of process 1 is $\left[1-\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^k\right]$. Or,

$$\mathrm{P}\left(Y_1^k > X_2\right) = 1-\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^k \tag{7}$$

$$\mathrm{P}\left(Y_1^k < X_2\right) = \left(\frac{\mu_1}{\mu_1+\mu_2}\right)^k \tag{8}$$

Now,

$$Z = \max\left[Y_1^k, Y_1^k + W\right]$$

where

$$W = X_2 - Y_1^k$$

or

$$Z = Y_1^k + V$$

where

$$V = \begin{cases} 0 & \text{if } W \le 0 \\ W & \text{otherwise} \end{cases}$$

Therefore

$$\begin{aligned} \mathrm{E}[Z] &= \mathrm{E}\left[Y_1^k\right] + \mathrm{E}[V] \\ &= \frac{k}{\mu_1} + \mathrm{E}[V] \end{aligned}$$

but

$$\begin{aligned} \mathrm{E}[V] &= \mathrm{E}[W|W>0]\,\mathrm{P}(W>0) \\ &= \mathrm{E}\left[X_2 - Y_1^k | X_2 > Y_1^k\right]\mathrm{P}\left(X_2 > Y_1^k\right) \end{aligned}$$

Conditioned on the event that $X_2 > Y_1^k$, which is the same as the event that there is no arrival from process 2 before k arrivals from process 1, the mean wait for an arrival from process 2 after the k^{th} arrival from process 1 is $\frac{1}{\mu_2}$. Or

$$\begin{aligned} \mathrm{E}\left[X_2 - Y_1^k | X_2 > Y_1^k\right] &= \frac{1}{\mu_2} \\ \text{or} \quad \mathrm{E}[V] &= \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^k \\ \text{or} \quad \mathrm{E}[Z] &= \frac{k}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^k \end{aligned} \tag{9}$$

Table 1 lists the mean delay for a customer in system C as a function of the state of system A at the time of the customer's arrival. The threshold for system A is assumed to be m.

5.2 Calculation of the mean delay for two simple cases

Case 1: For this case we will assume that the operating threshold is 1. As earlier, the arrival rate λ is assumed to be 1, and the service rate is μ_1 for server 1 and μ_2 for server 2. The Markov chain for system A for this threshold is shown in figure 3.

The detailed balance equations for this chain is as follows,

$$\mathrm{p}_{1B}\,(1+\mu_2) = \mathrm{p}_{2B}\mu_1 \tag{10}$$

Table 1: Table of the mean delay of a customer as a function of the state of system A at the time of the customer's arrival

Number of customers in system A at the time of the arrival	State of server 2	Mean overall delay for the customer
0	idle	$\frac{1}{\mu_1}$
1	idle	$\frac{2}{\mu_1}$
$\vdots$	$\vdots$	$\vdots$
i	idle	$\frac{i+1}{\mu_1}$
$\vdots$	$\vdots$	$\vdots$
$m-1$	idle	$\frac{m}{\mu_1}$
m	idle	$\frac{m}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^m$
1	busy	$\frac{1}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)$
2	busy	$\frac{2}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^2$
$\vdots$	$\vdots$	$\vdots$
j	busy	$\frac{j}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^j$
$\vdots$	$\vdots$	$\vdots$
m	busy	$\frac{m}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^m$
$m+1$	busy	$\frac{1}{\mu_1+\mu_2}+\frac{m}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^m$
$m+2$	busy	$\frac{2}{\mu_1+\mu_2}+\frac{m}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^m$
$\vdots$	$\vdots$	$\vdots$
n	busy	$\frac{n-m}{\mu_1+\mu_2}+\frac{m}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^m$
$\vdots$	$\vdots$	$\vdots$

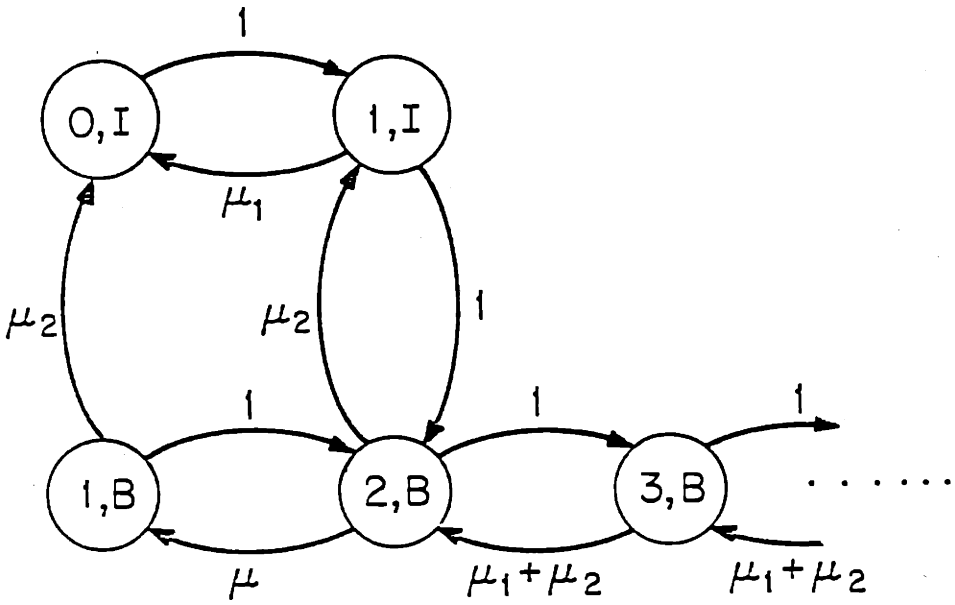

Figure 3: Markov chain of system A for threshold = 1

$$p_{1I} = (p_{1B} + p_{2B})\,\mu_2 \tag{11}$$

$$p_{0I} = p_{1I}\mu_1 + p_{1B}\mu_2 \tag{12}$$

and

$$p_{iB} = p_{(i-1)B}\frac{1}{\mu_1 + \mu_2} \qquad \text{for } i > 2 \tag{13}$$

The steady state probabilities are as follows,

$$p_{0I} = p_{2B}\frac{\mu_1\mu_2}{1+\mu_2}(2 + \mu_1 + \mu_2) \tag{14}$$

$$p_{1I} = p_{2B}\frac{\mu_2(1 + \mu_1 + \mu_2)}{1+\mu_2} \tag{15}$$

$$p_{1B} = p_{2B}\frac{\mu_1}{1+\mu_2} \tag{16}$$

$$p_{3B} = p_{2B}\frac{1}{\mu_1 + \mu_2} \tag{17}$$

$$p_{iB} = p_{2B}\left(\frac{1}{\mu_1 + \mu_2}\right)^{i-2} \qquad \text{for } i > 2 \tag{18}$$

and

$$p_{2B} = \left[\frac{\mu_1\mu_2(2 + \mu_1 + \mu_2)}{1+\mu_2} + \frac{(\mu_1 + \mu_2)^2}{\mu_1 + \mu_2 - 1}\right]^{-1} \tag{19}$$

where p_{iJ} is the steady state probability of the state (i, J).

Let d_{iJ} denote the mean delay of a customer that arrives when system A is in state (i, J); then,

$$d_{0I} = \frac{1}{\mu_1}$$

$$d_{1I} = \frac{1}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1 + \mu_2}\right)$$

$$
\begin{aligned}
d_{1B} &= \frac{1}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right) \\
d_{2B} &= \frac{1}{\mu_1} + \frac{1}{\mu_2} \\
d_{iB} &= \frac{i-2}{\mu_1+\mu_2} + \frac{1}{\mu_1} + \frac{1}{\mu_2} \qquad \text{for } i > 2
\end{aligned}
$$

Therefore the mean delay is

$$
\begin{aligned}
\overline{D} &= \sum d_{iJ} p_{iJ} \\
&= \left[\left(\frac{1}{\mu_1+\mu_2} - \frac{1}{\mu_2}\right)\left(\frac{\mu_1\mu_2}{1+\mu_2}\right)(2+\mu_1+\mu_2) + \frac{\mu_1+\mu_2}{(\mu_1+\mu_2-1)^2}\right] p_{2B} \\
&\quad + \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} - \frac{1}{\mu_1+\mu_2}\right) \qquad (20)
\end{aligned}
$$

where p_{2B} is given by equation 19.

