NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.
ANSE OF AUTHOR/NOM DE L'AUTEUR: Juan William Taylor

TITLE OF THESIS/TITRE DE LA THÈSE: Adaptive Routing Using the Expected Time in Tandem Queues

UNIVERSITY/UNIVERSITÉ: Carleton University

DEGREE FOR WHICH THESIS WAS PRESENTED/GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE: Master of Science

YEAR THIS DEGREE CONFERRED/ANNÉE D'OBTENTION DE CE GRADÉ: 1984

NAME OF SUPERVISOR/NOM DU DIRECTEUR DE THÈSE: Dr. C.M. Woodside

Permission is hereby granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

DATED/DATÉ: 30 April 1984 SIGNED/SIGNÉ: J. William

PERMANENT ADDRESS/RÉSIDENCE FIXE: 1313 Birchmont Cres

Ginchester, ON

(after 1 August 1984)
ADAPTIVE ROUTING USING THE EXPECTED TIME IN TANDEM QUEUES

by

Ivan W. Taylor

A thesis submitted to the Faculty of Graduate Studies in partial fulfilment of the requirements for the degree of Master of Science

Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario
1984
LICENSE TO CARLETON UNIVERSITY

In the interests of facilitating research by others at this institution and elsewhere, I hereby grant a licence to:

CARLETON UNIVERSITY

to make copies of my thesis:

Adaptive Routing Using the Expected Time in Random Queues

or substantial parts thereof, the copyright which is invested in me, provided that the licence is subject to the following conditions:

Only single copies shall be made or authorized to be made at any one time, and only in response to a written request from the library of any university or similar institution on its own behalf or on behalf of one of its users.

This licence shall continue for the full term of the copyright, or for so long as may be legally permitted.

The Universal Copyright Notice shall appear on the title page of all copies made under the authority of this licence.

This licence does not permit the sale of authorized copies at a profit, but does permit the collection by the institution or institutions concerned of charges covering actual costs.

All copies under the authority of this licence shall bear a statement to the effect that the copy in question is being made available in this form by the authority of the copyright owner solely for the purpose of private study and research and may not be copied or reproduced except as permitted by the copyright laws without written authority from the copyright owner.

The foregoing shall in no way preclude the granting by the author of a licence to the National Library of Canada to reproduce the thesis and to lend or to sell copies of the same. For this purpose it shall also be permissible for Carleton University to submit the abovementioned thesis to the National Library of Canada.

Signature of Witness  Signature of Student
Amanda  I. Antaki
L.  L.  Signature
St.  Sign

30 April 84  1984  M.S.c  Systems  Engineering
Date  Degree  Department of Student
The undersigned recommend to the Faculty
of Graduate Studies acceptance of the thesis:
"Adaptive Routing Using the Expected Time in Tandem Queues"
in partial fulfilment of the requirements for
the degree of Master of Science.

C. M. [signature]
Thesis Supervisor

B. [signature]
Chairman, Department of Systems
and Computer Engineering

Carleton University
April 1984
ABSTRACT

In a communication network, messages travelling from a source to a destination can be modelled as tokens passing through tandem queues. The time to travel through the system can be estimated with knowledge of the queue lengths seen on arrival. If there are alternative paths, then the message could be routed to the path with the shortest expected travel time in order to reduce the overall message delay.

A Markov Chain model is presented to estimate the time in tandem queues. This model is extended to consider finite storage, external arrivals and departures, and blocking. Unfortunately the computational requirements of the Markov Chain method are quite large. A fluid flow model is developed for infinite tandem queues with external arrivals and departures. The computations in this approach were much easier to automate.

