THE UNIVERSITY OF CALGARY

THEORY AND SIMULATION OF FEEDBACK QUEUES WITH PRIORITY

by

RAJESWARI NEELAKANTAN

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF MATHEMATICS AND STATISTICS

CALGARY, ALBERTA

July, 1996

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "Theory and Simulation of Feedback Queues with Priority" submitted by Rajeswari Neelakantan in partial fulfillment of the requirements for the degree of Master of Science.

Chairman, Dr. Ernest G. Enns
Department of Mathematics and Statistics

Dr. P. Ehlers
Department of Mathematics and Statistics

Dr. K. Loose
Department of Computer Science

96/07/15

Abstract

A mathematical model is developed to describe a particular feedback queueing situation with priority and the solution is obtained using time dependent and Markov chain techniques. The characteristics of such a model are presented in this thesis.

Moreover the same queueing situation is simulated using the language S-PLUS and checked against the theoretical values and the results are presented.

Acknowledgements

The author would like to express genuine thanks and appreciation to her supervisor, Dr.Ernest G. Enns, for his guidance and sincere encouragement which has made this work a success.

The author expresses her indebtedness to his family members for their support.

The author also would like to thank Mr.Bruce Bauslaugh for helping her to explore the language S-PLUS.

Financial assistance provided by the University of Calgary is also thankfully acknowledged.

Table of Contents

Chapter 1

Introduction

Queueing situations arise in all aspects of work and life. In normal context, they mean to most of us delays which, though some-what unpleasant, can at least be tolerated. Basically, a queueing or waiting-line phenomenon is described by the following model: Units emanating from a source arrive at a service facility, wait if necessary, and depart after receiving service. So understanding queueing theory and its concepts is thus basic to all personnel concerned with service situations.

The basic elements of a queueing problem are

1) input process - the manner in which customers arrive.

2) queue discipline - the order in which customers are served.

3) service mechanism - the manner in which the queue is being resolved.

These three can vary as follows:

Input Process

The number of customers emanate from finite or infinite sources. Also, the customers may arrive at the service facility in batches of fixed size or of variable size or one by one. In the case when more than one arrival is allowed to enter the system simultaneously, the input is said to occur in batches.

If the customer decides not to enter the queue because of its huge length, he is said to have balked. On the other hand, a customer may enter the queue, but after sometime loses patience and decides to leave. In this case he is said to have reneged. In the case when there are two or more parallel queues, customers may move from one queue to another for their personal economic gain, that is jockey for position.

The Queue Discipline

It is a rule according to which customers are selected for service when a queue has been formed. The most common discipline is the "first come, first served" (FCFS) under which the customers are serviced in the strict order of their arrivals. Other queue disciplines include: "Last in First out" (LIFO) rule according to which the last arrival in the system is serviced first, "Selection for service in Random Order" (SIRO) rule according to which the arrivals are serviced randomly irrespective of their arrival times in the system. Customers may also be given priorities, so that a customer of priority higher than another is always served before the other. Priority service may be pre-emptive, in which case a service is interrupted if a customer of higher priorities arrives. Non-preemptive priority means a service is always completed before taking the next highest priority customer.

The Service Mechanism

This means the arrangement of server's facility to serve the customers. If there are infinite number of servers then all the customers are served instantaneously on arrival and there will be no queue.

If the number of servers are finite, then the customers are served according to a specific order. Further, the customers may be served in batches of fixed size or of variable size rather than individually by the same server. this service system is termed as bulk service system.

1.1 Thesis Objective

The objective of this thesis is to develope a mathematical model to describe a particular queueing situation, and to find a solution using time-dependent and Markov chain techniques. Further the same queueing situation is simulated and checked against the theoretical results.

1.2 Thesis Outline

1) Chapter 2, describes a particular queueing situation and a mathematical model that has been developed using time dependent techniques.

2) Chapter 3, describes a mathematical model and associated solutions that are obtained using imbedded Markov chain techniques.

3) Chapter 4, deals with the simulated results that are obtained using time dependent techniques as described in Chapter 2 using the language S_PLUS.

4) Chapter 5, deals with the simulated results that are obtained using imbedded Markov chain techniques as described in chapter three using the language S-PLUS.

5) Chapter 6, describes the variance reduction techniques that are applied to the simulation and the results are presented.

6) In the final chapter simulation results are compared with the theory.

Chapter 2

Single Channel With Priority

2.1 Introduction

In the mechanism of a queueing process, customers arrive at service counter and are attended by one or more of the servers. As soon as a customer is served, they depart from the system. Thus a queueing system can be described as composed of customers arriving for service, waiting for service if not immediate, and if having waited for service, leaving the system after being served.

Sometimes the queue discipline may be such that some types of customer receive priority. For example, the cost per unit time of keeping certain customers queueing may be particularly high and it may then be reasonable to give them priority.

Any priority discipline must therefore, specify the rules for making the following two decisions:

(1) Which unit to select for service once the server has completed a service.

(2) Whether to continue or discontinue the service of the unit being serviced.

The decision of selecting the next unit for service may depend only upon the knowledge of the priority class to which a unit belongs.

The priority system in which a customer, once at the service-point remains there until his service is complete is called the **non-preemptive priority.** In other words, if a customer of higher priority arrives when a customer of lower priority is being served,the server never interrupts the current service. The service principle is as follows:If there are customers in the system, upon the service completion the server chooses the customer who has arrived first among the customers of highest priority present in the system.

The serving principle for the **preemptive priority** is as follows:If there are customers in the system,the server is busy and is serving the customer who arrived first among the customers of highest priority in the system.

2.2 Statement Of The Problem

Assumptions:

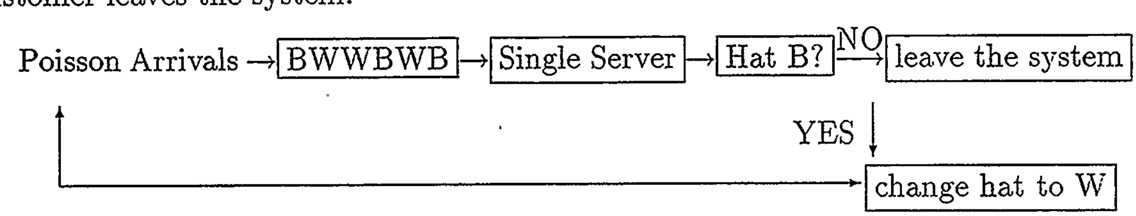
1 Poisson arrival

2. Exponential service

3. Single channel

4. FCFS queue discipline

5. Customers arrive in Poisson fashion at a single channel facility. As they come in they put on a black hat and join the queue. At the service counter they get served. The service time is defined by an exponential distribution. Upon completion of the service they put on a white hat and rejoin the queue. After the next service, the customer leaves the system.



2.3 Symbols And Notations

m = number of customers wearing black hats in the system.

n = number of customers wearing white hats in the system.

λ = arrival rate of customers.

μ = service rate of customers.

$\frac{\lambda}{\mu} = \rho$ = traffic intensity.

E_{xy} = the state in which there are x customers wearing black hats and y customers wearing white hats in the queueing system.

E_{00} = the state in which there are zero customers wearing black hats and zero customers wearing white hats in the queueing system.

$P_{mn}(t)$ = probability that there are exactly m customers wearing black hats and n customers wearing white hats in the system at time t.

$Q_{m_k n_k}$ = probability that there are exactly m customers wearing black hats and n customers wearing white hats in the queueing system at the completion of the kth service.

P_w = probability that there is exactly one customer wearing a white and zero customers wearing black hats in the system .

P_{ww} = probability that there are exactly two customers wearing white hats and zero customers wearing black hats in the system, where a customer wearing a white hat is at the service counter and a customer wearing a white hat is in the queue waiting for service.

P_{www} = probability that there are exactly three customers wearing white hats and zero customers wearing black hats in the system, where a customer wearing a white hat is at the service counter and the other two customers wearing white hats are waiting in the queue for their service.

P_{bb} = probability that there are exactly two customers wearing black hats in the system, where a customer wearing a black hat is at the service counter and a customer wearing a black hat is in the queue waiting for service.

P_{wb} = probability that there are exactly two customers in the system, one wearing a black hat and the other wearing a white hat, where the customer wearing a black hat is at the service counter and the customer wearing a white hat is waiting for service.

P_{wwb} = probability that there are exactly three customers in the system, one wearing a black hat and two wearing a white hat, where a customer wearing a black hat is at the service counter, the customer wearing a white hat is waiting for the next service, and other customer wearing white hat is waiting in the queue for his service.

P_{wbb} = probability that there are exactly three customers in the system, one wearing a white hat and two wearing a black hat, where a customer wearing a black hat is at the service counter, the customer wearing a black hat is waiting for the next service and the customer wearing a white hat is waiting behind for his service.

P_0 = probability that exactly zero customers are in the system = $P(E_{00})$.

E(m) = average number of customers wearing black hats waiting for service in the system.

E(n) = average number of customers wearing white hats waiting for service in the system.

V(m) = variance of the number of customers wearing black hats in the system.

V(n) = variance of the number of customers wearing white hats in the system.

$\overline{m}$ = average number of customers wearing black hats in the system using simulation.

$\overline{n}$ = average number of customers wearing white hats in the system using simulation.

S_m = sample variance in the number of customers wearing black hats in the system using simulation.

S_{m1} = sample variance in the number of customers wearing black hats in the system using variance reduction technique..

S_n = sample variance in the number of customers wearing white hats in the system using simulation.

S_{n1} = sample variance in the number of customers wearing white hats in the system using variance reduction technique.

% = percentage of variance reduction.

2.4 Model One

Customers are being served on the 'first come first served basis' (i.e.) Served customers with a black hat change to a white hat and rejoin the end of the queue.

