A level-crossing approach to waiting times in M/M/C queues with priorities.

Mado Bachan

University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation
Bachan, Mado, 'A level-crossing approach to waiting times in M/M/C queues with priorities.' (1986). Electronic Theses and Dissertations. 6803.
https://scholar.uwindsor.ca/etd/6803
A LEVEL-CROSSING APPROACH TO WAITING TIMES
IN M/M/C QUEUES WITH PRIORITIES

by

Mado Bachan

A Thesis submitted to the Faculty of Graduate Studies and Research through the Department of Industrial Engineering in Partial Fulfillment of the requirements for the Degree of Master of Applied Science at the University of Windsor

Windsor, Ontario, Canada

1986
UMI Number: EC54792

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction. In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI

UMI Microform EC54792
Copyright 2010 by ProQuest LLC
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
I hereby declare that I am the sole author of this thesis.
I authorize the University of Windsor to lend this thesis to other institutions or individuals for the purpose of scholarly research.

M. Bachan

I further authorize the University of Windsor to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

M. Bachan
The University of Windsor requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.
ABSTRACT

This thesis deals with a variant of an M/M/2 queue using a level-crossing approach to model the virtual waiting time process. The queueing system considered in detail consists of two types of customers arriving independently and demanding service from either server, with priority given to one class. The service-time distribution is different for each class of customers; a situation not previously dealt with in the literature.

Two "system-point" processes are defined; one for each customer-type. Typical sample paths are developed to illustrate the modelling technique. Accurate estimates are computed for the virtual waiting time probability density function and cumulative distribution function for each unit-type using a system-point Monte Carlo computation approach. A discussion is presented as to how larger systems can be modelled to obtain similar information.
To my wife,
RADICA,
and daughter,
MALINI
ACKNOWLEDGEMENTS

I am deeply indebted to Prof. Percy H. Brill for his astute guidance during all phases of this thesis. My sincere thanks go out to him for the numerous contributions, suggestions, criticisms and for helping to provide a stimulating environment, all of which were instrumental in the successful completion of this work.

I would also like to thank Dr. R. S. Lashkari and Dr. M. Hlynke for reviewing this thesis.

My sincere appreciation is extended to my wife, Radica who has been inspirational in her understanding, encouragement and patience during the many long days and nights that were spent completing this work. A special thanks go out to Mr. & Mrs. M. Lutchmeesingh for motivating me to pursue graduate studies and their constant support.

Finally, I am most grateful to the Canadian Commonwealth Scholarship and Fellowship Committee for providing financial support during my M.A.Sc. program.
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example of an SP(1) Sample Path</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>Generating an Ordinary Type-(1,1) Busy Period - B_{11}^0</td>
<td>46</td>
</tr>
<tr>
<td>3</td>
<td>Generating a Delayed Type-(1,1) Busy Period - B_{11}^\delta</td>
<td>47</td>
</tr>
<tr>
<td>4</td>
<td>Generating a Delayed &quot;mixed&quot; Busy Period - B_{12}^\delta</td>
<td>48</td>
</tr>
<tr>
<td>5</td>
<td>SP(2) Sample Path with Jump Sizes Distributed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>as B_{11}^0, e(\mu_1+\mu_2), e(\mu_1+\mu_2) + B_{11}^\delta_1 \delta_2,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>e(\mu_1+\mu_2) + B_{12}^\delta_1 \delta_2</td>
<td>49</td>
</tr>
<tr>
<td>6</td>
<td>SP(2) Sample Path with Jump Sizes Distributed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>as e(2\mu_2), e(\mu_1+\mu_2) and e(2\mu_2) + B_{12}^\delta_2</td>
<td>50</td>
</tr>
<tr>
<td>7</td>
<td>Probability Density Function for SP(1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(T=1000 min.)</td>
<td>58</td>
</tr>
<tr>
<td>8</td>
<td>Cumulative Distribution Function for SP(1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(T=1000 min.)</td>
<td>59</td>
</tr>
<tr>
<td>9</td>
<td>Probability Density Function for SP(2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(T=1000 min.)</td>
<td>65</td>
</tr>
<tr>
<td>10</td>
<td>Cumulative Distribution Function for SP(2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(T=1000 min.)</td>
<td>66</td>
</tr>
<tr>
<td>11</td>
<td>Theoretical and SP(1) Estimated pdf's for</td>
<td></td>
</tr>
<tr>
<td></td>
<td>an Approximated Simple M/M/2 Queue</td>
<td>73</td>
</tr>
<tr>
<td>12</td>
<td>Theoretical and SP(2) Estimated pdf's for</td>
<td></td>
</tr>
<tr>
<td></td>
<td>an Approximated Simple M/M/2 Queue</td>
<td>75</td>
</tr>
</tbody>
</table>
13. Bounds on Type-k CDF for an N-Class Priority System ........................................ 90

14. Bounds on Type-2 CDF for an M/M/2 3-Class Priority Queue ......................... 91
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Results from the $SP^{(1)}$ Monte Carlo Computation ($T=1000$ min.)</td>
<td>56</td>
</tr>
<tr>
<td>2. Results from the $SP^{(2)}$ Monte Carlo Computation ($T=1000$ min.)</td>
<td>62</td>
</tr>
<tr>
<td>3. Theoretical Values and S-P Estimates for an Approximated Simple $M/M/2$ Queue</td>
<td>71</td>
</tr>
<tr>
<td>4. Theoretical and $SP^{(1)}$ Estimates of pdf for an Approximated Simple $M/M/2$ Queue</td>
<td>72</td>
</tr>
<tr>
<td>5. Theoretical and $SP^{(2)}$ Estimates of pdf for an Approximated Simple $M/M/2$ Queue</td>
<td>74</td>
</tr>
<tr>
<td>6. Theoretical Values and S-P Estimates for a Non-Preemptive Priority $M/M/2$ Queue ($\lambda_1=15$, $\lambda_2=20$, $\mu_1=\mu_2=20$)</td>
<td>77</td>
</tr>
<tr>
<td>7. Theoretical Values and S-P Estimates for a Non-Preemptive Priority $M/M/2$ Queue ($\lambda_1=6$, $\lambda_2=12$, $\mu_1=\mu_2=24$)</td>
<td>78</td>
</tr>
<tr>
<td>8. System Simulation Results with Model Parameters $\lambda_1=15$, $\lambda_2=12$, $\mu_1=20$, $\mu_2=15$</td>
<td>84</td>
</tr>
<tr>
<td>9. $M/M/2$, 2-Class Priority Model Results</td>
<td>85</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

ABSTRACT ........................................................................ vi
DEDICATION ....................................................................... vii
ACKNOWLEDGEMENTS .................................................... viii
LIST OF FIGURES ........................................................... ix
LIST OF TABLES ............................................................ xi

Chapter | Page
--------|------
I. INTRODUCTION ......................................................... 1
  1.1 Overview of the Priority Queueing Discipline .......... 1
  1.2 Importance of Waiting-Time Distribution .. 2
  1.3 Motivation for this Study ...................... 4
  1.4 Various Approaches for Studying Priority Queueing Systems 5
  1.5 The System-Point Method .................... 9
II. LITERATURE REVIEW .................................................. 13
III. SYSTEM-POINT THEORY .............................................. 18
  3.1 The Virtual Waiting Time Process ............... 18
  3.2 The System-Point Process for Single-Server Queues ........ 19
  3.3 The System-Point Process for Multiple-Server Queues ... 21
  3.4 Methods of Solution .............................................. 22
IV. SYSTEM-POINT MODELS FOR THE DIFFERENT PRIORITY CLASSES ........................................ 24
  4.1 System-Point Process for Type-1 Customers - SP(1) .......... 25
  4.1.1 Illustration of a Typical SP(1) Sample Path ................. 29
  4.2 System-Point Process for Type-2 Customers - SP(2) .......... 35
  4.2.1 The System-Point Process Definition for Type-2 Customers ... 35
  4.2.2 Kinds of System Busy Periods ...... 38

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
4.2.3 Possible Jump Sizes for SP(2) Sample Path ....................... 39
4.2.4 Example of Typical SP(2) Sample Paths ....................... 41

V. SYSTEM-POINT MONTE CARLO COMPUTATION ......................... 51

5.1 Illustration of the System-Point Monte Carlo Computation Approach .................. 53
5.1.1 SP(1) Monte Carlo Computation ...... 54
5.1.2 SP(2) Monte Carlo Computation ...... 60
5.2 Verification of the System-Point Priority M/M/2 Computer Programs .......... 68

VI. SYSTEM-POINT MODEL RESULTS ........................................... 80

6.1 Check on the System-Point Priority M/M/2 Results .......................... 80
6.1.1 Check on the Waiting Time Estimates ......................... 81
6.2 Estimates for the Pr(any server idle) .... 86

VII. GENERALIZATIONS AND SUMMARY .................................... 88

7.1 Approximation of the CDF in an M/M/2 N-Class System .......................... 88
7.2 M/M/3 Priority System with Two Customer-Types .......................... 92
7.2.1 System-Point Process for Type-1 Units ............................ 93
7.2.2 System-Point Process for Type-2 Units ............................ 94
7.3 Summary .................................. 95

REFERENCES ................................................................. 97

APPENDICES

A. SAMPLE CALCULATIONS BASED ON ANALYTICAL MODELS .......................... 101

B. SP(1) FLOWCHART AND COMPUTER PROGRAM LISTING .......................... 107

- xiii -
C. $\text{SP}^{(2)}$ FLOWCHART AND COMPUTER PROGRAM LISTING .................................. 129
D. COMPUTER PROGRAM LISTING FOR PDF AND CDF COMPUTATION ............................ 151
E. SYSTEM SIMULATION MODEL ................................................................. 154

VITA AUCTORIS ......................................................................................... 177
CHAPTER I

INTRODUCTION

Many real-life queueing systems involve customers with priorities based on some external characteristic, such as, length of service required, importance of customer demanding service, etc. These characteristics are assumed to be independent of the state of the system. Therefore the use of priority-discipline models would provide a welcome refinement over the more usual non-priority queueing counterparts because they would distinguish between waiting times and other measures of performance for the different types of customers. Unfortunately, the inclusion of priorities makes the mathematical analysis sufficiently complicated, so that only limited results are presently available; the majority being for single server queues.

1.1 Overview of the Priority Queueing Discipline

There are two general situations in first come first served priority queueing disciplines. In the first, called preemptive, an arriving customer with a higher
priority is allowed to enter service immediately, possibly interrupting a customer with lower priority who is already in service when the higher priority customer enters the system. There are two possible variations; the preempted customer’s service when resumed can either continue from the point of preemption or start anew. In the second general priority situation, called non-preemptive, (also known as head-of-the-line [Kleinrock (1976)]), the highest priority waiting customer goes to the head of the queue but cannot interrupt the customer presently in service, even though this customer may have a lower priority.

1.2 Importance of Waiting-Time Distributions

The waiting time in a queue as a measure of performance can be used to design and control queueing systems when the objective is to minimise the cost of delay [Gross & Harris (1985)]. Depending on the context, this cost may be thought of as disutility, penalty cost, loss of goodwill, opportunity cost, postponement of revenue, customer dissatisfaction, storage cost, poorness of service, or some equivalent [Greenberger (1966)]. If the cost for each customer waiting in line is some constant per unit time, then the average waiting time can be used as input to these optimization problems. In the event of non-linear or step-function cost functions it would be necessary to know
the entire probability density function of the waiting time in order to compute expected cost of waiting.

The available analytical approaches generally provide the average measures of performance of a queueing system. Thus, if the cost structure of delay for a queueing system is non-linear we cannot attempt to optimize the system with respect to the cost of delay using the results from analytical approaches.

This thesis examines exponential multi-server, non-preemptive priority queueing systems. The model will consider systems where each customer-type demanding service has its own service-time distribution; this variant has not been solved analytically in the literature to date. The first (or main) objective is to compute the waiting time probability density function (pdf) for each customer-type in a two-server two-class priority system using a system-point Monte Carlo computation approach, Brill (1983). Having computed the pdf we then proceed to find the cumulative distribution function (cdf) and expected wait for each unit-type. Additional information in the form of \( \Pr(\text{an empty system}) \) and \( \Pr(\text{a single unit of either type in the system}) \) will be generated by virtue of the modelling approach adopted. The thesis also considers the question of generalizations of the two-server two-class priority system.
1.3 Motivation for this Study

As stated earlier, there are many real-life situations where customers of different priorities demand service from the available server(s).

A few real-life examples of queueing systems involving priorities are:

i) service and maintenance systems; where short jobs are given priority over long jobs,

ii) communication systems; where emergency messages are given priority over routine messages,

iii) health systems; where the condition of the patient determines the priority in which he/she is serviced,

iv) manufacturing (job-shop) systems; where work-orders from a particular customer are given priority over work-orders from another customer.

v) airport systems; where landing aircraft are given priority over departing aircraft.

All the above queueing applications can be modelled as M/M/C systems with priority queueing discipline. Here it is assumed that priority is externally determined and is independent of the state of the system. For example, consider the following scenario of a job-shop: A turret lathe department with two machines, receives two kinds of
jobs, namely government jobs and commercial jobs. Whenever a turret lathe finishes a job, it starts a government job if one is waiting; if not, it starts a commercial job if any is waiting. Jobs of the same type are taken on a first-come, first-served basis. Both types of jobs arrive at the department with independent Poisson arrival rates and the service times appear to be exponential with independent means [Hillier & Lieberman (1980)].

1.4 Various Approaches for Studying Priority Queueing Systems

The techniques used to study queueing systems can be broadly divided into the following:

1) analytic
2) numerical/computational (with approximations)
3) simulation

The analytic approach has been primarily used to study single server priority queueing systems. Explicit formulas are available for the measures of performance in the case of M/M/1 and M/G/1 [Cobham (1954), Jaiswal (1968)]. The analysis for mean values in the M/M/1 system is relatively straightforward but for the M/G/1 priority system extensive use is made of the Laplace transformation and the inversion procedure can be complex [Jaiswal (1962)]. Obtaining the
waiting time pdf for either the M/M/1 or the M/G/1 priority system is not a simple task since the method relies heavily on integral transforms.

Neuts' (1978) theory of matrix-geometric invariant probability vectors provides significant computational improvements over the analytic approach. Miller (1981) demonstrated that the special structure of this process yields explicit recursive formulas for the steady-state probabilities in an M/M/1 priority system.

Another approach that is commonly used to study queueing systems is to simulate the actual operation. For this simulation approach, a model is developed which is made to represent the system as closely as possible. The model is then driven with certain inputs so that the system operates as it would in real-life. By observing the corresponding outputs, inferences can be made about the true characteristics of the system.

We can broadly classify systems into two types, discrete-time and continuous-time. A discrete-time system is one for which the state variables change only at a countable (or finite) number of points in time, whereas a continuous-time system is one for which the state variables change continuously with respect to time [Law & Kelton (1982)]. Queueing systems are commonly modelled as
discrete-time systems since events which cause a change in the state of the system occurs at only a countable number of points in time. There are three alternative ways of building discrete-time models [Kiviat (1967)]. The event scheduling approach emphasizes a detailed description of the steps that occur when an individual event takes place. Each type of event naturally has a distinct set of steps associated with it. The activity scanning approach emphasizes a review of all activities in a simulation to determine which can be begun or terminated each time an event occurs. The process interaction approach emphasizes the progress of an entity through a system from its arrival event to its departure event.

In a discrete-time simulation model it is necessary to keep track of the current value of the simulated time and to advance simulated time from one value to another. Two approaches are used, next-event time advance and fixed-increment time advance [Law & Kelton (1982)]. The next-event time advance approach advances the simulation clock to the time of the next event, at which the state of the system is updated. The fixed-increment time advance method advances the simulation clock by a fixed time $\Delta t$ at which point a check is made on the system state to see if an event has occurred. Gafarian & Ancker (1966), have given a procedure for determining the value of the fixed time $\Delta t$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Simulation as a tool for studying complex systems has its advantages and disadvantages; a list is provided in Law & Kelton (1982). The major advantage of this approach is that it can be used to solve problems which are not analytically tractable and can produce any of the desired measures of performance for priority queueing systems. The main disadvantage of the system simulation approach is that it often requires a detailed model which can be expensive and time-consuming to develop. Also, such models can be incorrectly verified and validated. Because a simulation model is driven by random numbers, the output from a simulation run is only an estimate of a model’s true characteristic for a given set of input parameters. Hence, due to the stochastic nature of the output we need a number of replications for a given set of input parameters if we are to estimate the various measures of performance with some level of statistical confidence. On the other hand, an analytical model produces the actual true characteristics of that model for a given set of input parameters and is preferable to a simulation model when available. Based on the replications, a histogram can be constructed for a given measure of performance and the data fitted to a probability density function. Again, due to the stochastic nature of the output this may prove quite cumbersome and costly.
1.5 The System-Point Method

The system-point method (or level-crossing technique) [Brill (1975)] allows for computing the stationary probability density function of the virtual waiting time or number in the system in a broad class of queueing systems using either an analytic or Monte Carlo computation approach. To date this technique has been successfully applied in an analytic manner to the following: variations of the M/M/1 queue [Brill & Posner (1977)], variations of the M/M/R queue [Brill & Posner (1981a)], variants of the M/M/2 queue [Brill & Posner (1981b), Brill & Green (1984)] and the GI/G/1 queue [Brill (1979)].

Brill (1983) introduced a Monte Carlo computation approach, based on system-point theory for computing accurate estimates of the stationary probability density function (pdf) and cumulative distribution function (cdf) of the virtual waiting time. Using this approach, Covert (1985) successfully modelled and solved for the waiting time distribution in an M/G/1 priority queueing system with two classes of customers. The estimates were found to be close approximations to the analytical values.

Brill (1975) and Brill (1983) demonstrates that the system-point analytic approach and the system-point Monte Carlo computation approach, respectively, approximate the
true probability density function and cumulative
distribution function as $t \to \infty$, with probability one.
Thus, in one sample path generation we can evaluate the
entire probability density function (pdf) and cumulative
distribution function (cdf), expected waiting time, and the
probability that a customer will wait zero time. (Depending
on the nature of the system other probabilities can be
determined; for example, in the M/M/2 priority system the
probability of a single customer in the system can also be
found). In exponential models, usually the pdf of the
number in the system can also be computed. The effect of
altering the initial seed is minimal on the computed
probability density function, the cumulative distribution
function and the expected values of other measures of
performance, since results hold with probability one.

In the system-point Monte Carlo computation approach
adopted in this thesis a system-point model is constructed
for each customer-type in a k-class system. Thus, the time
required for modelling, validation, verification and
computation may become substantial as the number of
customer-types increases. On the other hand, since each
class of customer has its own associated system-point model
once verification and validation is completed for that
class there is no need to be concerned about that model's
correctness any more, and the class can be analysed
independently of the others. Another benefit of building a system-point model for each class of customer is that it forces an improved understanding of what is really happening as far as the dynamics and structure of the system is concerned.

In this thesis, a system-point model will be developed for each customer-type and the corresponding waiting time density function and cumulative distribution function computed via a system-point Monte Carlo computation. After constructing a typical sample path (Virtual waiting time vs. Time), the method explicitly connects the stationary probability density function of the virtual waiting time evaluated at any level \( w \), with the number of times the sample path crosses level \( w \).

The following is a brief outline of the topics to be dealt with in the remainder of this thesis. Chapter 2 presents a review of the relevant literature and illustrates the importance of this research. Chapter 3 briefly outlines some aspects of System-Point theory which are applicable to this study. In Chapter 4 the two-server two-class priority system under study is modelled by a system-point process for each customer-type. Examples of typical sample paths for the respective system-point processes illustrate the main ideas of the modelling approach adopted in this study. Chapter 5 concerns the
illustration of the system-point Monte Carlo computation approach to obtain the waiting time pdf (and other information); and verification of the computer programs written to generate sample paths for the respective processes. Making use of these verified computer programs, Chapter 6 contains "steady-state" results for the priority system modelled in Chapter 4. The results are checked against those obtained from a system simulation model. The final chapter discusses generalizations of the two-server two-class priority system to a case with three servers and a case with more than two customer-types.
CHAPTER II

LITERATURE REVIEW

Though there are many examples of multi-server non-preemptive priority queueing systems in real-life, there is relatively little literature on their analysis. The literature survey given in this section serves two important purposes: (i) to provide a framework for verification and validation of the system-point models to be formulated and their subsequent solutions and (ii) to show where the proposed work fits into the general picture of non-preemptive priority queueing systems. With this as our objective, the literature review will deal specifically with M/M/1, M/G/1 and M/M/C non-preemptive priority queueing systems.

Cobham (1954) was the first to derive formulas for the expected waiting times for units of each priority level in a non-preemptive priority queueing system. He considered both single and multiple channel systems; for the single server case the system was M/G/1, whilst for the multi-server case the system was M/M/C. For the multiple channel system, assuming that each priority class has the
same exponential service time probability distribution function with mean \( 1/\mu \), Cobham derived the expected waiting time for a customer from any priority class (but not the entire pdf). The results of Cobham's analysis for the multiple channel system (assuming a finite number of channels and priority) are given below.

The expected waiting time of a unit of priority \( p \) is given by:

\[
\bar{W}_p = \frac{E[T_0]}{ \left[ 1 - (1/(c\mu)) \sum_{k=1}^{p-1} \lambda_k \right] \left[ 1 - (1/(c\mu)) \sum_{k=1}^{p} \lambda_k \right]}
\]

where,

\[
E[T_0] = \frac{(c\rho)^c/(c\mu)}{c!(1-\rho) \left[ \sum_{j=0}^{c-1} (c\rho)^j/j! + (c\rho)^c/(c!(1-\rho)) \right]}
\]

\( \rho = \lambda/(c\mu) \)

\( p = \) priority of unit (smaller the integer \( p \), higher the priority; \( p = 1, 2, ... \))

\( \lambda_k = \) Poisson arrival rate of \( p \)-th priority unit

\( \mu = \) exponential service rate (hazard rate of service time pdf)

\( c = \) number of service channels

Equation 2.1 for \( \bar{W}_p \) is valid when \( (1/(c\mu)) \sum_{k=1}^{p} \lambda_k < 1 \).

Intuitively this says that the aggregate arrival rate of customers of priority \( p \) or higher, must be less than the
capacity service rate of the servers, for the stationary distribution to exist.

Morse (1958) considered a single exponential channel where arriving units are designated to be a member of one of two priority classes. Starting with balance equations, Morse derived the number-in-the-system-probability generating function for two cases: (i) both classes of customers have the same service time distribution and (ii) priority one customers are served at rate \( \mu_1 \) and the priority two customers at rate \( \mu_2 \).

Miller (1960) derived number-in-the-system-probability generating functions and Laplace transforms for the waiting time and busy period distributions for the non-preemptive M/G/1 priority queue by using the imbedded Markov chain technique. On assuming an exponential service time distribution, (M/M/1), the results obtained are different from the results of Morse (1958). Jaiswal (1968) observed that the probability distribution obtained by considering the queueing process in continuous time (as in Morse) is different from the one obtained by considering the queue length at those points at which a customer departs (as in Miller).

Jaiswal (1968) applied the supplementary variable technique to obtain expected waiting times and expected
number in the queue for a variety of systems. The introduction of supplementary variables makes the queueing process Markovian in continuous time. The measures of performance were evaluated for priority models with $k$ customer classes, various service time distributions, finite and infinite sources and preemptive and non-preemptive disciplines. The measures of performance are expressed in terms of the Laplace transforms of the density functions of the busy period and the occupation time of the server.

Marks (1972) analysed two queueing models, one with preemptive and the other with non-preemptive priority for the $M/M/1$ system, each model assuming two levels of priority. For both models a set of recursion formulas were derived that allow exact calculation of the equilibrium state probabilities.


Davis (1966) extended Cobham's work ($M/M/C$ priority queue) by deriving the equilibrium waiting time distribution function for the same model. Making use of
balance equations, Laplace transform and contour integration, Davis obtains an equation for the waiting time distribution. It should be noted that the result given by Davis is for a system where the service rates are the same for all classes.

The review of the literature presented above, indicates that although much work has been done on single-server priority queueing systems [Jaiswal (1968)], little work has been carried out on their multi-server counterparts. This can be attributed to the mathematical complexities involved in the analysis of such queueing systems. Most of the available results are for the average values of the measures of performance. It would be more useful to obtain the probability density (or distribution) function on which a measure of performance is based, since a mean value hardly captures and summarizes response, especially when the response distribution is highly skewed. The entire density (or distribution) function is most useful in cases where the cost of delay is non-linear. By knowing the distribution one can more accurately analyse a priority queueing system than just using the first and/or second moments. At this point it should be noted that for the M/M/C non-preemptive priority system, results are available only for exponential service time distributions with equal means [Cobham (1954), Davis (1966)]. Thus this research has theoretical as well as practical significance.
CHAPTER III

SYSTEM-POINT THEORY

This chapter reviews and summarizes some basic concepts and results of System-Point theory which is used to model and analyse the queueing system considered in this thesis. The material presented here is not intended to give an in depth coverage of System-Point theory. The reader interested in delving into this relatively new methodology for analysing stochastic systems is referred to Brill (1975), (1979), (1983) and Brill et al. (1977), (1981a), (1981b), (1984). Section 3.1 introduces the important concept of the virtual waiting time process which is generalized by the system-point method. This generalization is discussed in Sections 3.2 and 3.3. In the final section of this chapter we discuss methods of solving for the probability density function of the generalized virtual waiting time.

3.1 The Virtual Waiting Time Process

The time a potential customer would have to wait in queue before starting service were he to arrive at time t...
is called the virtual waiting time, denoted by $W(t)$. For single-server queues this can also be defined as the unfinished work to be done by the server at time $t$. A sample path of the virtual wait process \{W(t), \, t \geq 0\} decreases with slope $-1$ over time until it hits level zero, or until the next arrival. At an arrival epoch $T_n$, it jumps by an amount equal to the service time requirement of the arriving customer. If it hits level zero, it remains there until the next arrival [Takács (1962)]. This stochastic process is a continuous-state continuous-time Markov process subject to discontinuous jumps [Kleinrock (1975), Takács (1962)]. These jumps are at arrival epochs or possibly at other essential time points, depending on the model treated.

For multiple server queues, in sample paths of the virtual waiting time, the jumps sizes are equal to the "inter start-of-service departure" time. This is the time measured from the start of service epoch until the first departure from the system, thereafter.

3.2 The System-Point Process for Single-Server Queues

A generalization of the standard virtual waiting time process leads to the modelling of queues by what is called the System-Point process. For single server queues, with first-come, first-served discipline the system-point
process sample functions are identical to those of the usual virtual wait process. Brill (1975) introduced the following theorem which forms the basis of System-Point theory for single server queues and simple multiple server queues. For theorems which apply to more complex queues see Brill (1975).

**Basic System-Point Theorem for Single-Server Queues**

With probability 1

\[ \lim_{t \to \infty} \frac{D_t(w)}{t} = g(w); \quad w > 0. \]

\[ \lim_{t \to \infty} \frac{I_t}{t} = \lim_{w \to 0} g(w) = g(0^+) = \lambda P_0 \]

where

- \( D_t(w) \): the number of system-point (sample path) downcrossings of level \( w > 0 \) during time interval \([0,t]\)
- \( I_t \): the number of system-point (sample path) impacts (level-zero hits) during \([0,t]\)

This theorem essentially relates the limiting waiting time density function, \( g(w) \), \( w > 0 \) to the long-run average rate at which the system-point crosses level \( w \). Balancing the long-run average rates of downcrossings and upcrossings of level \( w \) yield an equation, usually a Volterra integral equation, that will involve the density function.
3.3 The System-Point Process for Exponential Multiple-Server Queues

For exponential multiple server queues the sample functions of the system-point process take values in a more general state space. Here, the system-point process is a Markov process denoted by \( \{<W(t),M(t)>, t \geq 0\} \), where \( W(t) \) is the virtual wait and \( M(t) \) is a random vector with countably many possible states. Brill (1975) referred to \( M(t) \) as the system configuration and introduced the concept of "lines", and "pages" to describe possible states of the system. For a further explanation of system configuration see Brill & Posner (1981a) and for applications see Brill & Posner (1981a, 1981b) and Brill & Green (1984).

The system-point method explicitly connects the stationary probability density function of the generalized virtual waiting time evaluated at \( w > 0 \), with the number of times that sample function crosses level \( w \) in the state space per unit time. The geometric point which traces out the sample function over time is called the "system-point".