Simulation was used to test the validity of these estimation methods. It was also used to examine the usefulness of these techniques for adaptive routing in queueing networks. The Markov Chain method performed poorly compared to other simpler methods. This may have been caused by problems in estimating the model parameters. The fluid flow model performed well when the system was moderately loaded and deserves consideration in future studies of adaptive routing.
ACKNOWLEDGEMENTS

The author wishes to express his thanks to the following persons.

a. Dr. Murray Woodside for his advice and encouragement as my supervisor;

b. Mr. Andrew Vernon for his assistance with the simulation programs;

c. Mr. Gordon Giddings for his suggestions concerning fluid flow methods and his help in printing the graphs;

d. Mr. Fred Kaudel, Mr. Ramiro Montealegre and Dr. Jim Tom for their assistance in editing an early version of this thesis;

e. Dr. Dave Stanford and Dr. Faruk Hadziomerovic for their helpful comments concerning the final version of this thesis; and

f. Ms. Barbara Maize for her support and encouragement and her many hours of word processing.

Financial support from the National Science and Engineering Research Council and the Department of National Defence was greatly appreciated.
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................... i  
ACKNOWLEDGEMENTS .................................................................................. ii  
LIST OF FIGURES ........................................................................................ iv  
LIST OF TABLES .............................................................................................. vi  
LIST OF NOTATION ......................................................................................... vii  

CHAPTER 1: INTRODUCTION AND MOTIVATION ........................................ 1  
CHAPTER 2: THE EXPECTED TIME IN TANDEM QUEUES ...................... 11  
CHAPTER 3: A MODEL OF TANDEM QUEUES WITH CROSS-TRAFFIC, FINITE STORAGE AND FULL-SERVER BLOCKING .......... 27  
CHAPTER 4: ALTERNATIVE MODELS INVOLVING RETRANSMISSION AND EMPTY-SERVER BLOCKING ......................... 42  
CHAPTER 5: AN ALTERNATIVE METHOD OF SOLVING THE MARKOV CHAIN MODELS ......................................................... 55  
CHAPTER 6: FLUID APPROXIMATIONS TO THE TIME IN TANDEM QUEUES ................................................................. 62  
CHAPTER 7: A SIMULATION EXPERIMENT ON ADAPTIVE ROUTING IN A NETWORK OF FINITE QUEUES WITH CROSS-TRAFFIC ......................................................... 78  
CHAPTER 8: CONCLUDING MATERIAL ...................................................... 89  
REFERENCES ................................................................................................. 92  
ANNEX A: DERIVATION OF SECOND MOMENT OF TIME TO ABSORPTION ................................................................. 95
LIST OF FIGURES

1. Communication Network Description ........................................... 2
2. Example of Two Parallel Servers .................................................. 6
3. Average Message Delay in Two Parallel Server System
   \( \mu_1 = 0.05, \mu_2 = 0.012 \) .................................................. 8
4. A Path Modelled as Tandem Queues .............................................. 11
5. a. Mean Time in System of Two Tandem Queues
   \( \mu_1 = 0.035, \mu_2 = 0.08 \) ............................................. 15a
   b. Standard Deviation for Time in System of Two
      Tandem Queues \( \mu_1 = 0.035, \mu_2 = 0.08 \) ....................... 15b
6. Network with Two Paths and Three Nodes ..................................... 16
7. Average Message Delay in Three Node System
   \( \mu_1 = 0.05, \mu_2 = 0.035, \mu_3 = 0.08 \) ................................ 18
8. Comparison of Queue Length and Expected Time Decision
   Rules for Three Node System \( \mu_1 = 0.05, \mu_2 = 0.035, \mu_3 = 0.08 \) 20
9. Average Message Delay for Three Node System
   \( \mu_1 = 0.035, \mu_2 = 0.05, \mu_3 = 0.08 \) ................................ 22
10. Network with Two Paths, Two Nodes Each .................................... 24
11. Average Message Delay for Four Node System
    \( \mu_1 = 0.05, \mu_2 = 0.05, \mu_3 = 0.08, \mu_4 = 0.02 \) ................. 25
12. A Path with Cross-Traffic ..................................................... 28
13. Markovian Model of Three Tandem Queues With
    Cross-Traffic ........................................................................ 30
14. Three Finite Tandem Queues with Full-Server Blocking ................. 34
15. Mean Time in System of Three Finite Queues
    with Full-Server Blocking ................................................... 38
16. a. Histogram of Time in System of Three Finite
    Queues .............................................................................. 40
    b. Error in Estimation of Time in System of
    Three Finite Queues .................................................................. 41
17. Three Finite Tandem Queues with Repeated Service .... 45
18. a. Three Finite Tandem Queues with Retransmission: The Queueing Model ......................... 47
20. Markov State-Transition Diagram for Three Finite Tandem Queues with Empty-Server Blocking \( (K_2=5, K_3=3) \). 54
21. Example of Two Finite Tandem Queues with Cross-Traffic ........................................... 58
22. Example of Five Tandem Queues ................................. 63
23. Backlog at Nodes as a Function of Time: Five Tandem Queue Example ............................... 65
24. Departure Processes for Five Tandem Queue Example ... 67
25. Example of Four Tandem Queues with Cross-Traffic .... 69
26. Backlog Functions for Four Tandem Queue Example ..... 70
27. Fluid Flow Model with Added Service Times ............... 76
28. Network Consisting of Two Paths and Five Nodes ....... 79
29. Average Message Delay in Five Node System ............. 85
## LIST OF TABLES