Case 2: For this case we will assume that the operating threshold is 2. As earlier, the arrival rate λ is assumed to be 1, and the service rate is μ_1 for server 1 and μ_2 for server 2. The Markov chain for system A for this threshold is shown in figure 4.

The detailed balance equations for this case are as follows,

$$
\begin{aligned}
p_{1B}(1+\mu_2) &= p_{2B}\mu_1 && (21) \\
p_{2B}(1+\mu_2) &= p_{3B}\mu_1 - p_{1B}\mu_2 && (22) \\
p_{2I} &= (p_{1B} + p_{2B} + p_{3B})\mu_2 && (23) \\
p_{1I} &= p_{2I}\mu_1 + (p_{1B} + p_{2B})\mu_2 && (24) \\
p_{0I} &= p_{1I}\mu_1 + p_{1B}\mu_2 && (25)
\end{aligned}
$$

and

$$
p_{iB} = p_{(i-1)B}\frac{1}{\mu_1+\mu_2} \qquad \text{for } i > 3 \qquad (26)
$$

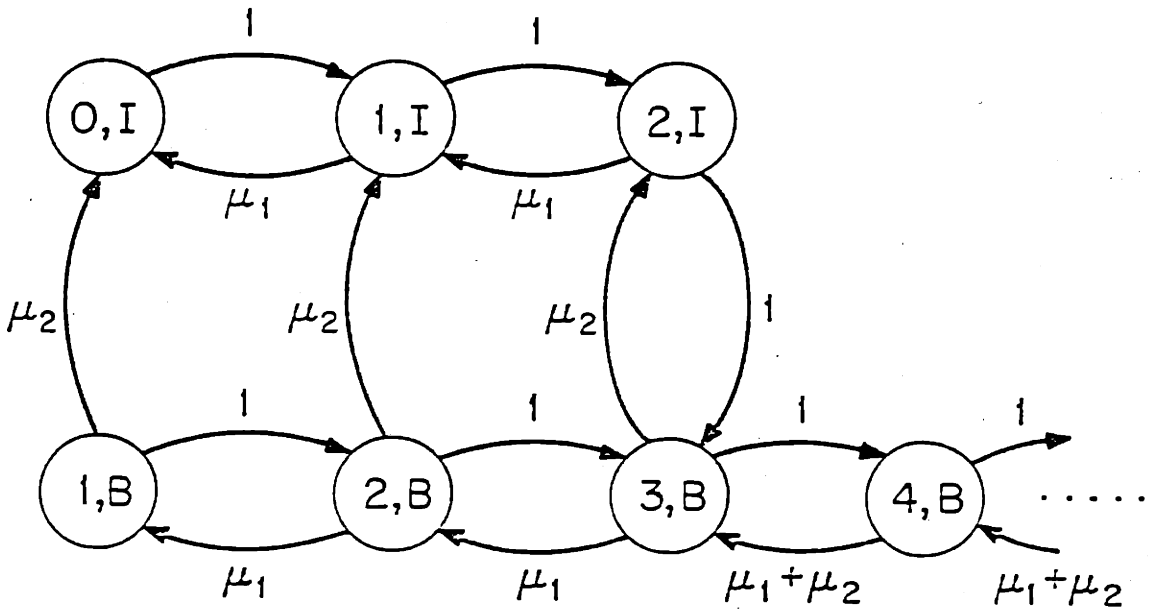

Figure 4: Markov chain of system A for threshold = 2

Solving for the steady state probabilities we get the following

$$p_{0I} = p_{3B}\mu_1^2\left[\mu_1+\mu_2+\frac{(2\mu_2-\mu_1+\mu_2^2)}{1+\mu_2(2+\mu_1+\mu_2)}\right] \quad (27)$$

$$p_{1I} = p_{3B}\mu_1\left[\mu_1+\mu_2+\frac{\mu_2-\mu_1+\mu_2^2}{1+\mu_2(2+\mu_1+\mu_2)}\right] \quad (28)$$

$$p_{2I} = p_{3B}\left[\mu_1+\mu_2-\frac{\mu_1(1+\mu_2)}{1+\mu_2(2+\mu_1+\mu_2)}\right] \quad (29)$$

$$p_{1B} = p_{3B}\left[\frac{\mu_1^2}{1+\mu_2(2+\mu_1+\mu_2)}\right] \quad (30)$$

$$p_{2B} = p_{3B}\left[\frac{\mu_1(1+\mu_2)}{1+\mu_2(2+\mu_1+\mu_2)}\right] \quad (31)$$

$$p_{iB} = p_{3B}\frac{1}{(\mu_1+\mu_2)^{i-3}} \qquad \text{for } i>3 \quad (32)$$

and

$$p_{3B} = \left[\left(1+\mu_1+\mu_1^2\right)(\mu_1+\mu_2)+\frac{\mu_1^2(2\mu_2-\mu_1+\mu_2^2)}{1+\mu_2(2+\mu_1+\mu_2)} + \frac{\mu_1\mu_2(1+\mu_2)}{1+\mu_2(2+\mu_1+\mu_2)}+\frac{\mu_1+\mu_2}{\mu_1+\mu_2-1}\right]^{-1} \quad (33)$$

and from table 1

$$d_{0I} = \frac{1}{\mu_1}$$

$$d_{1I} = \frac{2}{\mu_1}$$

$$d_{2I} = \frac{2}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^2$$

$$d_{1B} = \frac{1}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)$$

$$d_{2B} = \frac{2}{\mu_1}+\frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^2$$

$$d_{iB} = \frac{i-2}{\mu_1+\mu_2} + \frac{2}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^2 \qquad \text{for } i \geq 3$$

Therefore the mean delay, i.e.

$$\begin{aligned}\overline{D} &= \sum d_{iJ}p_{iJ} \\ &= \left[\frac{2}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^2\right] - p_{3B}F \qquad (34)\end{aligned}$$

where

$$\begin{aligned}F &= \left[\frac{1}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^2\right]\left[\mu_1^2(\mu_1+\mu_2) + \frac{\mu_1^2(2\mu_2-\mu_1+\mu_2^2)}{1+\mu_2(2+\mu_1+\mu_2)}\right] \\ &+ \left[\frac{\mu_1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^2\right]\left[\mu_1+\mu_2+\frac{(\mu_2-\mu_1+\mu_2^2)}{1+\mu_2(2+\mu_1+\mu_2)}\right] \\ &+ \left[\mu_1 - \frac{\mu_1^3}{(\mu_1+\mu_2)^2}\right]\left[\frac{1}{1+\mu_2(2+\mu_1+\mu_2)}\right] \\ &- \left[\frac{\mu_1+\mu_2}{(\mu_1+\mu_2-1)^2}\right] \qquad (35)\end{aligned}$$

and p_{3B} is as given by equation 33.

As is obvious from the calculations above, this system does not lend itself to nice expressions for the mean system delay and therefore it is difficult to find closed form expressions for the optimum threshold. In the next subsection we will look at a way to numerically compute the mean delay.

5.3 Numerical Computation

To compute the mean delay of system C the following two values are needed: the steady state probabilities of system A and the mean delay of a customer

as a function of the state of system A at the time of its arrival.

The mean delay of a customer as a function is listed in table 1. The steady state probabilities for system A are computed by using the detailed balance equations for the Markov chain across the cutsets shown in figure 5.

We start by assuming

$$p_{1B} = 1$$

The balance of flow across cutset 1 gives

$$\mu_1 p_{2B} = p_{1B}(1 + \mu_2)$$

Similarly by equating the flow across cutset $n(\leq m)$ gives

$$\mu_1 p_{(n+1)B} = \mu_2 \sum_{i=1}^{n} p_{iB} + p_{nB}$$

Since at each step the terms on the right hand side are known, we calculate one $p_{(n+1)B}$ at each step until we reach $p_{(m+1)B}$.