To be in state E_{00} at time t+Δt, the system could have been in state E_{00} at time t and no arrivals during Δt. (no service since the system is empty), or the system could have been in E_{01} at time t and have no arrivals during Δt but one service completion during Δt. So we can easily see that,

$P_{00}(t+\Delta t) = P_{00}(t)P[\text{ no arrivals during } \Delta t] + P_w(t)P[\text{ one service during } \Delta t]$

This equation can be rewritten as:

$P_{00}(t+\Delta t) = P_{00}(t)\ [1-\lambda\Delta t + o(\Delta t)] + P_w(t)\ [\mu\Delta t + o(\Delta t)]$

Or in the limit,

$\dot{P}_{00} = -\lambda P_{00} + \mu P_w$ where $\dot{P}_{00} = \left(\frac{dP_{00}(t)}{dt}\right)$

Similarly,

$\dot{P}_w \ = -\lambda P_w - \mu P_w + \mu P_{ww}$

$\dot{P}_{ww} = -\lambda P_{ww} - \mu P_{ww} + \mu P_{www} + \mu P_{wb}$

$\dot{P}_{www} = -\lambda P_{www} - \mu P_{www} + \mu P_{wwww} + \mu P_{wwb}$

$\dot{P}_{wb} = -\lambda P_{wb} - \mu P_{wb} + \mu P_{bb}$

$\dot{P}_{wwb} = -\lambda P_{wwb} - \mu P_{wwb} + \mu P_{wbb}$

To find P_w we should know P_{ww}; to find P_{ww} we need P_{www} and P_{wb}. So it is not possible by known techniques to solve these difference equations and find a solution for this model. So we will take cases that are solvable using known techniques such as priority models.

2.5 Model Two

Customers wearing white hats have higher (non-preemptive) priority over customers wearing black hats.

Let

m_t = number of black hats worn by customers at time t.

n_t = number of white hats worn by customers at time t.

$P(m_t = m, n_t = n) = P_{mn}(t)$.

Let $P_{mn}(t)$ be the probability that there are m customers wearing black hats and n customers wearing white hats in the system at time t. Then to write the difference equation for P_{00} we first consider how the system can get to state E_{00} at time t+Δt. To be in state E_{00} at time $t+\Delta t$, the system could have been in E_{00} at time t and no arrivals during Δt. (no service since the system is empty), or the system can be in E_{01} at time t and have no arrivals but one service completion during Δt. Since arrival and service are independent of each other, we can easily see that,

$\dot{P}_{00} = -\lambda P_{00} + \mu P_{01}$

$\dot{P}_{m0} = -\lambda P_{m0} - \mu P_{m0} + \lambda P_{m-1,0} + \mu P_{m1}$

$\dot{P}_{01} = -\lambda P_{01} - \mu P_{01} + \mu P_{10}$

$\dot{P}_{m1} = -\lambda P_{m1} - \mu P_{m1} + \lambda P_{m-1,1} + \mu P_{m+1,0}$

To solve the differential-difference equation, we make use of the generating function defined as,

$$G_t(z_1, z_2) = \sum_{m=0}^{\infty} \sum_{n=0}^{1} P_{mn}(t) z_1^m z_2^n$$

On multiplying the above differential difference equations by the corresponding powers of z_i, i = 1,2 and taking the summation over m=0 to ∞ and n = 0 to 1 and using the generating function defined above, we get

$$\left(\frac{dG}{dt}\right) = -\lambda G - \mu G + \lambda z_1 G + \left(\frac{\mu}{z_2}\right) G + \mu P_{00} - \left(\frac{\mu}{z_2}\right) \sum_{m=0}^{\infty} P_{m0} z_1^m + \left(\frac{\mu z_2}{z_1}\right) \sum_{m=0}^{\infty} P_{m0} z_1^m - \left(\frac{\mu z_2}{z_1}\right) P_{00}$$

Note that after a sufficiently long period of time the state probabilities are independent of the initial conditions. Now the system is said to be in statistical equilibrium. And an important characteristic of this is that it is stationary; that is, the state probabilities do not vary with time.

So when $\lambda < \mu/2$ and t$\rightarrow \infty$, then $P_{mn}(t) \rightarrow P_{mn}$ and $\frac{d}{dt} P_{mn}(t) \rightarrow 0$.

When $(\frac{dG}{dt}) = 0$ then,

$$G = \left(\frac{P_{00} z_2 (z_1 - z_2) + (z_2^2 - z_1) \sum_{m=0}^{\infty} P_{m0} z_1^m}{\rho z_1 z_2 (1 - z_1) + z_1 (z_2 - 1)} \right)$$

When the denominator is zero then $z_2 = \left(\frac{1}{1+\rho(1-z_1)}\right)$ or $z_1 = 0$ where it is obvious that $z_2 < 1$, for all $0 < z_1 < 1$. Now by definition $G(z_1, z_2)$ is analytic and $0 \leq G(z_1, z_2) \leq 1$ for $0 \leq z_i \leq 1$. Hence by Rouche's theorem the numerator must vanish when denominator vanishes. Hence,

$$\sum_{m=0}^{\infty} P_{m0} z_1^m = P_{00} \frac{1 - z_1 \left[1 + \rho(1 - z_1)\right]}{1 - z_1 \left[1 + \rho(1 - z_1)\right]^2}$$

Therefore,

$$G(z_1, z_2) = \left(\frac{P_{00} \{ z_2 (z_1 - z_2)(1 - z_1[1 + \rho(1 - z_1)]^2) + (z_2^2 - z_1)(1 - z_1[1 + \rho(1 - z_1)]) \}}{z_1 (z_2 \rho (1 - z_1) + z_2 - 1)\ (1 - z_1[1 + \rho(1 - z_1)]^2)} \right)$$

2.6 Characteristics Of Model Two

1) Average number of customers wearing black hats in the system.

First by substituting $z_1 = z$ and $z_2 = 1$ in the generating function defined above we get,

$G(z, 1) = E(z^m) = \left(\dfrac{(1 - 2\rho)\ (1 + \rho(1 - z))}{1 - 2\rho z - \rho^2 z (1 - z)} \right)$

Now,

$E(m) = \left(\dfrac{\rho(1 + \rho)}{1 - 2\rho} \right)$

2) The fluctuation (variance) in the customers wearing black hats in the system.

$V(m) = \left(\dfrac{\rho + 2\rho^3 - 3\rho^4}{(1 - 2\rho)^2} \right)$

3) Average number of customers wearing white hats in the system.

By substituting $z_1 = 1$ and $z_2 = z$ we get,

$G(1, z) = E(z^n) = 1 - \rho + \rho z$

Now,

$E(n) = \rho$

4) The fluctuation in the customers wearing white hats in the system.

$V(n) = \rho - \rho^2$

5) Covariance

$Cov(m, n) = E(mn) - E(m)E(n) = \rho^2$

6) Correlation

$r(m,n) = \left(\frac{Cov(m,n)}{\sqrt{V(m)V(n)}}\right) = \left(\frac{\rho^2(1-2\rho)}{\sqrt{(\rho-\rho^2)(\rho+2\rho^3-3\rho^4)}}\right)$

where the range is $0 \leq r(m, n) \leq .1368$, for $0 < \rho < .5$

2.7 Model Three

Customers wearing black hats have higher (non-preemptive) priority than customers wearing white hats.

The differential difference equations for this model are as follows:

$\dot{P}_{00} = -\lambda P_{00} + \mu P_{01}$

$\dot{P}_{0n} = -\lambda P_{0n} - \mu P_{0n} + \mu P_{1,n-1} + \mu P_{0,n+1}$ $, n \geq 1$

$\dot{P}_{m0} = -\lambda P_{m0} - \mu P_{m0} + \lambda P_{m-1,0}, m \geq 1$

$\dot{P}_{mn} = -\lambda P_{mn} - \mu P_{mn} + \mu P_{m+1,n-1} + \lambda P_{m-1,n}, n \geq 1, m \geq 1$

To solve the above differential-difference equations we make use of the generating function define as

$$G_t(z_1, z_2) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} z_1^m z_2^n P_{mn}(t)$$

On multiplying the above differential difference equations by the corresponding powers of z_i i = 1,2 and taking the summation over n = 0 to ∞ and m = 0 to ∞ and also using the generating function defined above we get,

$$\left(\frac{dG}{dt}\right) = \lambda(z_1-1)G+\mu[\left(\frac{z_2}{z_1}\right)-1]G+\mu P_{00}\left[1-\left(\frac{1}{z_2}\right)\right]+\mu\sum_{n=0}^{\infty} z_2^n P_{0n}\left[\left(\frac{1}{z_2}\right)-\left(\frac{z_2}{z_1}\right)\right]$$

Note that after a sufficiently long period of time the state probabilities are independent of the initial conditions. Now the system is said to be in statistical equilibrium. And an important characteristic of this is that it is stationary; that is, the state probabilities do not vary with time.

So when $\lambda < \mu/2$ and t$\rightarrow \infty$, then $P_{mn}(t) \rightarrow P_{mn}$ and $\frac{d}{dt}P_{mn}(t) \rightarrow 0$.

When $(\frac{dG}{dt}) = 0$ then,

$$G(z_1,z_2) = \frac{P_{00}z_1\left[1-z_2\right]+\sum_{n=0}^{\infty} z_2^n P_{0n}[z_2^2-z_1]}{z_2[\rho z_1^2 - z_1(1+\rho)+z_2]}$$

When the denominator equals to zero then, $z_1 = \left(\frac{1+\rho-\sqrt{(1+\rho)^2-4\rho z_2}}{2\rho}\right)$ or $z_2 = 0$ where it is obvious that $z_1 < 1$ for all $0 < z_2 < 1$. Now by definition $G(z_1,z_2)$ is analytic and $0 \leq G(z_1,z_2) \leq 1$ for $0 \leq z_i \leq 1$. Hence by Rouche's theorem the numerator must vanish when the denominator vanishes. Hence,

$$\sum_{n=0}^{\infty} z_2^n P_{0n} = P_{00}(z_2-1)\left(\frac{[1+\rho]-\sqrt{(1+\rho)^2-4\rho z_2}}{2\rho z_2-(1+\rho)+\sqrt{(1+\rho)^2-4\rho z_2}}\right)$$

Therefore,

$$G(z_1,z_2) = \left(\frac{P_{00}\left[h(z_2)+2\rho z_1\right]\left[z_2-z_2^2\right]}{\left[\rho z_1^2-(1+\rho)z_1+z_2\right]\left[2\rho z_2^2+h(z_2)\right]}\right)$$

where $h(z_2) = \sqrt{(1+\rho)^2-4\rho z_2}-(1+\rho)$

2.8 Characteristics Of Model Three

1) Average number of customers wearing black hats in the system.