| I   | Simulation Results for Two Parallel Servers ($\mu_1 = 0.05$, $\mu_2 = 0.012$) | ....................................... | 9 |
| II  | Results of Simulation of Three Queue System ($\mu_1 = 0.05$, $\mu_2 = 0.035$, $\mu_3 = 0.08$) | ....................................... | 19 |
| III | Results of Simulation Comparing Blocking and Retransmission | ....................................... | 50 |
| IV  | Special Notation Used in Absorbing Subspace Method | ....................................... | 55a |
| V   | Special Notation Used in Fluid Flow Model | ....................................... | 62a |
| VI  | The Accuracy of Fluid Approximations | ....................................... | 74 |
| VII | Average Message Delay Found in Simulation of Adaptive Routing Systems | ....................................... | 84 |
| VIII| Comparison of Parameter Estimates and Observed Values for Adaptive Routing Methods | ....................................... | 86 |
LIST OF NOTATION

The following notation is used throughout the thesis. However, additional symbols were used in Chapters 5 and 6 and separate lists are provided. When the same symbol has two possible definitions the context should clarify which is intended.

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>DEFINITION</th>
<th>TYPICAL USEAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_i )</td>
<td>Channel Capacity at node ( i )</td>
<td>6</td>
</tr>
<tr>
<td>( D_j )</td>
<td>Destination node for messages of class ( (S_i,D_j) )</td>
<td>27</td>
</tr>
<tr>
<td>( E(T/n_1,n_2) )</td>
<td>Expected time to travel through a system of two tandem queues when the queues lengths seen on arrival are ( n_1 ) and ( n_2 )</td>
<td>25</td>
</tr>
<tr>
<td>( E(T_i) )</td>
<td>Expected time to travel on path ( i )</td>
<td>18</td>
</tr>
<tr>
<td>( F(\bar{h}) )</td>
<td>Fluid flow approximation of the time in system of tandem queues when the initial state is ( \bar{h} )</td>
<td>82</td>
</tr>
<tr>
<td>( i )</td>
<td>Arbitrary node in system of ( k ) tandem queues (( j ) also used)</td>
<td>29</td>
</tr>
<tr>
<td>( j )</td>
<td>Arbitrary enumerated state in the Markov Chain (( i,k ) also used)</td>
<td>13</td>
</tr>
<tr>
<td>( k )</td>
<td>Number of queues in tandem</td>
<td>11</td>
</tr>
<tr>
<td>( K_i )</td>
<td>Maximum number of tokens that can be held at node ( i ) including the one service</td>
<td>33</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>System arrival rate</td>
<td>7</td>
</tr>
<tr>
<td>( \lambda_i )</td>
<td>Arrival rate to node ( i )</td>
<td>7</td>
</tr>
<tr>
<td>( \lambda'_{ij} )</td>
<td>Arrival rate for messages in class ( (S_i,D_j) )</td>
<td>29</td>
</tr>
<tr>
<td>SYMBOL</td>
<td>DEFINITION</td>
<td>TYPICAL USAGE</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>m&lt;sub&gt;j&lt;/sub&gt;</td>
<td>Mean time to absorption for Markov Chain with initial state j</td>
<td>12</td>
</tr>
<tr>
<td>m̄</td>
<td>Vector whose components are the mean time to absorption for all transient states</td>
<td>12</td>
</tr>
<tr>
<td>m&lt;sup&gt;(2)&lt;/sup&gt;</td>
<td>Second moment of the time to absorption from state j</td>
<td>14</td>
</tr>
<tr>
<td>l/μ</td>
<td>Average message length</td>
<td>6</td>
</tr>
<tr>