The basic theorem given in Section 3.2 now becomes

For \( m \in M \) (all possible configurations for which the waiting time is or can be greater than zero), with probability one

\[
\lim_{t \to \infty} \frac{E[D_t(w,m)]}{t} = f(w,m) \quad w > 0
\]

\[
\lim_{t \to \infty} \frac{E[I_t(m)]}{t} = \lim_{w \to 0} f(w,m) = f(0^+,m)
\]
The above theorem relates the joint limiting density \( f(w,m) \), \( w > 0, m \in M \) to the long-run average rate at which the system-point crosses level \( w \) in the coordinate system corresponding to configuration \( m \) in the state space. Balancing the expected entrance and exit rates of sets whose boundary is level \( w > 0 \) for each possible configuration yields Volterra type integral equations involving the density function of the waiting time. These theorems have several corollaries and implications discussed in Brill (1975) or Brill & Posner (1981a).

3.4 Methods of Solution

Brill (1975) and Brill & Posner (1981a) shows that application of the basic system-point theorem yield mathematical functions of the probability density function evaluated at level \( w \). This gives rise to two distinct possible solution methodologies viz. analytical and via a Monte Carlo computational approach.

In the analytical approach, the above-mentioned integral equations for the pdf are solved to yield analytical solutions (see for example, Brill & Posner (1981a)). However, for models that yield equations which are difficult to solve, the system-point Monte Carlo computational approach may be used to compute the probability density function. Brill (1983) proved the
following theorem: For any sample-path

\[ D_t(w) = P_t^{>w} - T_t^{>w} \]
\[ I_t = P_t^{>0} - T_t^{>0} \]

In words, this says that for a sample path the number of
downcrossings of a level \( w > 0 \), is equal to the number of
peaks minus the number of troughs above that level. A
similar result holds for level-zero hits. Making use of
the above theorem followed by the basic system-point
theorem allows for an estimate of the pdf to be obtained.

By utilizing random numbers, a sample path for \( W(.) \) on
a finite interval \([0,t]\) is generated. The heights of peaks
and troughs are obtained dynamically as the sample path is
generated over time. Direct application of the above
theorems then yield estimates of the probability density
function of the waiting time. From the estimated density
function, and formulas in Brill (1983), estimates for
moments of the waiting time and the cumulative distribution
function can be found.

In this thesis, we adopt a Monte Carlo computational
approach to compute the probability density function of the
waiting time for each customer-type. In the next chapter
we define a system-point process for each customer-type and
construct typical sample functions for the respective
processes.
CHAPTER IV

SYSTEM-POINT MODELS FOR THE DIFFERENT
PRIORITY CLASSES

The queueing system considered is a variant of the
M/M/2 queue in which two types of customers arrive
independently with Poisson arrival rates $\lambda_j (j=1,2)$ and
require an exponentially distributed amount of service time
with mean $1/\mu_j (j=1,2)$ from either server. The queueing
discipline within each class is first-come, first-served
(FCFS); type-1 customers having service priority over
type-2 customers and service of either class, once started,
is never interrupted (non-preemptive). Customers arriving
when the system is empty go into either server with equal
probability.

To analyse this queueing system we use two stochastic
processes: system-point processes for type-1 (SP(1)) and
type-2 (SP(2)) customers, respectively. Section 4.1
defines the type-1 process, which is used in Chapter 5 to
calculate the probability density function and cumulative
distribution function of the waiting time of the type-1
customers. Similarly, Section 4.2 defines the type-2
process, which is used in the next chapter to compute the
corresponding type-2 characteristics. Typical sample paths are illustrated for both processes. Understanding the construction of sample paths is essential for being able to write a computer program to generate sample paths.

4.1 System-Point Process for Type-1 Customers—SP\(^{(1)}\)

This section treats the waiting-time characteristics for type-1 customers. The system-point process for the type-1 customers is the stochastic process

\[\{(W(1)(t); M(1)(t)), t \geq 0\}\],

where:

- \(W(1)(t)\) is the virtual wait of a type-1 customer arriving at time \(t\).
- \(M(1)(t)\) is the system configuration at time \(t\) defined by the vector \(M(1)(t) = (n_S, n_W, k)\) if a type-1 customer arriving at time \(t\) would "see" \(n_S\) type-2 customers in service, and \(n_W\) type-2 customers waiting \((n_S = 0, 1, 2; n_W = 0, 1, 2, \ldots)\). Variable \(k\) may assume the value 0, 1 or 2. If \(k = 1\) or 2, it represents the unit-type in the other server at the time-\(t\) arrival's (possibly future) service-starting epoch; if \(k = 0\), the time-\(t\) arrival would enter service alone. Recapitulating,

\[n_S = \text{number of type-2 customers in service at the arrival epoch } t \text{ of a potential type-1 customer,}\]

\[n_S = 0, 1, 2\]
\( n_w = \text{number of type-2 customers waiting (not in service)} \)

at the arrival epoch \( t \) of a potential type-1 customer, \( n_w = 0, 1, 2, \ldots \)

\( k = \text{unit-type in the other server at the (possibly future) service-starting epoch of a type-1 customer arriving at time } t, k = 1, 2; k = 0 \text{ indicates the other server is empty when the type-1 time-} t \text{ arrival enters service.} \)

Possible states for the system configuration:

\((0, 0; 0)\) — type-1 starts service as the only customer in the system

\((n_s, n_{w}; 1)\) — type-1 starts service with a type-1 customer in other server; arrives when 
\( n_s = 0, 1, 2; n_w = 0, 1, 2, \ldots \)

\((n_s, n_{w}; 2)\) — type-1 starts service with a type-2 customer in other server; arrives when 
\( n_s = 1, 2; n_w = 0, 1, 2, \ldots \)

States \((n_s, n_{w}; 0)\), with \( n_s = 1, 2; n_w \geq 0 \) and \((0, n_{w}; 2)\), with \( n_w \geq 0 \) are infeasible. The reason for the infeasibility is as follows. For state \((n_s, n_{w}; 0)\), with \( n_s = 1, 2; n_w \geq 0 \); if an arriving type-1 went into service alone, the system must have been empty when he arrived. No type-2 customers can be in the system when the system is empty. State \((0, n_{w}; 2)\), with \( n_w \geq 0 \) is not possible since
in order for a type-1 customer to have as his neighbor in service a type-2 customer, due to the priority discipline the type-2 must be in service at the type-1’s arrival epoch.

The system is said to be in state \( <w; (n_s, n_w; k)> \) at time \( t \) if the virtual wait is \( W(1)(t) = w \), and the system configuration is \( (n_s, n_w; k) \). State \( <0; (0,0;0)> \) indicates an empty system. If the system is in state \( <w; (n_s, n_w; k)> \) at time \( t \), the SP\(^{(1)}\) process sample path may be pictured as having planar Cartesian co-ordinate \( (t, w) \) corresponding to the particular configuration \( (n_s, n_w; k) \). For a system with \( N \) distinct configurations there will be up to \( N \) co-ordinate systems, called "pages" in SP theory [Brill & Posner (1981)]. The "pages" correspond to configurations for which \( W(1)(t) \) can have positive values. \( N \) may be countably infinite - as in this model. Some configurations may be represented by "lines". The "lines" correspond to configurations for which \( W(1)(t) \) can have the value zero only. The "system-point" (the leading point of the sample path) moves from page to page at arrival events and possibly at service completions of customers, tracing out a sample path with time - thus describing the state of the system at any time \( t \geq 0 \). The virtual waiting time \( W(1)(t) \), is defined as the time that a potential type-i arrival to the system at
time \( t \) would have to wait in queue before starting service. Thus it is a continuous-time process, \( t \geq 0 \). The system-point process is a generalization of the virtual waiting time process.

We now introduce the following definitions of the stationary partial cumulative distribution function and probability density function of the waiting time for the type-1 customers.

\[
F^1(w, (n_S, n_W; k)) = \text{partial waiting time distribution associated with configuration } (n_S, n_W; k) = \Pr \{ \text{a type-1 arriving customer waits } \leq w, \text{ and the configuration is } (n_S, n_W; k) \}, w \geq 0 \\
= \lim_{t \to \infty} \Pr\{W(1)(t) \leq w, M(1)(t) = (n_S, n_W; k)\}, w \geq 0 \\
f^1(w, (n_S, n_W; k)) = \text{"mixed" joint density function of the waiting time associated with configuration } (n_S, n_W; k) = \frac{dF^1(w, (n_S, n_W; k))}{dw}, w > 0, \text{ whenever the derivative exists.}
\]

\[
G^1(w) = \text{total waiting time distribution} \\
= \sum_{\text{all } (n_S, n_W; k)} F^1(w, (n_S, n_W; k)), w \geq 0 \\
g^1(w) = \text{total waiting time probability density function} \\
= \sum_{\text{all } (n_S, n_W; k)} f^1(w, (n_S, n_W; k)), w > 0
\]
Let \( P^1_{000} \), \( P^1_{001} \), and \( P^1_{102} \) denote the probabilities that an arriving type-1 customer does not wait and the configuration is \((0,0;0)\), \((0,0;1)\), or \((1,0;2)\) respectively at the time of arrival. Hence

\[
P^1_0 = \Pr\{ W^{(1)}(t) = 0 \text{ in steady-state } \}
\]

\[
= \lim_{t \to \infty} \Pr\{ W^{(1)}(t) = 0 \}
\]

\[
= G^1(0)
\]

\[
= P^1_{000} + P^1_{001} + P^1_{102}
\]

\( P^1_0 \) is the probability of zero waiting time in queue for a type-1 customer. It is the sum of all the probabilities of a zero waiting time state for the system. \( P^1_{000} \) is the probability that the system is empty and can be found by the use of balance equations (to be done in Chapter 5).

4.1.1 Illustration of a Typical SP\((1)\) Sample Path

We now demonstrate the construction of a typical sample path for the type-1 customers. Throughout the following description, tacit use is made of the memoryless property of the exponential distribution and the independence of the service times in the two servers. Let \( \tau_n \), \( n \geq 1 \) be the arrival epochs of the customers (either type).

The following description is for the sample path shown in
Figure 1. Its purpose is to show how an arbitrary sample path for $SP^{(1)}$ can be constructed.

$\tau_0$ - At $t=0$, the system is empty and so the leading point of the sample path, denoted by $SP^{(1)}$, is on line $(0,0;0)$.

$\tau_1$ - First arrival to the system; a type-1 customer. The system-point jumps to $(\tau_1^+,0)$ on page $(0,0;1)$. This new arrival goes into service immediately. Any newly arriving customer would go into service immediately, and his neighbor in service would be a type-1.

$\tau_2$ - A type-1 customer arrives before the first customer ends service. He goes into service immediately. $SP^{(1)}$ stays on page $(0,0;1)$ and jumps by an amount denoted by $\exp(2\mu_1)$, since any newly arriving type-1 would have to wait a time exponentially distributed with mean $1/2\mu_1$.

$\tau_3$ - Say another type-1 customer arrives. This customer must wait until one of the previous two customers complete service. $SP^{(1)}$ stays on page $(0,0;1)$ and jumps by an amount $\exp(2\mu_1)$, for the reason as at $\tau_2$.

$\tau_4$ - A type-2 customer arrives. The system-point jumps to page $(0,1;1)$, but maintains the same height above zero since any newly arriving type-1 customer must wait before entering service and this wait is unaffected by the type-2's presence due to the priority discipline employed.

$\tau_5$ - Another type-2 customer arrives. The system-point jumps to page $(0,2;1)$ but remains at the same height.
above zero, for the same reason given previously at $\tau_4$.

A - Assume no new type-1 arrivals occur between the time $\tau_5$ and $A$. At $t = A$, there would be a single type-1 customer in service since an arriving type-1 would wait zero time. One of the waiting type-2's would start service.

At $t = A^+$, the jump is to page $(1,1;1)$ with co-ordinate $(A^+, W(1)(A^+))$, where $W(1)(A^+)$ is exponentially distributed with mean $1/(\mu_1+\mu_2)$.

(The jump would be to page $(1,1;1)$ or page $(1,1;2)$ with probabilities $\mu_2/(\mu_1+\mu_2)$ and $\mu_1/(\mu_1+\mu_2)$ respectively since a potential type-1 customer arriving at time $A^+$ would "see" a waiting type-2 and a servicing type-2 and would start service with a type-1 or a type-2 as his neighbor, depending on whether the type-1 or type-2 jointly in service, completes service earlier. In Fig.1 it is assumed that the type-2 service was less).

$\tau_6$ - A type-1 customer arrives. Since $W(1)(\tau_6) > 0$, the type-2 customer that arrived at $\tau_5$ is waiting in queue, whilst this type-1 customer will go into service at time $\tau_6 + W(\tau_6)$. $SP(1)$ remains on page $(1,1;1)$ and jumps up an amount $\exp(2\mu_1)$ for a similar reason as at $\tau_2$ or $\tau_3$.

B - At $t = B$, the type-2 customer that started servicing at $t = A^+$ completes its service. Any newly arriving type-1 would "see" a waiting type-2 and would enter service some time later with his neighbor being a type-1. Hence, at
t=B⁺, \( SP(1) \) jumps to page \((0,1;1)\) but remains at the same height above zero.

C - Assume no new type-1 arrivals occur between \( t=B \) and \( t=C \). At \( t=C \) there would be a single type-1 customer in service. The waiting type-2 would start service. At \( t=C⁺ \), the jump is to page \((1,0;2)\) with co-ordinate \((C⁺,W(1)(C⁺))\), where \( W(1)(C⁺) \) is exponentially distributed with mean \( 1/(\mu₁+\mu₂) \). This jump to page \((1,0;2)\) occurs with probability \( \mu₁/(\mu₁+\mu₂) \) - the probability that a type-1 completes service before a type-2. Any potential type-1 arrival would "see" a type-2 in service and would start service with a type-2 as his neighbor.

D - Assume no new type-1 arrival occur between \( t=C \) and \( t=D \). At \( t=D \), the type-1 jointly in service with a type-2 completes its service. The \( SP(1) \) stays on page \((1,0;2)\). A newly arriving customer would go into service immediately with its neighbor being a type-2 customer. This represents a zero-waiting state for type-1 customers.

\( \tau_7 \) - A type-2 customer arrives before service completion of the type-2 presently receiving service and goes into service immediately. \( SP(1) \) jumps to page \((2,0;2)\) with co-ordinate \((\tau_7⁺,W(1)(\tau_7⁺))\), where \( W(1)(\tau_7⁺) \) is exponentially distributed with mean \( 1/(2\mu₂) \).

\( \tau_8 \) - A type-1 customer arrives. At \( t=\tau_8⁺ \), the jump is to
page (2,0;1) with co-ordinate \((\tau_8^+, W^{(1)}(\tau_8^+))\), where 
\[W^{(1)}(\tau_8^+) = W^{(1)}(\tau_8) + \text{an amount which is exponentially distributed with mean } 1/(\mu_1+\mu_2).\]  
The shown jump to page (2,0;1) occurs with probability \(1/(\mu_1+\mu_2)\), for any newly arriving type-1 at \(\tau_8^+\) would "see" two type-2's in service when he arrived but would start service in the future with a type-1 in the other server.

**E** - At \(t=E\), a type-2 customer departs. Any newly arriving type-1 would "see" one type-2 in service, zero type-2's waiting and would enter service with a type-1 as his neighbor. Thus at \(t=E^+\), \(SP^{(1)}\) jumps to page (1,0;1) but maintains the same height above zero.

**F** - The last type-2 customer in the system completes service leaving a single type-1 customer in the system. \(SP^{(1)}\) jumps to page (0,0;1) with co-ordinate \((F^+, 0)\). Any newly arriving customer would go into service immediately, and his neighbor would be a type-1.
Figure 1: Example of an $SP^{(1)}$ Sample Path

- $W_1(t)$ on page $(0,0,1)$
- $W_1(t)$ on page $(0,1,1)$
- $W_1(t)$ on page $(0,2,1)$
- $W_1(t)$ on page $(1,1,1)$
- $W_1(t)$ on page $(1,0,1)$
- $W_1(t)$ on page $(1,0,2)$
- $W_1(t)$ on page $(2,0,1)$
- $W_1(t)$ on page $(2,0,2)$

Customer Arrivals by Types
- $\circ$: $SP^{(1)}$ exit
- $\ast$: $SP^{(1)}$ enter
- $e(\cdot)$: exponential

$t$: Time

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
4.2 System-Point Process for Type-2 Customers—SP(2)

This section treats the waiting time characteristics for type-2 customers. Due to the priority discipline employed, the appropriate process requires a definition different from that used for type-1 customers (SP(1)).

4.2.1 The System-Point Process Definition for Type-2 Customers

The system-point process for the type-2 customers can be represented by the stochastic process

\( \{ \langle W(2)(t); M(2)(t) \rangle, t \geq 0 \} \)

where:

- \( W(2)(t) \) is the virtual waiting time of a type-2 customer arriving at time \( t \), and
- \( M(2)(t) \) is the system configuration at time \( t \) defined by \( M(2)(t) = k \); \( k \) is the type of customer the time-\( t \) type-2 arrival "sees" in the other server when he enters service at time \( t + W(2)(t) \), \( k = 1,2 \); or \( k = 0 \) if the system is empty when the type-2 arrives.

Based on this definition of the system configuration at time \( t \), the SP(2) process can be fully described by a line and two pages. The line represents the configuration which occurs at arrival epochs of customers who find the
system empty while pages 1 and 2 depicts the system configurations when a type-2 customer starts service with a type-1 or a type-2 as his neighbor respectively. The $Sp^{(2)}$ process is said to be in state $<w;k>$ at time $t$ if the virtual wait is $W^{(2)}(t) = w$, and the system configuration is $M^{(2)}(t) = k$. State $(0;0)$ is the empty system and is represented by a line. As in the case of $Sp^{(1)}$, the system-point process sample path for the type-2 customers $(Sp^{(2)})$ can be pictured as having planar Cartesian co-ordinate $(t,w)$ corresponding to the particular configuration $k = 1,2$.

Define for $Sp^{(2)}$, the following:

$F^{(2)}(w,k) = \text{partial waiting time distribution associated with configuration } k, k = 1,2$

$= Pr\{\text{type-2 unit waits } \leq w, \text{ and the configuration is } k\}, w \geq 0, k = 1,2$

$f^{(2)}(w,k) = \text{"mixed" joint density function for the waiting time associated with configuration } k$

$= dF^{(2)}(w,k)/dw, w > 0, \text{ for which the derivative exists.}$

$G^{(2)}(w) = \text{total waiting time distribution (stationary cumulative distribution function)}$

$= \sum_{k=1,2} F^{(2)}(w,k), w \geq 0$

$g^{(2)}(w) = \text{total waiting time density function (stationary probability density function).}$
\[ f(w, k) = \sum_{k=1,2} f^{(2)}(w, k), \quad w > 0 \]
\[ = f^{(2)}(w, 1) + f^{(2)}(w, 2), \quad w > 0 \]

Let \( P^{200}, P^{201} \) and \( P^{202} \) denote the probabilities that an arriving type-2 customer does not wait and the configuration is 0, 1, or 2 respectively, at the time of arrival. Further, let \( P^{20} \) be the probability that a type-2 customer will not wait on arrival to the system. Thus,

\[ P^{20} = \Pr\{W(2) = 0 \text{ in steady-state}\} = \lim_{t \to \infty} \Pr\{W(2)(t) = 0\} = G^{(2)}(0) = P^{200} + P^{201} + P^{202} \]

Due to the non-preemptive priority queueing discipline under which the system operates, newly arriving or waiting type-2 customers (low priority) are required to wait until all but one of any type-1 customers in the system complete servicing, before they can start being serviced. This is apparent in the construction of the \( SP^{(1)} \) sample path shown in Figure 1. There an arriving or waiting type-2 may only start service during periods when there is at most one type-1 present (if any) in the system, or if he (an arrival) "sees" an empty system. Hence, type-1 customers
serve as interference to the $SP^2$ system. In the next section, the possible kinds of "system busy periods" will be defined.

4.2.2 Kinds of System Busy Periods

Three different kinds of system busy periods need to be considered since jump sizes of the $SP^2$ sample path will be distributed like them. The scenarios giving rise to the three kinds of busy periods are now described.

(a) A type-1 customer is in the system alone and another type-1 arrives at time $\tau$ say, before this first unit completes service. These two type-1's in service initiate a "type-(1,1) busy period"; the first time from $\tau$ until there is exactly one type-1 in the system again. This time is distributed like an ordinary busy period in an M/M/1 queue except that the hazard rate is $2\mu_1$. Such a busy period is denoted by $B_{11}^0$. Figure 2 demonstrate a $B_{11}^0$ busy period and the jump size due to $B_{11}^0$ is illustrated by Example 1 in 4.2.4.

(b) A type-1 customer is in the system alone and a type-2 arrives at time $\tau$ say, before the type-1 completes service. These two units jointly in service initiate a "mixed busy period"; the first passage time from $\tau$ until a server becomes free and there are no type-1's
waiting. (The reverse arrival order yields the same "busy period"). The distribution of the jump size due to this kind of busy period is denoted by $B_{11}^\delta$ or $B_{12}^\delta$ and will be discussed in 4.2.3 and illustrated by Examples 1 & 2 in 4.2.4.

(c) A type-2 customer is in the system alone, followed by another type-2 before the initial type-2 completes its servicing. This initiates a busy period denoted by $B_{12}^\delta$; the distribution of which is given in 4.2.3 and illustrated by Example 2 in 4.2.4.

The "system" busy periods (which are jump sizes in the $SP(2)$ sample path) described above will be simulated during the generation of the $SP(2)$ process sample path. The following section defines the sizes of the jumps in terms of the possible system busy periods.

4.2.3 Possible Jump Sizes for $SP(2)$ Sample Path

Consider the following definitions:

\[ \exp(h) = \text{an exponential random variable with hazard rate } h \]  
\[ \text{(or mean } 1/h), \]
\[ \delta_{ij} = \exp(\mu_1+\mu_j) - \exp(\lambda_1), \]
\[ B_{11}^0 = \text{an ordinary busy period in an M/M/1 queue with hazard rate } 2\mu_1, \]
a delayed busy period in an M/M/1 queue with
hazard rate $2\mu_1$, and delay $\delta$, if $\delta > 0$,
(see Figure 3),
$B_{11}^\delta = 0$, if $\delta < 0$,

\[ B_{12}^\delta = \begin{cases} 
    \text{a delayed "mixed" busy period, if } \delta > 0, \\
    B_{12}^\delta = 0, \text{ if } \delta < 0,
\end{cases} \]

The distribution of the delayed "mixed" busy period,
$B_{12}^\delta$ can be one of the following three cases,

Case (1) : All jumps that propagate this busy period are
$\exp(\mu_1 + \mu_2)$, with a type-1 in service at the
end of the busy period.

Case (2) : Same as Case (1), except that a type-2 remains
in service at the end of the busy the period.

Case (3) : The jumps are a combination of $\exp(2\mu_1)$ and
$\exp(\mu_1 + \mu_2)$, the first string of jumps being
$\exp(\mu_1 + \mu_2)$, followed by a string of
$\exp(2\mu_1)$ jumps. Hence a type-1 will remain
in service at the end of the busy period.

Cases (1), (2) and (3) are illustrated in Figure 4.

The jumps of the virtual waiting time for the $SP(2)$
sample path will be one of the following six possible
sizes.

1) $B_{11}^0$ (special case is $\exp(2\mu_1)$)
2) \(\exp(\mu_1+\mu_2) + B_{11}^{\delta_{12}}\)
3) \(\exp(\mu_1+\mu_2) + B_{12}^{\delta_{12}}\)
4) \(\exp(\mu_1+\mu_2)\) (special case of (2) or (3) when \(\delta_{12}<0\))
5) \(\exp(2\mu_2) + B_{12}^{\delta_{22}}\)
6) \(\exp(2\mu_2)\) (special case of (5) when \(\delta_{22}<0\)).

In order to consolidate the foregoing concepts, two typical sample paths for the \(SP(2)\) process will be constructed and described in the next section. The jump sizes of the \(SP(2)\) sample path generated will be a reflection of the unit-type(s) of the first two customers initiating the "busy periods". Making use of the concept of the ordinary type-\((1,1)\) busy period, delayed type-\((1,1)\) busy period and delayed "mixed" busy period, typical sample paths of the virtual waiting time for the type-2 customers can be constructed as described next.

### 4.2.4 Examples of Typical \(SP(2)\) Sample Paths

#### Example 1

An \(SP(2)\) Sample Path with Jump Sizes Distributed as \(B_{11}^0, \exp(\mu_1+\mu_2), \exp(\mu_1+\mu_2) + B_{11}^{\delta_{12}}\), \(\exp(\mu_1+\mu_2) + B_{12}^{\delta_{12}}\). and the \(SP(2)\) moving from page 1 to page 2, (Figure 5).

\(\tau_0\) – At \(t=0\), the system is empty and so the leading point of the sample path, denoted by "\(SP(2)\)" is on line 0.
\( \tau_1 \) - A type-1 unit arrives and goes into service immediately. The system-point jumps to page 1 from line 0 and has coordinate \((\tau_1^+, 0)\). Any newly arriving customer would go into service immediately and his neighbor in service would be a type-1.

\( \tau_2 \) - Another type-1 arrives before the previous type-1 completes service. The jump size generated is an ordinary type-(1,1) busy period, \( B_{11}^0 \) as defined in 4.2.2(a) and illustrated in Figure 2. \( B_{11}^0 \) happens to be \( \exp(2\mu_1) \) in the case illustrated. Any newly arriving type-2 would go into service at A. The system becomes empty at \( t=B \) and the system-point jumps to line 0.

\( \tau_3 \) - A type-1 arrives to an empty system and enters service immediately. \( SP(2) \) jumps from line 0 to page 1 with coordinates \((\tau_3^+, 0)\).

\( \tau_4 \) - Say a type-2 arrives during the previous type-1 service time. Suppose the type-2 service time is less than the type-1 service and that \( \delta_{12} = \exp(\mu_1+\mu_2) - \exp(\lambda_1) > 0 \). Then a jump of size \( \exp(\mu_1+\mu_2) + B_{11} \delta_{12} \) occurs. The system becomes empty at \( t=F \); \( SP(2) \) jumps to line 0.

\( \tau_7 \) - A type-1 unit arrives and enters service immediately with \( SP(2) \) jumping from line 0 to page 1 and having coordinate \((\tau_7^+, 0)\).

\( \tau_8 \) - Another type-1 unit arrives during the service time of the previous type-1. The sample path jumps up by an amount \( B_{11}^0 \) again.
\( \tau_9 \) - A type-2 unit arrives. The \( SP(2) \) jumps up by an amount \( \exp(\mu_1+\mu_2) + B_{12} \delta_{12} \) since in this realization the type-1 jointly in service with the type-2, left first. \( SP(2) \) stays on page 1 in this realization (any newly arriving type-2 would enter service in future and "see" a type-1 as his neighbor, Case (1) or (3) of \( B_{12} \delta_{12} \)).

\( \tau_{12} \) - A type-2 unit arrives. The sample path jumps up by an amount \( \exp(\mu_1+\mu_2) \) and stays on page 1. This occurs with probability \( \mu_2/(\mu_1+\mu_2) \) since there are no type-1 arrivals between \( t=J \) and \( t=K \). The system becomes empty at \( t=L \). \( SP(2) \) jumps to line 0.

\( \tau_{13} \) - A type-1 arrives and starts service immediately. \( SP(2) \) jumps from line 0 to page 1 having coordinate \( (\tau_{13+},0) \).

\( \tau_{14} \) - Another type-1 arrives during the service time of the type-1 at \( \tau_{13} \), and enters service. The sample path jumps up by an amount \( B_{11}^0 \) once more.

\( \tau_{15} \) - A type-2 unit arrives. The sample path jumps by an amount \( \exp(\mu_1+\mu_2) + B_{12} \delta_{12} \) and moves to page 2 in this realization. Case (2) of \( B_{12} \delta_{12} \).

\( \tau_{18} \) - A type-2 unit arrives. The sample path jumps up by an amount \( \exp(2\mu_2) \) since in this realization his neighbor is a type-2 when he starts service at \( t=P \) (no type-1's arrive between \( t=P \) and \( t=Q \)). At \( t=R \) the system becomes empty once more; \( SP(2) \) jumps to line 0 and remains in this configuration until a new arrival occurs.
Example 2: An SP(2) Sample Path with Jump Sizes Distributed as \( \exp(2\lambda_2) \), \( \exp(2\lambda_2) + B_{12}^{\delta_{22}} \), \( \exp(\mu_1+\mu_2) \) and SP(2) moving from page 2 to page 1, (Figure 6).