Cutset number $m+1$ gives the following equation

$$p_{mI} = \mu_2 \sum_{i=1}^{m+1} p_{iB}$$

which readily gives p_{mI}.

Similarly cutset number $m+j\,(1 < j \leq m+1)$ gives

$$p_{(m+1-j)I} = \mu_1 p_{(m+2-j)I} + \mu_2 \sum_{i=1}^{m+1-j} p_{(m+1-j)B}$$

In this equation also, all the right hand terms are known from the previous

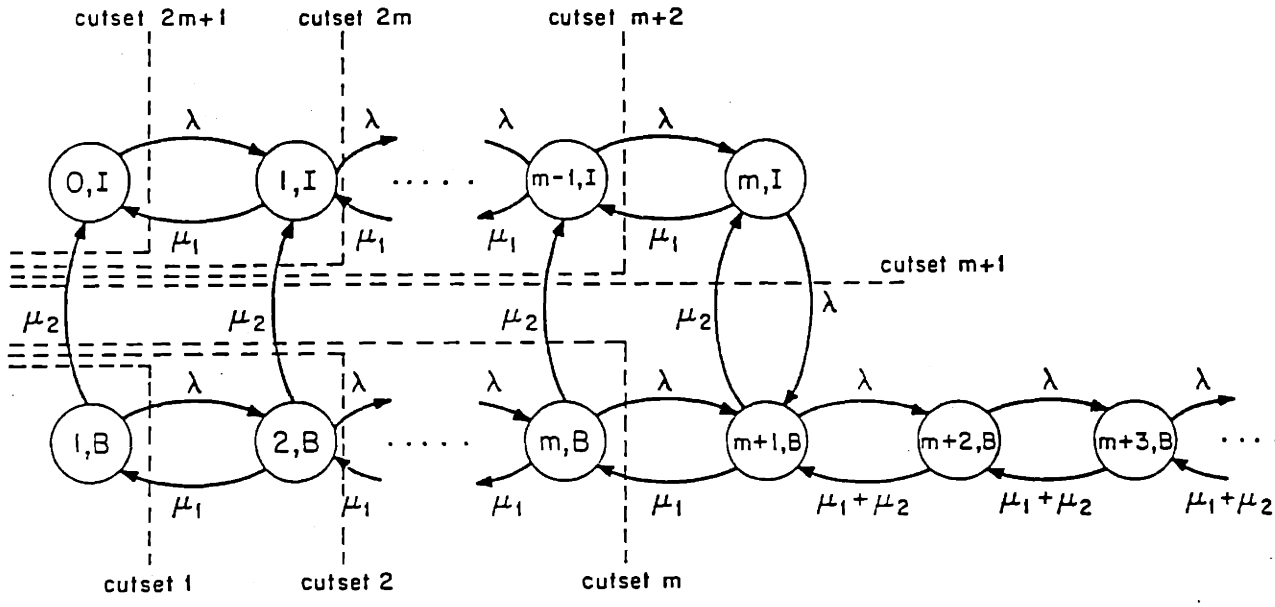

Figure 5: Markov chain, showing the cutsets used for the detailed balance equations for numerical computation

calculations. Therefore by solving the balance equation across the cutset one by one we can calculate p_{iI} for $i \leq m$ and p_{iB} for $i \leq m+1$.

For $i > m+1$, p_{iB} can be calculated by

$$p_{iB} = \frac{1}{\mu_1 + \mu_2} p_{(i-1)B}$$

In this way we get a scaled version of the steady state probabilities of system A. After this we sum all the computed values of the steady state probabilities. This sum will not be equal to 1. By dividing the computed values by this sum we get the correct values of steady state probabilities for system A.

The expression for the mean delay

$$\overline{D} = \sum p_{iJ} d_{iJ}$$

is used to compute the mean delay for system C.

6 Results

In this section the results from the numerical calculation will be discussed.

Figure 6 shows the plot of delay vs. threshold for a particular set of values of the service rates. The delay is computed only for integer values of the threshold and the points are connected by straight lines. For this plot μ_1 and μ_2 are 2.0 and 1.0 respectively. The time unit is chosen in such a way that the arrival rate λ is 1. The delay is also given in the same time unit. This curve shows two interesting features typical of such plots.

(i) There is some threshold which minimizes the delay. The minimum occurs at one or two thresholds but in the cases when two thresholds minimize the delay, the two thresholds differ by 1.

(ii) As the threshold increases, mean delay approaches a limit (for $\mu_1 > 1$) which is equal to the mean delay of the system when only the faster server is present.

Intuitively the second feature can be explained as follows: As the threshold becomes larger and larger, fewer and fewer customers are served by the slow server and the system works almost like a single-server queue with only the fast server.

Figure 7 shows the comparison between the delays for a heterogeneous two-server queue with or without the First-in-First-out restriction. As should be expected, the delay for the queue with the FIFO restriction is higher than that without the FIFO restriction because of the re-ordering delay in the former.