<td>N&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Service rate at node i (μ&lt;sub&gt;i&lt;/sub&gt; = μC&lt;sub&gt;i&lt;/sub&gt;)</td>
<td>7</td>
</tr>
<tr>
<td>n</td>
<td>State of system of tandem queues defined as a vector (n&lt;sub&gt;1&lt;/sub&gt;, ..., n&lt;sub&gt;k&lt;/sub&gt;) where n&lt;sub&gt;i&lt;/sub&gt; is the number of customers at node i including the one in service</td>
<td>11</td>
</tr>
<tr>
<td>Q&lt;sub&gt;0&lt;/sub&gt;</td>
<td>Initial state of system of tandem queues</td>
<td>11</td>
</tr>
<tr>
<td>N</td>
<td>Total number of states in Markov Chain</td>
<td>35</td>
</tr>
<tr>
<td>P&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Probability of token departing the system of tandem queues after completion of service at node i</td>
<td>29</td>
</tr>
<tr>
<td>P&lt;sub&gt;jk&lt;/sub&gt;</td>
<td>Probability of transition in Markov Chain from state j to state k</td>
<td>12</td>
</tr>
<tr>
<td>P</td>
<td>Transition probability matrix for all states in Markov Chain</td>
<td>13</td>
</tr>
<tr>
<td>Q</td>
<td>Transition probability matrix for set of transient states in Markov Chain</td>
<td>12</td>
</tr>
<tr>
<td>R&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Probability of being routed to path i in random routing schemes</td>
<td>6</td>
</tr>
<tr>
<td>R</td>
<td>Probability matrix for transitions from set of transient states to set of absorbing states</td>
<td>13</td>
</tr>
<tr>
<td>S&lt;sub&gt;i&lt;/sub&gt;</td>
<td>ith estimate of the mean time to absorption vector in Gauss-Seidel method</td>
<td>37</td>
</tr>
<tr>
<td>SYMBOL</td>
<td>DEFINITION</td>
<td>TYPICAL USAGE</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>$S_i$</td>
<td>Source of messages in class $(S_i, D_j)$</td>
<td>27</td>
</tr>
<tr>
<td>$(S_i, D_j)$</td>
<td>Message class, denoting all messages originating at source $S_i$ and headed to destination $D_j$</td>
<td>27</td>
</tr>
<tr>
<td>$t_j$</td>
<td>Mean time to transition out of state $j$</td>
<td>12</td>
</tr>
<tr>
<td>$\bar{t}$</td>
<td>Vector of mean times to next transition for all transient states</td>
<td>12</td>
</tr>
<tr>
<td>$t_j^{(2)}$</td>
<td>Second moment of the time to next transition from state $j$</td>
<td>14</td>
</tr>
<tr>
<td>$T$</td>
<td>Average message delay in system</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Set of Transient States</td>
<td>14</td>
</tr>
<tr>
<td>$T_i$</td>
<td>Steady-state time for token to pass through path $i$</td>
<td>7</td>
</tr>
<tr>
<td>$T(\lambda_1)$</td>
<td>Function representing the steady-state average message delay in a system of two paths where routing is random and the proportions being routed to paths 1 and 2 respectively are $(\lambda_1/\lambda)$ and $1 - (\lambda_1/\lambda)$</td>
<td>15</td>
</tr>
<tr>
<td>$T'(\lambda_1)$</td>
<td>First derivative of $T(\lambda_1)$ with respect to $\lambda_1$</td>
<td>15</td>
</tr>
<tr>
<td>$v_j$</td>
<td>Variance associated with the mean time to absorption from state $j$</td>
<td>14</td>
</tr>
<tr>
<td>$\bar{r}_i$</td>
<td>Average service time at node $i$</td>
<td>7</td>
</tr>
<tr>
<td>$\emptyset$</td>
<td>The absorbing state in the Markov Chain, $\emptyset = (0, \ldots, 0)$</td>
<td>12</td>
</tr>
</tbody>
</table>
ADAPTIVE ROUTING USING THE EXPECTED TIME IN TANDEM QUEUES