\( \tau_0 \) - At \( t=0 \), the system is empty and so the leading point of the sample path, denoted by "SP(2)", is on line 0.

\( \tau_1 \) - A type-2 unit arrives and starts service immediately. The system-point jumps to \( (\tau_1^+,0) \) on page 2 since any newly arriving unit would have to wait zero time and his neighbor in service would be a type-2.

\( \tau_2 \) - Another type-2 arrives before the previous type-2 completes service, and starts service immediately. SP(2) stays on page 2 in this realization (since no type-1's arrive during \( \exp(2\mu_2) \)) and jumps by an amount \( \exp(2\mu_2) \). The system becomes empty at \( t=B \) and SP(2) jumps to line 0.

\( \tau_3 \) - Say another type-2 arrives. The SP(2) behaves as at \( \tau_1 \) and for the same reason.

\( \tau_4 \) - Another type-2 arrives during the previous type-2 service time. The jump size at \( \tau_4 \) is \( \exp(2\mu_2) + B_{12}^{\delta_{22}} \) and stays on page 2 in this realization. Case (2) of \( B_{12}^{\delta_{22}} \).

\( \tau_5 \) - A type-2 arrives. The jump size at \( \tau_5 \) is \( \exp(2\mu_2) \) since no type-1's arrive during \( t=D \) and \( t=E \). The system is empty at \( t=F \); SP(2) jumps to line 0.

\( \tau_7 \) - A type-2 arrives and starts service immediately.
SP(2) jumps from line 0 to page 2 since any newly arriving unit would enter service having zero wait and his neighbor would be a type-2.

τ8 - Another type-2 arrives during the service time of the type-2 at τ7 and starts service. SP(2) jumps to page 1 at τ8+ with co-ordinate (τ8+, W(2)(τ8+)), where W(2)(τ8+) consist of exp(2μ2) + B12 δ12

Case (1) or (3) of B12 δ12.

τ9 - Say another type-2 arrives. The jump size of SP(2) at τ9 is exp(μ1+μ2) and stays on page 1 in this realization. (This occurs with probability μ2/(μ1+μ2), since no type-1’s arrive during t = H and t = I).

If a type-1 unit had arrived between t=H and t=I a delayed "mixed" busy period would have to be added to the peak at τ9, i.e., exp(μ1+μ2) + B12 δ12 and the page on which the jump terminates depends on the unit-type ending the B12 δ12 busy period, i.e., Case (1), (3) - type-1 or Case (2) - type-2.
Figure 2: Generating an Ordinary Type-(1,1) Busy Period $B_{11}$

Customer Arrivals by Types

$B_{11}^0$

$B_{11}^0$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 3: Generating a Delayed Type-(1,1) Busy Period - $B_{11}^\delta$
Figure 4: Generating a Delayed 'Mixed' Busy Period - $B^\delta_{12}$

- Ends with a type-1 in service
- Ends with either a type-1 or a type-2 in service
Figure 5: $SP_{(2)}$ Sample Path with Jump Size Distributed as $B_{11} e^{(\mu_1+\mu_2)}$, $e^{(\mu_1+\mu_2)} B_{12}$, $e^{(\mu_1+\mu_2)} B_{21}$, $e^{(\mu_1+\mu_2)} B_{22}$

Customer Arrivals by Types

- $SP_{(2)}$ exit
- $SP_{(2)}$ enter, $e(\cdot)$: exponential

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 6: $SP^{(2)}$ Sample Path with Jump Sizes Distributed as $e(2\mu_2), e(2\mu_2) \delta_{22}$ and $e(\mu_1+\mu_2)$
CHAPTER V

SYSTEM-POINT MONTE CARLO COMPUTATION

The objective of this chapter is twofold: firstly to demonstrate the application of the system-point Monte Carlo computation technique used to evaluate the probability density function and cumulative distribution function of the virtual waiting time for each class of customer, and secondly to check on the correctness of the computer programs written.

Two FORTRAN programs were written to generate sample paths for the $SP^1$ and $SP^2$ processes, respectively. Recall that the $SP^1$ and $SP^2$ processes, model the waiting time characteristics of the high priority customers (type-1 units) and low priority customers (type-2 units), respectively. The computer programs were written in a modular manner to facilitate easy debugging and tracing. Appendix B provides a flowchart and the computer program listing for $SP^1$ while Appendix C contains similar information for $SP^2$. Using the output from either program, a third program computes the probability density function and cumulative distribution function for the
respective waiting times, and plot the resulting functions. A listing of this program is given in Appendix D.

To test the system—point computer programs written for the M/M/2 two—class priority system the following hypothetical parameters were assumed:

\[
\begin{align*}
\lambda_1 &= 15 \text{ customers/hour} \\
\lambda_2 &= 12 \text{ customers/hour} \\
\mu_1 &= 20 \text{ customers/hour} \\
\mu_2 &= 15 \text{ customers/hour}
\end{align*}
\]

In addition to the probability density function and cumulative distribution function, the following values will be computed:

1) \( P_{0s} = \Pr(\text{an empty system}) \)
2) \( P_{1s} = \Pr(\text{a single customer in the system of either type}) \)
3) \( E(W_{q1}) = \text{expected waiting time for a type-i (i=1,2) customer in queue} \)

where, the superscript "s" refers to the system.

During the validation and verification process the above three values will be used to check against known analytic results or those obtained via the "usual" system simulation approach. A brief outline of the system simulation approach for the queueing system under study is provided in Appendix E.
5.1 Illustration of the System-Point Monte Carlo Computation Approach

The purpose of this section is to illustrate how the system-point Monte Carlo computation approach is used to obtain the virtual waiting time probability density function and the cumulative distribution function for the M/M/2 non-preemptive priority queueing system. The approach is due to Brill (1983) and only the relevant aspects are illustrated here. Since this section is merely for illustration purposes, with no intention of achieving steady-state results, the system-point computer programs (SP(1) and SP(2)) were run for an arbitrary time of T = 1000 minutes.

The approach adopted in this study is to first select a set of waiting time values denoted by \( \{W_j\} \). Let \( P^{>W_j} \) and \( T^{>W_j} \), \( W_j > 0 \) denote the number of "peaks" and "troughs" above level \( w \) respectively, for the sample path generated. The number of downcrossings of level \( W_j > 0 \) in time \( t \) is given by

\[
D_t(W_j) = P^{>W_j} - T^{>W_j} \quad \text{...Brill (1983)}
\]

and an estimate of the density function by

\[
\hat{g}(W_j) = D_t(W_j)/t \quad \text{...Brill (1983)}
\]

In order to compute the cumulative distribution of the virtual waiting time there are two nonzero probabilities
that the wait of a type-i customer will be zero, that must be evaluated. This will be done individually for the respective system-point model.

5.1.1 \(\text{SP}^{(1)}\) Monte Carlo Computation

Table 1 gives the results for the generation of the \(\text{SP}^{(1)}\) sample path. Using the theory in Brill (1975) and Brill (1983), the following balance equations can be written for the \(\text{SP}^{(1)}\) process (the left side and right side are the \(\text{SP}^{(1)}\) exit and entrance rates of the state).

**State:**

\[
\begin{align*}
000 & : \quad \lambda P_{000} = \mu_1 P_{001} + \mu_2 P_{102} \\
001 & : \quad (\lambda + \mu_1)P_{001} = f_{001}(0^+) + f_{101}(0^+) + \lambda P_{000} \\
& \quad \quad \quad = I_{001}(t)/t + I_{101}(t)/t + T_{000\rightarrow001}(t)/t \\
102 & : \quad (\lambda + \mu_2)P_{102} = f_{102}(0^+) + f_{202}(0^+) + \lambda P_{000} \\
& \quad \quad \quad = I_{102}(t)/t + I_{202}(t)/t + T_{000\rightarrow102}(t)/t
\end{align*}
\]

where,

- \(P_{000}\) is the probability of an empty system,
- \(P_{001}\) is the probability of a single customer in the system and it's a type-1,
- \(P_{102}\) is the probability of a single customer in the system and it's a type-2,
- \(I_{ijk}(t)\) is the number of sample path level zero hits for configuration \((i,j;k)\) during a simulated time \(t\),
- \(f_{ijk}(0^+)\) is the estimated probability density function of the
waiting time at level $0^+$ for configuration
$(i,j;k)$.

$T_{000\to ijk}(t) =$ number of system-point transitions from
configuration $(0,0;0)$ to configuration $(i,j;k)$
during a simulated time $t$

$$\lambda = \lambda_1 + \lambda_2$$

$t =$ simulated time

On applying the above balance equations and making use
of the results available in Table 1, we can estimate the
probabilities of interest. Thus

$$\hat{P}_{001} = (0.021 + 0.021 + 0.036)/(0.45 + 0.3333)$$

$$= 0.0995748$$

$$\hat{P}_{102} = (0.038 + 0.027 + 0.029)/(0.45 + 0.25)$$

$$= 0.1342857$$

$$\hat{P}_{000} = \left[0.3333(0.0995748) + 0.25(0.1342857)\right]/0.45$$

$$= 0.1483615$$

The above estimates which are based on the $SP(1)$
process sample path can now be used to determine measures
for the system. Thus an estimate of the probability of an
empty system, $P_0^S$, is $\hat{P}_{000} = 0.1483615$. The
probability of a single customer in the system of either
type, $P_1^S$, is $\hat{P}_{000} + \hat{P}_{102} = 0.2338605$ and the
probability that a type-1 unit would wait zero time in
queue before entering service is $\hat{P}_{000} + \hat{P}_{001} + \hat{P}_{102} = 0.382222$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 1: Results from the SP(1) Monte Carlo Computation (T = 1000 min.)

\[ \lambda_1 = 15, \lambda_2 = 12, \mu_1 = 20, \mu_2 = 15. \]

- \# of Type-001 Impacts = 21
- \# of Type-101 Impacts = 21
- \# of Type-102 Impacts = 38
- \# of Type-202 Impacts = 27
- \# of 000 --> 001 Transitions = 36
- \# of 000 --> 102 Transitions = 29

<table>
<thead>
<tr>
<th>w</th>
<th>( p^{&gt;w} )</th>
<th>( t^{&gt;w} )</th>
<th>( g(w) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0+</td>
<td>376</td>
<td>147</td>
<td>0.2290</td>
</tr>
<tr>
<td>0.5</td>
<td>309</td>
<td>126</td>
<td>0.1830</td>
</tr>
<tr>
<td>1.0</td>
<td>258</td>
<td>105</td>
<td>0.1530</td>
</tr>
<tr>
<td>1.5</td>
<td>225</td>
<td>93</td>
<td>0.1320</td>
</tr>
<tr>
<td>2.0</td>
<td>195</td>
<td>72</td>
<td>0.1230</td>
</tr>
<tr>
<td>2.5</td>
<td>156</td>
<td>56</td>
<td>0.1000</td>
</tr>
<tr>
<td>3.0</td>
<td>133</td>
<td>48</td>
<td>0.0850</td>
</tr>
<tr>
<td>3.5</td>
<td>104</td>
<td>37</td>
<td>0.0670</td>
</tr>
<tr>
<td>4.0</td>
<td>83</td>
<td>30</td>
<td>0.0530</td>
</tr>
<tr>
<td>4.5</td>
<td>70</td>
<td>25</td>
<td>0.0450</td>
</tr>
<tr>
<td>5.0</td>
<td>55</td>
<td>19</td>
<td>0.0360</td>
</tr>
<tr>
<td>5.5</td>
<td>46</td>
<td>16</td>
<td>0.0300</td>
</tr>
<tr>
<td>6.0</td>
<td>41</td>
<td>12</td>
<td>0.0290</td>
</tr>
<tr>
<td>6.5</td>
<td>33</td>
<td>10</td>
<td>0.0230</td>
</tr>
<tr>
<td>7.0</td>
<td>28</td>
<td>9</td>
<td>0.0190</td>
</tr>
<tr>
<td>7.5</td>
<td>24</td>
<td>9</td>
<td>0.0150</td>
</tr>
<tr>
<td>8.0</td>
<td>19</td>
<td>7</td>
<td>0.0120</td>
</tr>
<tr>
<td>8.5</td>
<td>15</td>
<td>6</td>
<td>0.0090</td>
</tr>
<tr>
<td>9.0</td>
<td>14</td>
<td>5</td>
<td>0.0090</td>
</tr>
<tr>
<td>9.5</td>
<td>11</td>
<td>5</td>
<td>0.0060</td>
</tr>
<tr>
<td>10.0</td>
<td>9</td>
<td>3</td>
<td>0.0060</td>
</tr>
<tr>
<td>10.5</td>
<td>4</td>
<td>1</td>
<td>0.0030</td>
</tr>
<tr>
<td>11.0</td>
<td>3</td>
<td>1</td>
<td>0.0020</td>
</tr>
<tr>
<td>11.5</td>
<td>3</td>
<td>0</td>
<td>0.0030</td>
</tr>
<tr>
<td>12.0</td>
<td>3</td>
<td>0</td>
<td>0.0030</td>
</tr>
<tr>
<td>12.5</td>
<td>3</td>
<td>0</td>
<td>0.0030</td>
</tr>
<tr>
<td>13.0</td>
<td>3</td>
<td>0</td>
<td>0.0030</td>
</tr>
<tr>
<td>13.5</td>
<td>3</td>
<td>0</td>
<td>0.0030</td>
</tr>
<tr>
<td>14.0</td>
<td>2</td>
<td>0</td>
<td>0.0020</td>
</tr>
<tr>
<td>14.5</td>
<td>2</td>
<td>0</td>
<td>0.0020</td>
</tr>
<tr>
<td>15.0</td>
<td>1</td>
<td>0</td>
<td>0.0010</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
From the results given in Table 1, a graph of the probability density function for the $SP(1)$ ($T = 1000$ min.) can be constructed. Brill (1983) shows that the estimate of the probability density function, $\hat{g}(w)$, is a step-function in variable $w$ ($w > 0$). Figure 7 shows the plot for the density function of the waiting time for the $SP(1)$ system.

Having estimated the probability that a type-1 unit would wait zero time in queue when he arrives and the probability density function, the cumulative distribution of the waiting time for the high priority customers can now be obtained using the relevant formulas from Brill (1983), as illustrated in Figure 8.

The estimated probability density function of the waiting time for type-1 customers, namely its a step-function [Brill (1983)] can be used to find the expected wait for this class of units. For the results presented in Table 1 the expected wait in queue for the type-1 units, $\hat{E}(W_{q1})$ works out to be 0.0315603 hrs.
Figure 7: Probability Density Function for $\text{SP}^{(1)}$ ($T=1000$ min.)

$\lambda_1 = 15$, $\lambda_2 = 12$, $\mu_1 = 20$, $\mu_2 = 15$. 
Figure 8: Cumulative Distribution Function for $SP^{(1)}$  
($T=1000$ min.)  

$\lambda_1=15$, $\lambda_2=12$, $\mu_1=20$, $\mu_2=15$.  

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
5.1.2 $S P^{(2)}$ Monte Carlo Computation

As in the case of $S P^{(1)}$, before computing the cumulative distribution function of the virtual waiting time for the type-2 customers, the two nonzero probabilities that the wait of a type-2 customer will be zero must be evaluated. The required probabilities can be computed from the following set of balance equations for the $S P^{(2)}$ process.

State:

$00 : \quad \lambda P_{00} = \mu_1 P_{01} + \mu_2 P_{02}$

$01 : \quad (\lambda + \mu_1) P_{01} = f_{01}(0^+) + \lambda P_{00}$

$\hat{=} I_{01}(t)/t + T_{00 \rightarrow 01}(t)/t$

$02 : \quad (\lambda + \mu_2) P_{02} = f_{02}(0^+) + \lambda P_{00}$

$\hat{=} I_{02}(t)/t + T_{00 \rightarrow 02}(t)/t$

where,

$P_{00} = \text{Pr}(\text{an empty system})$

$P_{01} = \text{Pr}(\text{a single customer in the system and it's a type-1})$

$P_{02} = \text{Pr}(\text{a single customer in the system and it's a type-2})$

$I_{0k}(t) = \text{number of sample path level zero hits for configuration } k, \ k = 1,2 \ \text{during a simulated time } t$

$f_{0k}(0^+) = \text{estimated probability density function of the waiting time at level } 0^+ \text{ for configuration } k$
ToQ_0 > O_k(t) = number of system-point transitions from
configuration (0,0) to configuration (0,k)
during a simulated time t.

\[ \lambda = \lambda_1 + \lambda_2 \]
\[ t = \text{simulated time} \]

Application of the above balance equations, together
with the results presented in Table 2, allows for the
estimation of the non-zero probabilities. Thus

\[ \hat{P}_{01} = \frac{(0.041 + 0.031)}{(0.450 + 0.333)} = 0.0919188 \]
\[ \hat{P}_{02} = \frac{(0.058 + 0.032)}{(0.450 + 0.250)} = 0.1285714 \]
\[ \hat{P}_{00} = \frac{0.333(0.0919188) + 0.250(0.1285714)}{0.450} = 0.1395097 \]

Making use of the above estimates based on the SP(2)
process sample path, the following values for the system
can be determined. \( P_0^S \), the probability of an empty
system is \( \hat{P}_{00} = 0.135097 \). The probability of a single
customer in the system of either type, \( P_1^S \), is \( \hat{P}_{01} + \hat{P}_{02} = 0.2204902 \) and the probability that a type-2 unit
would wait zero time in queue before its start of service
is \( \hat{P}_{00} + \hat{P}_{01} + \hat{P}_{02} = 0.359999 \).

Table 2 is a listing of the results for the generation
of the SP(2) sample path (T = 1000 min.) from which a
Table 2: Results from the SP(2) Monte Carlo Computation (T = 1000 min.)

\[ \lambda_1 = 15, \quad \lambda_2 = 12, \quad \mu_1 = 20, \quad \mu_2 = 15. \]

- \# of Type-1 Impacts = 41
- \# of Type-2 Impacts = 58
- \# of 00 --> 01 Transitions = 31
- \# of 00 --> 02 Transitions = 32

<table>
<thead>
<tr>
<th>w</th>
<th>p &gt; w</th>
<th>t &gt; w</th>
<th>( \hat{g}(w) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0+</td>
<td>241</td>
<td>145</td>
<td>0.0960</td>
</tr>
<tr>
<td>0.5</td>
<td>207</td>
<td>145</td>
<td>0.0620</td>
</tr>
<tr>
<td>1.0</td>
<td>193</td>
<td>141</td>
<td>0.0520</td>
</tr>
<tr>
<td>1.5</td>
<td>183</td>
<td>132</td>
<td>0.0510</td>
</tr>
<tr>
<td>2.0</td>
<td>175</td>
<td>125</td>
<td>0.0500</td>
</tr>
<tr>
<td>2.5</td>
<td>161</td>
<td>115</td>
<td>0.0460</td>
</tr>
<tr>
<td>3.0</td>
<td>152</td>
<td>108</td>
<td>0.0440</td>
</tr>
<tr>
<td>3.5</td>
<td>148</td>
<td>100</td>
<td>0.0480</td>
</tr>
<tr>
<td>4.0</td>
<td>139</td>
<td>94</td>
<td>0.0450</td>
</tr>
<tr>
<td>4.5</td>
<td>134</td>
<td>86</td>
<td>0.0480</td>
</tr>
<tr>
<td>5.0</td>
<td>123</td>
<td>82</td>
<td>0.0410</td>
</tr>
<tr>
<td>5.5</td>
<td>116</td>
<td>79</td>
<td>0.0370</td>
</tr>
<tr>
<td>6.0</td>
<td>109</td>
<td>77</td>
<td>0.0320</td>
</tr>
<tr>
<td>6.5</td>
<td>99</td>
<td>75</td>
<td>0.0240</td>
</tr>
<tr>
<td>7.0</td>
<td>94</td>
<td>73</td>
<td>0.0210</td>
</tr>
<tr>
<td>7.5</td>
<td>90</td>
<td>69</td>
<td>0.0210</td>
</tr>
<tr>
<td>8.0</td>
<td>87</td>
<td>66</td>
<td>0.0210</td>
</tr>
<tr>
<td>8.5</td>
<td>84</td>
<td>62</td>
<td>0.0220</td>
</tr>
<tr>
<td>9.0</td>
<td>81</td>
<td>61</td>
<td>0.0200</td>
</tr>
<tr>
<td>9.5</td>
<td>75</td>
<td>57</td>
<td>0.0180</td>
</tr>
<tr>
<td>10.0</td>
<td>72</td>
<td>56</td>
<td>0.0160</td>
</tr>
<tr>
<td>10.5</td>
<td>69</td>
<td>54</td>
<td>0.0150</td>
</tr>
<tr>
<td>11.0</td>
<td>68</td>
<td>53</td>
<td>0.0150</td>
</tr>
<tr>
<td>11.5</td>
<td>63</td>
<td>51</td>
<td>0.0120</td>
</tr>
<tr>
<td>12.0</td>
<td>61</td>
<td>51</td>
<td>0.0100</td>
</tr>
<tr>
<td>12.5</td>
<td>60</td>
<td>51</td>
<td>0.0100</td>
</tr>
<tr>
<td>13.0</td>
<td>60</td>
<td>48</td>
<td>0.0120</td>
</tr>
<tr>
<td>13.5</td>
<td>58</td>
<td>45</td>
<td>0.0130</td>
</tr>
<tr>
<td>14.0</td>
<td>57</td>
<td>42</td>
<td>0.0150</td>
</tr>
<tr>
<td>14.5</td>
<td>54</td>
<td>41</td>
<td>0.0130</td>
</tr>
<tr>
<td>15.0</td>
<td>51</td>
<td>40</td>
<td>0.0110</td>
</tr>
<tr>
<td>15.5</td>
<td>48</td>
<td>39</td>
<td>0.0090</td>
</tr>
</tbody>
</table>
Table 2 (continued)

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Width</th>
<th>Height</th>
<th>Thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.0</td>
<td>45</td>
<td>38</td>
<td>0.0070</td>
</tr>
<tr>
<td>16.5</td>
<td>43</td>
<td>37</td>
<td>0.0060</td>
</tr>
<tr>
<td>17.0</td>
<td>43</td>
<td>37</td>
<td>0.0060</td>
</tr>
<tr>
<td>17.5</td>
<td>43</td>
<td>36</td>
<td>0.0070</td>
</tr>
<tr>
<td>18.0</td>
<td>43</td>
<td>35</td>
<td>0.0080</td>
</tr>
<tr>
<td>18.5</td>
<td>40</td>
<td>33</td>
<td>0.0070</td>
</tr>
<tr>
<td>19.0</td>
<td>39</td>
<td>33</td>
<td>0.0060</td>
</tr>
<tr>
<td>19.5</td>
<td>38</td>
<td>33</td>
<td>0.0050</td>
</tr>
<tr>
<td>20.0</td>
<td>38</td>
<td>31</td>
<td>0.0070</td>
</tr>
<tr>
<td>20.5</td>
<td>38</td>
<td>30</td>
<td>0.0080</td>
</tr>
<tr>
<td>21.0</td>
<td>38</td>
<td>29</td>
<td>0.0090</td>
</tr>
<tr>
<td>21.5</td>
<td>36</td>
<td>29</td>
<td>0.0070</td>
</tr>
<tr>
<td>22.0</td>
<td>35</td>
<td>28</td>
<td>0.0070</td>
</tr>
<tr>
<td>22.5</td>
<td>35</td>
<td>28</td>
<td>0.0070</td>
</tr>
<tr>
<td>23.0</td>
<td>34</td>
<td>27</td>
<td>0.0070</td>
</tr>
<tr>
<td>23.5</td>
<td>34</td>
<td>27</td>
<td>0.0070</td>
</tr>
<tr>
<td>24.0</td>
<td>31</td>
<td>27</td>
<td>0.0040</td>
</tr>
<tr>
<td>24.5</td>
<td>31</td>
<td>27</td>
<td>0.0040</td>
</tr>
<tr>
<td>25.0</td>
<td>31</td>
<td>26</td>
<td>0.0050</td>
</tr>
<tr>
<td>25.5</td>
<td>31</td>
<td>25</td>
<td>0.0060</td>
</tr>
<tr>
<td>26.0</td>
<td>31</td>
<td>25</td>
<td>0.0060</td>
</tr>
<tr>
<td>26.5</td>
<td>31</td>
<td>25</td>
<td>0.0060</td>
</tr>
<tr>
<td>27.0</td>
<td>31</td>
<td>25</td>
<td>0.0060</td>
</tr>
<tr>
<td>27.5</td>
<td>30</td>
<td>23</td>
<td>0.0070</td>
</tr>
<tr>
<td>28.0</td>
<td>30</td>
<td>22</td>
<td>0.0080</td>
</tr>
<tr>
<td>28.5</td>
<td>30</td>
<td>22</td>
<td>0.0080</td>
</tr>
<tr>
<td>29.0</td>
<td>30</td>
<td>21</td>
<td>0.0090</td>
</tr>
<tr>
<td>29.5</td>
<td>29</td>
<td>21</td>
<td>0.0080</td>
</tr>
<tr>
<td>30.0</td>
<td>29</td>
<td>21</td>
<td>0.0080</td>
</tr>
<tr>
<td>30.5</td>
<td>29</td>
<td>19</td>
<td>0.0100</td>
</tr>
<tr>
<td>31.0</td>
<td>28</td>
<td>16</td>
<td>0.0120</td>
</tr>
<tr>
<td>31.5</td>
<td>28</td>
<td>15</td>
<td>0.0130</td>
</tr>
<tr>
<td>32.0</td>
<td>24</td>
<td>12</td>
<td>0.0120</td>
</tr>
<tr>
<td>32.5</td>
<td>22</td>
<td>12</td>
<td>0.0100</td>
</tr>
<tr>
<td>33.0</td>
<td>19</td>
<td>10</td>
<td>0.0090</td>
</tr>
<tr>
<td>33.5</td>
<td>17</td>
<td>9</td>
<td>0.0080</td>
</tr>
<tr>
<td>34.0</td>
<td>15</td>
<td>7</td>
<td>0.0080</td>
</tr>
<tr>
<td>34.5</td>
<td>14</td>
<td>7</td>
<td>0.0070</td>
</tr>
<tr>
<td>35.0</td>
<td>13</td>
<td>7</td>
<td>0.0060</td>
</tr>
<tr>
<td>35.5</td>
<td>13</td>
<td>7</td>
<td>0.0060</td>
</tr>
<tr>
<td>36.0</td>
<td>11</td>
<td>7</td>
<td>0.0040</td>
</tr>
<tr>
<td>36.5</td>
<td>9</td>
<td>4</td>
<td>0.0050</td>
</tr>
<tr>
<td>37.0</td>
<td>9</td>
<td>4</td>
<td>0.0050</td>
</tr>
<tr>
<td>37.5</td>
<td>9</td>
<td>4</td>
<td>0.0050</td>
</tr>
<tr>
<td>38.0</td>
<td>8</td>
<td>4</td>
<td>0.0040</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 2 (continued)

<table>
<thead>
<tr>
<th>Value</th>
<th>Count</th>
<th>Another Count</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>38.5</td>
<td>7</td>
<td>3</td>
<td>0.0040</td>
</tr>
<tr>
<td>39.0</td>
<td>7</td>
<td>3</td>
<td>0.0040</td>
</tr>
<tr>
<td>39.5</td>
<td>7</td>
<td>3</td>
<td>0.0040</td>
</tr>
<tr>
<td>40.0</td>
<td>7</td>
<td>3</td>
<td>0.0040</td>
</tr>
<tr>
<td>40.5</td>
<td>7</td>
<td>3</td>
<td>0.0040</td>
</tr>
<tr>
<td>41.0</td>
<td>7</td>
<td>3</td>
<td>0.0040</td>
</tr>
<tr>
<td>41.5</td>
<td>7</td>
<td>3</td>
<td>0.0040</td>
</tr>
<tr>
<td>42.0</td>
<td>7</td>
<td>3</td>
<td>0.0040</td>
</tr>
<tr>
<td>42.5</td>
<td>6</td>
<td>3</td>
<td>0.0030</td>
</tr>
<tr>
<td>43.0</td>
<td>6</td>
<td>3</td>
<td>0.0030</td>
</tr>
<tr>
<td>43.5</td>
<td>6</td>
<td>2</td>
<td>0.0030</td>
</tr>
<tr>
<td>44.0</td>
<td>6</td>
<td>2</td>
<td>0.0030</td>
</tr>
<tr>
<td>44.5</td>
<td>6</td>
<td>2</td>
<td>0.0040</td>
</tr>
<tr>
<td>45.0</td>
<td>6</td>
<td>2</td>
<td>0.0040</td>
</tr>
<tr>
<td>45.5</td>
<td>3</td>
<td>1</td>
<td>0.0020</td>
</tr>
<tr>
<td>46.0</td>
<td>3</td>
<td>1</td>
<td>0.0020</td>
</tr>
<tr>
<td>46.5</td>
<td>3</td>
<td>1</td>
<td>0.0020</td>
</tr>
<tr>
<td>47.0</td>
<td>3</td>
<td>1</td>
<td>0.0020</td>
</tr>
<tr>
<td>47.5</td>
<td>3</td>
<td>0</td>
<td>0.0030</td>
</tr>
<tr>
<td>48.0</td>
<td>3</td>
<td>0</td>
<td>0.0030</td>
</tr>
<tr>
<td>48.5</td>
<td>2</td>
<td>0</td>
<td>0.0020</td>
</tr>
<tr>
<td>49.0</td>
<td>1</td>
<td>0</td>
<td>0.0010</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 9: Probability Density Function for $SP^{(2)}$ 
($T=1000$ min.)