By substituting $z_1 = z, z_2 = 1$ we get,

$$G(z,1) = E(z^m) = \left(\frac{1-\rho}{1-\rho z}\right)$$

Therefore,

$$E(m) = \left(\frac{\rho}{1-\rho}\right)$$

2) The fluctuation in the customers wearing black hats in the system.

$$V(m) = \left(\frac{\rho}{(1-\rho)^2}\right)$$

3) Average number of customers wearing white hats in the system.

By substituting $z_1 = 1, z_2 = z$,we get,

$$G(1,z) = E(z^n) = \left(\frac{(1-2\rho)\ z\ (1-\rho-\sqrt{(1+\rho)^2-4\rho z})}{2\rho z^2 - 1 - \rho + \sqrt{(1+\rho)^2-4\rho z}}\right)$$

Now,

$$E(n) = \left(\frac{\rho+\rho^2}{1-3\rho+2\rho^2}\right)$$

4) The fluctuation in the customers wearing white hats in the system.

$$V(m) = \left(\frac{\rho\ (\rho^4 - 8\rho^3 + 2\rho^2 + 1)}{(1-\rho)\ (1-3\rho+2\rho^2)^2}\right)$$

5) Covariance

$$Cov(m,n) = E(mn) - E(m)E(n) = \left(\frac{\rho^2}{(1-\rho)^3}\right)$$

6) Correlation

$$r(m,n) = \left(\frac{Cov(m,n)}{\sqrt{V(m)V(n)}}\right) = \left(\frac{\rho(1-2\rho)}{\sqrt{1+\rho-2\rho^2-8\rho^3+9\rho^4-\rho^5}}\right)$$

where the range is $0 \leq r(m,n) \leq .1216$ for $0 < \rho < .5$.

2.9 Number In The System

In our system customers wearing black hats get served twice while customers wearing white hats get the service once.

Let K = number of services in the system = 2(number of customers wearing black hats) + (number of customers wearing white hats) = $2m + n$.

We can also think of K as the number of customers in a queue who arrive in batches of two where the customers within a batch are served one at a time, and, the service times of the customers are independent identically distributed random variables. This random variable K does not depend on any queue discipline.

For batch arrivals of two the differential difference equations are as follows:

$\dot{P}_0 = -\lambda P_0 + \mu P_1$

$\dot{P}_1 = -\lambda P_1 - \mu P_1 + \mu P_2$

$\dot{P}_2 = -\lambda P_2 - \mu P_2 + \lambda P_0 + \mu P_3$

Similarly

$\dot{P}_n = -\lambda P_n - \mu P_n + \lambda P_{n-2} + \mu P_{n+1}$

We define the generating function $G(z)$ as

$$G(z) = \sum_{n=0}^{\infty} z^n P_n$$

On multiplying the above defined differential difference equations by the corresponding powers of z and taking summation over $n = 0$ to ∞ and using the generating function defined above, we get $G(z) = \dfrac{1-2\rho}{1-\rho z-\rho z^2}$

In our system we have $K = 2m + n$.

Taking expectation we get, $E(z^k) = E(z^{2m+n}) = E(\left[z^2\right]^m z^n)$

Since the number in the system is not affected by the priority, substituting $z_1 = z^2$ and $z_2 = z$ in $G(z_1, z_2)$ both white hat priority and black hat priority yields

$$G(z^2, z) = \frac{1 - 2\rho}{1 - \rho z - \rho z^2}$$

as they should.

Now K $= 2m + n$

Therefore,

$E(K) = 2E(m) + E(n) = \left(\dfrac{3\rho}{1 - 2\rho}\right)$

$V(K) = 4V(m) + V(n) + 4COV(m, n) = \left(\dfrac{\rho(5 - \rho)}{(1 - 2\rho)^2}\right)$

So it is clear that both $E(K)$ and $V(K)$ are independent of the queue discipline.

Chapter 3

Imbedded Markov Chain

3.1 Introduction

Let us consider the system at the moment following the completion of a service. (say, the kth service completion by the server).

Let us define,

m_k = number of customers wearing a black hat in the queue at the completion of the kth service.

n_k = number of customers wearing a white hat in the queue at the completion of the kth service.

l = number of arrivals in the system during the (k+1)th service.

Now we have,

$m_{k+1} = m_k + l - \varepsilon_1$

$n_{k+1} = n_k + \varepsilon_2$

where ε_1 and ε_2 depend on the queue discipline.

Here m_{k+1} and n_{k+1} depends upon the current value m_k and n_k respectively and not upon any previous values. Moreover it behaves like a Markov chain at the completion of each service, so it is called the imbedded Markov chain.

3.2 Model One

Customers are being served on the 'first come and first served' basis. (i.e)Served customers with a black hat change to a white hat and rejoin the end of the queue.

We define the generating function

$$H_k(z_1, z_2) = \sum_{m_k=0}^{\infty} \sum_{n_k=0}^{\infty} z_1^{m_k} z_2^{n_k} Q_{m_k n_k}$$

When we assume **no priority** in our model we get the following table which gives the values for ε_1 *and* ε_2.

The table

m_k	n_k	ε_1	ε_2
0	0	0	1
+	0	1	1
0	+	0	-1
+	+		
+	+		

Since the model does not use all the information available in the queue, we can not fill the table. Therefore we can not solve this by this method.

3.3 Model Two

Customers wearing white hats have higher (non-preemptive) priority over customers wearing black hats.

The table

m_k	n_k	ε_1	ε_2
0	0	0	1
+	0	1	1
0	+	0	-1
+	+	0	-1

Define the generating function as

$$H_k(z_1,z_2) = \sum_{m_k=0}^{\infty} \sum_{n_k=0}^{1} z_1^{m_k} z_2^{n_k} Q_{m_k n_k}$$

Now,

$E[z_1^{m_{k+1}} z_2^{n_{k+1}}] = E[z_1^{m_k+l-\varepsilon_1} z_2^{n_k+\varepsilon_2}]$

Since l is independent of m_k and ε_1 we have,

$E[z_1^{m_{k+1}} z_2^{n_{k+1}}] = E[z_1^l]\{ \sum_{m_k=0}^{\infty} \sum_{n_k=0}^{1} z_1^{m_k-\varepsilon_1} z_2^{n_k+\varepsilon_2} Q_{m_k n_k}\}$

where $E[z_1^l] = \sum_{l=0}^{\infty} z_1^l \int_0^{\infty} \left(\frac{e^{-\lambda t}(\lambda t)^l \mu e^{-\mu t}}{l!} \right) dt = \left(\frac{1}{1+\rho-\rho z_1} \right)$

Therefore $E\left[z_1^{m_{k+1}} z_2^{n_{k+1}}\right] = \frac{1}{1+\rho-\rho z_1} \left\{ z_2 Q_{00} + \sum_{m_k=1}^{\infty} z_1^{m_k-1} z_2 Q_{m_k 0} + Q_{01} + z_1^{m_k} Q_{m_k 1} \right\}$

Now, $lim_{k\to\infty} E[z_1^{m_{k+1}} z_2^{n_{k+1}}] = lim_{k\to\infty} E[z_1^{m_k} z_2^{n_k}] = H(z_1,z_2)$, and $Q_{m_k n_k} \to Q_{mn}$ when $k \to \infty$.

Hence,

$H(z_1,z_2) = \left(\frac{Q_{00} z_2^2 (z_1 - 1) + \sum_{m=0}^{\infty} z_1^m Q_{m0}(z_2^2 - z_1)}{z_1(z_2-1) + \rho z_1 z_2 (1 - z_1)} \right)$

When the denominator is zero then $z_2 = \left(\frac{1}{1+\rho(1-z_1)} \right)$, where it obvious that $z_2 < 1$ for all $0 < z_1 < 1$. Now by definition $H(z_1,z_2)$ is analytic and $0 \leq H(z_1,z_2) \leq 1$ for $0 \leq z_i \leq 1$. Hence by Rouche's theorem the numerator must vanish when the denominator vanishes.Hence,

$$\sum_{m=0}^{\infty} z_1^m Q_{m0} = \left(\frac{Q_{00}(1-z_1)}{1-z_1(1+\rho(1-z_1))^2}\right)$$

Therefore,

$$H(z_1,z_2) = \left(\frac{Q_{00}\{(z_2^2 - z_1 - z_2^2(1-2\rho z_1 - \rho^2 z_1(1-z_1))) + z_1 z_2^2(1-2\rho z_1 - \rho^2 z_1(1-z_1))\}}{(z_1(z_2-1)+\rho z_1 z_2\,(1-z_1))\,(1-2\rho z_1 - \rho^2 z_1(1-z_1))\,.}\right)$$

3.4 Characteristics Of Model Two

1) Average number of customers wearing black hats in the system.

By substituting $z_1 = z, z_2 = 1$ in $H(z_1, z_2)$ we get,

$H(z,1) = E(z^m) = \left(\frac{(1-2\rho)\,(2\rho z + \rho^2 z(1-z))}{2\rho z\,(1-2\rho z - \rho^2 z\,(1-z))}\right)$

Therefore,

$E(m) = \left(\frac{3\rho}{2\,(1-2\rho)}\right)$

2) The fluctuation in the customers wearing black hats in the system.

$V(m) = \left(\frac{\rho\,(4\rho^2 - 5\rho + 6)}{4\,(1-2\rho)^2}\right)$

3) Average number of customers wearing white hats in the system.

By substituting $z_1 = 1, z_2 = z$ in $H(z_1, z_2)$ we get,

$H(1,z) = E(z^n) = \left(\frac{z+1}{2}\right)$

Therefore,

$E(n) = \left(\frac{1}{2}\right)$

4) The fluctuation in the customers wearing white hats in the system.