CHAPTER 1
INTRODUCTION AND MOTIVATION

Anyone who has experienced a traffic jam can understand that the most direct route from one place to another may not be the fastest. Sometimes it may be faster to take a slightly longer alternate route to avoid the delays caused by traffic congestion. If one could predict the expected transit time for each route based on available information on speed limits, congestion and traffic arrivals, the best route would be the one with the smallest expected time. How does one estimate the expected time for each route? Unfortunately, the state of modelling automobile traffic flow has not been developed sufficiently to solve this problem (see Ref. 1, p. 225). However in the following thesis, this rationale will be applied to communication networks where good models are available.

Communication networks can be modelled using networks of queues (Refs. 2, 3, 4). Figure 1(a) shows a communication network represented as a graph. The nodes in this graph represent communication centers which receive, store and transmit messages (Fig. 1(b)). Each node is geographically separate but is connected to other nodes via communication channels represented as edges on the graph. These channels may vary in speed of transmission (capacity) as shown on

-1-
a) COMMUNICATIONS NETWORK

\[ c_{ij} \text{ = two-way channel capacity} \]

b) INTERFACE MESSAGE PROCESSOR NODE

\[ C = \text{channel servers} \]
\[ B = \text{switch to block traffic from Host} \]

FIGURE 1: COMMUNICATION NETWORK DESCRIPTION
Figure 1(a). Messages can be thought of as tokens generated at a source and travelling through the network to a destination. At each node, storage space is available in case a message arrives when the desired channel is busy and must wait to be transmitted to the next node on its path. Assuming that time at the node CPU is negligible, we have a network in which each channel is a server, each communication center is a switch and a group of queues, and messages travel through the system requesting service from the channels on their path.

Although in the most general case this model is mathematically intractable (see Ref. 2) with the following assumptions the accuracy of the model is retained and the mathematics is greatly simplified.

a. Messages at each source are generated by a Poisson process;

b. Message length (and thus the service time on each channel) is distributed according to an exponential distribution;

c. Each node has infinite storage capacity;

d. Propagation delay between nodes is negligible; and

e. The nodes and channels are perfectly reliable.

An extension to b. has been referred to as Kleinrock's Independence Assumption which states that:

"Each time a message is received at a node within the network, a new (independent) length is chosen.
for this message from an exponential probability density function" (Ref. 2, p. 50).
With this assumption, all of the service times in the model become independent and well-known results from queueing theory can be applied (as in Ref. 5).

This model has been used extensively in the design of communication networks. It has provided insight into the relationship between the design parameters and system performance. The performance measures include average message delay, cost, throughput and reliability. The design considerations include:

a. The assignment of capacity to the various channels;

b. The method of avoiding network overload through limiting the number of messages in the system (flow control); and

c. The method of choosing a route through the network from a source to a destination.

Although all of these problems must be considered in an optimal design, we will concentrate our effort into the area of routing procedures and in particular, routing procedures which can adapt to changes in the state of the network. Using a method similar to Rubin's decomposition approach (Ref. 3), a portion of this model will be extracted to study the usefulness of a routing scheme which is based on the expected time in tandem queues.

It is worthwhile to begin by examining a simple example
such as shown in Figure 2. In this case, we have two parallel servers with exponential service rates $\mu_1$ and $\mu_2$, $\mu_1 > \mu_2$. Tokens arrive at this system with Poisson rate $\lambda$. On arrival they pass through a switch that directs them to either server 1 or server 2. Various switching schemes could be envisioned:

a. Send all tokens to the fastest server (fixed);
b. Send tokens to the servers randomly with a pre-specified probability $r_1$ of being sent to node 1 and probability $r_2 = (1 - r_1)$ of being sent to node 2 (random); and
c. Send tokens to the servers based on knowledge of the current queue lengths and/or service rates (adaptive).