Another interesting thing to be noted from this plot is that the minimum

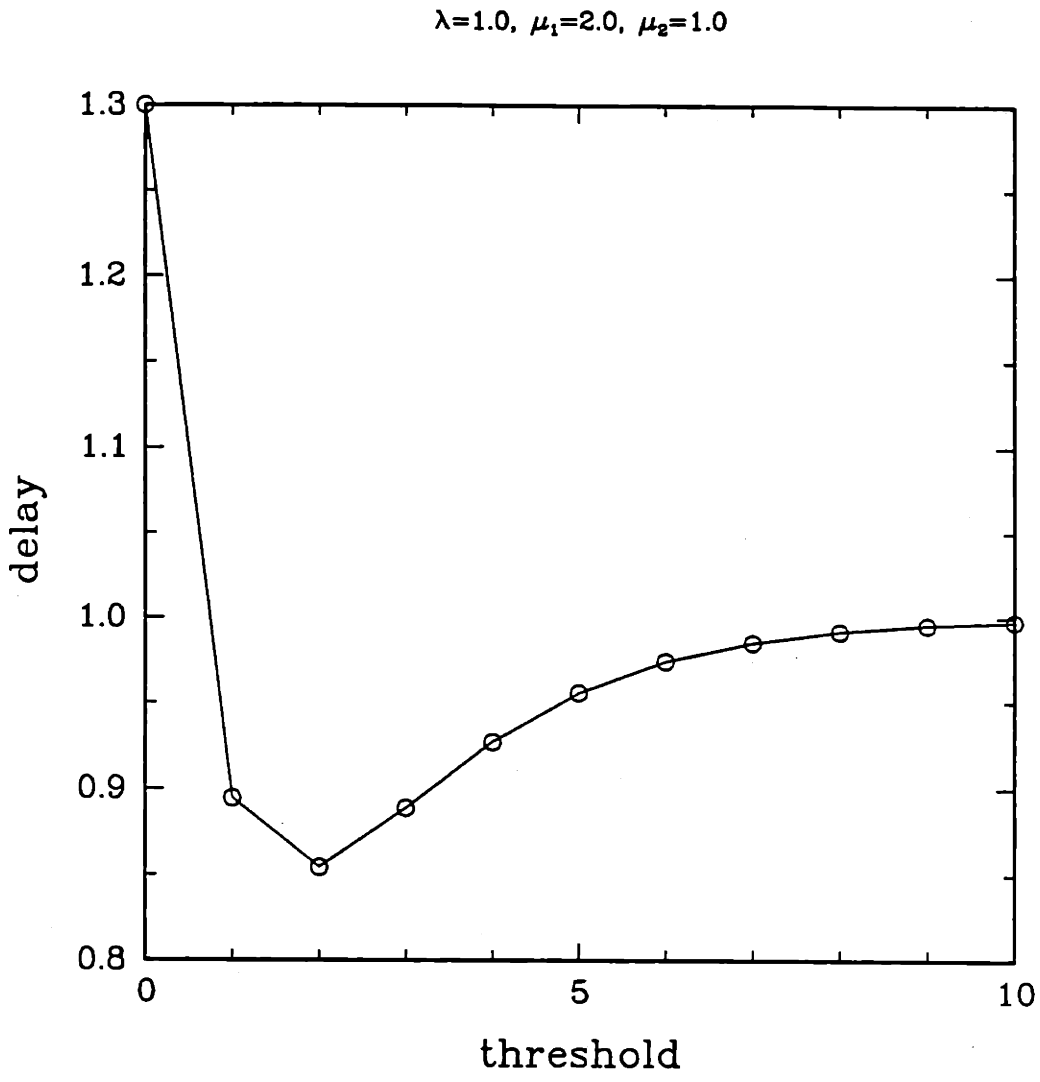

Figure 6: Delay vs. threshold for a non-FIFO queue

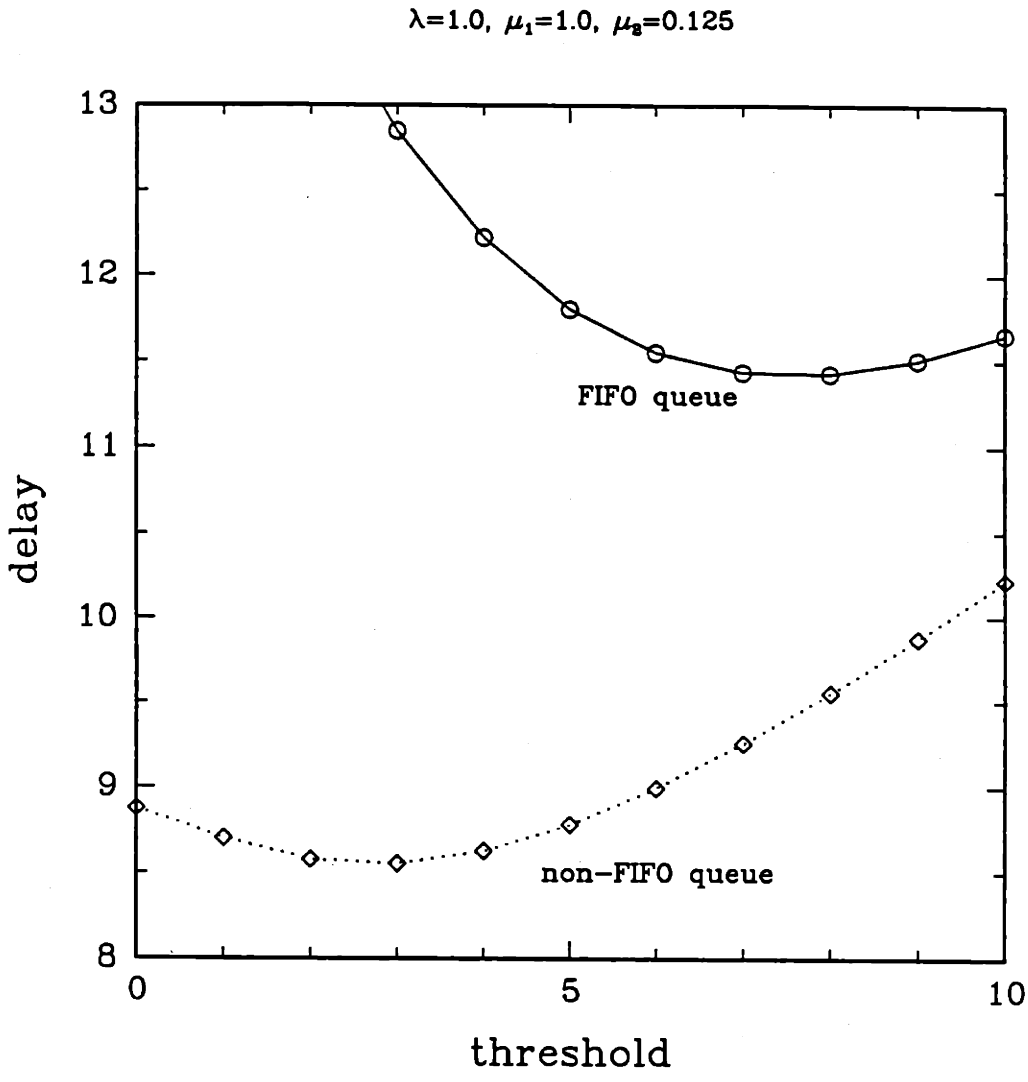

Figure 7: Comparison between the delays for a FIFO and a non-FIFO heterogeneous two-server queue

occurs at a higher threshold in the FIFO queue than the non-FIFO queue. An intuitive argument for this will be given later.

Figure 8 shows the different regions where different values of the threshold are optimum. The most striking feature here is that the curves marking the boundaries between the regions where two different values of the threshold are optimum are almost straight lines. This plot gives the optimum threshold scheme for any given pair of service rates, i.e. μ_1 and μ_2.

Figure 9 compares the optimum threshold regions of the FIFO heterogeneous two-server queue with that of the non-FIFO queue. The most noticeable thing here is that for all the tested values of μ_1 and μ_2, the optimum threshold for the FIFO case is higher than the threshold for the non-FIFO case. The same thing was noticed in figure 7 as well. An intuitive reasoning is as follows.

Let us suppose that we have a FIFO and a non-FIFO queue with the same arrival and service rates in the exact same state with server 2 being idle. If all the customers wait for server 1 then their mean delay is the same in both the queues. However, if customer number k goes to server 2, in the FIFO queueing system, its expected delay is $\mathrm{E}[\max(Y_1^{k-1}, X_2)]$, but in the non-FIFO queue, its expected delay is $\mathrm{E}[X_2]$. Since

$$\mathrm{E}[\max(Y_1^{k-1}, X_2)] > \mathrm{E}[X_2]$$

therefore, sending a customer to server 2 is a worse alternative in the FIFO queue compared to the non-FIFO queue. Hence in the FIFO queue, a customer should be sent to server 2 only if the wait for server 1 is too large. This implies that the threshold for the FIFO queue should be higher than