$\lambda_1=15$, $\lambda_2=12$, $\mu_1=20$, $\mu_2=15$
Figure 10: Cumulative Distribution Function for $SP_2(T=1000 \text{ min.})$

$\lambda_1=15$, $\lambda_2=12$, $\mu_1=20$, $\mu_2=15$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
graph of the probability density function is constructed as illustrated in Figure 9. Using a similar approach to that for SP(1), the cumulative distribution of the waiting time for the low priority customers can be obtained as shown in Figure 10. Using the results given in Table 2 the expected wait in queue for the type-2 units, $E[W_{q2}]$ is found to be 0.140059 hrs.

Some interesting observations can be made on the basis of the results obtained from SP(1) and SP(2). A higher value of $E[W_{q2}]$ was obtained as is expected since newly arriving or waiting type-2 units (low priority) are required to wait until all but one of any type-1 units in the system completes servicing, before they can start being serviced. Estimates of $P_0^s$ and $P_1^s$ by either system-point models are close together in value as expected since they both represent system probabilities for all unit-types (despite lack of steady-state). Also, estimates of the Pr(zero wait) for both unit-types are quite close together in value as might be expected since this probability is equal to $P_0^s + P_1^s$ which is independent of the unit-type. The Pr(type-1 unit wait zero) = $P_0^s + P_1^s$, since a newly arriving type-1 to the system will have to wait zero if the system is empty or if there is a single unit of either type in service when he arrives. Similarly, Pr(type-2 unit waits zero) = $P_0^s + P_1^s$, 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
since a newly arriving type-2 unit to the system will have a zero wait if there are no units in service (empty system) or a single unit of either type in service.

Finally, it must be emphasized here that the estimated values of $E[W_{q1}]$, $g^i(w)$, $G^i(w)$, $P_0$, and $P_1$ are not "steady-state" values since the time duration of the sample path generation was limited to $T = 1000$ minutes. The next chapter will be concerned with establishing "steady-state" results for the system-point models.

5.2 Verification of the System-Point Priority M/M/2 Computer Programs

The intention of this section is to verify the system-point Monte Carlo computation computer programs written for the system under consideration. The approach taken is to adjust the model parameters so that it represents very closely a system with known analytical solution and compare this with the model results. Based on the literature review we undertake two tests in the verification process.

In the first instance, the model parameters are selected so that the $SP^{(1)}$ and the $SP^{(2)}$ models have to handle only a small proportion of type-2 customers (low...
priority) and type-1 customers (high priority), respectively. The service-time distribution is made the same for both classes. Hence the models will operate almost as non-priority M/M/2 queueing systems for the respective customer-type. For such non-priority M/M/2 systems there are known analytic results.

Using this approach, the SP(1) was checked with the following parameters:

\[
\begin{align*}
\lambda_1 &= 15 \text{ units/hour} \\
\lambda_2 &= 0.01 \text{ units/hour} \\
\mu_1 &= 15 \text{ units/hour} \\
\mu_2 &= 15 \text{ units/hour}
\end{align*}
\]

and the SP(2) with

\[
\begin{align*}
\lambda_1 &= 0.01 \text{ units/hour} \\
\lambda_2 &= 12 \text{ units/hour} \\
\mu_1 &= 15 \text{ units/hour} \\
\mu_2 &= 15 \text{ units/hour}
\end{align*}
\]

Sample paths were generated over a time duration of \( T = 1000 \) minutes. Table 3 shows both the theoretical and estimated values for \( \text{SP}(1) \) and \( \text{SP}(2) \) of expected wait in queue, probability of an empty system and the probability of a single customer in the system of either type. The results given in Table 3 indicates that the \( \text{SP}(1) \) model produces an extremely accurate estimate of the average waiting time in queue and only slightly overestimates the probability of an empty system and the probability of a single customer in the system of either
type. The difference between $E(W_{q1})$ and $\bar{W}_{q1}$ (within 0.05%) is almost negligible whilst the values of $P_0^S$ and $\hat{P}_1^S$ are close to $\bar{P}_0^S$ and $\bar{P}_1^S$ (within 2.56%).

As for the results produced by $SP^{(2)}$, we notice that the estimate of the average waiting time for type-2 customers differ from the theoretical by 5.49%. A possible explanation for this is the fact that $T = 1000$ minutes is too small a simulated time for the $SP^{(2)}$ process to attain "steady-state". On the other hand, the estimates of the probability of an empty system and the probability of a single unit in the system of either type are extremely accurate.

As a further means of checking the $SP^{(1)}$ and $SP^{(2)}$ modelling, we can make use of the well known equation for the pdf of the waiting time in the simple $M/M/2$ queue [Gross & Harris (1985)] by comparing it with the system-point estimated pdfs. Tables 4 and 5 shows a set of $t > 0$ values with the corresponding theoretical and $SP^{(i)}$ ($i = 1,2$) estimated pdfs, respectively. Plots of the data are shown in Figures 11 and 12. Both plots indicate that the estimated pdfs lies closely on either sides of the theoretical values, which is expected since $E[W_{q1}]$ values are very close to the theoretical, $\bar{W}_{q1}$, discussed above.
Table 3: Theoretical Values and S-P Estimates for an Approximated Simple M/M/2 Queueing System with Parameters $\lambda_1=15$, $\lambda_2=0.01$, $\mu_1=\mu_2=15$ for Type-1 (SP(1)) and $\lambda_1=0.01$, $\lambda_2=12$, $\mu_1=\mu_2=15$ for Type-2 (SP(2)) (Unit of Time = 1 hour)

<table>
<thead>
<tr>
<th>Method of Analysis</th>
<th>Measure of Performance</th>
<th>Customer Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Type-1</td>
</tr>
<tr>
<td>Analytical*</td>
<td>$W_{qi}$ (hrs.)</td>
<td>0.022222</td>
</tr>
<tr>
<td>Eqs. (a.1.1)</td>
<td>$P_0^S$</td>
<td>0.333333</td>
</tr>
<tr>
<td>(a.1.2), (a.1.3)</td>
<td>$P_1^S$</td>
<td>0.333333</td>
</tr>
<tr>
<td>System-Point</td>
<td>$E(W_{qi})$ (hrs.)</td>
<td>0.022233</td>
</tr>
<tr>
<td>Computation</td>
<td>$P_0^S$</td>
<td>0.341658</td>
</tr>
<tr>
<td>$T = 1000$ mins.</td>
<td>$P_1^S$</td>
<td>0.341886</td>
</tr>
</tbody>
</table>

* The theoretical calculations are performed in Appendix A.
Table 4: Theoretical and $SP^{(1)}$ Estimates ($T = 1000$ min.) of PDF for a Simple M/M/2 Queue with Parameters $\lambda_1=15$, $\lambda_2=0.01$, $\mu_1=\mu_2=15$ (Unit of Time = 1 hour)

<table>
<thead>
<tr>
<th>$t$ (min.)</th>
<th>Theoretical*</th>
<th>$SP^{(1)}$ Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0649</td>
<td>0.0560</td>
</tr>
<tr>
<td>2</td>
<td>0.0506</td>
<td>0.0430</td>
</tr>
<tr>
<td>3</td>
<td>0.0394</td>
<td>0.0310</td>
</tr>
<tr>
<td>4</td>
<td>0.0307</td>
<td>0.0320</td>
</tr>
<tr>
<td>5</td>
<td>0.0239</td>
<td>0.0270</td>
</tr>
<tr>
<td>6</td>
<td>0.0186</td>
<td>0.0230</td>
</tr>
<tr>
<td>7</td>
<td>0.0145</td>
<td>0.0180</td>
</tr>
<tr>
<td>8</td>
<td>0.0113</td>
<td>0.0140</td>
</tr>
<tr>
<td>9</td>
<td>0.0088</td>
<td>0.0100</td>
</tr>
<tr>
<td>10</td>
<td>0.0069</td>
<td>0.0090</td>
</tr>
<tr>
<td>11</td>
<td>0.0053</td>
<td>0.0060</td>
</tr>
<tr>
<td>12</td>
<td>0.0042</td>
<td>0.0010</td>
</tr>
<tr>
<td>13</td>
<td>0.0032</td>
<td>0.0010</td>
</tr>
<tr>
<td>14</td>
<td>0.0025</td>
<td>0.0010</td>
</tr>
</tbody>
</table>

* Values obtained using Eq. (a.1.4) in Appendix A.
Figure 11: Theoretical and $S^{(1)}$ Estimate for Simple M/M/2 Queue (T=1000 min.)
Table 5: Theoretical and $SP^{(2)}$ Estimates ($T = 1000$ min.) of PDF for a Simple M/M/2 Queue with Parameters $\lambda_1=0.01$, $\lambda_2=12$, $\mu_1=\mu_2=15$ (Unit of Time = 1 hour)

<table>
<thead>
<tr>
<th>$t$ (min.)</th>
<th>Theoretical*</th>
<th>$SP^{(2)}$ Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0509</td>
<td>0.0550</td>
</tr>
<tr>
<td>2</td>
<td>0.0377</td>
<td>0.0330</td>
</tr>
<tr>
<td>3</td>
<td>0.0279</td>
<td>0.0250</td>
</tr>
<tr>
<td>4</td>
<td>0.0207</td>
<td>0.0170</td>
</tr>
<tr>
<td>5</td>
<td>0.0153</td>
<td>0.0140</td>
</tr>
<tr>
<td>6</td>
<td>0.0114</td>
<td>0.0110</td>
</tr>
<tr>
<td>7</td>
<td>0.0084</td>
<td>0.0110</td>
</tr>
<tr>
<td>8</td>
<td>0.0062</td>
<td>0.0100</td>
</tr>
<tr>
<td>9</td>
<td>0.0046</td>
<td>0.0100</td>
</tr>
<tr>
<td>10</td>
<td>0.0034</td>
<td>0.0070</td>
</tr>
<tr>
<td>11</td>
<td>0.0025</td>
<td>0.0050</td>
</tr>
<tr>
<td>12</td>
<td>0.0019</td>
<td>0.0030</td>
</tr>
<tr>
<td>13</td>
<td>0.0014</td>
<td>0.0010</td>
</tr>
</tbody>
</table>

* Values obtained using Eq. (a.1.4) in Appendix A.
Figure 12: Theoretical and $SP(2)$ Estimate for
Simple M/M/2 Queue (T=1000 min.)
In the second instance, the model parameters are selected so that the service rates for both classes are the same. The resultant model is a non-preemptive priority queueing system with the same service time distribution for either class. This queueing system was solved analytically by Cobham (1954) and Davis (1966).

Thus, (see Table 6) for the purpose of verification, both system-point models were run with the following parameters:

\[
\begin{align*}
\lambda_1 &= 15 \text{ customers/hour} \\
\lambda_2 &= 12 \text{ customers/hour} \\
\mu_1 &= 20 \text{ customers/hour} \\
\mu_2 &= 20 \text{ customers/hour}
\end{align*}
\]

For the model parameters given above, the sample path generations were performed for \( T = 1000 \) minutes. Table 6 below illustrates the theoretical and system-point estimated values for \( SP(1) \) and \( SP(2) \) of expected wait in queue, probability of an empty system and the probability of a single unit of either type in the system. From the entries in Table 6, it is clearly obvious that the \( SP(1) \) and \( SP(2) \) models produce an accurate estimate of the average waiting time in queue, \( P_0^S \) and \( P_1^S \), respectively.

Table 7 present results for another set of model parameters in which the arrival rate of the type-2's is twice that of the type-1's. Similar conclusions can be made for the results in this table as were stated for Table 6.
Table 6: Theoretical Values and S-P Estimates for a Non-Preemptive Priority Queueing System with Parameters $\lambda_1=15$, $\lambda_2=12$, $\mu_1=\mu_2=20$
(Unit of Time = 1 hour).

<table>
<thead>
<tr>
<th>Method of Analysis</th>
<th>Measure of Performance</th>
<th>Customer type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Type-1</td>
</tr>
<tr>
<td>Analytical*</td>
<td>$\bar{W}_{qi}$ (hrs.)</td>
<td>0.021761</td>
</tr>
<tr>
<td>Eqs. (a.2.1),</td>
<td>$\bar{P}_0^S$</td>
<td>0.194029</td>
</tr>
<tr>
<td>(a.1.1), (a.1.3)</td>
<td>$\bar{P}_1^S$</td>
<td>0.261940</td>
</tr>
<tr>
<td>System-Point</td>
<td>$E[\bar{W}_{qi}]$ (hrs.)</td>
<td>0.022279</td>
</tr>
<tr>
<td>Computation</td>
<td>$\hat{P}_0^S$</td>
<td>0.189107</td>
</tr>
<tr>
<td>T=1000 mins.</td>
<td>$\hat{P}_1^S$</td>
<td>0.255319</td>
</tr>
</tbody>
</table>

* The theoretical calculations are performed in Appendix A.
Table 7: Theoretical Values and S-P Estimates for a Non-Preemptive Priority Queueing System with Parameters \( \lambda_1 = 6, \lambda_2 = 12, \mu_1 = \mu_2 = 24 \) (Unit of Time = 1 hour).

<table>
<thead>
<tr>
<th>Method of Analysis</th>
<th>Measure of Performance</th>
<th>Customer Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Type-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type-2</td>
</tr>
<tr>
<td>Analytical</td>
<td>( \bar{W}_{qi} ) (hrs.)</td>
<td>0.004673</td>
</tr>
<tr>
<td></td>
<td>( \hat{P}_0^s )</td>
<td>0.454545</td>
</tr>
<tr>
<td></td>
<td>( \hat{P}_1^s )</td>
<td>0.340909</td>
</tr>
<tr>
<td>System-Point Computation</td>
<td>( \hat{E}[W_{qi}] ) (hrs.)</td>
<td>0.004730</td>
</tr>
<tr>
<td></td>
<td>( \hat{P}_0^s )</td>
<td>0.449524</td>
</tr>
<tr>
<td></td>
<td>( \hat{P}_1^s )</td>
<td>0.337143</td>
</tr>
<tr>
<td>T=1000 mins.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This section attempted to verify the system-point computer programs written to generate sample paths of the stochastic processes designed to model the queueing system under study. Based on the two tests performed in this section it is obvious that the system-point computer programs produce accurate estimates of the measures of performance of interest. Since the estimates are very close to the analytical values this gives us some degree of confidence in the correctness of the computer programs and our system-point model definitions. In the next chapter the verified computer programs will be used to establish "steady-state" results for the queueing system which concerns us.
CHAPTER VI

SYSTEM-POINT MODEL RESULTS

The preceding chapter was concerned with illustrating the system-point Monte Carlo computation technique and verification of the computer programs written to generate sample paths for the stochastic processes. Our objective in this chapter is to provide "steady-state" results of the M/M/2 two-class non-preemptive priority system having different service-time distribution for each customer-type (the main model treated in this thesis). A check was performed to demonstrate that the system-point approach accurately models the priority queueing system under study. Since for such a system theoretical results are non-existent, to do this check we resort to a large-scale system simulation model to verify our results obtained using the system-point Monte Carlo computation approach.

6.1 Check on the System-Point Priority M/M/2 Results

As stated above, the verification involves the comparison of the system-point results with those obtained via a system simulation modelling approach. For this
purpose a large-scale discrete-event system simulation model was written to simulate the M/M/2 priority system with two classes of customers. A brief discussion of this modelling approach is provided in Appendix E. Average waiting time of each customer-type and the probability that a given server is idle, are the measures of performance analysed. Antithetic sampling was used as a variance reduction technique. Five independent replications were made, independence between runs being accomplished by using a different seed each time for the random number generator. For each replication, the simulation was also performed using its complementary random numbers. Thus, five pairs of simulation runs were performed, each pair being correlated. Since we are interested in long-run measures, the system simulation was run for an arbitrary long time of $T = 10000$ minutes using the hypothetical model parameters $\lambda_1 = 15$, $\lambda_2 = 12$, $\mu_1 = 20$, $\mu_2 = 15$ (unit of time = 1 hour).

6.1.1 Check on the Waiting Time Estimates

Table 8 below shows the mean value of the waiting time, its standard error and a 95% confidence interval of the mean waiting time for each customer-type. All calculations for the system simulation are performed in Appendix E. A difference of means test was conducted on the output of the
simulation and the conclusion is that type-1 and type-2 mean waiting times are significantly different at $\alpha = 0.05$. This difference in waiting time is expected because of the priority discipline under which the system operates. Further, with 95% confidence we can conclude that type-1 mean waiting time is smaller than type-2 mean waiting time, the difference is within a range of 0.075080 to 0.115999 hours.

Initially the system-point models were allowed to operate for $T = 5000$ minutes with the assumed model parameters. Estimates of the average waiting time values, $\hat{E}[W_q^i]$, $Pr$ (an empty system), $\hat{P}_0^S$ and $Pr$ (a single customer of either type in the system) = $\hat{P}_1^S$, generated by the system-point method for the priority system are given in the first row of Table 9. A comparison of row one with row three (which contains the system simulation estimates), reveals that the average waiting time values generated by the system-point method for both customer-type fall within the 95% confidence interval produced by the system simulation model. Thus the system-point Monte Carlo computation approach generates an accurate estimate for the virtual waiting time probability density function for both type-1 and type-2 units.
To further observe the behavior of the system-point models, the time duration was increased to $T = 10000$ minutes. In so doing, the system-point computer programs now handle a greater number of customers of either type. The second row of Table 9 provides the results for this case. It demonstrates that the average waiting time value generated by the system-point method for the type-1 units fall outside the 95% confidence interval of the system simulation model. However, there is only a 0.81% difference between the value of $\hat{E}[W_{q1}]$ and the upper 95% confidence interval point. With regards to the type-2 units, the average waiting time value generated by the system-point method falls within the 95% confidence interval generated by the system simulation.

From Table 9 we observe that there are no significant changes in the estimates produced by either $SP^{(1)}$ or $SP^{(2)}$ by increasing the time duration under which the system-point models operate. This suggest that the models are operating under "steady-state" conditions and consequently our estimates of average waiting times are "steady-state" values.
Table 8: System Simulation Results (T = 10000 min.)
with Model Parameters \( \lambda_1=15, \mu_2=12, \mu_2=20, \mu_2=15 \)
(Unit of Time = hours).

<table>
<thead>
<tr>
<th>Customer Type</th>
<th>Type-1</th>
<th>Type-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Waiting Time (hrs.)</td>
<td>0.030127</td>
<td>0.125667</td>
</tr>
<tr>
<td>Standard Error of the Mean Waiting Time (hrs.)</td>
<td>0.000895</td>
<td>0.007332</td>
</tr>
<tr>
<td>Confidence Interval for Mean Waiting Time (hrs.) ( \alpha = 0.05 )</td>
<td>[0.027643, 0.032611]</td>
<td>[0.105314, 0.146019]</td>
</tr>
<tr>
<td>Confidence Interval of the Difference Between Type-1 &amp; Type-2 Waiting Times (hrs.) ( \alpha = 0.05 )</td>
<td>[0.075080, 0.115999]</td>
<td></td>
</tr>
<tr>
<td>Difference of Means Test ( \alpha = 0.05 )</td>
<td>table ( t = 2.770 )</td>
<td>test ( t = 12.9351 )</td>
</tr>
</tbody>
</table>
Table 9: M/M/2, 2-Class Priority Model Results.
(Model Parameters are $\lambda_1=15$, $\lambda_2=12$, $\mu_1=20$, $\mu_2=15$, (Unit of Time = 1 hour)).

<table>
<thead>
<tr>
<th>Method of Analysis</th>
<th>Measure of Performance</th>
<th>Customer Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Type-1</td>
</tr>
<tr>
<td>System-Point Computation</td>
<td>$E[W_q]$ hrs. $P_0^S$</td>
<td>0.032370</td>
</tr>
<tr>
<td>$T = 5000$ min.</td>
<td>$P_1^S$</td>
<td>0.129595</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.204176</td>
</tr>
<tr>
<td>System-Point Computation</td>
<td>$E[W_q]$ hrs. $P_0^S$</td>
<td>0.032876</td>
</tr>
<tr>
<td>$T = 10000$ min.</td>
<td>$P_1^S$</td>
<td>0.126477</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.197967</td>
</tr>
<tr>
<td>System Simulation</td>
<td>95% Confidence Interval for Average Waiting (hrs.)</td>
<td>[0.027643, 0.032611]</td>
</tr>
<tr>
<td>$T = 10000$ min.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>95% Confidence Interval for $Pr$(any server idle)</td>
<td>[0.2331, 0.2462]</td>
</tr>
</tbody>
</table>
6.2 Estimates for the Pr(any server idle)

Whilst comparing the system-point estimates obtained for Pr(0 units in the system) to that obtained via another method would be interesting, the unavailability of such a result forces us to resort to the following measure. For the two-server queueing system considered, a given server is idle if there are no customers in the system or if there is one customer (of either type) in the system and that customer is being served by the other server. That is

\[ \text{Pr(any server idle)} = P_0^S + \frac{1}{2} P_1^S \]

From the system simulation, a 95% confidence interval for \( \text{Pr(any server idle)} \) lies in the range 0.2331 to 0.2462. Estimates of \( P_0^S \) and \( P_1^S \) generated by the system-point method for both \( \text{SP}^{(1)} \) and \( \text{SP}^{(2)} \) are given in Table 9 for the two simulated time periods viz., \( T = 5000 \) and \( T = 10000 \) minutes.

Consider the case when \( T = 5000 \) minutes:

For \( \text{SP}^{(1)} \) an estimate of \( \text{Pr(any server idle)} \) = 0.23168 and for \( \text{SP}^{(2)} \) it is 0.24873. Whilst both estimates fall outside the 95% confidence interval generated by the system simulation, they are quite close to each other (as they should, since they are both probability estimates of the same event). Also it is worth noting that one system simulation replication produced a value lower than while two replications produced values greater than those
generated by SP(1) and SP(2), respectively.

Now consider the case when $T = 10000$ minutes.
In this case SP(1) estimates the $Pr(\text{any server idle})$ as 0.22546 while SP(2) produces a value of 0.24404. Here a similar type of result is obtained for SP(1) as for $T = 5000$ min. It is interesting to note however that SP(2) estimate of $Pr(\text{any server idle})$ falls within the 95% confidence interval given by the system simulation.
CHAPTER VII

GENERALIZATIONS AND SUMMARY

In this chapter we discuss extensions of the system-point model for the two-server two-class priority system analysed in the previous chapters. It is not uncommon for a priority system of the type considered in this thesis to have more than two servers and/or more than two customer-types. Thus it will be of significance if some insight can be provided as to how modelling these more complex cases can be accomplished. In Section 7.1 we present an approximation for the waiting time distribution of customers of any priority level in an M/M/2 queue with an arbitrary number of priority levels. In 7.2 we model an M/M/3 queue with two customer-types. The final section presents a summary of the work undertaken in this thesis and results obtained.

7.1 Approximation of the CDF in an M/M/2 N-Class System

Use can be made of the M/M/2 2-class priority model analysed in this thesis to obtain both upper and lower bounds for the cumulative distribution function of the
virtual waiting time of type-\(k\) units for an M/M/2 class model. Suppose there are \(N\) Poisson arrival rates \(\lambda(1), \ldots, \lambda(k), \ldots, \lambda(N)\) and \(N\) exponential service rates \(\mu(1), \ldots, \mu(k), \ldots, \mu(N)\). To obtain an upper bound for the virtual waiting time cumulative distribution function of type-\(k\) units consider the above system as a "two-class" priority system as follows.

Let type-1 to type-\(k\) comprise high priority customers. The arrival rate of the high priority customers is then the sum of the type-1 to type-\(k\) arrival rates, that is,

\[
\lambda^{(1)} = \lambda(1) + \ldots + \lambda(k) \tag{7.1.1}
\]

The expected service rate of the high priority units is taken to be the maximum value of type-1 to type-\(k\) service rates, that is,

\[
\mu^{(1)} = \max(\mu(1), \ldots, \mu(k)) \tag{7.1.2}
\]

Further, let type-(\(k+1\)) to type-\(N\) comprise the low priority customers. Thus,

\[
\lambda^{(2)} = \lambda(k+1) + \ldots + \lambda(N) \tag{7.1.3}
\]

\[
\mu^{(2)} = \max(\mu(k+1), \ldots, \mu(N)) \tag{7.1.4}
\]

Based on this approximation the type-\(k\) waiting time will be stochastically less than the true type-\(k\) waiting time in the actual model since type-\(k\)’s will now wait less. Hence the approximate cumulative distribution function of the virtual waiting time will lie above the true distribution function as depicted in Figure 13.

To derive a lower bound for the cumulative distribution
Figure 13: Bounds on Type-k CDF for an N-Class Priority System
Figure 14: Bounds on Type-2 CDF for an M/M/2 3-Class Priority Queue.

$\lambda_1 = 4$, $\lambda_2 = 5$, $\lambda_3 = 3$, $\mu_1 = 18$, $\mu_2 = 15$, $\mu_3 = 20$. 

Upper Bound

Lower Bound
function of the virtual waiting time we proceed in the following manner. Let type-1 to type-(k-1) comprise the high priority units and type-k to type-N the low priority units. Also, let
\begin{align*}
    \mu^{(1)} &= \min(\mu_1, \ldots, \mu_{k-1}) \quad (7.1.5) \\
    \mu^{(2)} &= \min(\mu_k, \ldots, \mu_N) \quad (7.1.6)
\end{align*}
In this case the type-k waiting time will be stochastically larger than the type-k waiting time in the actual model since type-k's will wait longer. Thus the approximate cumulative distribution function of the virtual waiting time will lie below the true distribution function as shown in Figure 13.

The above approach gives us a procedure for finding bounds on the distribution function for any unit-type in an M/M/2 queue with any number of priority levels using the S-P models formulated in Chapter 4. An actual case is presented in Figure 14 for N=3 and k=2. It can provide useful information for one interested in a good (not necessary exact) idea of the distribution for a given unit-type. The possibility of obtaining sharper bounds will be investigated in future work.

7.2 M/M/3 Priority System with Two Customer-Types

In this section an outline is given of the model
definitions necessary for analysing a priority system with three independent servers and two classes of customers via a system-point approach. In Sections 7.2.1 and 7.2.2 we model the type-1 and type-2 virtual waiting time processes, respectively.

7.2.1 System-Point Process for Type-1 Units

Define the system-point process for the type-1 units by the stochastic process

$\{\langle W^{(1)}(t); M^{(1)}(t) \rangle, t \geq 0 \}$

where

$W^{(1)}(t)$ is the virtual wait of a type-1 unit arriving at time $t$.

$M^{(1)}(t)$ is the system configuration at time $t$ defined by the vector $M^{(1)}(t) = (n_s, n_w, k)$, where

$n_s =$ number of type-2's in service at time $t$, $n_s = 0, 1, 2, 3$.