$V(n) = \left(\frac{1}{2}\right)$

5) Covariance

$Cov(m,n) = E(mn) - E(m)E(n) = \left(\frac{-\rho}{4}\right)$

6) Correlation

$$r(m,n) = \left(\frac{Cov(m,n)}{\sqrt{V(m)V(n)}}\right) = \left(\frac{-\rho(1-2\rho)\sqrt{2}}{2\sqrt{6\rho-5\rho^2+4\rho^3}}\right)$$

where the range is $0 \leq r(m,n) \leq -.0415$ for $0 < \rho < .5$.

3.5 Model Three

Customers wearing black hats have higher (non-preemptive) priority than customers wearing white hats.

The table

m_k	n_k	ε_1	ε_2
0	0	0	1
+	0	1	1
0	+	0	-1
+	+	1	1

We define the generating function $H(z_1, z_2)$ as

$$H_k(z_1, z_2) = \sum_{m_k=0}^{\infty}\sum_{n_k=0}^{\infty} z_1^{m_k} z_2^{n_k} Q_{m_k n_k}$$

Now,

$$E[z_1^{m_{k+1}} z_2^{n_{k+1}}] = E[z_1^{m_k+l-\varepsilon_1} z_2^{n_k+\varepsilon_2}]$$

Since l is independent of m_k and ε_1 we have,

$$E[z_1^{m_{k+1}} z_2^{n_{k+1}}] = E[z_1^l].[\sum_{m_k=0}^{\infty}\sum_{n_k=0}^{\infty} z_1^{m_k-\varepsilon_1} z_2^{n_k+\varepsilon_2} Q_{m_k n_k}]$$

$$= \left(\frac{1}{1+\rho-\rho z_1}\right)\left[z_2 Q_{00} + z_2 \sum_{m_k=1}^{\infty} z_1^{m_k-1} Q_{m_k 0} + \sum_{n_k=1}^{\infty} z_2^{n_k-1} Q_{0 n_k} + \sum_{m_k=1}^{\infty}\sum_{n_k=1}^{\infty} z_1^{m_k-1} z_2^{n_k-1} Q_{m_k n_k}\right.$$

Now $lim_{k\to\infty} E[z_1^{m_{k+1}} z_2^{n_{k+1}}] = lim_{k\to\infty} E[z_1^{m_k} z_2^{n_k}] = H(z_1, z_2)$, and when $k \to \infty$, $Q_{m_k n_k} \to Q_{mn}$.

Therefore,
$H(z_1, z_2) = \left(\frac{Q_{00}z_1(z_2^2-1) + \sum_{n=0}^{\infty} z_2^n Q_{0n}(z_1 - z_2^2)}{(1+\rho-\rho z_1)z_1 z_2 - z_2^2}\right)$
When the denominator is zero we get,
$z_1 = \left(\frac{1+\rho-\sqrt{(1+\rho)^2 - 4\rho z_2}}{2\rho}\right)$ where it obvious that $z_1 < 1$ for all $0 < z_2 < 1$.Now by definition $H(z_1, z_2)$ is analytic and $0 \le H(z_1, z_2) \le 1$ for $0 \le z_i \le 1$. Hence by Rouche's theorem the numerator must vanish when the denominator vanishes. Hence,

$$\sum_{n=0}^{\infty} z_2^n Q_{0n} = Q_{00}(z_2^2 - 1)\left(\frac{1+\rho-\sqrt{(1+\rho)^2 - 4\rho z_2}}{2z_2^2\rho - 1 - \rho + \sqrt{(1+\rho)^2 - 4\rho z_2}}\right)$$

Therefore,

$$H(z_1, z_2) = \left(\frac{Q_{00}\left(2\rho z_1 - 1 - \rho + \sqrt{(1+\rho)^2 - 4\rho z_2}(z_2 - z_2^3)\right)}{(\rho z_1^2 - z_1(1+\rho) + z_2)\left(2\rho z_2^2 - 1 - \rho + \sqrt{(1+\rho)^2 - 4\rho z_2}\right)}\right)$$

3.6 Characteristics Of Model Three

1) Average number of customers wearing black hats in the system.

By substituting $z_1 = z, z_2 = 1$ in $H(z_1, z_2)$we get,
$H(z, 1) = E(z^m) = \left(\frac{(1-\rho)}{(1-\rho z)}\right)$
Now,
$E(m) = \left(\frac{\rho}{1-\rho}\right)$
2) The fluctuation in the number of customers wearing black hats in the system.
$V(m) = \left(\frac{\rho}{(1-\rho)^2}\right)$
3) Average number of customers wearing white hats in the system.

By substituting $z_1 = 1, z_2 = z$ in $H(z_1, z_2)$ we get,

$$H(1,z) = E(z^n) = \left(\frac{(1-2\rho)z(z+1)\{1-\rho-\sqrt{(1+\rho)^2-4\rho z}\}}{2\{2\rho z^2-1-\rho+\sqrt{(1+\rho)^2-4\rho z}\}}\right)$$

Now,

$$E(n) = \left(\frac{1-\rho+4\rho^2}{2-6\rho+4\rho^2}\right)$$

4) The fluctuation in the number of customers wearing white hats in the system.

$$V(n) = \left(\frac{1-3\rho+19\rho^2-17\rho^3-16\rho^4}{4\ (1-\rho)\ (2\rho^2-3\rho+1)^2}\right)$$

5) Covariance

$$Cov(m,n) = E(mn) - E(m)E(n) = \left(\frac{\rho^2}{(1-\rho)^3}\right)$$

6) Correlation

$$r(m,n) = \left(\frac{Cov(m,n)}{\sqrt{V(m)V(n)}}\right) = \left(\frac{2\rho^2(1-2\rho)}{\sqrt{\rho-4\rho^2+22\rho^3-36\rho^4+\rho^5+16\rho^6}}\right)$$

where the range is $0 \leq r(m,n) \leq .1422$ for $0 < \rho < .5$.

Chapter 4

Simulation For The Time Dependent Model

4.1 Introduction

Simulation is a word with which many people are familiar, at least in a general way. Experimentation of the real system, simulation and mathematical modeling are the set of three alternatives that can be used in problem solving. When we experiment with the real system realism can be achieved, however the real system must exist before experiments can be performed on it, whereas the objective might be to design a system that does not yet exist. When we consider mathematical modeling, it involves a high degree of abstraction. The disadvantage of mathematical modeling is that it can require a relatively high level of mathematical sophistication on the part of the problem solver. Now consider simulation, which we regard as "an experiment in which we attempt to understand how something will behave in reality by imitating its behavior in an artificial environment that approximates reality as closely as possible". Between the two extremes of real-system experimentation and mathematical modeling, simulation has some of advantages and disadvantages of the two extremes. Some of the advantages of simulation are time compression, non-existent systems and training. The main disadvantage is the cost for providing a simulation capability.

To simulate the mathematical model in chapter two, we use the language S-PLUS. S-PLUS is useful for data analysis and graphics. It encourages to look at the data, and program interactively, with quick feedback to learn and understand. It is useful in financial analysis, statistical research and also for analytical computing and data analysis.

Simulating the equivalent of eight hours of real-system operation on a computer with different values of ρ, and doing 500 simulation runs, the results are given below.

The equivalent of eight hours of real-system operation can be simulated in five minutes on a computer.

4.2 Model One

Customers are being served on the 'first come first served basis'.

Even though it is not possible by standard methods to develop a mathematical model for this case it is quite easy to simulate. The program for this case is given in appendix A.

4.3 Results

(a) The mean, variance and covariances for the number of white and black hats in the system.

ρ	$\overline{m}$	S_m	$\overline{n}$	S_n	$COV(mn)$
.1	.1132	.1284	.1333	.1523	.0093
.2	.3521	.4273	.4458	.3858	.0457
.3	.8971	.9821	.8746	1.1721	.0922
.4	2.589	3.3656	1.9211	5.8302	.1849

(b) 90 % confidence interval for the mean using simulation.

ρ	.1	.2	.3	.4
$E(m)$	[.0868, .1396]	[.3040, .4002]	[.8242, .9700]	[2.4540, 2.7240]
$E(n)$	[.1046, .1620]	[.4001, .4915]	[.7950, .9542]	[1.7447, 2.0987]

4.4 Model Two

Customers wearing white hats having higher priority over customers wearing black hats.

Simulation program for this is given in appendix B.

4.5 Results

(a) The mean, variance and the covariances for the number of white and black hats in the system using simulation.

ρ	$\overline{m}$	S_m	$\overline{n}$	S_n	$COV(mn)$
.1	.1237	.1348	.1034	.0983	.0096
.2	.3986	.4679	.2151	.1458	.0399
.3	.9082	2.0021	.3021	.1986	.0743
.4	2.6678	8.7516	.4126	.2253	.1489

(b) Results from the theory.

ρ	$E(m)$	$V(m)$	$E(n)$	$V(n)$	$COV(mn)$
.1	.1375	.1589	.1	.09	.01
.2	.4	.5867	.2	.16	.04
.3	.975	2.0606	.3	.21	.09
.4	2.8	11.28	.4	.24	.16

(c) 90% confidence intervals for the mean.

ρ	.1	.2	.3	.4
$E(m)$	[.0967,.1507]	[.3483,.4489]	[.8041,1.0123]	[2.4502,2.8854]
$E(n)$	[.0803,.1265]	[.1870,.2432]	[.2693,.3349]	[.3377,.4475]

4.6 Model Three

Customers wearing black hats have higher priority than customers wearing white hats.

Simulation program for this model is given in appendix C.

4.7 Results

(a) The mean, variance and the covariances for the number of white and black hats in the system using simulation.

ρ	$\overline{m}$	S_m	$\overline{n}$	S_n	$COV(mn)$
.1	.1048	.1046	.1436	.1998	.0092
.2	.265	.39	.5243	1.3203	.0768
.3	.412	.5879	1.3283	5.2420	.2181
.4	.6862	.9987	5.0226	34.5108	.5524

(b) Results from the theory.

ρ	$E(m)$	$V(m)$	$E(n)$	$V(n)$	$COV(mn)$
.1	.1111	.1235	.1528	.2169	.0137
.2	.25	.3125	.5	1.1042	.0781
.3	.4286	.6122	1.3929	5.312	.2624
.4	.6667	1.1111	4.6669	38.5926	.7407

(c) 90% confidence interval for the mean.