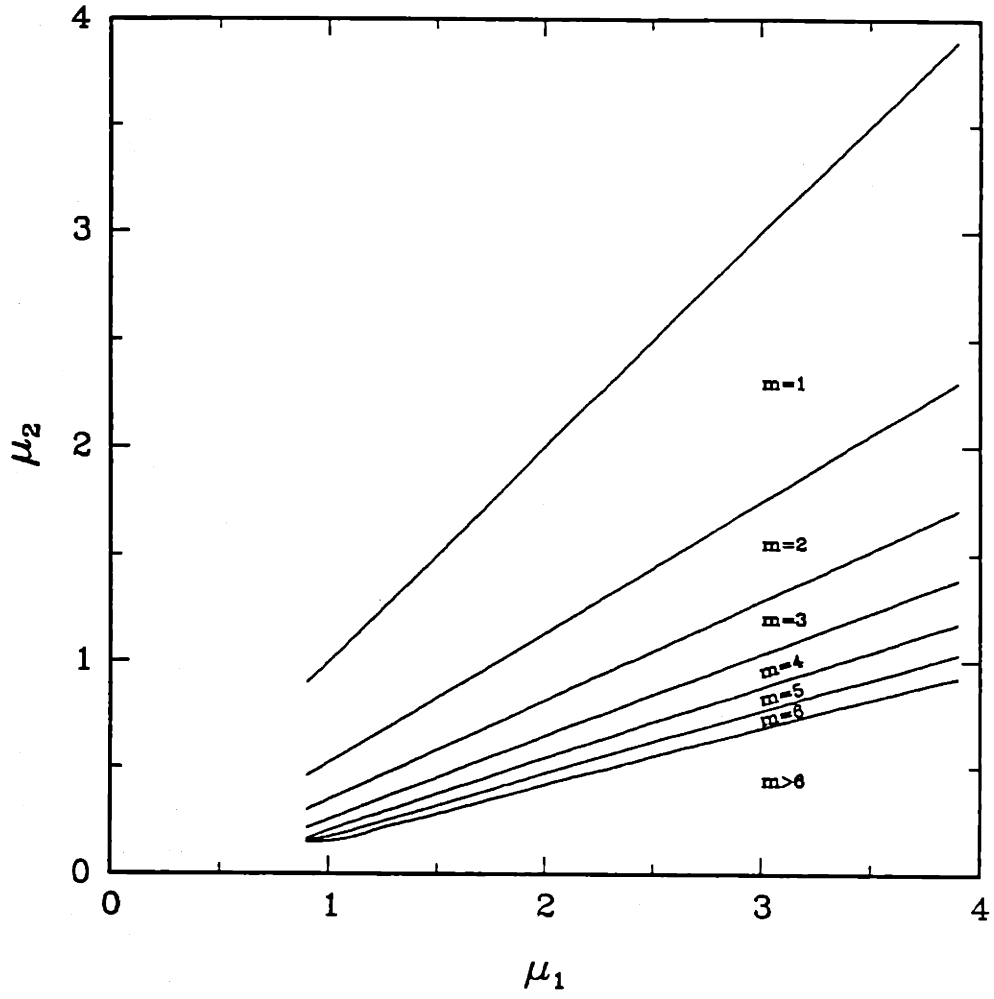

Figure 8: $\mu_1\mu_2$−plane, partitioned into different optimal threshold regions for a FIFO queue.($\lambda = 1$)

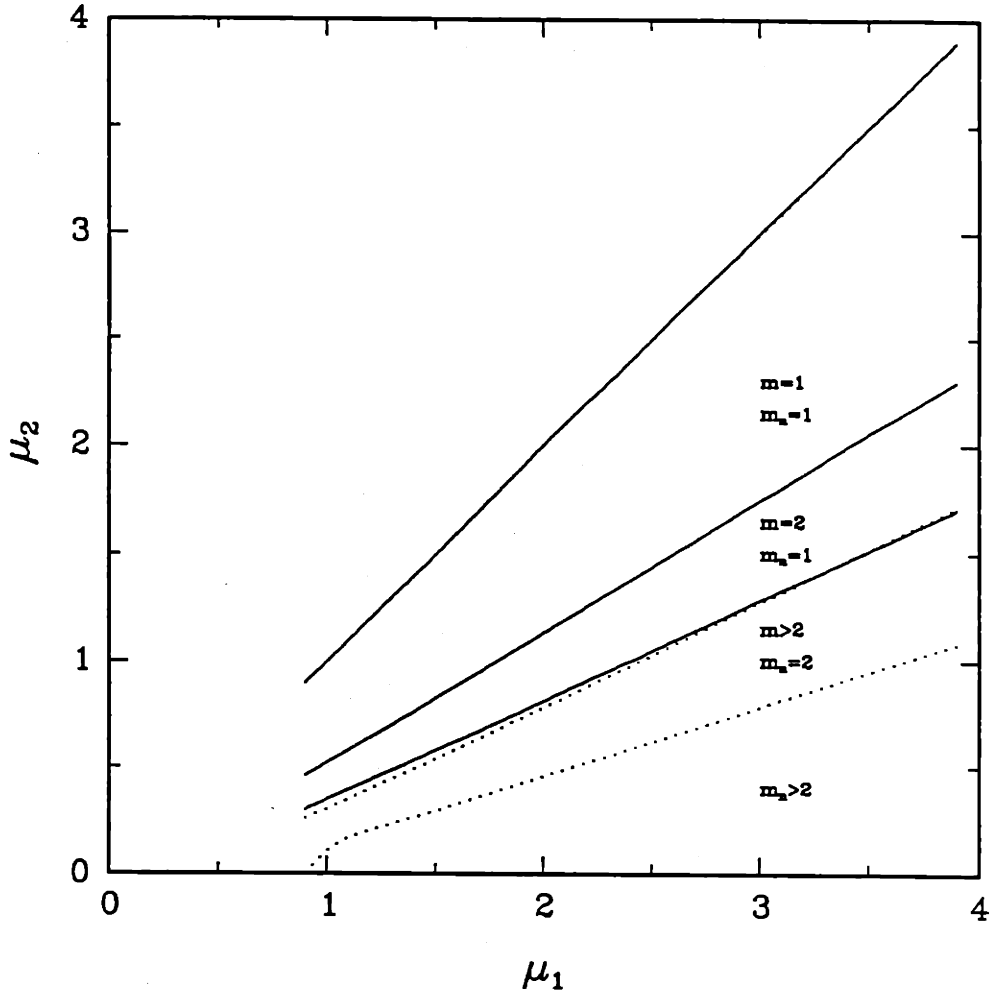

Figure 9: Comparison between the optimal threshold for a FIFO and a non-FIFO queue

Solid lines represent the boundaries of optimal threshold regions for a FIFO queue and the dotted lines represent the boundaries for a non-FIFO queue. m is the optimal threshold for the FIFO queue and m_n is the optimal threshold for the non-FIFO queue.

the threshold for the non-FIFO queue.

However, the argument is not rigorous because a change in the threshold changes all the state probabilities as well as the delays and it is not obvious that a comparison can be made between the two systems with different thresholds.

6.1 Greedy vs. optimal threshold for a non-FIFO queue

The threshold given by the greedy algorithm for a non-FIFO queue is

$$\left\lfloor \frac{\mu_1}{\mu_2} \right\rfloor$$

This is because when server 2 is idle, the expected delay of customer number j is $\frac{j}{\mu_1}$ if it waits for server 1 and $\frac{1}{\mu_2}$ if it goes to server 2. Therefore, when server 2 is idle, customer number $\left\lfloor \frac{\mu_1}{\mu_2} \right\rfloor + 1$ will be the customer sent to server 2 according to the algorithm described in section 3.2.

Figure 10 compares the threshold regions according to the greedy algorithm with the optimal threshold regions on the $\mu_1\mu_2-$plane. As can be seen from the graph, the optimal threshold is either lower than or equal to the threshold given by the greedy algorithm.

A proof is given below to show why the optimal threshold should be lower than the greedy threshold when $\frac{\mu_1}{\mu_2}$ is greater than an integer by a very small amount.

Let us consider a situation in which $\mu_1 = (k+\epsilon)\mu_2$, where k is an integer and ϵ is a very small positive number. We will assume that ϵ is equal to 0 for all purposes except when determining the greedy threshold. The greedy algorithm in this case suggests a threshold of k.