$n_w =$ number of type-2's waiting at time $t$, $n_w = 0, 1, 2, ...$

$k =$ $(k_1, k_2)$ where,

$k_1 =$ number of other servers with type-1 in them at service starting epoch of the time-$t$ type-1 arrival $k_1 = 0, 1, 2$.

$k_2 =$ number of other servers with type-2 in them at service starting epoch of the time-$t$ type-1 arrival $k_2 = 0, 1, 2$.

Thus $0 \leq k_1 + k_2 \leq 2$. If $k_1 + k_2 = 0$, then a newly arriving
type-1 enters an empty system. Notice that this is similar to the $SP^{(1)}$ process for two servers, except that $k$ is now a vector. Clearly this is generalizable to any finite number of servers.

### 7.2.2 System-Point Process for Type-2 Units

Define the system-point process by

$$\{<W^{(2)}(t), M^{(2)}(t)>, t \geq 0\}$$

where

- $W^{(2)}(t)$ is the virtual wait of a type-2 unit arriving at time $t$.
- $M^{(2)}(t)$ is the system configuration at time $t$ defined by the vector $M^{(2)}(t) = k$. Vector $k = (k_1, k_2)$, represents the number of each unit-type occupying the other two servers respectively, when a time-$t$ type-2 arrival enters service at $t + W^{(2)}(t)$.

Possible configurations $k$ for $SP^{(2)}$ include:

- $(0, 0)$ — arrives to an empty system.
- $(1, 0)$ — starts service with a type-1 in service.
- $(2, 0)$ — starts service with two type-1's in service.
- $(1, 1)$ — starts service with types 1 and 2 in service.
- $(0, 1)$ — starts service with a type-2 in service.
- $(0, 2)$ — starts service with two type-2's in service.
The SP*(1) and SP*(2) sample paths can be constructed in a similar manner as illustrated for the M/M/2 2-class priority system discussed in Chapter 4. By increasing the number of servers from two to three while keeping the number of unit-types constant at two, the size of the system-point model definitions increases. For example, to model the virtual waiting time of the type-2 units would require three "lines" and three "pages". Despite the increase in model size, the fundamental structure for the system-point processes allows this modelling approach to remain a viable option.

7.3 Summary

In this thesis the virtual waiting time of each customer-type in an M/M/2 non-preemptive priority queueing system with 2-classes of customers was computed. To do so a system-point model was developed to describe the virtual waiting time process for the respective unit-type. Examples of typical sample paths for the respective system-point processes illustrate the main ideas of the modelling approach adopted in this study. By adopting a system-point Monte Carlo computation approach, the virtual waiting time probability density function, cumulative distribution function and expected waiting time for each customer-type were derived. Also computed were some important system probabilities. Various checks were performed to ensure correct modelling of
the system; these checks generally proved to be satisfactory. Computational results obtained via the system-point models were validated by comparing them with a large-scale system simulation developed for this purpose. The estimates were found to be quite close in value to the system simulation results.

The modelling approach presented in this thesis is useful since it allows for the computation of the entire probability density function of the waiting time for each customer-type. This is a necessary input to the solution of optimization problems where the cost of waiting is non-linear. The model developed here can be easily extended to include cases where the service time depends on waiting time and/or the servers are heterogeneous. Other areas for future research using the system-point Monte Carlo computation approach includes:

a) a k-class preemptive priority M/G/1 and M/M/c queueing system,

b) priority queueing systems where priorities are state dependent, for example, selection of a unit-type for service depends on the number of units of that type waiting.
REFERENCES


APPENDIX A

SAMPLE CALCULATIONS BASED ON ANALYTICAL MODELS
This Appendix provides sample calculations based on known analytical results that are used to verify the system-point computer programs written.

A.1 M/M/2 Non-Priority Queue

(Refers to Table 3, Table 4 and Table 5)

In this case the system-point model parameters are adjusted so as to represent an M/M/2 non-priority queueing system for either class of customers. For such a queueing system it is well known that the probability of an empty system is

\[ P_0 = \left[ \sum_{n=0}^{c-1} \frac{1}{n!} \left( \frac{\lambda}{\mu} \right)^n + \frac{1}{c!} \left( \frac{\lambda}{\mu} \right)^c \left( \frac{c \mu}{c \mu - \lambda} \right) \right]^{-1} \]  

(a.1.1)

\[ w_q = \frac{(\lambda/\mu)c \mu}{(c-1)! (c \mu - \lambda)^2} \frac{1}{P_0} \]  

(a.1.2)

\[ p_1 = (\lambda/\mu) \frac{1}{P_0} \]  

(a.1.3)

\[ f(w) = \frac{(\lambda/\mu)c \mu e^{-(c \mu - \lambda)w}}{(c-1)!} \frac{1}{P_0} \quad (w > 0) \]  

(a.1.4)

... Gross & Harris (1985)

A.1.1 Analytic Results to check on SP(1) Computer

Program

Parameters used:
\[ \lambda_1 = 15 \text{ units/hour} \]
\[ \lambda_2 = 0.01 \text{ units/hour} \]
\[ \mu_1 = 15 \text{ units/hour} \]
\[ \mu_2 = 15 \text{ units/hour} \]

The theoretical measures of performance are:

\[ P_0^s = \left[ \frac{1}{0!} \left( \frac{15}{15} \right)^0 + \frac{1}{1!} \left( \frac{15}{15} \right)^1 + \frac{1}{2!} \left( \frac{15}{15} \right)^2 \left( \frac{30}{15} \right) \right]^{-1} \]
\[ = \frac{1}{3} \]
\[ = 0.33333 \]

\[ P_1^s = \left( \frac{15}{15} \right) \left( \frac{1}{3} \right) \]
\[ = 0.33333 \]

\[ W_{q1} = \frac{\left( \frac{15}{15} \right)^2 \left( \frac{15}{15} \right)}{\left( \frac{2-1}{1!} \left( \frac{30}{15} \right)^2 \right)} \left( \frac{1}{3} \right) \]
\[ = 0.02222 \text{ hrs.} \]

A.1.2. **Analytic Results to check on SP(2) Computer Program**

Parameters used:

\[ \lambda_1 = 0.01 \text{ units/hour} \]
\[ \lambda_2 = 12 \text{ units/hour} \]
\[ \mu_1 = 15 \text{ units/hour} \]
\[ \mu_2 = 15 \text{ units/hour} \]

The theoretical measures of performance are:

\[ P_0^s = \left[ \frac{1}{0!} \left( \frac{12}{15} \right)^0 + \frac{1}{1!} \left( \frac{12}{15} \right)^1 + \frac{1}{2!} \left( \frac{15}{12} \right)^2 \left( \frac{30}{18} \right) \right]^{-1} \]
\[ = 0.428571 \]
\[ P_1^s = \left( \frac{12}{15} \right) \left( \frac{1}{2.3333} \right) \]
\[ = 0.342857 \]
\[ W_{q2} = \frac{15 \left( \frac{12}{15} \right)^2}{(2-1)! (30-12)^2} \left( 0.42857 \right) \]
\[ = 0.012698 \text{ hrs.} \]

### A.2 M/M/2 Priority Queue

(Refers to Table 6, Table 7)

In this approach the system point model parameters are adjusted so that the service rate for either class is the same. For this non-preemptive priority queueing system with the same service time distribution for either class Cobham (1954) has derived the expected waiting time in queue for each class as follows:

The expected waiting time of a unit of priority \( p \) is given by:

\[ \bar{W}_p = \frac{E[T_0]}{1- \left( \frac{1}{\rho} \right) \sum_{k=1}^{p-1} \lambda_k} \frac{1- \left( \frac{1}{\rho} \right) \sum_{k=1}^{p} \lambda_k}{1- \left( \frac{1}{\rho} \right) \sum_{k=1}^{p-1} \lambda_k} \]  \hspace{1cm} (a.2.1)

where,

\[ E[T_0] = \frac{(c_0)^c/c_0}{c! (1-\rho) \left[ \sum_{j=0}^{c-1} \left( c_0 \right)^j / j! + (c_0)^c / (c! (1-\rho)) \right]} \]

\[ \rho = \lambda/c_0 \]

\( p = \) priority of unit (smaller the integer, higher the priority)
\( \lambda_k \) = poisson arrival rate of \( p \)-th priority unit
\( \mu \) = exponential service rate
\( c \) = number of service channels
\( \lambda = \lambda_1 + \lambda_2 \)

Equation 2.1 for \( \bar{W}_p \) is valid when \( (1/c\mu) \sum_{k=1}^{p} \lambda_k < 1 \).

To compute \( \bar{P}_0^S = \Pr(\text{an empty system}) \) and \( \bar{P}_1^S = \Pr(\text{a single unit of either type in the system}) \) we make use of equations a.1.1 and a.1.3 respectively from Section A.1

Using this approach (Table 6) the system-point models were run with the following parameters

\( \lambda_1 = 15 \text{ customers/hour} \)
\( \lambda_2 = 12 \text{ customers/hour} \)
\( \mu_1 = 20 \text{ customers/hour} \)
\( \mu_2 = 20 \text{ customers/hour} \)

Hence the theoretical results of interest are:

\[ \bar{W}_{q1} = \frac{E[T_0]}{[1-0][1-15/40]} \]

where,

\[ E[T_0] = \frac{[2(27/40)]^{2/40}}{2(1-27/40)[1 + 2(27/40) + 2(27/40)/2(1-27/40)]} \]
\[ = 0.0136007 \]

\[ \bar{W}_{q1} = (0.0136007/0.625) \]
\[ = 0.0217611 \text{ hrs.} \]
\[
\bar{w}_{q2} = \frac{E[T_0]}{[1-15/40] \times [1-27/40]}
\]
\[
= (0.0136007/0.2030312)
\]
\[
= 0.0669575 \text{ hrs.}
\]
\[
\bar{P}_0^S = \left[1 + 2(0.675) + \frac{(2\times0.675)^2}{2(1-0.675)}\right]^{-1}
\]
\[
= 0.1940298
\]
\[
\bar{P}_1^S = \left(\frac{27}{20}\right) \bar{P}_0^S
\]
\[
= 0.2619402.
\]
APPENDIX B

SP\((1)\) FLOWCHART AND
COMPUTER PROGRAM LISTING
Figure B.1: Flowchart for $SP^{(1)}$ Sample Path Generation

Start

Read and Initialize Parameters

Generate arrival and its type

is it a type-1?

yes

Generate a type-1 departure epoch on page (0,0;1)

Generate next arrival and its type

Is arrival a type-1?

yes

Is arrival epoch < departure epoch?

yes

Generate jump of size $\exp(\mu_1+\mu_2)$

no

System becomes empty, line (0,0;0)

no

Is arrival epoch < departure epoch?

yes

Generate jump of size $\exp(2\mu_1)$ on page (0,0;1)

no

Generate a type-2 departure epoch on page (1,0;2)

system becomes empty, line (0,0;0)

no
Figure B.1 (continued)

Generate next arrival and its type

Is arrival a type-1?

Is arrival epoch < departure epoch?

Generate jump of size $\exp(U_1 + U_2)$

System becomes empty line (0,0;0)

Generate jump of size $\exp(2U_2)$

System becomes empty line (0,0;0)
Figure B.1 (continued)

Which customer type completes servicing first?

- Type-1
  - Jump is to page (1,0;2)
  - Generate next arrival and type
    - If arrival epoch < departure epoch:
      - Yes: Type-1
        - Generate an $\exp(U_1 + U_2)$ jump
      - No: Type-2
        - Jump to page (1,n,2) at the same ht.
  - Is there any waiting type-2's?
    - No: Go back
    - Yes: Generate an $\exp(2U_2)$ jump on page (2,n;2)
- Type-2
  - Jump is to page (1,0;1)
  - Generate next arrival and type
- Generate an $\exp(U_1 + U_2)$ jump

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure B.1 (continued)

Is arrival epoch < departure epoch?  

Is there waiting type-2's?  

Type-1?  

Generate an \( \exp(W_i+W_z) \) jump  

Generate an \( \exp(2u_1) \) jump on page (0,n;1)  

Jump to page (1,n:1) at the same ht.  

Generate a type-1 departure epoch  

Is the previous arrival epoch < departure epoch?  

System becomes empty, (0,0;0)  

Type-1?  

Generate an \( \exp(\mu_1+\mu_2) \) jump  

Generate an \( \exp(2\mu_1) \) jump  

\( H \)  

\( F \)  

\( I \)  

\( G \)  

\( E \)  

\( C \)
SPONE JOB (R240,SL2,5), 'MADO BACHAN', CLASS=1
EXEC WATFIV
//60.SYSIN DD *
$JOB WATFIV

*****************************************
C$ THIS PROGRAM IS DESIGNED TO GENERATE THE SYSTEM-POINT PROCESS $C
C$ SAMPLE PATH FOR THE TYPE-1 CUSTOMERS (SP(11)) USING THE $C
C$ SYSTEM-POINT MONTE CARLO COMPUTATION APPROACH. $C
C$ TYPE-1 CUSTOMERS ARE THE HIGH PRIORITY CUSTOMERS OF AN M/M/2 $C
C$ TWO-CLASS NON-PREEMPTIVE PRIORITY QUEUEING SYSTEM. TYPE-1 $C
C$ UNITS HAVE SERVICE PRIORITY OVER THE TYPE-2 UNITS AND SERVICE $C
C$ RATES ARE DIFFERENT FOR EACH CLASS. $C
*****************************************
C
THE FOLLOWING ARE THE MAJOR NOTATIONS TO BE USED:
C
C  LAMDA1 = ARRIVAL RATE OF TYPE-I (1=1,2) CUSTOMERS.
C  AMUI = SERVICE RATE OF TYPE-I (1=1,2) CUSTOMERS.
C  PROBT1 = PROBABILITY THAT A TYPE-1 ARRIVES BEFORE A TYPE-2.
C  PRDEP = PROBABILITY THAT A TYPE-1 DEPARTS BEFORE A TYPE-2.
C  CLOCK = LENGTH OF SIMULATION RUN (MINS.)
C  AT = INTERARRIVAL TIME OF A CUSTOMER.
C  SEED = INITIAL VALUE FOR RANDOM # GENERATOR.
C  YFL = UNIFORMLY DISTRIBUTED RANDOM NUMBER BETWEEN 0-1.
C  P = ARRIVAL EPOCHS OF CUSTOMERS.
C  Z = DEPARTURE POINTS OF CUSTOMERS.
C  TAIIS = VARIABLE TO KEEP TRACK OF SIMULATED TIME TO DATE.
C  TYP = INDICATE CUSTOMER TYPE.
C  NPAGE = # OF TYPE-2 CUSTOMERS WAITING IN QUEUE.
C  NPKK(W) = # OF PEAKS ABOVE LEVEL W.
C  NTRR(W) = # OF TROUGHS ABOVE LEVEL W.
C  PKK(W) = ASSIGNED HEIGHT OF LEVEL W TO CHECK & RECORD PEAK HT.
C  TRR(W) = ASSIGNED HEIGHT OF LEVEL W TO CHECK & RECORD TROUGH HT.
C  DIFF(W) = # OF PEAKS - # OF TROUGHS ABOVE LEVEL W.
C  DWX(W) = DOWN-CROSSING RATE OF LEVEL W = DIFF(W)/CLOCK.
C  IMPK = # OF LEVEL ZERO HITS.
C  NT = # OF SYSTEM-POINT TRANSITIONS
C
C $DECLARATION AND INITIALIZATION OF VARIABLES.
C
C INTEGER FLAG,SEED,NPKK(1000),NTRR(1000),DIFF(1000),TYP
REAL PKK(1000),TRR(1000),SI(1000),LAMDA1,LAMDA2
REAL DWX(1000),CLOCK,DX(1000),TT(1000)
READ,LAMDA1,LAMDA2,AMUI,AMU2,CLOCK,SEED,W,NN
IX=SEED
TAII$=IMPK1=IMPK2=IMPK3=IMPK4=#NPAGE=HTRR=NT1=NT2=0.
PROBT1=LAMDA1/(LAMDA1+LAMDA2)
PRDEP=AMUI/(AMUI+AMU2)
DO 5 J=1,1000

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
PKK(J) = (J-1)*W
NPKK(J) = 0
TRR(J) = (J-1)*W
NTRR(J) = 0
DIFF(J) = 0
DWX(J) = 0

5 CONTINUE

C *** GENERATE AN ARRIVAL, CHECK FOR CUSTOMER TYPE AND GENERATE
C *** ITS SERVICE TIME.
C
EXECUTE RANDOM
AT = TAXIS + (-1./((LAMDA1+LAMDA2))*ALOG(YFL))
TAXIS = AT
EXECUTE RANDOM
IF (YFL.LE.PROBT1) THEN
  NT1 = NT1 + 1
  EXECUTE RANDOM
  Z = TAXIS + (-1./AMU1)*ALOG(YFL)
ELSE
  NT2 = NT2 + 1
  EXECUTE RANDOM
  Z = TAXIS + (-1./AMU2)*ALOG(YFL)
  GO TO 100
ENDIF
C
C *** GENERATE AN ARRIVAL WITH A SINGLE TYPE-1 CURRENTLY IN SERVICE
C *** AND CHECK WHETHER OR NOT A JUMP OCCURS.
C
10 EXECUTE ARRIVAL
IF (P.GT.CLOCK) GO TO 99
IF (TYC.EQ.1.AND.Z.GE.P) THEN
  HTRR = 0.
  EXECUTE JUMP11
  FLAG = 1
ELSEIF (TYC.EQ.1.AND.Z.LT.P) THEN
  NT1 = NT1 + 1
  EXECUTE OLEV1
  GO TO 10
ELSEIF (TYC.EQ.2.AND.Z.GE.P) THEN
  HTRR = 0.
  EXECUTE JUMP12
  FLAG = 2
ELSEIF (TYC.EQ.2.AND.Z.LT.P) THEN
  NT2 = NT2 + 1
  EXECUTE OLEV2
  GO TO 100
ENDIF
GO TO 200
C
C *** GENERATE AN ARRIVAL WITH A TYPE-2 CURRENTLY IN SERVICE AND CHECK
C ### WHETHER OR NOT A JUMP OCCURS.
C 100 EXECUTE ARRIVAL
   IF (P.GT.CLOCK) GO TO 99
   IF (TYP.EQ.1.AND.Z.GE.P) THEN
     HTRR=0.
     EXECUTE JUMP12
     FLAG=2
   ELSEIF (TYP.EQ.1.AND.Z.LT.P) THEN
     NT1=NT1+1
     EXECUTE OLEV1
     GO TO 10
   ELSEIF (TYP.EQ.2.AND.Z.GE.P) THEN
     HTRR=0.
     EXECUTE JUMP22
     FLAG=3
   ELSEIF (TYP.EQ.2.AND.Z.LT.P) THEN
     NT2=NT2+1
     EXECUTE OLEV2
     GO TO 100
   ENDIF
C C ### GENERATE AN ARRIVAL WHEN BOTH SERVERS ARE BUSY.
C 200 EXECUTE ARRIVAL
C C ### CHECK FOR END OF SIMULATION CONDITION.
C   IF (P.GT.CLOCK) GO TO 99
C C ### IF A TYPE-1 ARRIVES DURING SERVICING OF PREVIOUS TYPE-1'S
C   IF (TYP.EQ.1.AND.Z.GE.P.AND.FLAG.EQ.1) THEN
     HTRR=Z-P
     EXECUTE THROUGH
     EXECUTE JUMP11
     FLAG=1
     GO TO 200
C C ### IF A TYPE-1 ARRIVES AFTER SERVICING OF PREVIOUS TYPE-1'S.
C   ELSEIF (TYP.EQ.1.AND.Z.LT.P.AND.FLAG.EQ.1) THEN
     EXECUTE CHECK1
     IF (IFLAG.EQ.1) GO TO 200
     IF (IFLAG.EQ.2) GO TO 10
     IF (IFLAG.EQ.3) GO TO 210
   210 EXECUTE FCHECK
     IF (IFLAG.EQ.4 .OR. IFLAG.EQ.1) GO TO 200
     IF (IFLAG.EQ.2) GO TO 10
     IF (IFLAG.EQ.3) GO TO 210
     IF (IFLAG.EQ.5) GO TO 220
EXECUTE DOUBLE
   IF(IFLAG.EQ.6 .OR. IFLAG.EQ.4) GO TO 200
   IF(IFLAG.EQ.5) GO TO 220
   IF(IFLAG.EQ.2) GO TO 10

C
C *** IF A TYPE-2 ARRIVES BEFORE SERVICE COMPLETION OF PREVIOUS TYPE-1.
C
ELSEIF(TYP.EQ.2 .AND. .GE. P .AND. FLAG.EQ.1) THEN
   NPAGE=NPAGE+1
   TAXIS=P
   FLAG=1
   GO TO 200

C
C *** IF A TYPE-2 ARRIVES AFTER SERVICING OF PREVIOUS TYPE-1'S.
C
ELSEIF(TYP.EQ.2 .AND. .LT. P .AND. FLAG.EQ.1) THEN
   EXECUTE CHECK3
   IF(IFLAG.EQ.7) GO TO 2300
   IF(IFLAG.EQ.4) GO TO 200
   IF(IFLAG.EQ.8) GO TO 100

C
C *** IF A TYPE-1 ARRIVES DURING THE SERVICING OF A TYPE-1 AND A TYPE-2
C
ELSEIF(TYP.EQ.1 .AND. .GE. P .AND. FLAG.EQ.2) THEN
   HTRR=Z-P
   EXECUTE TROUGH
   EXECUTE RANDOM
   IF(YFL.LE.PRDEP) THEN
      EXECUTE JUMP12
      FLAG=2
      GO TO 200
   ELSE
      EXECUTE JUMP11
      FLAG=1
      GO TO 200
   ENDIF

C
C *** IF A TYPE-1 ARRIVES AFTER THE SERVICE COMPLETION OF EITHER A
C *** TYPE-1 OR A TYPE-2.
C
ELSEIF(TYP.EQ.1 .AND. .LT. P .AND. FLAG.EQ.2) THEN
   EXECUTE RANDOM
   IF(YFL.LE.PRDEP) THEN
      FLAG=13
      EXECUTE CHECK2
      IF(IFLAG.EQ.4) GO TO 200
      IF(IFLAG.EQ.2) GO TO 10
      IF(IFLAG.EQ.5) GO TO 260
   EXECUTE DOUBLE
   IF(IFLAG.EQ.6 .OR. IFLAG.EQ.4) GO TO 200
   IF(IFLAG.EQ.2) GO TO 10
IF(FLAG.EQ.5) GO TO 260
ELSE
  FLAG=12
  EXECUTE CHECK1
  IF(FLAG.EQ.1) GO TO 200
  IF(FLAG.EQ.2) GO TO 10
  IF(FLAG.EQ.3) GO TO 270
  EXECUTE FCHECK
  IF(FLAG.EQ.1.OR.FLAG.EQ.4) GO TO 200
  IF(FLAG.EQ.2) GO TO 10
  IF(FLAG.EQ.3) GO TO 270
  IF(FLAG.EQ.5) GO TO 280
  EXECUTE DOUBLE
  IF(FLAG.EQ.4.OR.FLAG.EQ.6) GO TO 200
  IF(FLAG.EQ.5) GO TO 280
  IF(FLAG.EQ.2) GO TO 10
ENDIF
C
C *** IF A TYPE-2 ARRIVES DURING SERVICING OF A TYPE-1 AND A TYPE-2.
C
ELSEIF(TYP.EQ.2.AND.Z.GE.P.AND.FLAG.EQ.2) THEN
  NPAGE=NPAGE+1
  TAXIS=P
  FLAG=2
  GO TO 200
C
C *** IF NEXT TYPE-2 ARRIVAL OCCURS AFTER SERVICE COMPLETION OF EITHER
C *** A TYPE-1 OR A TYPE-2 JOINTLY IN SERVICE.
C
ELSEIF(TYP.EQ.2.AND.Z.LT.P.AND.FLAG.EQ.2) THEN
  EXECUTE RANDOM
  IF(YFL.LT.PREP) THEN
    FLAG=13
    EXECUTE CHECK4
    IF(FLAG.EQ.9) GO TO 290
    IF(FLAG.EQ.4) GO TO 200
    IF(FLAG.EQ.10) GO TO 100
  290  EXECUTE CHECK9
    IF(FLAG.EQ.6) GO TO 99
    IF(FLAG.EQ.2) GO TO 10
    IF(FLAG.EQ.4) GO TO 200
    IF(FLAG.EQ.5) GO TO 300
    IF(FLAG.EQ.9) GO TO 290
    IF(FLAG.EQ.10) GO TO 100
  300  EXECUTE DOUBLE
    IF(FLAG.EQ.4.OR.FLAG.EQ.6) GO TO 200
    IF(FLAG.EQ.5) GO TO 300
    IF(FLAG.EQ.2) GO TO 10
ELSE
  FLAG=12
  EXECUTE CHECK3
IF(IFLAG.EQ.7) GO TO 2300
IF(IFLAG.EQ.4) GO TO 200
IF(IFLAG.EQ.8) GO TO 100
ENDIF

C *** NEXT TYPE-1 ARRIVAL OCCURS DURING TWO TYPE-2'S SERVICING.

ELSEIF(TYP.EQ.1.AND.Z.GE.P.AND.FLAG.EQ.3) THEN
  MTRA=Z-P
  EXECUTE TRROUGH
  EXECUTE JUMP12
  FLAG=2
  GO TO 200

C *** NEXT TYPE-1 ARRIVAL OCCURS AFTER A TYPE-2 COMPLETES SERVICE AND
C *** WHOSE NEIGHBOR WAS A TYPE-2.

ELSEIF(TYP.EQ.1.AND.Z.LT.P.AND.FLAG.EQ.3) THEN
  FLAG=14
  EXECUTE CHECK2
    IF(IFLAG.EQ.5) GO TO 400
    IF(IFLAG.EQ.4) GO TO 200
    IF(IFLAG.EQ.2) GO TO 10
  400  EXECUTE DOUBLE
    IF(IFLAG.EQ.4.OR.IFLAG.EQ.6) GO TO 200
    IF(IFLAG.EQ.5) GO TO 400
    IF(IFLAG.EQ.2) GO TO 10

C *** IF A TYPE-2 ARRIVAL OCCURS DURING TWO TYPE-2'S SERVICING.

ELSEIF(TYP.EQ.2.AND.Z.GE.P.AND.FLAG.EQ.3) THEN
  NPAGE=NPAGE+1
  TAXIS=P
  FLAG=3
  GO TO 200

C *** IF A TYPE-2 ARRIVAL OCCURS AFTER A TYPE-2 COMPLETES SERVICE AND
C *** WHOSE NEIGHBOR WAS A TYPE-2.