ρ	.1	.2	.3	.4
$E(m)$	[.081,.1286]	[.2191,.3109]	[.3556,.4684]	[.6125,.7595]
$E(n)$	[.1107,.1765]	[.4398,.6088]	[1.1599,1.4967]	[4.5904,5.4548]

In each case the confidence interval spans the theoretical value.

Chapter 5

Simulation For The Markov Chain Model

5.1 Introduction

We simulate the model given in chapter three using S-PLUS. Simulating this 1000 times for different values of ρ the results are given below.

We have,

$m_{k+1} = m_k + l - \varepsilon_1$

$n_{k+1} = n_k + \varepsilon_2$

5.2 Model One

Customers wearing white hats have higher priority over customers wearing black hats.

The simulation program for this case is given in appendix D.

5.3 Results

(a) Mean and variance and covariances for the number of white and black hats in the system for different values of ρ using simulation.

ρ	$\overline{m}$	S_m	$\overline{n}$	S_n	$COV(mn)$
.1	.185	.1930	.5	.2503	−.0214
.2	.461	.6182	.5	.2503	−.0502
.3	1.058	2.2388	.5	.2503	−.0685
.4	2.856	9.2617	.5	.2503	−.7982

(b) Results from the theory.

ρ	$E(m)$	$V(m)$	$E(n)$	$V(n)$	$COV(mn)$
.1	.1875	.2164	.5	.5	−.025
.2	.5	.7167	.5	.5	−.05
.3	1.125	2.2781	.5	.5	−.075
.4	3	11.6	.5	.5	−.1

(c) 90 % confidence intervals for the mean.

ρ	.1	.2	.3	.4
$E(m)$	[.1621,.2079]	[.42,.502]	[.9802,1.1358]	[2.6977,3.0143]
$E(n)$	[.474,.526]	[.474,.526]	[.474,.526]	[.474,.526]

5.4 Model Two

Customers wearing black hats have higher priority than customers wearing white hats.

The program for this case is given in appendix E.

5.5 Results

(a) Mean, variance and covariances for the number of white and black hats in the system for different values of ρ using simulation.

ρ	$\overline{m}$	S_m	$\overline{n}$	S_n	$COV(mn)$
.1	.102	.1037	.622	.3935	.0086
.2	.279	.3912	1.054	1.5040	.0682
.3	.409	.5843	1.85	4.4059	.2186
.4	.683	.9955	5.4114	32.996	.5041

(b) Results from the theory.

ρ	$E(m)$	$V(m)$	$E(n)$	$V(n)$	$COV(mn)$
.1	.111	.1235	.6528	.4669	.0137
.2	.25	.3125	1	1.3542	.0781
.3	.4286	.6122	1.8929	5.5640	.2624
.4	.6667	1.111	5.1667	38.7977	.7407

(c) 90 % confidence intervals for the mean.

ρ	.1	.2	.3	.4
$E(m)$	[.0852,.1188]	[.2465,.3115]	[.3692,.4488]	[.6311,.7349]
$E(n)$	[.5894,.6546]	[.9902,1.1178]	[1.7408,1.9592]	[5.1126,5.7102]

In each case the confidence interval spans the theoretical value.

Chapter 6

VARIANCE REDUCTION TECHNIQUE

6.1 Introduction

Suppose we have generated U_1 and U_2, identically distributed random variables, then

Var$\left(\frac{U_1+U_2}{2}\right) = \left(\frac{Var(U_1)+Var(U_2)+2Cov(U_1,U_2)}{4}\right)$

Now the variance would be reduced if U_1 and U_2 are negatively correlated.

Suppose that U_1 is a function is given as, $U_1 = g(V_1, ..., V_k)$ where $V_1, ..., V_k$ are k independent random numbers. If V is a random number that is uniformly distributed on (0,1) - then so is 1-V. Hence the random variable $U_2 = g(1 - V_1, ..., 1 - V_k)$ has the same distribution as U_1. Moreover 1-V is negatively correlated with V so U_1 will be negatively correlated with U_2 when "g" is a monotonic function in most of its variables. To compute U_1, we first generate $V_1, ..., V_k$ then to generate U_2 we just use the set $1 - V_1, ..., 1 - V_k$. By doing this we not only save the time but also the resulting estimator will have smaller variance.

6.2 Variance Reduction Technique For The Markov Chain Model

Simulating the model for 1000 times with different values of ρ, the results are given below.

Model One *Customers wearing white hats having higher priority over customers wearing black hats.*

The program for this is given in appendix F.

ρ	$V(m)$	S_m	S_{m1}	%
.1	.2164	.1930	.0920	52.3
.2	.7167	.6182	.3396	45.1
.3	2.2781	2.3817	.6018	73.1
.4	11.6	9.2617	3.829	58.7

ρ	$V(n)$	S_n	S_{n1}	%
.1	.5	.2503	.2503	0
.2	.5	.2503	.2503	0
.3	.5	.2503	.2503	0
.4	.5	.2503	.2503	0

Model Two *Customers wearing black hats have higher priority than customers wearing white hats.*

The program for this given in appendix G.

ρ	$V(m)$	S_m	S_{m1}	%
.1	.1235	.1037	.0405	61
.2	.3125	.3912	.0997	74.5
.3	.6122	.5843	.2599	55.5
.4	1.111	.9955	.3974	60.1

ρ	$V(n)$	S_n	S_{n1}	%
.1	.4669	.3935	.2803	28.8
.2	1.3542	1.5040	.6706	55.4
.3	5.5645	4.4059	2.3784	46
.4	38.7977	32.996	5.7581	82.5

6.3 Variance Reduction Technique For The Time Dependent Model

Simulating the model for eight hours with different values of ρ and 500 runs the results are given below.

Model one Customers wearing white hats have higher priority over customers wearing black hats.

The program for this simulation is given in appendix H.

ρ	$V(m)$	S_m	S_{m1}	%
.1	.1589	.1348	.0427	68.3
.2	.5867	.4679	.2119	54.7
.3	2.0606	2.0021	.4351	78.3
.4	11.28	8.7516	1.8613	78.7

ρ	$V(n)$	S_n	S_{n1}	%
.1	.09	.0983	.0362	63.2
.2	.16	.1458	.0783	46.3
.3	.21	.1986	.0982	50.1
.4	.24	.2253	.1068	52.6

Model Two Customers wearing black hats have higher priority than customers wearing white hats.

The program for this case is given in appendix I.

ρ	$V(m)$	S_m	S_{m1}	%
.1	.1235	.1046	.0321	69.3
.2	.3125	.390	.1002	74.3
.3	.6122	.5879	.2947	49.9
.4	1.1111	.9987	.3316	66.8

ρ	$V(n)$	S_n	S_{n1}	%
.1	.2169	.1998	.0673	66.3
.2	1.1042	1.3203	.5028	61.9
.3	5.312	5.2420	1.2374	76.4
.4	38.5926	34.5108	5.2897	84.7

Chapter 7

VERIFICATION OF THE RESULTS

7.1 Introduction

In this chapter we verify that the results for the theory and simulation are close to each other by showing that the 90 % confidence interval for the means spans the theoretical value..

7.2 Model One

7.3 Time Dependent Technique

(a) Customers wearing white hats have higher priority over customers wearing black hats.

ρ	$E(m)$	$\overline{m}$	$E(n)$	$\overline{n}$	90% C.I for $E(m)$	90% C.I for $E(n)$
.1	.1375	.1237	.1	.1034	[.0967,.1597]	[.0803,.1265]
.2	.4	.3986	.2	.2151	[.3483,.4489]	[.1870,.2432]
.3	.975	.9082	.3	.3021	[.8041,1.0123]	[.2693,.3349]
.4	2.8	2.6678	.4	.4126	[2.4502,2.8854]	[.3777,.4475]

(b) Customers wearing black hats have higher priority over customers wearing white hats.

ρ	$E(m)$	$\overline{m}$	$E(n)$	$\overline{n}$	90% C.I for $E(m)$	90% C.I for $E(n)$
.1	.1111	.1048	.1528	.1436	[.0810,.1286]	[.1107,.1765]
.2	.25	.265	.5	.5243	[.2191,.3109]	[.4398,.6088]
.3	.4286	.412	1.3929	1.3283	[.3556,.4684]	[1.1599,1.4967]
.4	.6667	.6862	4.6669	5.0226	[.6125,.7595]	[4.5904,5.4548]

7.4 Model Two

7.5 Imbedded Markov Chain Technique

(a) Customers wearing white hats having higher priority than customers wearing black hats.

ρ	$E(m)$	$\overline{m}$	$E(n)$	$\overline{n}$	90% C.I for $E(m)$	90% C.I for $E(n)$
.1	.1875	.185	.5	.5	[.1621,.2079]	[.474,.526]
.2	.5	.461	.5	.5	[.4200,.5020]	[.474,.526]
.3	1.125	1.038	.5	.5	[.9602,1.116]	[.474,.526]
.4	3	2.856	.5	.5	[2.6977,3.0143]	[.474,.526]

(b) Customers wearing black hats have higher priority over customers wearing white hats.

ρ	$E(m)$	$\overline{m}$	$E(n)$	$\overline{n}$	90% C.I for $E(m)$	90% C.I for $E(n)$
.1	.111	.102	.6528	.622	[.0852,.1188]	[.5894,.6546]
.2	.25	.279	1	1.054	[.2465,.3115]	[.9902,1.1178]
.3	.4286	.409	1.8929	1.850	[.3692,.4488]	[1.7408,1.9592]
.4	.6666	.683	5.1667	5.4114	[.6311,.7349]	[5.1126,5.7102]

The above tables show that the 90 % confidence intervals span the theoretical means.

7.6 Conclusions

Mathematical models have been presented in this thesis for a particular queueing problem using time dependent and Markov chain techniques. Since it was not possible to find a solution by known techniques for both time dependent and Markov chain model for the assumption that the customers are serviced in the strict order of their arrivals, a solution was determined using a certain type of priority condition. The characteristics of such a model have been presented in this Thesis.

The queueing situation using priority conditions was simulated using the language S-PLUS and verified that the 90 % confidence interval spans the theoretical means for all cases.