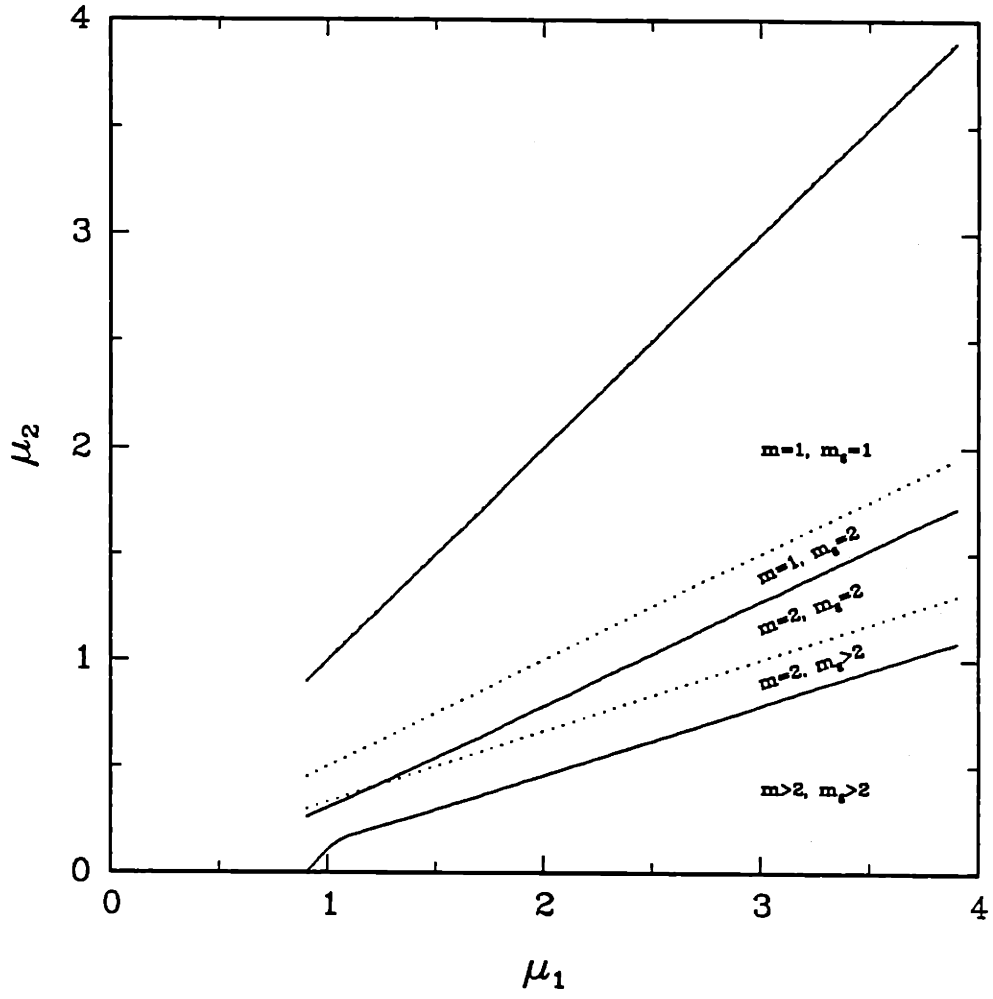

Figure 10: Comparison between the thresholds determined by the greedy algorithm and the optimal thresholds in a non-FIFO queue.

Solid lines represent the boundaries of optimal threshold regions and dotted lines represent the boundary of the greedy threshold regions. In each region m is the optimal threshold and m_g is the greedy threshold.

Now let us look at the following situation. Suppose there are exactly k customers in the system and server 2 is idle. Let us number the customers from 1 to k according to their arrival time, number 1 being the one that arrived earliest.

Let us compare the mean delay of the two systems: the original system that uses the greedy threshold k all the time, and the modified system that uses a threshold of $k-1$ in the beginning, sending customer number k to server 2 and then switching to the greedy threshold.

In the modified system, if no service completion takes place at server 1 before the next arrival to the system, then at the time of the next arrival, customer number k can still be receiving service from server 2. Since the service time at server 2 has a memoryless distribution, the state of the system at the time of the latest arrival is exactly the same as in the original system. Hence, the expected delay of the customers remains unchanged by the decision to use a threshold of $k-1$ at the present instant.

However, if no service completion takes place at server 1 before the next arrival but a service completion at server 2 takes place in the modified system, then the mean delay of the new arrival is unchanged because in the original system it would have gone to server 2 and experienced a mean delay of $\frac{1}{\mu_2}$, but in the modified system, it waits for server 1 and experiences a mean delay of $\frac{k}{\mu_1}$. But the number of customers in the system is smaller at the time of the new arrival for the modified system. Since the same threshold is going to be used in both the systems after this point, the average number of customers in the modified system will be less than or equal to the average number of customers in the original system at each

time instant in the future. This means that the mean delay will be smaller for the modified system.

In all the other cases the mean delay of all the customers either stays the same or becomes smaller in the modified system. This implies that one use of threshold $k-1$ followed by the use of threshold k is better than the use of threshold k. Using the result from Markov decision theory [2] we conclude that use of threshold $k-1$ will give lower mean delay than the use of threshold k.

This will be true as ϵ increases until ϵ becomes big enough to offset the advantage gained by sending customer number k to server 2, by increasing the mean delay of customer k when it goes to server 2.

6.2 Greedy vs. optimal threshold for the FIFO queue

A careful analysis is needed to find the greedy threshold for a FIFO queue. When server 2 is idle, the expected system delay of customer number j is $\frac{j}{\mu_1}$ if it waits for server 1. However, if customer j goes to server 2 for service, then its delay will be distributed as $\max\left[Y_1^{j-1}, X_2\right]$ where Y_1^{j-1} and X_2 are as defined in section 5.1. Hence, in this case the expected delay of customer j is

$$\frac{j-1}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^{j-1}$$

Therefore, the greedy threshold for the FIFO queue is the smallest integer k for which

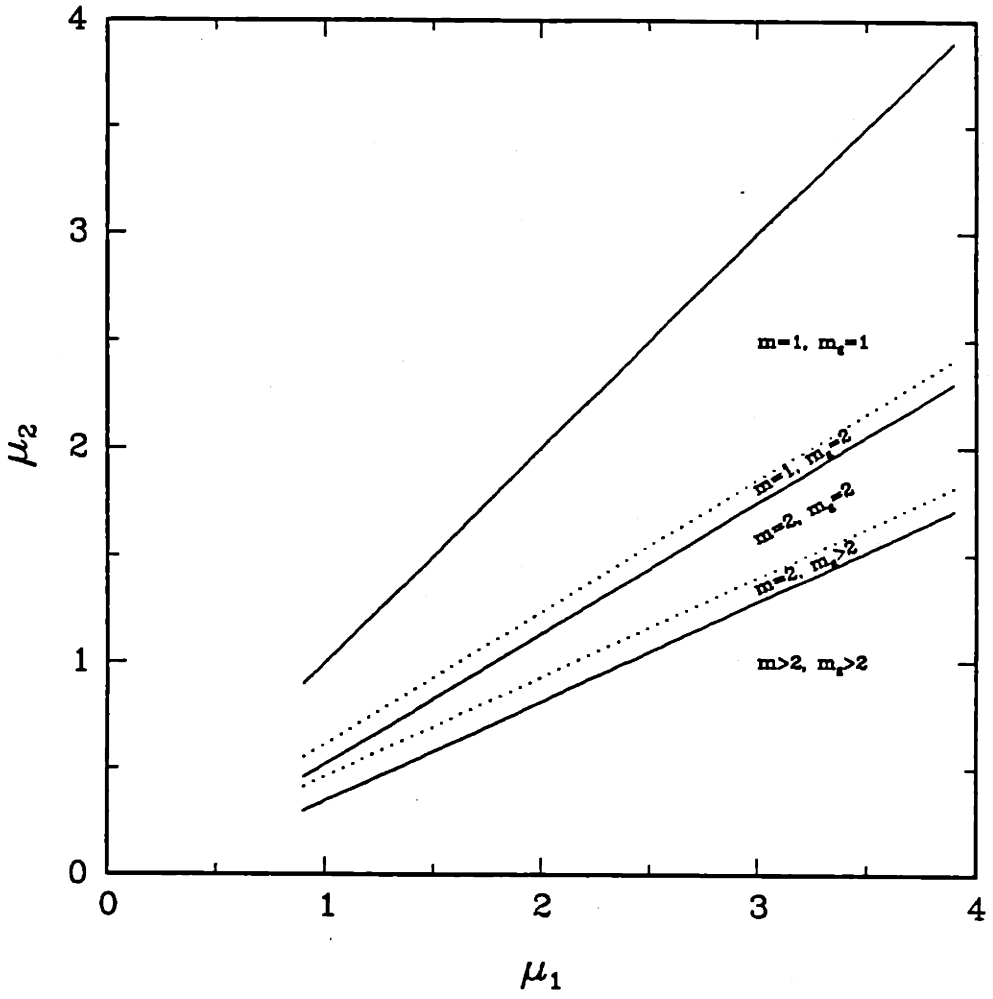

Figure 11: Comparison between the thresholds determined by the greedy algorithm and the optimal thresholds in a FIFO queue.