ELSEIF(TYP.EQ.2.AND.Z.LT.P.AND.FLAG.EQ.3) THEN
  FLAG=14
  EXECUTE CHECK4
    IF(IFLAG.EQ.9) GO TO 410
    IF(IFLAG.EQ.4) GO TO 200
    IF(IFLAG.EQ.10) GO TO 100
  410  EXECUTE CHECK9
    IF(IFLAG.EQ.6) GO TO 99
    IF(IFLAG.EQ.4) GO TO 200
    IF(IFLAG.EQ.2) GO TO 10
    IF(IFLAG.EQ.3) GO TO 420
    IF(IFLAG.EQ.9) GO TO 410
IF(IFLAG.EQ.10) GO TO 100

EXECUTE DOUBLE
IF(IFLAG.EQ.4 OR IFLAG.EQ.6) GO TO 200
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.5) GO TO 420

ENDIF

C

C ### THIS SECTION OF THE PROGRAM SIMULATE THE SAMPLE PATH WHEN A
C ### TYPE-2 STARTS SERVICE AT THE END OF A TYPE-1 BUSY PERIOD
C ### WITH ITS NEIGHBOR BEING A TYPE-1.
C

2300 IF(I.GE,P) THEN
NPAGE=NPAGE+1
TAXIS=P
EXECUTE ARRIVAL
IF(P.GT.CLOCK) GO TO 99
IF(I.GE,P AND TYP.EQ.1) THEN
EXECUTE CHECKS
IF(IFLAG.EQ.4) GO TO 200
ELSEIF(I.LT.P AND TYP.EQ.1) THEN
EXECUTE CHECK6
IF(IFLAG.EQ.1 OR IFLAG.EQ.4) GO TO 200
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.3) GO TO 240
IF(IFLAG.EQ.5) GO TO 235

235 EXECUTE DOUBLE
IF(IFLAG.EQ.4 OR IFLAG.EQ.6) GO TO 200
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.5) GO TO 235

240 EXECUTE FCHECK
IF(IFLAG.EQ.4 OR IFLAG.EQ.1) GO TO 200
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.3) GO TO 240
IF(IFLAG.EQ.5) GO TO 235
ELSEIF(I.GE,P AND TYP.EQ.2) THEN
NPAGE=NPAGE+1
TAXIS=P
EXECUTE CHECK7
IF(IFLAG.EQ.5) GO TO 99
IF(IFLAG.EQ.1 OR IFLAG.EQ.4) GO TO 200
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.3) GO TO 2600
IF(IFLAG.EQ.5) GO TO 2550
IF(IFLAG.EQ.7) GO TO 2300
IF(IFLAG.EQ.8 OR IFLAG.EQ.10) GO TO 100
IF(IFLAG.EQ.9) GO TO 2650

2550 EXECUTE DOUBLE
IF(IFLAG.EQ.4 OR IFLAG.EQ.6) GO TO 200
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.5) GO TO 2550

2600 EXECUTE FCHECK

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
IF(IFLAG.EQ.4, OR, IFLAG.EQ.1) GO TO 200
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.3) GO TO 2600
IF(IFLAG.EQ.5) GO TO 2550

EXECUTE CHECK9
IF(IFLAG.EQ.6) GO TO 99
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.4) GO TO 200
IF(IFLAG.EQ.5) GO TO 2550
IF(IFLAG.EQ.9) GO TO 2650
IF(IFLAG.EQ.10) GO TO 100

ELSEIF(Z.LT.P.AND.TYP.EQ.2) THEN
EXECUTE CHECK9
IF(IFLAG.EQ.4) GO TO 200
IF(IFLAG.EQ.7) GO TO 2300
IF(IFLAG.EQ.8.OR.IFLAG.EQ.10) GO TO 100
IF(IFLAG.EQ.9) GO TO 250

EXECUTE CHECK9
IF(IFLAG.EQ.6) GO TO 99
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.4) GO TO 200
IF(IFLAG.EQ.5) GO TO 255
IF(IFLAG.EQ.9) GO TO 250
IF(IFLAG.EQ.10) GO TO 100

EXECUTE DOUBLE
IF(IFLAG.EQ.4.OR.IFLAG.EQ.6) GO TO 200
IF(IFLAG.EQ.5) GO TO 255
IF(IFLAG.EQ.2) GO TO 10

ENDIF
ELSE
EXECUTE CHECK9
IF(IFLAG.EQ.4) GO TO 200
IF(IFLAG.EQ.7) GO TO 2300
IF(IFLAG.EQ.8.OR.IFLAG.EQ.10) GO TO 100
IF(IFLAG.EQ.9) GO TO 275

EXECUTE CHECK9
IF(IFLAG.EQ.6) GO TO 99
IF(IFLAG.EQ.4) GO TO 200
IF(IFLAG.EQ.5) GO TO 285
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.9) GO TO 275
IF(IFLAG.EQ.10) GO TO 100

EXECUTE DOUBLE
IF(IFLAG.EQ.4.OR.IFLAG.EQ.6) GO TO 200
IF(IFLAG.EQ.2) GO TO 10
IF(IFLAG.EQ.5) GO TO 255

ENDIF
C
99 EXECUTE RESULT
GO TO 999
C ########################################################################
C REMOTE BLOCK LISTING
C*************************************************************
C*************************************************************
C REMOTE BLOCK RANDOM
C*************************************************************
C *** GENERATE 0-1 UNIFORM RANDOM NUMBERS.
C
C CALL RANDU(IY,YFL)
II=IY
ENDBLOCK
C*************************************************************
C REMOTE BLOCK PEAK
C*************************************************************
C *** INCREMENT BY 1 EACH W WHICH IS LESS THAN HT. OF THE PEAK.
C
J=1
WHILE(PKK(J).LT.HPEAK)
  NPKK(J)=NPKK(J)+1
  J=J+1
END WHILE
ENDBLOCK
C*************************************************************
C REMOTE BLOCK TROUGH
C*************************************************************
C *** INCREMENT BY 1 EACH W WHICH IS LESS THAN HT. OF THE TROUGH.
C
K=1
WHILE(TRR(K).LT.HTRR)
  NTRR(K)=NTRR(K)+1
  K=K+1
END WHILE
ENDBLOCK
C*************************************************************
C REMOTE BLOCK ARRIVAL
C*************************************************************
C *** GENERATE ARRIVAL EPOCHS OF CUSTOMERS AND CHECK THEIR TYPE.
C
EXECUTE RANDOM
AT=-1./(LAMDA1+LAMDA2)*ALOG(YFL)
P = TAXIS+AT
EXECUTE RANDOM
IF(YFL.LE.PROBT1) THEN
  TYP=1
ELSE
  TYP=2
ENDIF
ENDBLOCK
C*************************************************************
C REMOTE BLOCK JUMP11
C*************************************************************
C *** DETERMINE THE HEIGHT OF PEAK WHEN THE JUMP SIZE IS DUE TO TYPE-1.
C
TAXIS=P
EXECUTE RANDOM
ST11=(-1./[2*AMU1])*#ALOG(YFL)
HPEAK=ST11+HTRR
EXECUTE PEAK
Z=TAXIS+HPEAK
ENDBLOCK

C ---------------------
REMOTE BLOCK OLEV1
C
C *** DETERMINE THE SERVICE TIME FOR A SINGLE TYPE-1 UNIT.
C
TAXIS=P
EXECUTE RANDOM
Z=TAXIS+(-1./AMU1)*#ALOG(YFL)
ENDBLOCK

C ---------------------
REMOTE BLOCK JUMP12
C
C *** DETERMINE THE HEIGHT OF PEAK WHEN THE JUMP SIZE IS DUE TO A
C *** TYPE-1 AND A TYPE-2.
C
TAXIS=P
EXECUTE RANDOM
ST12=(-1./(AMU1+AMU2))*#ALOG(YFL)
HPEAK=ST12+HTRR
EXECUTE PEAK
Z=TAXIS+HPEAK
ENDBLOCK

C ---------------------
REMOTE BLOCK OLEV2
C
C *** FIND THE SERVICE TIME FOR A TYPE-2 UNIT.
C
TAXIS=P
EXECUTE RANDOM
Z=TAXIS+(-1./AMU2)*#ALOG(YFL)
ENDBLOCK

C ---------------------
REMOTE BLOCK JUMP22
C
C *** DETERMINE THE HEIGHT OF PEAK WHEN THE JUMP SIZE IS DUE TO TWO
C *** TYPE-2'S.
C
TAXIS=P
EXECUTE RANDOM
ST22=(-1./(2*AMU2))*#ALOG(YFL)
HPEAK=ST22
EXECUTE PEAK

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
I=TAXIS+ST22
ENDBLOCK

C ---------------------------------------------
REMOTE BLOCK CHECK1
C ---------------------------------------------
C *** ANY WAITING TYPE-2 GOES INTO SERVICE ON A FCFS BASIS.
C
IF(NPAGE.GE.1) THEN
  TAXIS=Z
  NPAGE=NPAGE-1
  EXECUTE RANDOM
  ST12=((-1./((A11+A12)))*ALOG(YFL))
  HPEAK=ST12
  EXECUTE PEAK
  I=TAXIS+ST12
  FLAG=2
  IFLAG=3
ELSE
  IF(FLAG.EQ.1) IMPK1=IMPK1+1
  IF(FLAG.EQ.12) IMPK2=IMPK2+1
  EXECUTE RANDOM
  ST1=(1./A11)*ALOG(YFL)
  I=I+ST1
  IF(Z.GE.P) THEN
    HTRR=Z
    EXECUTE JUMP11
    FLAG=1
    IFLAG=1
  ELSE
    NT1=NT1+1
    EXECUTE OLEV1
    IFLAG=2
  ENDIF
ENDIF
ENDBLOCK

C ---------------------------------------------
REMOTE BLOCK FCHECK
C ---------------------------------------------

IF(Z.GE.P) THEN
  TAXIS=P
  HTRR=Z-P
  EXECUTE TROUGH
  EXECUTE RANDOM
  IF(YFL.LE.PREDP) THEN
    EXECUTE JUMP12
    FLAG=2
    IFLAG=4
  ELSE
    EXECUTE JUMP11
    FLAG=1

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
IFLAG=4
ENDIF
ELSEIF(I.LT.P) THEN
EXECUTE RANDOM
IF(YFL.GT.PROEP) THEN
FLAG=12
EXECUTE CHECK1
ELSE
FLAG=13
EXECUTE CHECK2
ENDIF
ENDIF
ENDBLOCK
C -------------------------
REMOTE BLOCK DOUBLE
C -------------------------
IF(I.GE.P) THEN
TAXIS=P
HTRR=Z-P
EXECUTE THROUG
EXECUTE JUMP12
FLAG=2
IFLAG=6
ELSE
FLAG=14
EXECUTE CHECK2
ENDIF
ENDBLOCK
C -------------------------
REMOTE BLOCK CHECK2
C -------------------------
C *** INITIATE SERVICING OF A TYPE-2 WITH ITS NEIGHOR BEING A TYPE-2
C *** AND NEXT ARRIVAL IS A TYPE-1
IF(NPAGE.GE.1) THEN
TAXIS=2
NPAGE=NPAGE-1
EXECUTE RANDOM
ST22=(-1./(2#AMU2))#ALOG(YFL)
HPEAK=ST22
EXECUTE PEAK
Z=TAXIS+HPEAK
FLAG=3
IFLAG=5
ELSE
IF(FLAG.EQ.13) IMPK3=IMPK3+1
IF(FLAG.EQ.14) IMPK4=IMPK4+1
HTRR=0.
EXECUTE RANDOM
ST2=(-1./AMU2)#ALOG(YFL)
Z=Z+ST2
IF(I.GE.P) THEN
EXECUTE JUMP12
   FLAG=2
   IFLAG=4
ELSE
   NT1=NT1+1
   EXECUTE OLEV1
   IFLAG=2
ENDIF
ENDIF
ENDBLOCK

C -----------------------------
REMOTE BLOCK CHECK3
C -----------------------------
C *** INITIATE SERVICING OF A TYPE-2 IF ONE IS WAITING WITH ITS
C *** NEIGHBOR BEING A TYPE-1, NEXT ARRIVAL IS A TYPE-2.
C
IF(NPAGE.GE.1) THEN
   TAXIS=Z
   NPAGE=NPAGE-1
   EXECUTE RANDOM
   ST12=(-1./(AMU1+AMU2))*ALOG(YFL)
   HPEAK=ST12
   EXECUTE PEAK
   Z=TAXIS+HPEAK
   FLAG=2
   IFLAG=7
ELSE
   IF(FLAG.EQ.1) IMPK1=IMPK1+1
   IF(FLAG.EQ.12) IMPK2=IMPK2+1
   EXECUTE RANDOM
   ST1=(-1./AMU1)*ALOG(YFL)
   Z=Z+ST1
   IF(Z.GE.P) THEN
      HTRR=0.
      EXECUTE JUMP12
      FLAG=2
      IFLAG=4
   ELSE
      NT2=NT2+1
      EXECUTE OLEV2
      IFLAG=8
   ENDIF
ENDIF
ENDBLOCK

C -----------------------------
REMOTE BLOCK CHECK4
C -----------------------------
C *** INITIATE SERVICING OF A TYPE-2 WITH ITS NEIGHBOR BEING A TYPE-2,
C *** NEXT ARRIVAL IS A TYPE-2.
C
IF(NPAGE.GE.1) THEN
TAIIS=Z
NPAGE=NPAGE-1
EXECUTE RANDOM
ST22=(-1./(2*AMU2))*ALOG(YFL)
HPEAK=ST22
EXECUTE PEAK
Z=TAIIS+HPEAK
IFLAG=9
ELSE
 IF(FLAG.EQ.13) IMPK3=IMPK3+1
 IF(FLAG.EQ.14) IMPK4=IMPK4+1
 HTRR=0.
 EXECUTE RANDOM
 ST2=(-1./AMU2)*ALOG(YFL)
 Z=Z+ST2
 IF(Z.GE.P) THEN
   EXECUTE JUMP22
   FLAG=3
   IFLAG=4
 ELSE
   NT2=NT2+1
   EXECUTE OLEV2
   IFLAG=10
 ENDIF
ENDIF
ENDBLOCK
C -------------------------------
REMOTE BLOCK CHECK5
C -------------------------------
C *** DETERMINE HEIGHT OF PEAK WHEN A TYPE-1 ARRIVAL FINDS A TYPE-1
C *** AND A TYPE-2 JOINTLY IN SERVICE.
C
HTRR=Z-P
EXECUTE TROUGH
EXECUTE RANDOM
IF(YFL.LE.PDWP) THEN
  EXECUTE JUMP12
  FLAG=2
  IFLAG=4
ELSE
  EXECUTE JUMP11
  FLAG=1
  IFLAG=4
ENDIF
ENDBLOCK

C -------------------------------
REMOTE BLOCK CHECK6
C -------------------------------
C *** DETERMINE WHETHER A TYPE-1 OR A TYPE-2 DEPARTS FIRST WHEN BOTH
C *** ARE JOINTLY IN SERVICE AND NEXT ARRIVAL IS A TYPE-1
C
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
EXECUTE RANDOM
IF(YFL.LE.PRDEP) THEN
   FLAG=13
   EXECUTE CHECK2
ELSE
   FLAG=12
   EXECUTE CHECK1
ENDIF
ENDBLOCK
C-------------------------------------------------
REMOTE BLOCK CHECK7
C-------------------------------------------------

1000 EXECUTE ARRIVAL
   IF(P.GT.CLOCK) THEN
      FLAG=9
      ELSEIF(Z.GE.P.AND.TYP.EQ.1) THEN
         EXECUTE CHECK5
      ELSEIF(Z.LT.P.AND.TYP.EQ.1) THEN
         EXECUTE CHECK6
      ELSEIF(Z.GE.P.AND.TYP.EQ.2) THEN
         NPAGE=NPAGE+1
         TAXIS=P
         GO TO 1000
      ELSEIF(Z.LT.P.AND.TYP.EQ.2) THEN
         EXECUTE CHECK9
   ELSEIF(Z.LT.P.AND.TYP.EQ.1) THEN
      EXECUTE CHECK3
   ENDIF
ENDIF
ENDBLOCK
C-------------------------------------------------
REMOTE BLOCK CHECK9
C-------------------------------------------------

C *** DETERMINE WHETHER A TYPE-1 OR A TYPE-2 DEPARTS FIRST WHEN BOTH
C *** ARE JOINTLY IN SERVICE AND NEXT ARRIVAL IS A TYPE-2
C
EXECUTE RANDOM
IF(YFL.LE.PRDEP) THEN
   FLAG=13
   EXECUTE CHECK4
ELSE
   FLAG=12
   EXECUTE CHECK3
ENDIF
ENDBLOCK
C-------------------------------------------------
REMOTE BLOCK CHECK9
C-------------------------------------------------

IF(Z.GE.P) THEN
   NPAGE=NPAGE+1
   TAXIS=P
   ELSEIF(Z.LT.P) THEN
      EXECUTE CHECK7
   ELSEIF(Z.LT.P.AND.TYP.EQ.1) THEN
      EXECUTE CHECK3
   ENDIF
ENDIF
ENDBLOCK
C-------------------------------------------------
REMOTE BLOCK CHECK7
C-------------------------------------------------

2000 EXECUTE ARRIVAL
   IF(P.GT.CLOCK) THEN
      FLAG=6
ELSEIF (Z .GE. P. AND. TYP. .EQ. 1) THEN
   HTRR = Z-P
   EXECUTE THROUGH
   EXECUTE JUMP12
   FLAG = 2
   IFLAG = 4
ELSEIF (Z .LT. P. AND. TYP. .EQ. 1) THEN
   EXECUTE CHECK2
ELSEIF (Z .GE. P. AND. TYP. .EQ. 2) THEN
   NPAGE = NPAGE + 1
   TAXIS = P
   GO TO 2000
ELSEIF (Z .LT. P. AND. TYP. .EQ. 2) THEN
   EXECUTE CHECK4
ENDIF
ELSE
   FLAG = 14
   EXECUTE CHECK4
ENDIF
END BLOCK

C -----------------------------------------------
REMOTE BLOCK RESULT
C -----------------------------------------------
C ** CALCULATE AND PRINT RESULTS.
C
S1(1) = 0.
DO 1 I = 2, NN
   S1(I) = S1(I-1) + W
1 CONTINUE
DO 2 J = 1, NN
   DIFF(J) = NPKK(J) - NTRR(J)
   DMX(J) = DIFF(J) / CLOCK
2 CONTINUE
PRINT 11
11 FORMAT (12I,'RESULTS FROM THE MONTE-CARLO SP SIMULATION')
   ' ', -------------------------------
   PRINT,'LAMDA1= ', LAMDA1, ', LAMDA2= ', LAMDA2
   PRINT,'AMUI= ', AMUI1, ', AMU2= ', AMU2
   PRINT,'SEED= ', SEED, ', CLOCK= ', CLOCK
   PRINT,'# OF TYPE-001 IMPACTS = ', IMPK1
   PRINT,'# OF TYPE-101 IMPACTS = ', IMPK2
   PRINT,'# OF TYPE-102 IMPACTS = ', IMPK3
   PRINT,'# OF TYPE-202 IMPACTS = ', IMPK4
   PRINT,'# OF (000— >001) TRANSITIONS= ', NT1
   PRINT,'# OF (000— >102) TRANSITIONS= ', NT2
   PRINT 22
22 FORMAT (//,7X,' WAIT ', 6X,' PEAK COUNT', 5X,' THROUGH COUNT', 8X,' DWI' )
   ') DO 33 I = 1, NN
   PRINT 44, S1(I), NPKK(I), NTRR(I), DMX(I)
44 FORMAT (6I, F5.1, 7X, I4, 12X, I4, 12X, F6.4)
33 CONTINUE
  DO 45 I=1,NN
    PRINT 46,I-1,DMX(I)
  46    FORMAT(5X,I4,5X,F6.4)
  45 CONTINUE
END BLOCK
999 STOP
END
ENTRY
0.25 0.20 0.333 0.25 1000 123456789 .5 40
*IBSYS
*STOP
//
APPENDIX C

$SP(2)$ FLOWCHART AND
COMPUTER PROGRAM LISTING

- 129 -

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure C.1: Flowchart for SP(2) Sample Path Generation

Start

Read and Initialize Parameters

Generate arrival and its type

C

yes

is it a type-1 ?

Generate a type-1 departure epoch on page 1

Generate next arrival and its type

A

no

Generate a type-2 departure epoch on page 2

system becomes empty, line 0

is arrival epoch < departure epoch ?

B

Is arrival epoch < departure epoch?

yes

Generate jump of size \( \exp(\mu_1 + \mu_2) \)

no

Generate jump of size \( \exp(2\mu_1) \) on page 1

F

System becomes empty, line 0

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure C.1 (continued)

A

Generate next arrival and its type

Is arrival a type-1?

yes

Is arrival epoch < departure epoch

yes

Generate jump of size \( \exp(\mu_1 + \mu_2) \)

no

System becomes empty line 0

no

Is arrival epoch < departure epoch

yes

Generate jump of size \( \exp(2\mu_2) \)

no

System becomes empty line 0

E

C

D

B


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure C.1 (continued)

Which customer type completes servicing first?

- type-2
  - Jump is to page 2
  - Generate next arrival and type
  - Is arrival epoch < departure epoch?
    - no
  - type-1
    - Jump is to page 1
    - Generate next arrival and type
    - Is there any waiting type-2's?
      - no
      - yes
        - Generate an $\exp(\mu_1 + \mu_2)$ jump
  - Generate an $\exp(2\mu_2)$ jump on page 2
  - Record arrival epoch
Figure C.1 (continued)

Is arrival epoch < departure epoch?
Yes

Type-1?
Yes

Generate an \( \exp(2\mu_1) \) jump on page 1

No

Is there waiting type-2's?
Yes

Generate an \( \exp(\mu_1+\mu_2) \) jump

No

Generate an \( \exp(2\mu_1) \) jump

Is the previous arrival epoch < departure epoch?

Yes

Type-1?
Yes

Generate an \( \exp(\mu_1+\mu_2) \) jump

No

Generate an \( \exp(2\mu_1) \) jump

No

Type-1?
Yes

Generate an \( \exp(\mu_1+\mu_2) \) jump

System becomes empty, line 0

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
THIS PROGRAM IS DESIGNED TO GENERATE THE SYSTEM-POINT PROCESS $IJOB WATFIV

C

C *** DECLARATION AND INITIALIZATION OF VARIABLES.
C

INTEGER SEED,NPKK(9000),NTRR(9000),DIFF(9000),TYP,TA(1000)
REAL PKK(9000),TRR(9000),S1(9000),LAMDA1,LAMDA2
REAL DWX(9000),CLOCK
READ,LAMDA1,LAMDA2,AMUI,AMU2,CLOCK,SEED,W,NN
II*SEED
TAXIS=IMPK1=IMPK2=HTRR=NA=HPKB1=AR=NT1=NT2=0.
PROBT1=LAMDA1/(LAMDA1+LAMDA2)
PRODP=AMUI/(AMUI+AMU2)

DO I J=1,9000
   PKK(J)=(J-1)*W
   TRR(J)=(J-1)*W
   NPKK(J)=0
   NTRR(J)=0
   DIFF(J)=0
   DWX(J)=0
CONTINUE

C *** GENERATE AN ARRIVAL, ITS TYPE AND SERVICE TIME
C

EXECUTE ARRIVAL
IF(TYP.EQ.1) THEN
   NT1=NT1+1
   TAXIS=AR
   EXECUTE RANDOM
   DP=AR + (-1./AMUI)*ALOS(YFL)
ELSE
   NT2=NT2+1
   TAXIS=AR
   EXECUTE RANDOM
   DP=AR + (-1./AMU2)*ALOS(YFL)
   GO TO 20
ENDIF

C *** GENERATE AN ARRIVAL WITH A SINGLE TYPE-1 IN SERVICE
C
10 EXECUTE ARRIVAL
IF(AR.GT.CLOCK) GO TO 99
IF(TYP.EQ.1.AND.DP.GE.AR) THEN
EXECUTE JUMP1
DP=AR+ST1
HPKB1=ST1
TAXIS=AR
FLAG=1
ELSEIF(TYP.EQ.1.AND.DP.LT.AR) THEN
NT1=NT1+1
EXECUTE OLEV1
GO TO 10
ELSEIF(TYP.EQ.2.AND.DP.GE.AR) THEN
HTRR=0.
EXECUTE JUMP12
DP=AR+HPEAK
FLAG=2
ELSEIF(TYP.EQ.2.AND.DP.LT.AR) THEN
NT2=NT2+1
EXECUTE OLEV2
GO TO 20
ENDIF
GO TO 30
C
C *** GENERATE AN ARRIVAL WITH A SINGLE TYPE-2 IN SERVICE
C
20 EXECUTE ARRIVAL
IF(AR.GT.CLOCK) GO TO 99
IF(TYP.EQ.1.AND.DP.GE.AR) THEN
HTRR=0
EXECUTE JUMP12
DP=AR+ST12
TAXIS=AR
FLAG=2
ELSEIF(TYP.EQ.1.AND.DP.LT.AR) THEN
NT1=NT1+1
EXECUTE OLEV1
GO TO 10
ELSEIF(TYP.EQ.2.AND.DP.GE.AR) THEN
HTRR=0
EXECUTE JUMP22
DP=AR+ST22
TAXIS=AR
FLAG=3
ELSEIF(TYP.EQ.2.AND.DP.LT.AR) THEN
NT2=NT2+1
EXECUTE OLEV2
GO TO 20
ENDIF
30 EXECUTE ARRIVAL
C
C *** CHECK FOR END OF SIMULATED TIME PERIOD
C
IF(AR.GT.CLOCK) GO TO 99
C
C *** NEXT TYPE-1 ARRIVAL OCCURS DURING A TYPE-1 BUSY PERIOD
C
IF(TYP.EQ.1.AND.DP.GE.AR.AND.FLAG.EQ.1) THEN
  EXECUTE JUMP11
  HPKBPI = HPKBPI + ST11
  DP = DP + ST11
  FLAG = 1
  GO TO 30
C
C *** NEXT TYPE-1 ARRIVAL OCCURS AFTER A TYPE-1 BUSY PERIOD
C
ELSEIF(TYP.EQ.1.AND.DP.LT.AR.AND.FLAG.EQ.1) THEN
  EXECUTE LORDBP
  GO TO 95
C
C *** NEXT TYPE-2 ARRIVES DURING A TYPE-1 BUSY PERIOD
C
ELSEIF(TYP.EQ.2.AND.DP.GE.AR.AND.FLAG.EQ.1) THEN
  NA = NA + 1
  TA(NA) = AR
  GO TO 1000
C
C *** IF A TYPE-2 ARRIVAL OCCURS AFTER A TYPE-1 BUSY PERIOD
C
ELSEIF(TYP.EQ.2.AND.DP.LT.AR.AND.FLAG.EQ.1) THEN
  EXECUTE LORDBP
  GO TO 70
C
C *** NEXT TYPE-1 ARRIVAL DURING A MIXED BUSY PERIOD
C
ELSEIF(TYP.EQ.1.AND.DP.GE.AR.AND.FLAG.EQ.2) THEN
  HPKBPI = HPEAK
  EXECUTE RANDOM
  IF(YFL.LE.PREDP) THEN
    HTRR = HPKBPI
    EXECUTE JUMP12
    HPKBPI = HPEAK
    DP = DP + ST12
    TAXIS = AR
    FLAG = 2
  ELSE
    EXECUTE JUMP11
    HPKBPI = HPKBPI + ST11
    DP = DP + ST11
    FLAG = 1
  ENDIF
GO TO 30

C *** IF A TYPE-1 ARRIVES DURING THE TIME PERIOD WHEN THERE IS A SINGLE
C *** TYPE-1 OR TYPE-1 IN SERVICE

C ELSEIF(TYPE .EQ. 1 .AND. DP .LT. AR .AND. FLAG .EQ. 2) THEN
    HPKBPI = HPEAK
    EXECUTE LORDBP
    EXECUTE RANDOM
    IF (YFL .LE. PRDEP) THEN
        GO TO 80
    ELSE
        GO TO 85
    ENDIF

C *** NEXT TYPE-2 ARRIVES DURING A MIXED BUSY PERIOD

C ELSEIF(TYPE .EQ. 2 .AND. DP .GE. AR .AND. FLAG .EQ. 2) THEN
    NA = NA + 1
    TA(NA) = AR
    GO TO 2000

C *** NEXT TYPE-2 ARRIVAL OCCURS DURING THE TIME PERIOD WHEN THERE IS
C *** A SINGLE TYPE-1 OR TYPE-2 IN SERVICE

C ELSEIF(TYPE .EQ. 2 .AND. DP .LT. AR .AND. FLAG .EQ. 2) THEN
    HPKBPI = HPEAK
    EXECUTE LORDBP
    EXECUTE RANDOM
    IF (YFL .LE. PRDEP) THEN
        GO TO 90
    ELSE
        GO TO 70
    ENDIF

C *** NEXT TYPE-1 ARRIVES DURING A TYPE-2 BUSY PERIOD

C ELSEIF(TYPE .EQ. 1 .AND. DP .GE. AR .AND. FLAG .EQ. 3) THEN
    HTTR = HPEAK
    EXECUTE JUMP12
    DP = DP + ST12
    FLAG = 2
    GO TO 30
ELSEIF(TYPE .EQ. 1 .AND. DP .LT. AR .AND. FLAG .EQ. 3) THEN
    HPKBPI = HPEAK
    EXECUTE LORDBP
    GO TO 80

C *** IF A TYPE-2 ARRIVES DURING A TYPE-1 BUSY PERIOD

C ELSEIF(TYPE .EQ. 2 .AND. DP .GE. AR .AND. FLAG .EQ. 3) THEN
NA=NA+1
TA(NA)=AR
HPKBPI=HPEAK
GO TO 60
ELSEIF (TYP.EQ.2.AND.DP.LT.AR.AND.FLAG.EQ.3) THEN
   HPKBPI=HPEAK
   EXECUTE LORDBP
   GO TO 90
ENDIF