Finally I do not understand the difference in the variances $V(n)$ and S_n for the white hat priority in the Markov chain model.

Directions Of Future Research

The following changes can be implemented in the queueing situation.

1. Instead of changing hats twice from black to white, a customer can change his hats thrice from black to white hat and then to a blue hat, and upon completion of the service leave the system.

2. Instead of a nonpreemptive priority one can assume a preemptive priority in the model.

3. Research new methods that may allow us to solve the FCFS model theoreti-

cally.

Appendix A

The simulation program for the time dependent model where customers are being served on the "first come first served basis".

Notation

arri.time = arrival time

ser.time = service time

stime = simulation time

ctime = current time

num = total number of simulation runs

vector("numeric",num) = represents a vector which contains numeric values and the size of the vector is determined by the input value "num".

rexp(1,arri.rate) = generate an exponential random variable with rate λ = arri.rate

append (Q,0) = attach zero to the vector Q.

Q[−1] = except the first element of the vector Q.

Input

The main function "simu" takes arrival time, service time, total simulation time and number of simulation we want to do as an input. Every time it calls the subroutine "queue" iteratively. The function "queue" takes arrival rate, service rate and the total simulation time as an input.

Output

It outputs the number of black hats and the number of white hats in a vector form with number of black hats in the first column and the number of white hats in the second column. We then assign the first column to a variable and the second column to another variable. We can find the mean and variances of the number of black and white hats in the system [(i.e) E(m),V(m),E(n) and V(n)] by using the commands summary( ) and var( ). To find E(mn), we should multiply the columns and then use the command summary( ). From that we can the covariance.

<u>Program</u>

<u>Function "simu"</u>

```
function(arri.time,serv.time,stime,num)
{
arri.rate ← 1/arri.time
serv.rate ← 1/serv.time
B ← vector("numeric",num)
W ← vector("numeric",num)
for(i in 1: num)
{
A ← queue(arri.rate,serv.rate,stime)
B[i] ← A[1]
W[i] ← A[2]
}
cbind(B,W)
}
```

<u>Sub-routine "queue"</u>

```
function(arri.rate,serv.rate,stime)
```

```
{
Q ← vector("numeric",0)
ctime ← 0
arrive ← rexp(1,arri.rate)
serve ← stime+1
while(ctime + arrive < stime || ctime + serve < stime)
{
if (length(Q) = = 0)
{
ctime ← ctime + arrive
arrive ← rexp(1,arri.rate)
serve ← rexp(1,serv.rate)
Q ← appened(Q,0)
print("B arrives")
}
else
{
        if (arrive < serve )
        {
        ctime ← ctime + arive
        serve ← serve − arrive
        arrive ← rexp(1,arri.rate)
        Q ← appened(Q,0)
        print("B arrives")
```

```
        }
        else
        {
        ctime ← ctime + serve
        arrive ← arrive − serve
        serve ← rexp(1,serv.rate)
        if(Q[1] = = 0)
        {
        Q ← appened(Q[−1],1)
        print("B served")
        }
        else
        {
        Q ← Q[−1]
        print("W served")
        }
}
}
c(length(Q[Q = = 0] ), length(Q[Q = = 1] ) )
}
```

Appendix B

The simulation program for the time dependent case where customers wearing white hats have higher priority over customers wearing black hats.

Notation

arri.time = arrival time

ser.time = service time

stime = simulation time

ctime = current time

num = total number of simulation runs.

numB = number of customers wearing black hats

numW = number of customers wearing white hats

nex = represents the next customer who is going to get the service

A[1] = first element of A.

cbind = column bind.

rexp(1,arri.time) = generate an exponential random variable with rate λ = arri.rate.

Input

The function "simu" takes arrival time, service time, total simulation time and number of simulation we want to do as an input Every time it calls the sub-routine "queuew" iteratively. The function "queuew" takes arrival rate, service rate and the total simulation time as an input.

Output

It outputs the number of black hats and the number of white hats in a vector form with number of black hats in the first column and the number of white hats in the second column. We then assign those two columns to two different variables. We can find the mean and variances of the number of black hats and white hats in the system [(i.e) E(m),E(n),V(m)and V(n)] by using the commands summary( ) and var( ). To find E(mn), we multiply the columns and then use the command summary( ). From that we can find the covariance.

Program

Function "simu"

```
function(arri.time,serv.time,stime,num)
{
arri.rate ← 1/arri.time
serv.rate ← 1/serv.time
B ← vector("numeric",num)
W ← vector("numeric",num)
for(i in 1: num)
{
A ← queuew(arri.rate,serv.rate,stime)
B[i] ← A[1]
W[i] ←A[2]
}
cbind(B,W)
}
```

Sub-routine "queuew"

```
function(arri.rate,serv.rate,stime)
{
numB ← 0
numW ← 0
ctime ← 0
arrive ← rexp(1,arri.rate)
serve ← stime+1
while(ctime + arrive < stime || ctime + serve < stime)
{
if (numB + numW = = 0)
{
ctime ← ctime + arrive
arrive ← rexp(1,arri.rate)
serve ← rexp(1,serv.rate)
numB ← 1
nex ← "B"
print("B arrives")
}
else
{
          if (arrive < serve )
          {
          ctime ← ctime + arrive
```

```
serve ← serve − arrive
arrive ← rexp(1,arri.rate)
numB ← numB + 1
print("B arrives")
}
else
{
ctime ← ctime + serve
arrive ← arrive − serve
serve ← rexp(1,serv.rate)
if (nex = = "B")
{
numB ← numB −1
numW ← numW +1
print("B served")
}
else
{
numW ← numW −1
print("W served")
}
if (numW > 0)
nex ← "W"
else nex ← "B"
```

```
            }
    }
    }
    }
    c(numB,numW )
    }
```

Appendix C

The simulation program for the time dependent case where customers wearing black hats have

higher priority than a customers wearing white hats.

<u>Notation</u>

arri.time = arrival time

ser.time = service time

stime = simulation time

ctime = current time

num = total number of simulation runs.

numB = number of customers wearing black hats

numW = number of customers wearing white hats

nex = represents the next customer who is going to get the service.

A[1] = first element of A.

cbind = column bind.

rexp(1,arri.rate) = generate an exponential random variable with rate λ = arri.rate.

<u>Input</u>

The function "simu" takes arrival time, service time, total simulation time and number of simulation we want to do as an input. Every time it calls the sub-routine "queueb" iteratively. The function "queueb" takes arrival rate, service rate and the total simulation time as an input.

Output

It outputs the number of black hats and the white hats in the system in a vector form with number of black hats in the first column and the number of white hats in the second column. We then assign the number of black hats and white hats to two different variables. Then we find the mean and variances of the number of black hats and white hats in the system [(i.e) E(m),V(m),E(n) and V(n)] by using the commands summary( ) and var( ). To find E(mn), we should multiply both columns and then use the command summary( ). From that we can find the covariance. .

Program

Function "simu"

```
function(arri.time, serv.time, stime,num)
{
arri.rate ← 1/arri.time
serv.rate ← 1/serv.time
B ← vector("numeric", num)
W ← vector("numeric", num)
for(i in 1: num)
{
A ← queuew(arri.rate, serv.rate, stime)
B[i] ← A[1]
W[i] ←A[2]
}
cbind(B, W)
}
```

<u>Sub-routine "queueb"</u>

```
function(arri.rate, serv.rate, stime)
{
numB ← 0
numW ← 0
ctime ← 0
arrive ← rexp(1, arri.rate)
serve ← stime+1
while(ctime + arrive < stime || ctime + serve < stime)
{
if (numB + numW = = 0)
{
ctime ← ctime + arrive
arrive ← rexp(1, arri.rate)
serve ← rexp(1, serv.rate)
numB ← 1
nex ← "B"
print("B arrives")
}
else
{
        if (arrive < serve )
        {
        ctime ← ctime + arrive
```

```
serve ← serve − arrive
arrive ← rexp(1, arri.rate)
numB ← numB + 1
print("B arrives")
}
else
{
ctime ← ctime + serve
arrive ← arrive − serve
serve ← rexp(1, serv.rate)
if (nex = = "B")
{
numB ← numB − 1
numW ← numW + 1
print("B served")
}
else
{
numW ← numW − 1
print("W served")
}
if (numB >0)
nex ← "B"
else nex ← "W"
```

```
            }
    }
    }
    }
    c(numB, numW )
    }
```

Appendix D

Simulation program for the white hat priority using imbedded Markov chain technique.

Notations

m,n = vectors

rep(0,z) = repeat the vector with "z" zeros.

r[1] = first element of r.

cbind = column bind.

runif(1) = generate an uniformly distributed random number.

trunc = represents integer division.

Input

The main function "gen" takes ρ and the number of simulation we want to do as an input. It iteratively calls the sub-routine "it" which calculates the values "m" and "n" and returns them back to the main function.

Output

It outputs the number of black hats and the number of white hats in the system in a vector form with number of black hats in the first column and the number of white hats in the second column. we then assign the black hats and white hats to two different variables, then find the mean and variances of the number of black hats and white hats in the system by using the commands summary( ) and var( ). To find E(mn), we should multiply the columns then use the command summary( ). From that we can find the covariance.

<u>Program</u>

<u>Function "gen"</u>

```
function(ρ, z)
{
m ← rep( 0, z )
n ← rep( 0, z )
for ( i in 1:(z−1) )
r ← it ( ρ, m[i], n[i] )
m[i+1] ← r[1]
n[i+1] ← r[2]
}
cbind(m,n)
}
```

<u>Sub-routine "it"</u>

```
function(ρ, m, n)
{
u ← runif(1)
l ← trunc ((log(u) / log(ρ/(1 + ρ)))
if( m = = 0 && n = = 0 )
{
r ← c( m+l , n+1 )
}
else if (m > 0 && n == 0)
```

```
{
r ← c(m + l − 1, n + 1)
}
else if ( m = = 0 && n >0 )
{
r ← c(m + l, n − 1)
}
else if (m > 0 && n > 0)
{
r ← c( m+l , n−1 )
}
r
}
```

Appendix E

Simulation for the black hat priority using the imbedded Markov chain technique.