Solid lines represent the boundaries of optimal threshold regions and dotted lines represent the boundary of the greedy threshold regions. In each region m is the optimal threshold and m_g is the greedy threshold.

$$\frac{k+1}{\mu_1} > \frac{k}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^k$$

$$\frac{1}{\mu_1} > \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^k \tag{36}$$

Hence, if

$$\frac{1}{\mu_1} = \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^r$$

or,

$$r = \frac{\log\mu_2 - \log\mu_1}{\log\mu_1 - \log(\mu_1+\mu_2)} \tag{37}$$

then,

$$k = \lceil r \rceil \tag{38}$$

Figure 11 compares the regions of different greedy thresholds with the regions of different optimum thresholds on the $\mu_1\mu_2-$ plane. This plot has two interesting features.

(i) The boundaries between two regions of different greedy thresholds are straight lines passing through the origin. This is because these boundaries satisfy

$$\frac{1}{\mu_1} = \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^r \qquad \text{for } r = 0,1,2\ldots \tag{39}$$

For a given r the points satisfying equation 39 also satisfy

$$\frac{\mu_2}{\mu_1} = z$$

where z is the solution to

$$z(1+z)^r = 1 \tag{40}$$

This shows that these boundaries should be straight lines passing through the origin with slopes given by the solutions of equation 40 for $r = 0, 1, 2 \ldots$.

(ii) Just as for the non-FIFO queue, in this case also, the optimal threshold is either lower than or equal to the greedy threshold. However, it is not intuitively obvious why this should be so.

The reason it is not obvious is the following. Suppose μ_1 and μ_2 satisfy

$$\frac{1}{\mu_1} = \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1 + \mu_2}\right)^{k-1} - \epsilon \qquad (41)$$

for some small $\epsilon > 0$. We will assume that ϵ is equal to 0 for all purposes except when determining the greedy threshold. The greedy algorithm gives a threshold of k in this case.

We try to look at the situation in which system A of the queue is in state (k, I). Let us compare the mean delay of the two systems: the original system that uses the greedy threshold k and the modified system that uses a threshold of $k-1$ in the beginning, sending customer number k to server 2 and then switching to the greedy threshold. We can look at the following different cases; case 3 below is the non-obvious one.

Case 1: If in the modified system, no service completion takes place at either server before the next arrival to the system, the decision to use a threshold of $k-1$ instead of k does not make any difference in mean delay because of the memoryless property of the service times.

Case 2: In the modified system, if a service completion takes place at server 2 before the next arrival to the queue, but no service completion takes place at server 1, the expected delay for the new arrival remains the same. This is because, in the modified system, the new customer will see $k-1$ customers in the system and therefore wait for server 1. Hence, it will experience a mean delay of $\frac{k}{\mu_1}$. In the original system, it would have seen k customers in front and would have gone to server 2. This would have made the mean delay for the new customer equal to $\frac{k-1}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)$. By assuming that $\epsilon = 0$ in equation 41 we conclude that the mean delay for the new customer remains unchanged in the modified system.

However, the number of customers in the modified system at the time of the new arrival is less than the number of customers in the original system. Since the two systems use the same threshold in the future, the average number of customers in the modified system will always be less than or equal to the corresponding number in the original system. This makes the modified system better.

Case 3: If one or more service completions take place at server 1, but none at server 2 before the next arrival to the system, then it is not clear whether it is better to use a threshold of $k-1$ or k. Depending upon the number of service completions at server 1, the modified system will have a higher or lower average delay for the future arrivals, compared to the original system. The appendix provides additional insight about this issue.

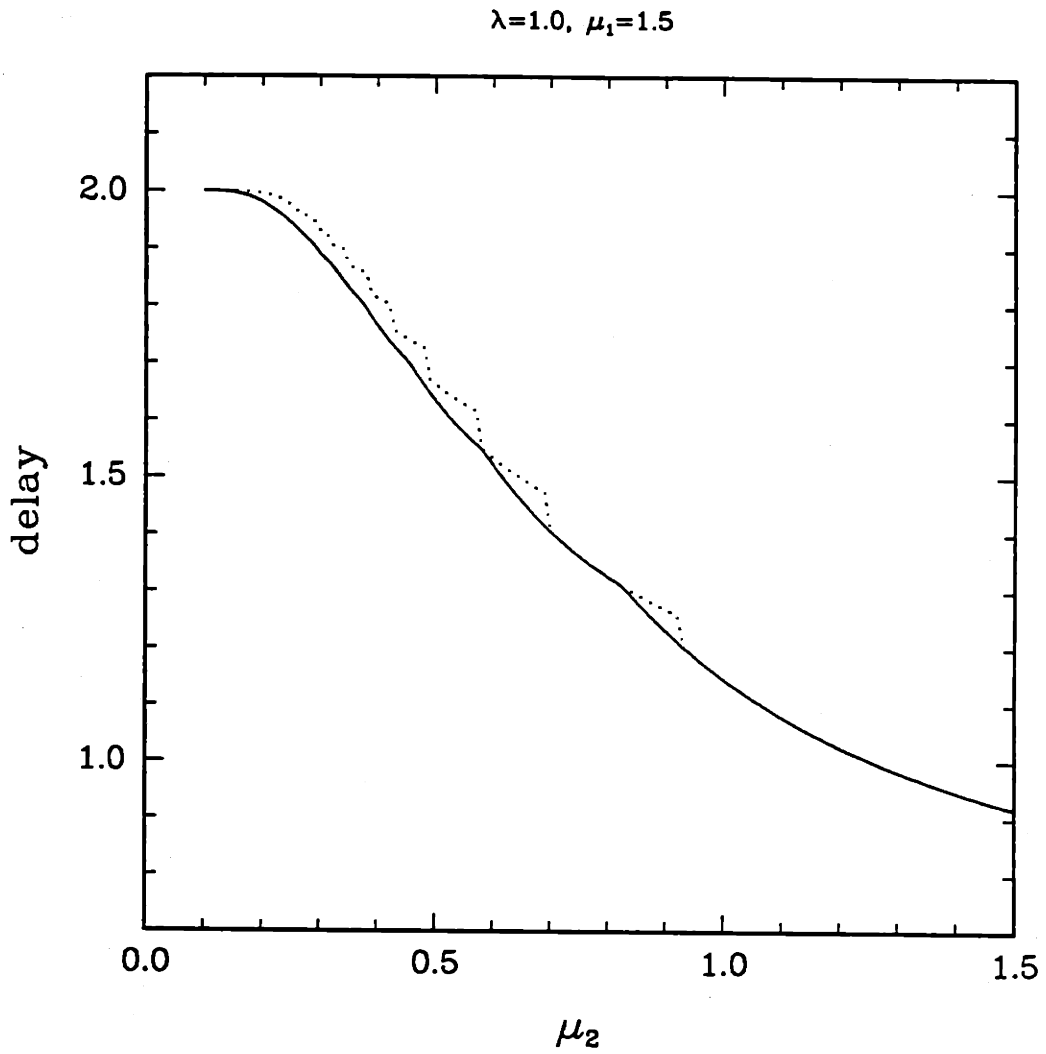

Figure 12: optimal and greedy threshold delays vs. μ_2 for $\mu_1 = 1.5$

Solid lines represent the delay for optimal threshold and the dotted lines represent the delay for the greedy threshold