1000 EXECUTE BPTYP1
   IF(IFLAG.EQ.3) GO TO 1000
   IF(IFLAG.EQ.4) GO TO 40
   IF(IFLAG.EQ.5) GO TO 45
   IF(IFLAG.EQ.6) GO TO 50
   IF(IFLAG.EQ.7) GO TO 55
   IF(IFLAG.EQ.8) GO TO 60
   IF(IFLAG.EQ.25) GO TO 99
   IF(IFLAG.EQ.9) GO TO 100
   IF(IFLAG.EQ.10) GO TO 95

40 EXECUTE MIDLAY
   GO TO 110

45 EXECUTE SEARCH
   IF(IFLAG.EQ.27) GO TO 45
   IF(IFLAG.EQ.28) GO TO 110
   IF(IFLAG.EQ.15) GO TO 80

50 EXECUTE ORDLAY
51 IF(NA.GE.1) THEN
   GO TO 1000
ELSE
   GO TO 30
ENDIF

55 EXECUTE CHECK1
   IF(IFLAG.EQ.30) GO TO 45
   IF(IFLAG.EQ.31) GO TO 55
   IF(IFLAG.EQ.32) GO TO 110
   IF(IFLAG.EQ.33) GO TO 65
   IF(IFLAG.EQ.35) GO TO 51

60 EXECUTE BPTYP2
   IF(IFLAG.EQ.11) GO TO 60
   IF(IFLAG.EQ.12) GO TO 1000
   IF(IFLAG.EQ.3) GO TO 110
   IF(IFLAG.EQ.2) GO TO 45
   IF(IFLAG.EQ.22) GO TO 100
   IF(IFLAG.EQ.26) GO TO 99

90 EXECUTE STYP22
   IF(IFLAG.EQ.21) GO TO 30
   IF(IFLAG.EQ.22) GO TO 20

70 EXECUTE STYP12
   IF(IFLAG.EQ.1) GO TO 30
   IF(IFLAG.EQ.2) GO TO 20
80 EXECUTE STYP21
    IF(IFLAG.EQ.1) GO TO 30
    IF(IFLAG.EQ.2) GO TO 10
85 EXECUTE STYP11
    IF(IFLAG.EQ.1) GO TO 30
    IF(IFLAG.EQ.2) GO TO 10
95 EXECUTE CHECK2
    IF(IFLAG.EQ.1.OR.IFLAG.EQ.2) GO TO 30
    IF(IFLAG.EQ.2) GO TO 10
    IF(IFLAG.EQ.40) GO TO 100
    IF(FLAG.EQ.41) GO TO 95
    IF(FLAG.EQ.45.CR.IFLAG.EQ.47) GO TO 60
    IF(FLAG.EQ.46) GO TO 1000
100 EXECUTE CHECK3
    IF(FLAG.EQ.21) GO TO 30
    IF(FLAG.EQ.22) GO TO 20
    IF(FLAG.EQ.40) GO TO 100
    IF(FLAG.EQ.47) GO TO 60
110 EXECUTE RANDOM
    IF(NA.EQ.1) THEN
        IF(YFL.LE.PRDEP) THEN
            GO TO 60
        ELSE
            GO TO 1000
        ENDIF
    ELSE
        GO TO 30
    ENDIF
2000 EXECUTE RANDOM
    IF(YFL.GT.PREDEP) THEN
        HPKBP1=HPEAK
        GO TO 1000
    ELSE
        HPKBP1=HPEAK
        HTRR=HPEAK
        GO TO 60
    ENDIF
99 EXECUTE RESULT
    GO TO 999
C ==========================================================================
C REMOTE BLOCK LISTING
C ==========================================================================
C
C ------------------------
C REMOTE BLOCK RANDOM
C ------------------------
C *** GENERATE 0-1 UNIFORM RANDOM NUMBERS.
C
C CALL RANDU(IX,IY,YFL)
C IX=IY
ENDBLOCK
C ------------------------
REMOTE BLOCK PEAK
C ------------------------
C ### INCREMENT BY 1 EACH W WHICH IS LESS THAN HT. OF THE PEAK.
C
J=1
WHILE(PKK(J).LT.HPEAK)
   NPkk(J)=NPkk(J)+1
   J=J+1
END WHILE
ENDBLOCK

C ------------------------
REMOTE BLOCK TROUGH
C ------------------------
C ### INCREMENT BY 1 EACH W WHICH IS LESS THAN HT. OF THE TROUGH.
C
K=1
WHILE(TRR(K).LT.HTRR)
   NTRR(K)=NTRR(K)+1
   K=K+1
END WHILE
ENDBLOCK

C ------------------------
REMOTE BLOCK ARRIVAL
C ------------------------
C ### GENERATE ARRIVAL EPOCHS OF CUSTOMERS AND CHECK THEIR TYPE.
C
EXECUTE RANDOM
AT=(-1./LAMDA1+LAMDA2)*ALOG(YFL)
AR = TAXIS+AT
EXECUTE RANDOM
IF(YFL.LE.PROBT1) THEN
   TYP=1
ELSE
   TYP=2
ENDIF
ENDBLOCK

C ------------------------
REMOTE BLOCK JUMP11
C ------------------------
C ### DETERMINE THE HEIGHT OF PEAK WHEN THE JUMP SIZE IS DUE TO TYPE-1.
C
TAXIS=AR
EXECUTE RANDOM
ST11=(-1./H1)*ALOG(YFL)
ENDBLOCK

C ------------------------
REMOTE BLOCK OLEV1
C ------------------------
C ### DETERMINE THE SERVICE TIME FOR A SINGLE TYPE-1 UNIT.
C
TAXIS=AR
EXECUTE RANDOM
DP{TAXIS+(-1./AMU1)*ALOG(YFL)}
ENDBLOCK
C ------------------------
REMOTE BLOCK JUMP12
C ------------------------
C *** DETERMINE THE HEIGHT OF PEAK WHEN THE JUMP SIZE IS DUE TO A
C *** TYPE-1 AND A TYPE-2.
C
TAXIS=AR
EXECUTE RANDOM
ST12={-1./(AMU1+AMU2)}*ALOG(YFL)
HPEAK=ST12+HTRR
ENDBLOCK
C ------------------------
REMOTE BLOCK OLEV2
C ------------------------
C *** FIND THE SERVICE TIME FOR A TYPE-2 UNIT.
C
TAXIS=AR
EXECUTE RANDOM
DP{TAXIS+(-1./AMU2)*ALOG(YFL)}
ENDBLOCK
C ------------------------
REMOTE BLOCK JUMP22
C ------------------------
C *** DETERMINE THE HEIGHT OF PEAK WHEN THE JUMP SIZE IS DUE TO TWO
C *** TYPE-2'S.
C
TAXIS=AR
EXECUTE RANDOM
ST22={-1./(2*AMU2)}*ALOG(YFL)
HPEAK=ST22+HTRR
ENDBLOCK
C ------------------------
REMOTE BLOCK STYP11
C ------------------------
C *** GENERATE SAMPLE PATH WHEN A TYPE-1 ARRIVAL SEES A SINGLE TYPE-1
C *** IN SERVICE
C
TAXIS=DP
IMPK1=IMPK1+1
EXECUTE RANDOM
STI={-1./AMU1}*ALOG(YFL)
DP=TAXIS+ST1
IF(DP,6E,AR) THEN
  EXECUTE JUMP11
HPKBP1=ST11
DP=AR+ST11
FLAG=1
IFLAG=1
ELSE
    EXECUTE OLEV11
    NT1=NT1+1
    IFLAG=2
ENDIF
END BLOCK

C -----------
REMOTE BLOCK STYP12
C -----------
C *** GENERATE SAMPLE PATH WHEN A TYPE-2 ARRIVAL SEES A SINGLE TYPE-2
C *** IN SERVICE
C
TAXIS=DP
IMPK1=IMPK1+1
EXECUTE RANDOM
ST1=(-1./AMU1)*ALOG(YFL)
DP=TAXIS+ST1
IF(DP.GE.AR) THEN
    HTRR=0.
    EXECUTE JUMP12
    DP=AR+HPEAK
    IFLAG=2
    IFLAG=1
ELSE
    EXECUTE OLEV12
    NT2=NT2+1
    IFLAG=2
ENDIF
END BLOCK

C -----------
REMOTE BLOCK STYP21
C -----------
C *** GENERATE SAMPLE PATH WHEN A TYPE-1 ARRIVAL SEES A SINGLE TYPE-2
C *** IN SERVICE
C
TAXIS=DP
IMPK2=IMPK2+1
EXECUTE RANDOM
ST2=(-1./AMU2)*ALOG(YFL)
DP=TAXIS+ST2
IF(DP.GE.AR) THEN
    HTRR=0.
    EXECUTE JUMP12
    DP=AR+HPEAK
    IFLAG=2
    IFLAG=1
ELSE
    EXECUTE OLEV11
    NT1=NT1+1
    IFLAG=2

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
ENDIF
ENDBLOCK
C ------------------------
REMOTE BLOCK STYP22
C ------------------------
C ### GENERATE SAMPLE PATH WHEN A TYPE-2 ARRIVAL SEES A SINGLE TYPE-2
C ### IN SERVICE
C
TAXIS=DP
IMPK2=IMPK2+1
EXECUTE RANDOM
ST2=(-1./AMU2) * ALOG(YFL)
DP=TAXIS+ST2
IF(DP.GE.AR) THEN
  HTRR=0,
  EXECUTE JUMP22
  DP=AR+HPEAK
  FLAG=3
  IFLAG=21
ELSE
  EXECUTE OLEVTL2
  NT2=NT2+1
  IFLAG=22
ENDIF
ENDBLOCK
C ------------------------
REMOTE BLOCK LORDBP
C ------------------------
C
HPEAK=HPKBPI
EXECUTE PEAK
HPKBPI=0.
ENDBLOCK
C ------------------------
REMOTE BLOCK ORDLAY
C ------------------------
C
TAXIS=AR
EXECUTE JUMP11
HPKBPI=HPEAK + ST11
DP=DP+ST11
FLAG=1
ENDBLOCK
C ------------------------
REMOTE BLOCK MIDLAY
C ------------------------
C
TAXIS=AR
EXECUTE RANDOM
ST12=(-1./AMU1*AMU2) * ALOG(YFL)
HPKBPI=HPEAK+ST12
HPEAK=HFKBP1
DP=DP+ST12
FLAG=2
ENDBLOCK

C --------------------------
REMOTE BLOCK UPDATE
C --------------------------

C
NA=NA-1
IF(NA.GE.1) THEN
   DO 26 I=1,NA
      II=I+1
      TA(II)=TA(I)
   26 CONTINUE
ENDIF
END BLOCK

C --------------------------
REMOTE BLOCK BPTYP1
C --------------------------

C ### GENERATE THE LENGTH OF A ORDINARY BUSY PERIOD
C
TAXIS=AR
EXECUTE ARRIVAL
IF(AR.GT.CLOCK) THEN
   IFLAG=25
ELSEIF(TYP.EQ.1.AND.DP.GE.AR) THEN
   EXECUTE JUMP11
   HPKBP1=HPKBP1+ST1
   DP=DP+ST1
   FLAG=1
   IFLAG=3
ELSEIF(TYP.EQ.1.AND.DP.LT.AR) THEN
   EXECUTE DLAY1
ELSEIF(TYP.EQ.2.AND.DP.GE.AR) THEN
   NA=NA+1
   TA(NA)=AR
   IFLAG=3
ELSEIF(TYP.EQ.2.AND.DP.LT.AR) THEN
   EXECUTE DLAY2
ENDIF
END BLOCK

C --------------------------
REMOTE BLOCK DLAY1
C --------------------------

C ### GENERATE THE LENGTH OF A DELAYED BUSY PERIOD
C
EXECUTE LORDBP
HTRR=(DP-TA(II))
EXECUTE THROUGH
EXECUTE JUMP12
DP=DP+ST12

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
EXECUTE UPDATE
EXECUTE RANDOM
IF(YFL.LE.PRDEP.AND.AR.LE.DP) THEN
  IFLAG=4
ELSEIF(YFL.LE.PRDEP.AND.AR.GT.DP) THEN
  IFLAG=5
  IND=10
ELSEIF(YFL.GT.PRDEP.AND.AR.LE.DP) THEN
  IFLAG=6
ELSE
  IFLAG=7
  IND=11
ENDIF
ENDBLOCK

C ------------------------
REMOTE BLOCK DLY2
C ------------------------
C
EXECUTE LORDBP
HTRR=DP-TA(1)
EXECUTE THROUGH
EXECUTE JUMP12
DP=DP+ST12
EXECUTE UPDATE
EXECUTE RANDOM
IF(YFL.LE.PRDEP.AND.AR.LE.DP) THEN
  HPKBPl=HPEAK
  NA=NA+1
  TA(NA)=AR
  IFLAG=8
ELSEIF(YFL.LE.PRDEP.AND.AR.GT.DP) THEN
  IFLAG=9
ELSEIF(YFL.GT.PRDEP.AND.AR.LE.DP) THEN
  HPKBPl=HPEAK
  NA=NA+1
  TA(NA)=AR
  IFLAG=3
ELSE
  IFLAG=10
ENDIF
ENDBLOCK

C ------------------------
REMOTE BLOCK BPTYP2
C ------------------------
C *** GENERATE THE LENGTH OF A MIXED BUSY PERIOD
C
taxi=ar
EXECUTE ARRIVAL
IF(AR.GT.CLOCK) THEN
  IFLAG=26
ELSEIF(TYP.EQ.1.AND.DP.GE.AR) THEN
HTRR=HPKBPI
EXECUTE JUMP12
HPKBPI=HPEEK
DP=DP+ST12
EXECUTE RANDOM
IF(YFL.LE.PRDEP) THEN
   IFLAG=11
ELSE
   IFLAG=12
ENDIF
ELSEIF(TYP.EQ.1.AND.DP.LT.AR) THEN
   EXECUTE LORDBP
   HTRR=DP-TA(1)
   EXECUTE THROUGH
   EXECUTE JUMP22
   DP=DP+ST22
   EXECUTE UPDATE
   IF(DP.LT.AR) THEN
      IFLAG=2
      IND=10
   ELSE
      HTRR=HPEEK
      EXECUTE JUMP12
      HPKBPI=HPEEK
      DP=DP+ST12
      IFLAG=2
      IFLAG=3
   ENDIF
ELSEIF(TYP.EQ.2.AND.DP.LE.AR) THEN
   NA=NA+1
   TA(NA)=AR
   IFLAG=11
ELSEIF(TYP.EQ.2.AND.DP.LT.AR) THEN
   EXECUTE LORDBP
   HTRR=DP-TA(1)
   EXECUTE THROUGH
   EXECUTE JUMP22
   DP=DP+ST22
   EXECUTE UPDATE
   IF(DP.LT.AR) THEN
      IFLAG=22
   ELSE
      HPKBPI=HPEEK
      NA=NA+1
      TA(NA)=AR
      IFLAG=11
   ENDIF
ENDIF
ENDBLOCK
C ---------------------------------
REMOTE BLOCK CHECK1
C -----------------------------------
C *** A WAITING TYPE-2 ENTERS SERVICE AT THE COMPLETION OF A TYPE-1
C *** BUSY PERIOD.
C

IF(NA.GE.1) THEN
  HPKBPI=HPEAK
  EXECUTE LORDBP
  HTRR=DP-TA(1)
  EXECUTE TROUGH
  EXECUTE JUMP12
  DP=DP+ST12
  EXECUTE UPDATE
  EXECUTE RANDOM
  IF(DP.LT.AR.AND.YFL.LE.PRDEP) THEN
    IFLAG=30
    IND=10
  ELSEIF(DP.LT.AR.AND.YFL.GT.PRDEP) THEN
    IFLAG=31
  ELSEIF(DP.GE.AR.AND.YFL.LE.PRDEP) THEN
    HTRR=HPEAK
    EXECUTE JUMP12
    HPKBPI=HPEAK
    DP=DP+ST12
    FLAG=2
    IFLAG=32
  ELSEIF(DP.GE.AR.AND.YFL.GT.PRDEP) THEN
    EXECUTE JUMP11
    HPKBPI=HPEAK+ST11
    FLAG=1
    IFLAG=35
  ENDIF
ELSE
  EXECUTE PEAK
  IF(IND.EQ.11) IFLAG=33
ENDIF
ENDBLOCK

C -----------------------------------
REMOTE BLOCK CHECK2
C -----------------------------------
C

IF(NA.GE.1) THEN
  HPKBPI=HPEAK
  EXECUTE LORDBP
  HTRR=DP-TA(1)
  EXECUTE TROUGH
  EXECUTE JUMP12
  DP=DP+ST12
  EXECUTE UPDATE
  EXECUTE RANDOM
  IF(YFL.LE.PRDEP.AND.DP.GE.AR) THEN
    NA=NA+1

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
TA(NA)=AR
IFLAG=45
ELSEIF(YFL.LE.PRDEP.AND.DP.LT.AR) THEN
  EXECUTE CHECK3
ELSEIF(YFL.GT.PRDEP.AND.DP.GE.AR) THEN
  HPKBPI=HPEAK
  NA=NA+1
  TA(NA)=AR
  IFLAG=46
ELSE
  IFLAG=41
ENDIF
ELSE
  EXECUTE PEAK
  EXECUTE STYP12
ENDIF
ENDBLOCK

REMOTE BLOCK CHECK3

+++ A WAITING TYPE-2 ENTERS SERVICE AT THE END OF A MIXED BUSY
+++ PERIOD WITH A TYPE-2 AS HIS NEIGHBOR.

IF(NA.GE.1) THEN
  HPKBPI=HPEAK
  EXECUTE LORDBP
  HTRR=DP-TA(1)
  EXECUTE THROUGH
  EXECUTE JUMP22
  DP=DP+ST22
  EXECUTE UPDATE
  IF(DP.GT.AR) THEN
    HPKBPI=HPEAK
    NA=NA+1
    TA(NA)=AR
    IFLAG=47
  ELSE
    IFLAG=40
  ENDIF
ELSE
  EXECUTE PEAK
  EXECUTE STYP22
ENDIF
ENDBLOCK

REMOTE BLOCK SEARCH

IF(NA.GE.1) THEN
  HPKBPI=HPEAK
  EXECUTE LORDBP

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
HTRR=DP-TA(1)
EXECUTE TROUGH
EXECUTE JUMP22
DP=DP+ST22
EXECUTE UPDATE
IF(DP.LT.AR) THEN
  HPKB1=HPEAK
  IFLAG=27
ELSE
  HTRR=HPEAK
  EXECUTE JUMP12
  HPKB1=HPEAK
  DP=DP+ST12
  FLAG=2
  IFLAG=28
ENDIF
ELSE
  EXECUTE PEAK
  IF(IND.EQ.10) IFLAG=15
ENDIF
ENDBLOCK

C -----------------------------------------------
REMOTE BLOCK RESULT
C -----------------------------------------------
C *** CALCULATE AND PRINT RESULTS.
C
S1(1)=0.
DO 101 I=2,NN
  S1(I)=S1(I-1)+W
101 CONTINUE
DO 102 J=1,NN
  DIFF(J)=NPK(J)-NTRR(J)
  DHX(J)=DIFF(J)/CLOCK
102 CONTINUE
PRINT 103
103 FORMAT(12X,'RESULTS FROM THE MONTE-CARLO SP SIMULATION')
PRINT,
' LAMDA1=' ,LAMDA1,'LAMDA2=' ,LAMDA2
PRINT,'AMU1=' ,AMU1,'AMU2=' ,AMU2
PRINT,'SEED=' ,SEED,'CLOCK=' ,CLOCK
PRINT,'# OF TYPE-1 IMPACTS =',IMPK1
PRINT,'# OF TYPE-2 IMPACTS =',IMPK2
PRINT,'# OF (00->01) TRANSITIONS=',NT1
PRINT,'# OF (00->02) TRANSITIONS=',NT2
PRINT 104
104 FORMAT(//,7X,' WAIT ',6X,'PEAK COUNT',5X,'TROUGH COUNT',8X,'DHX'
     1)
DO 105 I=1,NN
  PRINT 106,S1(I),NPK(I),NTRR(I),DHX(I)
106 FORMAT(6X,F5.1,7X,I4,12X,I4,12X,F6.4)
105 CONTINUE
DO 107 I=1,NN
   PRINT 108,I-1,DW(I)
108   FORMAT(5X,I4,5X,F6.4)
107   CONTINUE
END BLOCK
999 STOP
END

*ENTRY
0.25 0.20 0.333 0.250 1000 364820481 .5 150
*IBSYS
*STOP
//
APPENDIX D

COMPUTER PROGRAM LISTING

FOR PDF & CDF COMPUTATION
// MEAN JOB (R240, SL2), 'BACHAN', CLASS=I
// EXEC WATFIV
// EO.SYSIN DD $
$JOB WATFIV
C
C This program is designed to calculate the expected waiting time for type-I customers using the output from the SP(I) programs. It also plots the estimated PDF and CDF for Type-I customers.
C
REAL D(500), DX(500), T(500), TT(500), CDF(100), CCDF(100)
INTEGER WT(IOO)
REAL LAMDA1, LAMDA2
SUN1'SUM2>0
LAHDAl*LAHDA2*IHPKl«IMPK2«IMPK3«IMPK4*TRl«TR2»0
C
C Read data from the output of SP(I), I = 1, 2 program.
C
READ, LAMDA1, LAMDA2, AMU1, AMU2, CLOCK
READ, IMPK1, IMPK2, IMPK3, IMPK4, TR1, TR2, N, W
10 READ(S, t, ENO*SOO) I, CWX
C
C Calculate expected waiting time in queue using the downcrossing rate at level W, W > 0
C
SUM1=SUM1+I*DWX
SUM2=SUM2+DWX
D(I+1)=DWX
T(I+1)=I
GO TO 10
500 GSUM=0.01# SUM1+0.005*SUM2
PRINT 15, GSUM
15 FORMAT(5X, 'MEAN VALUE OF WAITING TIME = ', F9.6, ' MINUTES')
C
C Plot graph of the probability density function of the waiting time for type-I customers.
C
PRINT 20
20 FORMAT(///, 25X, 'GRAPH OF ESTIMATED PDF FOR SP(I): T=1000 MIN. ')
DO 25 J=1, N
DX(J)=O(J)
TT(J)=(J-1)*W
25 CONTINUE
CALL PLOT3(TT, DX, N)
C
C Calculate the probability of zero wait in queue.
C
SIGMA=LAMDA1+LAMDA2
A=IMPK1/CLOCK+IMPK2/CLOCK+TR1/CLOCK

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
C
C *** CALCULATE AND PLOT CDF OF THE WAITING TIME IN QUEUE FOR TYPE-I
C *** CUSTOMERS
C
J=2
CDF(1)=PO
DO 30 I=1,N,2
   R=D(I)+D(I+1)
   RR=0.5*R
   CDF(J)=CDF(J-1)+RR
   J=J+1
30 CONTINUE
M=N/2+1
DO 35 J=1,M
   CCDF(J)=CDF(J)
   WT(J)=(J-1)*I
   PRINT 40,WT(J),CCDF(J)
40 FORMAT(5X,I3,5X,F8.5)
35 CONTINUE
CALL PLOT3(WT,CCDF,M)
STOP
END
ENTRY
  0.25 0.2 0.333 0.25 1000
  41 00 58 00 31 32 98 0.5
#SYSIN
#STOP
//
E.1 Model Development

For the M/M/2 non-preemptive priority model with two customer types and parameters $\lambda_1, \lambda_2, \mu_1, \mu_2$ a discrete-event system simulation model was written in FORTRAN using the next-event time-advance approach. To facilitate development and debugging, the FORTRAN model was organized in a modular fashion using Remote Blocks.

The overall structure of the system simulation program is shown by the flowchart given in Figure E.1. The main program controls the flow of the event-scheduling, time-advance algorithm. In addition to a main program, the computer program consists of Remote Blocks and a Function subprogram. A program listing is given below; the listing provides the major FORTRAN variables the program uses (modelling variables include state variables, statistical counters and variables that are used to facilitate the writing of the program).

E.2 Variance Reduction Technique

For each random number stream, the method of antithetic sampling has been used as a variance reduction technique. This technique is applicable since we are dealing with a single system [Law & Kelton (1982)]. In this technique, the random number sequence $U_1, U_2, \ldots, U_n$ is used to
generate an input sequence on the first run, and \((1-U_1), (1-U_2), \ldots, (1-U_n)\) to generate the corresponding input sequence in the second run. The intuitive feeling here is that some of the negative correlation between inputs finds its way to the corresponding output and hence a variance reduction is realised. Antithetic sampling tries to induce this negative correlation by using complementary random numbers to drive the two runs in a pair.

### E.3 Model Verification

Verification is determining whether a simulation model performs as intended [Law & Kelton (1982)]. In order to determine whether the system simulation model is operating as intended, the model was run under simplifying assumptions for which the model's true characteristics are known analytically \((\mu_1 = \mu_2)\).

For the proposed system with different service rates for each class of customer, it is not possible to compute the desired system characteristics analytically. Therefore, one must resort to simulation. In order to verify the simulation model, we can run the general simulation model with the same service rate for each class of customers \((i.e. \mu_1 = \mu_2)\). The resulting model has known steady-state characteristics [Gross & Harris (1985)] (see Eq. (2.1) or (a.2.1)).
For the purpose of verification, the model was run with the following parameters:

\[
\begin{align*}
\lambda_1 &= 15 \text{ customers/hour} \\
\lambda_2 &= 12 \text{ customers/hour} \\
\mu_1 &= 20 \text{ customers/hour} \\
\mu_2 &= 20 \text{ customers/hour}
\end{align*}
\]

Given the model parameters above, we can calculate the steady-state expected waiting time for each class of customers analytically, using Equation (a.2.1) given in Appendix A. Table E.1 below illustrate the theoretical expected waiting times and also point estimates of this measure of performance from the system simulation run of length \( T = 1000 \) minutes. The table clearly indicates that the system simulation model produces very accurate estimates of the expected waiting times for either class of customers. Since the estimates are very close to the true values this gives us some degree of confidence in the correctness of the computer program.
Table E.1. Theoretical Values and System Simulation Estimates for a Non-Preemptive Priority Queueing System with Parameters $\lambda_1=15$, $\lambda_2=12$, $\mu_1=\mu_2=20$ (unit of time = 1 hour).

<table>
<thead>
<tr>
<th>Method of Analysis</th>
<th>Measure of Performance</th>
<th>Customer Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Type-1</td>
</tr>
<tr>
<td>Analytical Eq. (a.2.1)</td>
<td>$\bar{W}_{q_1}$ (hrs.)</td>
<td>0.021761</td>
</tr>
<tr>
<td>System Simulation T=1000 mins.</td>
<td>$\hat{E}[W_{q_1}]$ (hrs.)</td>
<td>0.022190</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
E.4 Statistical Estimation

The simulation was run for five independent random number streams; for each stream antithetic sampling was applied. In order to avoid having to deal with autocorrelation, the technique of independent replications was used. Each replication began with the same initial conditions, independence of replication being accomplished by using different random numbers for each replication (i.e. using different initial seeds). Now, for each replication, using its complementary random numbers we performed the simulation. Thus, five pairs of simulation runs were performed, each pair being correlated.