Notations

Same as in appendix D.

Input

The main function "gen1" takes the value for ρ and the number of simulation we want to do as an input. It calls the sub-routine "it1" iteratively which takes the values of ρ "m" and "n" as an input.

Output

It outputs the number of black hats and the white hats in the system in a vector form with number of black hats in the first column and the number of white hats in the second column. We then assign the black hats and the white hats to two different variables. We find the mean and variances of the number of black hats and white hats in the system by using the commands summary( ) and var( ). To find E(mn), we should multiply the columns and then use the command summary( ). From that we can find the covariance.

Program

Function "gen1"

```
function(ρ, z)
{
m ← rep( 0, z )
```

```
n ← rep( 0, z )
for ( i in 1:(z-1) )
r ← it1 ( ρ, m[i], n[i] )
m[i+1] ← r[1]
n[i+1] ← r[2]
}
cbind(m, n)
}
```

<u>Sub-routine "it1"</u>

```
function(ρ, m, n)
{
u ← runif(1)
l ← trunc ((log(u) / log(ρ/(1 + ρ))))
if( m = = 0 && n = = 0 )
{
r ← c( m+l , n+1 )
}
else if ( m > 0 && n = = 0 )
{
r ← c( m+l−1 , n+1 )
}
else if ( m = = 0 && n >0 )
{
r ← c( m+l , n−1 )
```

```
}
else if ( m > 0 && n > 0 )
{
r ← c( m+l−1 , n +1 )
}
r
}
```

Appendix F

Simulation program for the variance reduction technique using imbedded Markov chain method for the white hat priority.

Notations

u, $1-u$ = random numbers.

m = vector containing the number of black hats in the queue using u.

n = vector containing the number of white hats in the queue using u.

a = vector containing the number of black hats in the queue using (1− u).

b = vector containing the number of white hats in the queue using (1− u).

ρ = the input value

z = the number of simulation runs we want to do.

rep$(0, z)$ = repeat the vector with "z" zeros.

$r[1]$ = first element of the vector r.

cbind = column bind.

runif = generate an uniformly distributed random number.

Input

The main function "GENE" takes the values of ρ the number of simulation we want to do as an input. It calls the sub-routine"ITRATE" iteratively. The main function takes the value for ρ and the number of simulation run we want to do as an input which calculates the values of m, n, a, b and returns them back to the main function.

Output

It outputs the values of "m", "n", "a" and "b" in a vector form with "m" in the first column, "n" in the second column, "a" in the third column and "b" in the fourth column. Now add columns one and three, then divide by two then assign a new variable to it. Similarly, we add columns two and four and divide by two then assign a new variable to it. We then find the mean and variances for the number of black and white hats in the system by using the commands summary( ) and var( ).

Program

Function "GENE"

```
function (ρ, z)
{
m ← rep(0,z)
n ← rep(0,z)
a ← rep(0,z)
b ← rep(0,z)
for (i in 1:(z-1) ) {
        r ← ITRATE(ρ, m[i], n[i], a[i], b[i])
        m[i+1] ← r[1]
        n[i+1] ← r[2]
        a[i+1] ← r[3]
        b[i+1] ← r[4]
}
cbind(m,n,a,b)
}
```

<u>Sub-routine "ITRATE"</u>

```
function(ρ, m, n, a, b)
{
u ← runif(1)
l ← trunc(log(u)/log(ρ/1 + ρ)))
if (m = = 0 && n = =0)
{
r ← c(m+l , n+1)
}
else if (m > 0 && n = = 0)
{
r ←c(m+l−1 , n+1)
}
else if (m = = 0 && n > 0)
{
r ← c(m+l , n−1)
}
else if (m > 0 && n > 0)
{
r ← c(m+l , n−1)
}
k ← trunc(log(1−u)/log(ρ/(1 + ρ)))
if (a = = 0 && b = = 0)
{
```

```
s ← c(a+k , b+1)
}
else if (a > 0 && b = = 0)
{
s ← c(a+k−1 , b+1)
}
else if (a = = 0 && b > 0 )
{
s ← c(a+k , b−1)
}
else if ( a > 0 && b > 0)
{
s ← c(a+k , b−1)
}
c(r,s)
}
```

Appendix G

Simulation program for the variance reduction technique using imbedded Markov chain method for the black hat priority.

<u>Notations</u>

u, 1−u = random numbers.

m = vector containing the number of black hats in the queue using u.

n = vector containing the number of white hats in the queue using (1−u).

a = vector containing the number of black hats in the queue using u.

b = vector containing the number of white hats in the queue using (1−u).

ρ = the input value

z = the number of simulation runs we want to do.

rep = repeat the vector with "z " zeros.

r[1] = first element of the vector r

cbind = column bind

runif = generate a uniformly distributed random number.

<u>Input</u>

The main function "gene" takes the value for ρ and the number of simulation run we want to do as an input. It iteratively calls the sub-routine "itrate" which calculates the values of m,n,a,b and returns them back to the main function.

<u>Output</u>

It outputs values of "m", "n", "a" and "b" in a vector form with "m" in the first column, "n" in the second column, "a" in the third column and "b" in the fourth column. Add columns one and three then divide it by two and assign a new variable to it. Similarly we add columns two and four and divide it by two then assign a new variable to it. Now we can find the mean and variance of the number of white and black hats in the system by using the command summary( ) and var( ) respectively.

Program

Function "gene

```
function (ρ, z)
{
m ← rep(0,z)
n ← rep(0,z)
a ← rep(0,z)
b ← rep(0,z)
for (i in 1:(z-1) ) {
        r ← itrate(ρ, m[i], n[i], a[i], b[i])
        m[i+1] ← r[1]
        n[i+1] ← r[2]
        a[i+1] ← r[3]
        b[i+1] ← r[4]
}
cbind(m,n,a,b)
}
```

Sub-routine "itrate"

```
function(ρ, m, n, a, b)
```

```
{
u ← runif(1)
l ← trunc(log(u)/log(ρ/1 + ρ)))
if (m = = 0 && n = =0)
{
r ← c(m+l , n+1)
}
else if (m > 0 && n = = 0)
{
r ←c(m+l−1 , n+1)
}
else if (m = = 0 && n > 0)
{
r ← c(m+l , n−1)
}
else if (m > 0 && n > 0)
{
r ← c(m+l−1 , n+1)
}
k ← trunc(log(1−u)/log(ρ/(1 + ρ)))
if (a = = 0 && b = = 0)
{
s ← c(a+k , b+1)
}
```

```
else if (a > 0 && b = = 0)
{
s ← c(a+k−1 , b+1)
}
else if (a = = 0 && b > 0 )
{
s ← c(a+k , b−1)
}
else if ( a > 0 && b > 0)
{
s ← c(a+k−1 , b+1)
}
c(r,s)
}
```

Appendix H

The simulation program for the variance reduction technique for the time dependent case where customers wearing white hats have higher priority over customers wearing black hats.

<u>Notations</u>

u,1−u, v, 1−v = random numbers.

numB = number of customers wearing a black hat in the system using u.

numB1 = number of customers wearing a black hat in the system using 1−u.

numW = number of customers wearing a white hat in the system using v.

numW1 = number of customers wearing a white hat in the system using 1−v.

vector("numeric",num) = vector containing numeric values whose size is determined by the input "num"

cbind( ) = column bind.

arr.time = arrival time

ser.time = service time

stime = simulation time

ctime = current time

append(u,w) = attach w to u.

u[−1] = except the first element of u.

nex = represents the next customer who is going to get the service.

<u>Input</u>

The main function "simW" takes arrival time, service time, simulation time and the number of simulation we want to do as an input. At each run it calls the subroutine "QUEUEW" which takes the arrival time, service time, the simulation time as an input.

Output

It returns the values of numB, numW, numB1, mumW1 in a vector form with values of numB in column one, values of numW in column two, values of numB1 in column three, and the values of numW1 in column four. We then add column one and three the divide it by two. We then assign a new variable to it. Similarly we add columns two and four, divide it by two then assign a new variable to it. Now we an find the mean and the variances of the number of black hats and the umber of white hats in the system by using the commands summary( ) and var().

Function "simW"

```
function (arr.time,serv.time,stime,num) {
B = vector("numeric",num)
B1 = vector("numeric",num)
W = vector("numeric",num)
W1 = vector("numeric",num)
for ( i in 1:num) {
A ← QUEUEW(arr.time,serv.time,stime)
B[i] ← A[1]
W[i] ← A[2]
B1[i] ← A[3]
W1[i] ← A[4]
```

```
cbind(B,W,B1,W1)
}
```

<u>Sub-routine QUEUEW</u>

```
function (arr.time, serv.time, stime)
{
numB ← 0
numW ← 0
numB1 ← 0
numW1 ← 0
u ← runif (10)
u1 ← u
arrive ← (−arr.time) * (log(u[1])
arrive1 ← (−arr.time) * log (1−u1[1])
u ← u[−1]
u1←u1[−1]
v← runif (10)
v1 ← v
serve ← stime + 1
serve1 ← stime +1
nex ← "B"
nex1 ← "B"
while (ctime + arrive < stime || ctime + serve < stime || ctime + arrive1 < stime
|| ctime + serve1 < stime) {
if(length(u) < 3|| length(u1) < 3) {
```

```
w ← runif (10)
u ← append (u, w)
u1 ← append (u1, w)
}
if(length(v) < 3|| length (v1) < 3) {
w ← runif (10)
v ← append (v, w)
v1 ← append (v1, w)
}
if(arrive < serve && arrive < arrive1 && arrive < serve1) {
ctime ← ctime + arrive
if(numB + numW > 0)
serve ← serve − arrive
else {
serve ← (−serv.time) * log(v[1])
v ← v[−1]
}
arrive1 ← arrive1 − arrive
serve1 ← serve1 − arrive
arrive ← (−arr.time) * log(u[1])
u ← u[−1]
numB ← numB + 1
print("B arrives")
}
```

```
else if (serve < arrive1 && serve < serve1) {
ctime ← ctime +serve
arrive ←arrive − serve
arrive1 ← arrive1 − serve
serve1 ← serve1 − serve
serve ← (−serve.time) * log (v[1])
v ← v[−1]
if (nex == "B") {
numB ← numB −1
numW ← numW +1
print("B served")
}
else {
numW ← numW −1
print("W served")
}
if (numW > 0)
nex ← "W"
else nex ← "B"
if (numB + numW == 0 ) {
serve ← stime
nex ← "B"
}
}
```

```
else if (arrivel < servel) {
ctime ← ctime + arrivel
if(numB1 + numW1 > 0)
servel ← servel − arrivel
else {
servel ← (−serv.tme) * log(1− v1[1])
v1 ← v1[−1]
}
arrive ← arrive − arrivel
serve ← serve − arrivel
arrivel ← (−arr.time) * log(1 − u1[1])
u1 ← u1[−1]
numB1 ← numB1 +1
print("B arrives")
}
else {
ctime ← ctime + servel
arrivel ← arrivel− servel
arrive ← arrive − servel
serve ← serve − servel
servel ← (−serv.time) * log (1 −v1[1])
v1 ← v1[-1]
if (nex1 == "B") {
numB1 ← numB1 −1
```

```
numW1 ← numW1 +1
print("B served")
}
else {
numW1 ← numW1 −1
print("W served')
}
if (numW1 > 0)
nex1 ← "W"
else nex1 ← "B"
if(numB1 +numW1 == 0) {
serve1 ← stime
nex1 ← "B"
}
}
}
c(numB, numW, numB1, numW1)
}
```