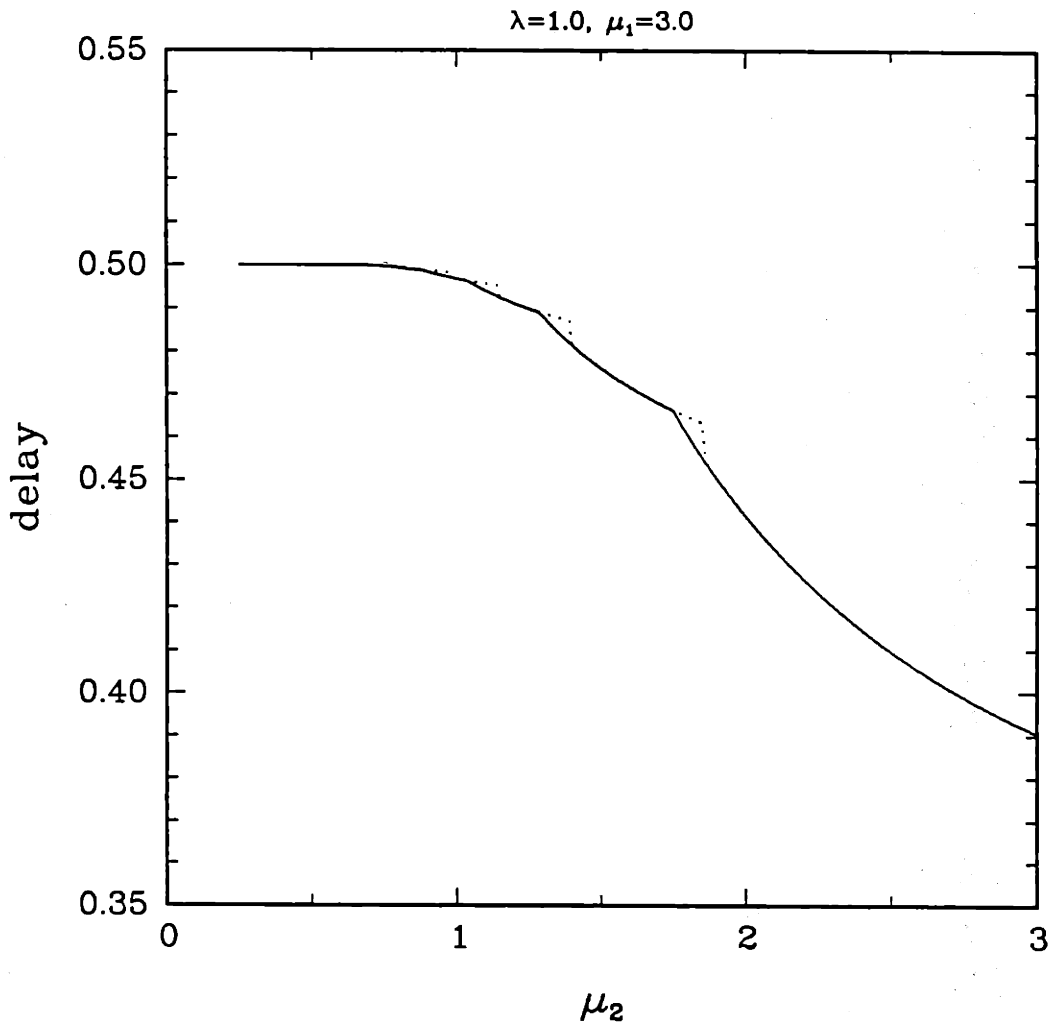

Figure 13: optimal and greedy threshold delays vs. μ_2 for $\mu_1 = 3.0$

Solid lines represent the delay for optimal threshold and the dotted lines represent the delay for the greedy threshold

Figures 12 and 13 plot the average delays for the greedy threshold and the optimal threshold vs. μ_2 for two different values of μ_1 for the non-FIFO queue. The regions of different optimal thresholds are delimited by the corner points of the delay curve for optimal threshold. As expected, the optimal threshold delay curve is continuous but the greedy threshold delay shows discontinuities at point where the greedy threshold differs from the optimal threshold.

7 Conclusion

We have presented the FIFO queueing model to analyze the overall delay of a packet, including the re-ordering delay, when more than one link is used to transmit different packets of a message.

A threshold strategy is proposed to use the links for the two link case. The optimal threshold is computed for the two server FIFO queue. Its performance is compared with a similar strategy for a non-FIFO queue. The optimal threshold is also compared with the threshold given by the intuitive *greedy algorithm.*

Appendix

Why should customer number $m+1$ go to server 2?

Let us assume that system A is operating at a threshold m. Let us look at the queue in system A when server 2 is idle and there are more than m customers in the system and let us number the customers starting from the customer receiving service at server 1. At this point server 2 has to take one of the customers into service.

Conjecture 1 *If it is better for server 2 to take customer number $i \leq m$ compared to taking customer number $m+1$ into service then m can not be the optimum threshold.*

Let us look at this situation as a Dynamic Programming problem. System A is in some particular state when server 2 is idle and a particular customer goes to server 2 to receive service according to some policy.

Let us look at the following two policies:

Policy 1: Customer number $m+1$ goes to server 2 to receive service.

Policy 2: Customer number $i \leq m$ goes to server 2 to receive service.

We know that if and only if one use of policy 2, followed by use of policy 1 all the time, is better than use of policy 1 all the time, then policy 2 is better than policy 1 in the long run [2]. Let us compare the expected delay of all the customers for the following two cases:

Case 1: One use of policy 2 followed by policy 1 all the time.

Case 2: Use of policy 1 all the time.

In both the cases, the delay for customer number 1 to $i-1$ will be distributed as $X_1, Y_1^2, Y_1^3 \ldots$ upto Y_1^{i-1} . In case 1, customer number i to m will have delays of $\max[X_2, Y_1^{i-1}]$ to $\max[X_2, Y_1^{m-1}]$ respectively, but in case 2, customer number i to m will have delays of Y_1^i to Y_1^m. In both the cases, customer number $m+1$ will have a delay of $\max[X_2, Y_1^m]$ and all the customers with numbers $j > m+1$ will have a delay of $Y_{12}^{j-m-1} + \max[X_2, Y_1^m]$. This means only customers with number i to m experience a difference in the delay in the two cases. The following table lists the mean delays of customer numbers 1 to m for the two different cases.

Customer no.	Mean Delay		Difference
	Case 1	Case 2	
1	$\frac{1}{\mu_1}$	$\frac{1}{\mu_1}$	0
2	$\frac{2}{\mu_1}$	$\frac{2}{\mu_1}$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i-1$	$\frac{i-1}{\mu_1}$	$\frac{i-1}{\mu_1}$	0
i	$\frac{i-1}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^{i-1}$	$\frac{i}{\mu_1}$	$\frac{1}{\mu_1} - \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^{i-1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	$\frac{m-1}{\mu_1} + \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^{m-1}$	$\frac{m}{\mu_1}$	$\frac{1}{\mu_1} - \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^{m-1}$

Let us now assume that case 1 is better than case 2. This means that the sum of all the differences should be positive.This implies that the difference for customer number m should be positive because it is the largest of all the differences listed above. This means that the greedy threshold for this system is less than or equal to $m-1$. But as can be seen from figure 11, the optimal threshold is always less than or equal to the greedy threshold.

Hence, m can not be the optimal threshold.

The above argument is rigorous except it relies on the optimal threshold being less than or equal to the greedy threshold.

We want to show that conjecture 1 also implies that the optimal threshold is less than or equal to the greedy threshold.

Let us suppose that the optimal threshold is greater than the greedy threshold. This means that for the optimal threshold m

$$\frac{1}{\mu_1} < \frac{1}{\mu_2}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^{m-1}$$

This means that all the terms in the difference column of the table on page 54 are negative, which in turn implies that case 1 is better than case 2. This implies that it is better for some customer number $i \leq m$ to go to server 2 which contradicts conjecture 1.

References

[1] Dimitri Bertsekas and Robert Gallager, *Data Networks*, Prentice-Hall, Inc., Englewood cliffs, New Jersey, 1987.

[2] Ronald A. Howard, *Dynamic Programming and Markov Processes*, Technology Press of Massachusetts Institute of Technology, Cambridge, 1960.

[3] R. L. Larsen and A. K. Agrawala, "Control of a Heterogeneous Two-Server Exponential Queueing System," *IEEE Transactions on Software Engineering*, Vol. SE-29, No. 4, pp. 522-526, July 1983.

[4] W. Lin and P. R. Kumar, "Optimal Control of a Queueing System with Two Heterogeneous Servers," *IEEE Transactions on Automatic Control*, Vol. AC-29, No. 8, pp. 696-703, August 1984.

[5] Ward Whitt, "Deciding Which Queue to Join: Some Counterexamples," *Operations Research*, Vol. 34, No. 1, pp. 55-62, January-February 1986.

[6] Wayne Winston, "Optimality of the Shortest Line Discipline," *Journal of Applied Probability*, Vol. 14, pp. 181-189, 1977.