E.4.1 Output Analysis

Using the model parameters $\lambda_1=15$, $\lambda_2=12$, $\mu_1=20$, $\mu_2=15$, the queueing system was simulated for a total of $T=10000$ min. on each replication for the purpose of estimating long-run mean waiting time in queue for each customer type. The simulation results of five statistically independent pairs of runs (each pair is correlated) are shown in Table E.2. The overall point estimate for the mean waiting time in queue for type-1 and type-2 units are 0.030127 and 0.125667 hr. respectively. The average value of the Pr(any server idle) is 0.2396.
Table E.2: System Simulation Output Summary

\((T = 10000 \text{ min.})\)

<table>
<thead>
<tr>
<th>seed</th>
<th>sequence</th>
<th>Mean Waiting Time (min.)</th>
<th>Pr(idle server)</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456789</td>
<td>U</td>
<td>1.6294</td>
<td>5.6183</td>
</tr>
<tr>
<td></td>
<td>1-U</td>
<td>1.7562</td>
<td>7.1216</td>
</tr>
<tr>
<td>364820481</td>
<td>U</td>
<td>1.7725</td>
<td>7.0813</td>
</tr>
<tr>
<td></td>
<td>1-U</td>
<td>1.7509</td>
<td>7.8483</td>
</tr>
<tr>
<td>764928413</td>
<td>U</td>
<td>1.7722</td>
<td>8.1945</td>
</tr>
<tr>
<td></td>
<td>1-U</td>
<td>1.7527</td>
<td>5.7949</td>
</tr>
<tr>
<td>284756873</td>
<td>U</td>
<td>1.8926</td>
<td>9.3860</td>
</tr>
<tr>
<td></td>
<td>1-U</td>
<td>1.7327</td>
<td>6.3936</td>
</tr>
<tr>
<td>475638471</td>
<td>U</td>
<td>1.8879</td>
<td>8.5881</td>
</tr>
<tr>
<td></td>
<td>1-U</td>
<td>2.1288</td>
<td>9.3735</td>
</tr>
<tr>
<td>Mean Value (\bar{X}_i)</td>
<td>1.80761</td>
<td>7.5400</td>
<td>0.2396</td>
</tr>
<tr>
<td>(\hat{\sigma}^2(X_i))</td>
<td>0.0028825</td>
<td>0.1935112</td>
<td>5.6526E-06</td>
</tr>
<tr>
<td>(\hat{\sigma}(X_i))</td>
<td>0.0536887</td>
<td>0.4398991</td>
<td>2.3775E-03</td>
</tr>
</tbody>
</table>
E.4.2 Confidence Interval for Point Estimates

Consider a 95% confidence interval for each measure of performance. In general for small sample sizes the confidence interval of the mean is given by

\[ \bar{X} \pm \left( t_{\alpha/2, f} \right) \hat{\sigma}(\bar{X}) \]

where \( \bar{X} \) and \( \hat{\sigma}(\bar{X}) \) are the mean and standard error of the point estimate \( (X_i) \), respectively and \( t_{\alpha/2, f} \) is the value of the t distribution with \( f \) degrees of freedom, leaving an area of \( \alpha/2 \) to the right.

**Mean waiting time in queue (min.):**

- **type-1 units:** \( \bar{X}_{q1} \pm t_{0.025, 4} \hat{\sigma}(\bar{X}_{q1}) \)
  
  1.80761 ± 2.776(0.0536887)
  
  1.80761 ± 0.14904
  
  1.65856 ≤ \( W_{q1} \) ≤ 1.95665

- **type-2 units:** \( \bar{X}_{q2} \pm t_{0.025, 4} \hat{\sigma}(\bar{X}_{q2}) \)
  
  7.5400 ± 2.776(0.439899)
  
  7.5400 ± 1.2211596
  
  6.31884 ≤ \( W_{q2} \) ≤ 8.76117

**Mean Pr(any server idle):**

\[ \bar{p} \pm t_{0.025, 4} \hat{\sigma}(\bar{p}) \]

0.2396 ± 2.776(0.0023775)

0.2396 ± 0.0065999

0.2331 ≤ \( p \) ≤ 0.2462

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
E.4.3 Confidence Interval for the Difference Between Waiting Times of Type-1 and Type-2 Units

Here we are dealing with a small sample and unequal population variances. The confidence interval of the difference between the mean waiting times, \((W_{q1} - W_{q2})\) for such properties is given by

\[
(X_{q1} - X_{q2}) \pm (t_{\alpha/2, f} \cdot \sqrt{\frac{\hat{\sigma}^2(X_{q1}) + \hat{\sigma}^2(X_{q2})}{n-1}})^{1/2}
\]

where \(\bar{X}_{q1}\) and \(\hat{\sigma}^2(X_{q1})\) are as defined in Section E.4.2, and \(t_{\alpha/2}\) is the value of the t distribution with \(f = \frac{(\hat{\sigma}^2(X_{q1}) + \hat{\sigma}^2(X_{q2}))^2}{(\hat{\sigma}^2(X_{q1}))^2/(n-1) + (\hat{\sigma}^2(X_{q2}))^2/(n-1)}\) degrees of freedom, leaving an area of \(\alpha/2\) to the right.

Thus

\[
f = \frac{(0.00288248 + 0.1935112)^2}{[(0.00288248)^2/4] + [(0.1935112)^2/4]} = 4.12
\]

C.I. = \((7.540 - 1.80761) \pm t_{0.025, 4.12}(0.196394)^{1/2}\)

= \([4.504829, 6.9599515]\]

Since the confidence interval for \((W_{q2} - W_{q1})\) is totally to the right of zero, there is a strong evidence for the hypothesis that \(W_{q2} > W_{q1}\). This will be tested formally in the next section.
E.4.4 Hypothesis Testing

At this point it would be interesting to test statistically, if there is a significant difference between the mean waiting time of the two classes of customers. Intuitively, due to the queueing discipline by which the system operates, one expects a difference between the waiting times.

Let

\[ H_0 : \bar{W}_{q1} = \bar{W}_{q2} \]
\[ H_1 : \bar{W}_{q1} \neq \bar{W}_{q2} \]

Choose \( \alpha = 0.05 \)

**Critical region:** \( T < -2.770 \) and \( T > 2.770 \)

where

\[ T = \frac{(\bar{X}_{q1} - \bar{X}_{q2}) - 0}{(\hat{\sigma}^2(\bar{X}_{q1}) + \hat{\sigma}^2(\bar{X}_{q2}))^{1/2}} \]

with \( f = 4.12 \) (degrees of freedom)

**Computations:**

\[ t = \frac{(7.5400 - 1.80761) - 0}{(0.00288248 + 0.1935112)^{1/2}} \]

\[ = 12.935167 \]

**Conclusion:** Since \( t > T \), \( H_0 \) is rejected and conclude that there is a significant difference between the waiting times of each customer type.
Figure E.1: Flowchart for the Overall Structure of the System Simulation

Start

Read input parameters

- Set simulation CLOCK = 0
- Initialize system state and statistical counters
- Initialize events list

- Call the Time-advance routine
- Call event routine

- Determine the next event type, i
- Advance the simulation CLOCK

- Update system state
- Update statistical counter
- Generate future events and add to the event list

Is the simulation run completed?

yes

- Compute estimates of the average waiting time
- Print statistics

Is this the first run with current seed?

yes

Replace Random # generator 'U' by (1-U)

no
- Compute estimate of the average waiting time for the complementary sequence
- Print statistics

Is this the last run?

- Perform confidence interval calculation from the average waiting times
- Print summary of statistics

Stop

- Go to top of program and read new seed
THE FOLLOWING ARE THE MAJOR NOTATIONS THAT WILL BE USED:

- MIAT(I) = MEAN INTERARRIVAL TIME OF CLASS-I, I = 1, 2
- AMU(I) = MEAN SERVICE TIME OF CUSTOMERS OF CLASS-I, I = 1, 2
- ST(I) = SERVICE TIME OF CLASS-I, I = 1, 2
- TNE(I) = TIME OF NEXT EVENT-I, I = 1, ..., 5
- TIME = CURRENT CLOCK TIME.
- ATIQ(I) = AVERAGE TIME IN QUEUE FOR CLASS-I, I = 1, 2
- AWQ(I) = AVERAGE TIME IN QUEUE FOR CLASS-I FOR THOSE WHO WAIT
- PROBIT = PROBABILITY OF AN IDLE TELLER.
- STATUS(I) = STATUS OF TELLER-I, I = 1, 2 (1 = BUSY, 0 = IDLE)
- TNA(I) = TOTAL # OF ARRIVAL OF CLASS-I.
- TNAWQ(I) = TOTAL # OF ARRIVAL OF CLASS-I WHO WAIT IN QUEUE
- TNIQ(I) = TOTAL # IN QUEUE OF CLASS-I.
- SEED = INITIAL SEED TO DRIVE THE R.N. GENERATOR.
- RATIQ(I) = AVERAGE TIME IN QUEUE FOR CLASS-I FOR EACH ANTITHETIC REPLICATION.
- RAWQ(I) = AVERAGE TIME IN QUEUE FOR CLASS-I FOR THOSE WHO WAIT FOR EACH ANTITHETIC REPLICATION.
- RPROB = AVERAGE PROBABILITY OF AN IDLE TELLER FOR EACH ANTITHETIC REPLICATION.
- ULJ(I) = UPPER LIMIT OF THE CONFIDENCE INTERVAL OF PARAMETER-I
- LLJ(I) = LOWER LIMIT OF THE CONFIDENCE INTERVAL OF PARAMETER-I
- TARRVL(I,J) = TIME OF ARRIVAL OF J-TH CUSTOMER OF TYPE-I
- YFL = UNIFORM 0-1 RANDOM NUMBER

---

REAL MIAT(5),AMU(5),TARRVL(5,2000),TOTDEL(5),ST(5),LL2(9),LL3
REAL TNE(5),DELAY(5),AMQ(5),AQL(5),ATIQ(5),UL1(9),UL2(9),LL(9)
REAL AAHQ(5,2000),SMATIQ(5),SMAWQ(5),RATIQ(5,10),RAWQ(5,10)
REAL RAWQ(5,2000),APROB(2000),RPROB(10),ATIQST(9),AWST(9)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
REAL ATIQ(9),AVOM(9),B1(9),B2(9),SSQ1(9),SSQ2(9),YY1(9),YY2(9)
INTEGER STATUS(2),Niq(5),SEED
INTEGER TNA(5),TNAHQ(5),IDEN(2)
COMMON SEED,II,IY,IND

C
C

C
C

*** SPECIFY THE NUMBER OF EVENT TYPES FOR THE TIMADV ROUTINE
C

NEVNTS=5
NM=0

C
C

C
C

*** EVENT 1 = DEPARTURE FROM SERVER-1
C
C

C
C

*** EVENT 2 = DEPARTURE FROM SERVER-2
C
C

C
C

*** EVENT 3 = ARRIVAL OF A TYPE-1 CUSTOMER
C
C

C
C

*** EVENT 4 = ARRIVAL OF A TYPE-2 CUSTOMER
C
C

C
C

*** EVENT 5 = SCHEDULE THE END OF THE SIMULATION
C
C

C
C

*** READ INPUT PARAMETERS
C

READ,(HIAT(I),I=3,4)
READ,(AMU(I),I=3,4)
READ,TEND
4000 READ(5,4,END=5000) SEED
   SMATQ(I)=SMAWQ(I)=SMPROB=0.
   3 CONTINUE

C
C

C
C

*** INITIALIZE THE SIMULATION
C

5 EXECUTE INITIAL
C

C
C

C
C

*** DETERMINE THE NEXT EVENT
C

10 EXECUTE TIMADV
C

C
C

C
C

*** EXECUTE THE APPROPRIATE EVENT ROUTINE
C

IF(NEXT.EQ.3.OR.NEXT.EQ.4) THEN
   EXECUTE ARRIVAL
   GO TO 10
ELSEIF(NEXT.EQ.1.OR.NEXT.EQ.2) THEN
   EXECUTE DEPART
   GO TO 10
ELSE
   GO TO 15
ENDIF

15 EXECUTE FTMADV
   IF(IFLAG.EQ.2) THEN
      GO TO 20
   ELSE
      EXECUTE DEPART

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
GO TO 15
ENDIF
20 EXECUTE STAT
EXECUTE RESULT
IF(INO.EQ.1) THEN
IND=2
EXECUTE ANTIT
GO TO 5
ELSE
EXECUTE ANTIT
EXECUTE REPLI
GO TO 4000
ENDIF
5000 EXECUTE CONCAL
EXECUTE STARES
GO TO 99
C -----------------------------
C -----------------------------
C *** REMOTE BLOCK LISTINGS ***
C -----------------------------
C -----------------------------
C *** REMOTE BLOCK 'INITIAL' INITIALIZES THE SIMULATION CLOCK, SYSTEM
C *** STATE AND OTHER VARIABLES TO BE USED IN THE SIMULATION.
C -----------------------------
REMOTE BLOCK INITIAL
C -----------------------------
C *** INITIALIZE THE SIMULATION CLOCK
C
TIME=0.
IFLAG=NFLAG=0
C
C *** INITIALIZE THE STATE VARIABLES
C
IX=SEED
TBT=0.
DO 110 J=1,2
   STATUS(J)=0
110 CONTINUE
DO 120 I=3,4
   NIQ(I)=0
   ST(I)=0.
120 CONTINUE
C
C *** INITIALIZE THE STATISTICAL COUNTERS.
C
DO 130 J=3,4
   TNA(J)=0
   TOTDEL(J)=0.
   TNAWQ(J)=0
130 CONTINUE
C
C
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
C #### INITIALIZE THE EVENT LIST. SINCE NO CUSTOMERS ARE PRESENT, THE
C #### TIME OF THE NEXT DEPARTURE IS SET AT A LARGE VALUE
C
DO 150 K=3,4
   TNE(K)=TIME+EXPON(MIAT(K))
150 CONTINUE
DO 160 J=1,2
   TNE(J)=1.E+30
160 CONTINUE
TNE(5)=TEND
ENDBLOCK
C
C #### REMOTE BLOCK 'TIMADV' DETERMINES THE TIME OF OCCURRENCE OF
C #### NEXT EVENT TYPE.
C
REMOTE BLOCK TIMADV
C
RMIN=1.E+29
NEXT=0
C
C #### DETERMINE THE EVENT TYPE OF THE NEXT EVENT TO OCCUR.
C
DO 210 I=1,NEVNTS
   IF(TNE(I).GE.RMIN) GO TO 210
   RMIN=TNE(I)
   NEXT=I
210 CONTINUE
C
C #### ADVANCE THE SIMULATION CLOCK
C
TIME=TNE(NEXT)
ENDBLOCK
C
C #### REMOTE BLOCK 'FTMADV' DETERMINES THE TIME OF DEPARTURE OF NEXT
C #### EVENT TYPE WHEN THE SIMULATION RUN LENGTH IS REACHED.
C
REMOTE BLOCK FTMADV
C
RMIN=1.E+29
DO 49 J=3,4
   TNE(J)=1.E+30
49 CONTINUE
NEXT=0
DO 59 I=1,4
   IF(TNE(I).GE.RMIN) GO TO 59
   RMIN=TNE(I)
   NEXT=I
59 CONTINUE
IF(NEXT.NE.0) THEN
   TIME=TNE(NEXT)
ELSE
IFLAG=2
ENDIF
ENDBLOCK

C
C  *** REMOTE BLOCK 'ARRIVAL' EXECUTES THE ARRIVAL EVENT WHEN IT OCCURS.
C  ----------------------------------------
C  REMOTE BLOCK ARRIVAL
C  ----------------------------------------
C  TNA(NEXT)=TNA(NEXT)+1
C
C  *** SCHEDULE THE NEXT ARRIVAL
C
D0 310 J=3,4
    TNE(J)=TIME+EXPON(MIAT(J))
310 CONTINUE

C  *** IF SERVERS ARE BUSY, ADD ONE TO THE NUMBER OF CUSTOMERS IN THE
C  *** PARTICULAR QUEUE AND STORE THE TIME OF ARRIVAL OF THE ARRIVING
C  *** CUSTOMER IN THE ARRAY 'TARRVL'
C
EXECUTE SERVER
    IF(STATUS(1).EQ.1.AND.STATUS(2).EQ.1) THEN
        NIQ(NEXT)=NIQ(NEXT)+1
        TNAW0(NEXT)=TNAW0(NEXT)+1
        TARRVL(NEXT,NIQ(NEXT))=TIME
    ELSE

C  *** IF A SERVER IS IDLE START SERVICE ON THE ARRIVING CUSTOMER
C
    DELAY(NEXT)=0.
    TOTDEL(NEXT)=TOTDEL(NEXT)+DELAY(NEXT)

C  *** MAKE A SERVER BUSY AT RANDOM BY EXECUTING THE ASSIGN BLOCK
C
EXECUTE ASSIGN
    STATUS(JJ)=1

C  *** SCHEDULE A DEPARTURE
C
    Y=EXPON(AMU(NEXT))
    TNE(JJ)=TIME+Y
    ST(NEXT)=ST(NEXT)+Y
ENDIF
ENDBLOCK

C
C  *** REMOTE BLOCK 'DEPART' EXECUTES THE DEPARTURE EVENT WHEN IT OCCURS
C  ----------------------------------------
C  REMOTE BLOCK DEPART
C  ----------------------------------------
C  *** IF QUEUE IS EMPTY MAKE SERVER IDLE AND SET THE TIME OF THE NEXT
C  *** DEPARTURE TO A LARGE VALUE
IF(NIQ(3).EQ.0.AND.NIQ(4).EQ.0) THEN
  STATUS(NEXT)=0
  TNE(NEXT)=1.E+30
ELSE

C *** IF QUEUE IS NOT EMPTY, SCHEDULE A DEPARTURE; TYPE-1 CUSTOMERS HAVING
C *** PRIORITY OVER TYPE-2'S

DO 425 J=3,4
  IF(NIQ(J).GE.1) THEN
    NIQ(J)=NIQ(J)-1
    DELAY(J)=TIME-TARRVL(J,1)
    TOTDEL(J)=TOTDEL(J)+DELAY(J)
    Y=EXPON(AMU(J))
    ST(J)=ST(J)+Y
    TNE(NEXT)=TIME+Y
    K=NIQ(J)
  IF(NIQ(J).GE.1) THEN
    DO 430 I=1,K
      II=I+1
      TARRVL(J,I)=TARRVL(J,II)
  430 CONTINUE
  ENDIF
GO TO 425
ENDIF
425 CONTINUE
435 CONTINUE
ENDIF
ENDBLOCK

C *** REMOTE BLOCK 'RANDOM' GENERATES 0-1 RANDOM NUMBERS USING IBM
C *** SUBROUTINE RANDU.

-----------------------------
REMOTE BLOCK RANDOM
-----------------------------
CALL RANDU(IX,IY,YFL)
IX=IY
IF(IND.EQ.2) YFL=1-YFL
ENDBLOCK

C *** REMOTE BLOCK 'SERVER' CHECKS FOR THE CURRENT STATUS OF THE SERVERS
-----------------------------
REMOTE BLOCK SERVER
-----------------------------
NIT=0
J=0
DO 510 I=1,2
  IF(STATUS(I).EQ.0) THEN
    NIT=NIT+1
    J=J+1
  ENDIF
510 CONTINUE
IDEN(J)=1
ENDIF
510 CONTINUE
IF(NIT.EQ.0) THEN
   DO 505 I=1,2
      STATUS(I)=1
   505 CONTINUE
ENDIF
ENDBLOCK

C ***REMOTE BLOCK 'ASSIGN' RANDOMLY Assigns AN ARRIVAL TO A FREE Server
C ---------------
REMOTE BLOCK ASSIGN
C ---------------
IF(NIT.EQ.1) THEN
   JJ=IDEN(1)
ELSEIF(NIT.EQ.2) THEN
   EXECUTE RANDOM
   IF(YFL.LE.0.5) THEN
      JJ=IDEN(1)
   ELSE
      JJ=IDEN(2)
   ENDIF
ENDIF
ENDBLOCK

C *** REMOTE BLOCK 'STAT' CALCULATES THE RELEVANT STATISTICS.
C -------------
REMOTE BLOCK STAT
C -------------
DO 610 J=3,4
   TBT=ST(J)+TBT
610 CONTINUE
DO 620 K=3,4
   ATIQ(K)=TOTDEL(K)/TNA(K)
   AWQ(K)=TOTDEL(K)/TNAWQ(K)
620 CONTINUE
PROBIT=(TIME#2)-TBT)/(TIME#2)
ENDBLOCK

C *** REMOTE BLOCK 'ANTIT' FINDS THE SUM OF THE MEANS OF EACH PAIR
C *** ANTITHETIC REPLICATIONS.
C --------------
REMOTE BLOCK ANTIT
C --------------
DO 980 K=3,4
   SMATIQ(K)=SMATIQ(K)+ATIQ(K)
   SMAWQ(K)=SMAWQ(K)+AWQ(K)
980 CONTINUE
SMPROB=SMPROB+PROBIT
ENDBLOCK

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
C
C *** REMOTE BLOCK REPLI FINDS THE MEAN FOR EACH PAIR OF REPLICATION
C ---------------------------------------------
C                    REMOTE BLOCK REPLI
C ---------------------------------------------
NM=NM+1
DO 981 J=3,4
   RATIQ(J,NM)=SMA IQ(J)/2.
   RAVQ(J,NM)=SMAVQ(J)/2.
   RPROB(NM)=SMPROB/2.
981 CONTINUE
ENDBLOCK
C
C *** REMOTE BLOCK CONCAL CALCULATES THE CONFIDENCE INTERVAL FOR THE
C *** MEASURE OF PERFORMANCE.
C ---------------------------------------------
C                    REMOTE BLOCK CONCAL
C ---------------------------------------------
DO 982 I=3,4
   RATIQG(I)=AWGQG(I)=PROBG=0.
982 CONTINUE
N=5
DO 983 J=1,N
   DO 984 L=3,4
      RATIQG(L)=ATIQG(L)+RATIQ(L,J)
      AWGQG(L)=AWGQG(L)+RAWQ(L,J)
   984 CONTINUE
   PROBG=PROBG+RPROB(J)
983 CONTINUE
DO 985 L=3,4
   ATIQM(L)=ATIQG(L)/N
   AWQM(L)=AWGQG(L)/N
985 CONTINUE
PROBM=PROBG/N
DO 1599 I=3,4
   B1(I)=B2(I)=B3=0.
1599 CONTINUE
DO 1600 I=1,N
   DO 1601 L=3,4
      B1(L)=B1(L)+(RATIQ(L,I)-ATIQM(L))*2
      B2(L)=B2(L)+(RAWQ(L,I)-AWQM(L))*2
   1601 CONTINUE
   B3=B3+(RPROB(I)-PROBM)**2
1600 CONTINUE
DO 1602 L=3,4
   SSQ1(L)=B1(L)/(N-1)
   SSQ2(L)=B2(L)/(N-1)
   SSQ3=B3/(N-1)
1602 CONTINUE
F=2.776
DO 1603 L=3,4

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
\[ YY1(L) = \frac{1}{2} \sqrt{SSQ1(L)/N} \]
\[ YY2(L) = \frac{1}{2} \sqrt{SSQ2(L)/N} \]
\[ YY3 = \frac{1}{2} \sqrt{SSQ3/N} \]

1603 CONTINUE
DO 1604 I=3,4
  UL1(I) = YY1(I) + ATIQM(I)
  UL2(I) = YY2(I) + AWQM(I)
  UL3 = YY3 + PR0BN
  LL1(I) = ATIQM(I) - YY1(I)
  LL2(I) = AWQM(I) - YY2(I)
  LL3 = PR0BN - YY3
1604 CONTINUE
ENDBLOCK

C
C $$$$ REMOTE BLOCK 'RESULT' PRINTS THE DESIRED PARAMETERS
C -------------------------

REMOTE BLOCK RESULT
C -------------------------

C $$$$ PRINT HEADING AND INPUT PARAMETERS
C

PRINT 710
710 FORMAT(/15X,'NON-PREEHTIVE PRIORITY QUEUEING MODEL')
PRINT 715
715 FORMAT(15X,'---------------------------------------------------------')
PRINT 720
720 FORMAT('/10X,'INPUT PARAMETERS TO THE SIMULATION MODEL')
PRINT 725
725 FORMAT(10X,'')
PRINT 730
730 FORMAT(/5X,'CUSTOMER TYPE',5X,'MEAN INTERARRIVAL TIME (MIN)',5X,'MEAN SERVICE TIME (MIN)')
DO 740 I=1,2
  PRINT 735,MIAT(I+2),AMU(I+2)
740 CONTINUE
PRINT 745,SEED
745 FORMAT(/5X,'INITIAL SEED FOR RANDOM NUMBER GENERATOR = ',I9)
PRINT,''
IF(IND.EQ.2) PRINT,' COMPLIMENTARY RANDOM NUMBER SEQUENCE USED'
PRINT 760,TIME
760 FORMAT(/5X,'SIMULATION RUNLENGTH = ',F8.1,3X, 'MINUTES')
PRINT 750
750 FORMAT(/15X,'RESULTS FROM THE SIMULATION')
PRINT 755
755 FORMAT(15X,'')
PRINT,' STEADY-STATE MEASURES OF PERFORMANCE'
PRINT,' '}
805 FORMAT(/5X,'CUSTOMER TYPE',5X,'AVERAGE TIME IN QUEUE BEFORE SERVIC
C
C *** REMOTE BLOCK 'STARES' PRINTS THE RESULTS OF THE STATISTICS.
C ---------------------------------------------------------
REMOTE BLOCK STARES
C ---------------------------------------------------------

PRINT 1312
1312 FORMAT(/)
    PRINT,
    SUMMARY OF STATISTICAL RESULTS'
    PRINT,
    'THE REPLICATION TECHNIQUE WAS APPLIED TO ESTIMATE THE'
    PRINT,
    'CONFIDENCE INTERVALS FOR THE MEASURES OF PERFORMANCE.'
    PRINT,
    'TO REDUCE THE VARIANCE OF THE ESTIMATORS ANTITHETIC'
    PRINT,
    'SAMPLING WAS APPLIED.'
1314 FORMAT('REPLICATIONS TYPE-1 TYPE-2')
1315 FORMAT(5X,I9,7X,F8.5,14X,F8.5)
1318 FORMAT(5X,'OVERALL POINT ESTIMATE FOR A TYPE-1,11,'=",F9.5)
PRINT 1312
    PRINT,
    'ANTITHETIC AVERAGE TIME IN QUEUE '
PRINT 1314
    DO 1326 J=1,N
        PRINT 1315,J,(RATIQ(I,J),1*3,4)
    1326 CONTINUE
    DO 1327 J=3,4
        I=J-2
        PRINT 1318,I,ATIQ(I)
        PRINT 1330,LL1(J),UL1(J)
    1327 CONTINUE
    PRINT 1312
    'ANTITHETIC AVERAGE TIME IN QUEUE FOR THOSE WHO'
    PRINT 1314
    DO 1331 J=1,N
        PRINT 1315,J,(RAWQ(I,J),1*3,4)
CONTINUE
  DO 1332 J=3,4
  I=J-2
  PRINT 1318,I,AWQM(J)
  PRINT 1333,LL2(J),UL2(J)
1333 FORMAT(5I,F8.4,'(== 95% C.I. FOR AVERAGE TIME IN QUEUE FOR THOSE
& WHO WAIT ==>)',F8.4)
1332 CONTINUE
  PRINT 1312
  PRINT,' ANTITHETIC REPLICATION PROBABILITIES OF IDLE TELLER'
  DO 1334 J=1,N
  1340 FORMAT(B%,I2,26X,F8.4)
  PRINT 1340,J,RRPR0(B)
  1334 CONTINUE
  PRINT,''
  PRINT 1341,PR0BM
  1341 FORMAT(5%,'OVERALL POINT ESTIMATE FOR PROBABILITY OF IDLE TELLER=
&',F8.4)
  PRINT 1337, LL3,UL3
1337 FORMAT(5I,F8.4,'(== 95% C.I. FOR PROBABILITY OF IDLE TELLER==>)
&',F8.4)
  PRINT 1312
END BLOCK
99 STOP
END

C
C *** THE FUNCTION EXPON GENERATES EXPONENTIAL INTERARRIVAL AND SERVICE
C *** TIMES (EXPONENTIAL RANDOM VARIATES) USING THE IBM RANDOM NUMBER
C *** GENERATOR. FOR THIS GENERATOR, WE NEED AN ODD SEED WIDTH NINE
C *** OR LESS DIGITS.

REAL FUNCTION EXPON(P)
INTEGER SEED,IX,IY
COMMON SEED,IX,IY,IND
CALL RANDU(IX,IY,YFL)
IX=IY
IF(IND.EQ.2) YFL=1-YFL
EXPON =-P*ALOG(YFL)
RETURN
END

ENTRY
4 5
3 4
1000
123456789
364820491
764928413
284756873
475638471
1IBSYS
STOP
//
VITA AUCTORIS

1959  Born in Trinidad, West Indies, on September 25th.

1979  Completed secondary education from ASJA Boys’ College and Naparima College, Trinidad.

1982  Graduated from the University of the West Indies, St. Augustine, Trinidad, with a Bachelor of Science degree in Chemical Engineering (First Class Honours).

1982  Worked as a Petroleum Engineer at Trinidad-Tesoro Petroleum Company Ltd., Trinidad.

1984  Awarded a Canadian Commonwealth Scholarship to pursue a Master’s degree in Industrial Engineering at the University of Windsor, Canada.

1986  Currently a candidate for Master of Applied Science in Industrial Engineering at the University of Windsor, Windsor, Ontario, Canada.