Appendix I

The simulation program for the variance reduction technique for the time dependent case where customers wearing black hats have higher priority than customers wearing white hats.

<u>Notations</u>

u,1−u, v, 1−v = random numbers.

numB = number of customers wearing a black hat in the system using u.

numB1 = number of customers wearing a black hat in the system using 1−u.

numW = number of customers wearing a white hat in the system using v.

numW1 = number of customers wearing a white hat in the system using 1−v.

vector("numeric",num) = vector containing numeric values whose size is determined by the input "num"

cbind( ) = column bind.

arr.time = arrival time

ser.time = service time

stime = simulation time

ctime = current time

append(u,w) = attach w to u.

u[−1] = except the first element of u.

nex = represents the next customer who is going to get the service.

<u>Input</u>

The main function "simB" takes arrival time, service time, simulation time and the number of simulation runs as an input. At each run it calls the sub-routine "QUEUEB" which takes arrival time, service time and simulation time as an input.

Output

It outputs the numB, numW, numB1 and numW1 in a vector form with numB in column one, numW in column two, numB1 in column three and numW1 in column four. We then add columns one and three then divide it by two. We then assign a new variable to this. Similarly,add columns two and four divide it by two then assign a new variable to it. Now we can find the mean and variances of the number of white and black hats in the system by using the commands summary() and var( ) respectively.

Function "simB"

```
function (arr.time,serv.time,stime,num) {
B = vector("numeric",num)
B1 = vector("numeric",num)
W = vector("numeric",num)
W1 = vector("numeric",num)
for ( i in 1:num) {
A ← QUEUEB(arr.time,serv.time,stime)
B[i] ← A[1]
W[i] ← A[2]
B1[i] ← A[3]
W1[i] ← A[4]
cbind(B,W,B1,W1)
```

```
}
```

<u>Function QUEUEB</u>

```
function (arr.time, serv.time, stime)
{
numB ← 0
numW ← 0
numB1 ← 0
numW1 ← 0
u ← runif (10)
u1 ← u
arrive ← (−arr.time) * (log(u[1])
arrive1 ← (−arr.time) * log (1−u1[1])
u ← u[−1]
u1 ← u1[−1]
v ← runif(10)
v1 ← v
serve ← stime + 1
serve1 ← stime +1
nex ← "B"
nex1 ← "B"
while (ctime + arrive < stime || ctime + serve < stime || ctime + arrive1 < stime
|| ctime + serve1 < stime) {
if (length(u) < 3 || length(u1) < 3) {
w ← runif (10)
```

```
u ← append (u, w)
u1 ← append (u1, w)
}
if(length(v) < 3 || length (v1) < 3) {
w ← runif (10)
v ← append (v, w)
v1 ← append (v1, w)
}
if(arrive < serve && arrive < arrive1 && arrive < serve1) {
ctime ← ctime + arrive
if(numB + numW > 0)
serve ← serve - arrive
else {
serve ← (−serv.time) * log(v[1])
v ← v[−1]
}
arrive1 ← arrive1 -arrive
serve1 ← serve1 - arrive
arrive ← (−arr.time) * log(u[1])
u ← u[−1]
numB ← numB + 1
print("B arrives")
}
else if (serve < arrive1 && serve < serve1) {
```

```
ctime ← ctime +serve
arrive ←arrive − serve
arrivel ← arrivel − serve
servel ← servel − serve
serve ← (−serve.time) * log (v[1])
v ← v[−1]
if (nex == "B") {
numB ← numB −1
numW ← numW +1
print("B served")
}
else {
numW ← numW −1
print("W served")
}
if (numB > 0)
nex ← "B"
else nex ← "W"
if (numB + numW == 0 ) {
serve ← stime
nex ← "B"
}
}
else if (arrivel < servel) {
```

```
ctime ← ctime +arrivel
if(numB1 + numW1 > 0)
servel ← servel − arrivel
else {
servel ← (− serv.tme) * log(1−v1[1])
v1 ← v1[−1]
}
arrive ← arrive − arrivel
serve ← serve − arrivel
arrivel ← (−arr.time) * log(1 − u1[1])
u1 ← u1[−1]
numB1 ← numB1 +1
print("B arrives")
}
else {
ctime ← ctime + servel
arrivel ← arrivel− servel
arrive ← arrive −servel
serve ← serve −servel
servel ← (−serv.time) * log (1 −v1[1])
v1 ← v1[−1]
if (nex1 == "B") {
numB1 ← numB1 −1
numW1 ← numW1 +1
```

```
print("B served")
}
else {
numW1 ← numW1 −1
print("W served")
}
if (numB1 > 0)
nex1 ← "B"
else nex1 ← "W"
if(numB1 +numW1 == 0) {
serve1 ←stime
nex1 ← "B"
}
}
}
c(numB, numW, numB1, numW1)
}
```

Appendix J

Important Results

Model One

White Hat Priority.

Time Dependent Technique.

$$E(m) = \left(\frac{\rho\ (1+\rho)}{1-2\rho}\right)$$

$$V(m) = \left(\frac{\rho + 2\rho^3 - 3\rho^4}{(1-2\rho)^2}\right)$$

$$E(n) = \rho$$

$$V(n) = \rho - \rho^2$$

$$COV(mn) = \rho^2$$

$$r(m,n) = \left(\frac{\rho^2(1-2\rho)}{\sqrt{(\rho-\rho^2)(\rho+2\rho^3-3\rho^4)}}\right)$$

Markov Chain Technique.

$$E(m) = \left(\frac{3\rho}{2\ (1-2\rho)}\right)$$

$$V(m) = \left(\frac{\rho\ (6-5\rho+4\rho^2)}{4\ (1-2\rho)^2}\right)$$

$$E(n) = \left(\frac{1}{2}\right)$$

$$V(n) = \left(\frac{1}{2}\right)$$

$$COV(mn) = \left(\frac{-\rho}{4}\right)$$

$$r(m,n) = \left(\frac{-\rho(1-2\rho)\sqrt{2}}{2\sqrt{6\rho-5\rho^2+4\rho^3}}\right)$$

Model Two

Black Hat Priority.

Time Dependent Technique

$$E(m) = \left(\frac{\rho}{1-\rho}\right)$$

$$V(m) = \left(\frac{\rho}{(1-\rho)^2}\right)$$

$$E(n) = \left(\frac{\rho+\rho^2}{1-3\rho+2\rho^2}\right)$$

$$V(n) = \left(\frac{\rho\,(1+2\rho^2-8\rho^3+\rho^4)}{(1-\rho)\,(1-3\rho+2\rho^2)^2}\right)$$

$$COV(mn) = \left(\frac{\rho^2}{(1-\rho)^3}\right)$$

$$r(m,n) = \left(\frac{\rho(1-2\rho)}{\sqrt{1+\rho-2\rho^2-8\rho^3+9\rho^4-\rho^5}}\right)$$

Markov Chain Technique

$$E(m) = \left(\frac{\rho}{1-\rho}\right)$$

$$V(m) = \left(\frac{\rho}{(1-\rho)^2}\right)$$

$$E(n) = \left(\frac{1-\rho+4\rho^2}{2-6\rho+4\rho^2}\right)$$

$$V(n = \left(\frac{1-3\rho+19\rho^2-17\rho^3-16\rho^4}{4\,(1-\rho)\,(1-3\rho+2\rho^2)^2}\right)$$

$$COV(mn) = \left(\frac{\rho^2}{(1-\rho)^3}\right)$$

$$r(m,n) = \left(\frac{2\rho^2(1-2\rho)}{\sqrt{\rho-4\rho^2+22\rho^3-36\rho^4+\rho^5+16\rho^6}}\right)$$

Bibliography

[1] Saaty, L.T., "Elements Of Queueing Theory With Applications". (1961), Mcgraw Hill Co.

[2] Kleinrock, L., "Queueing Systems". (1975), John Wiley & Sons.

[3] Cooper, B.R. , "Introduction To Queueing Theory". (1981), North Holland Co.

[4] Prabhu, N.U., "Queues And Inventories". (1965), Wiley.

[5] Feller, W., "An Introduction To Probability Theory and Its Applications". (1968), Volume One, Third Edition,John Wiley & Sons

[6] Jaiswal, N.K., "Priority Queues", (1968), New York Academic Press.

[7] Koenigsberg, E., "Cyclic Queues", (1958), Operational Research Quarterly.

[8] Barry, J.Y., "A Priority Queueing Problem", (1956), J.Operations Research Soc, Vol 4, P. 385-387.

[9] Cobham, A., "Priority Assignment In Waiting Line Problem", (1955), J.Operations Research Soc Vol 3, P.70-76.

[10] Holley, J.L., "Waiting Line Subject To Priorities", (1954), J.Operations Research Soc. Vol 2, P.341-343.