Two random routing methods were devised. In the first, the probability of routing to node 1 is proportional to the service rate at node 1. That is, $r_1 = \mu_1 / (\mu_1 + \mu_2)$. In the second, the proportion is based on a calculation which attempts to minimize the steady-state time in system given knowledge of the arrival rate $\lambda$. Let $\lambda_1 = r_1 \lambda$ and $\lambda_2 = (1 - r_1) \lambda$. Then the steady-state expected time in node 1 is $T_1 = 1 / (\mu_1 - \lambda_1)$ and the steady-state expected time in node 2 is $T_2 = 1 / (\mu_2 - \lambda_2)$. The overall mean time in system is

$$T = (\lambda_1 / \lambda)T_1 + (\lambda_2 / \lambda)T_2.$$
\[ \mu_1 = \mu C_1, \quad \mu_2 = \mu C_2 \]

where \( 1/\mu \) is the mean message length and \( C_1 \) is the channel capacity (we have used \( \mu_1 = 0.05 \) and \( \mu_2 = 0.012 \) in our computations).

The message length is often measured in bits, the channel capacity in bits per second and the arrival rate in messages per second. In this case, the average message delay would be in seconds. Although we will not refer to specific units in this thesis, these respective measures are assumed to be consistent.
Since $\lambda_2 = (\lambda - \lambda_1)$, we can solve
\[
\min T(\lambda_1) \text{ for } 0 \leq \lambda_1 \leq \lambda
\]
to find the proportions which minimize the expected steady-state time in system.

Two adaptive routing methods were also considered. The first will route to node 1 if $n_1 < n_2$ and to node 2 if $n_1 > n_2$ where $n_1$ and $n_2$ are the respective queue lengths (including the token in service). The second is based on the expected time in system. It will route the token to node 1 if $n_1 \bar{x}_1 < n_2 \bar{x}_2$ and to node 2 if $n_1 \bar{x}_1 > n_2 \bar{x}_2$, where $\bar{x}_1$ and $\bar{x}_2$ are the respective mean service times.

Along with the five routing methods discussed above, we have included an "ideal observer" scheme as suggested by Fultz and Kleinrock (Ref. 6). The ideal observer chooses the route based on complete knowledge of the system including the service times for the tokens in the system (see Ref. 19 for algorithm). Although this is not a practical routing scheme, it does provide a theoretical minimum with which to compare the other methods.

These schemes were studied using analytic and simulation methods for the case when $\mu_1 = 0.05$ and $\mu_2 = 0.012$. Figure 3 summarizes the change in average message delay as the arrival rate increases.

The average mean delay for the fixed routing method was obtained from the steady-state M/M/1 formula with $\mu = 0.05$. For both of the random routing methods, the values of $\lambda_1$ and
FIGURE 3: AVERAGE MESSAGE DELAY IN TWO PARALLEL SERVER SYSTEM

\(\mu_1 = 0.05, \mu_2 = 0.012\)
\( T_2 \) were obtained as described above. Then they were used in
the steady-state formulae with \( \mu_1 = 0.05 \) and \( \mu_2 = 0.012 \) to
obtain \( T_1 \) and \( T_2 \). The average message delay was computed
using the formula

\[
T = \left( \frac{A_1}{A} \right) T_1 + \left( \frac{A_2}{A} \right) T_2
\]

For the adaptive routing methods and the ideal observer,
simulation was used to obtain estimates of the average
message delay. Five runs were carried out, for each value of
\( A \) (0.01, 0.02, 0.03, 0.04, 0.05). In each run, 12,000
messages were generated and routed through the system. The
average time in system (message delay) was obtained and taken
to be an independent experimental observation. The mean of
these five observations is shown in Figure 3. Using the
method suggested by Law and Kelton (Ref. 10, p. 288), an
approximate 95% confidence interval was obtained for each
point. These confidence intervals did not overlap (see
Table I). This procedure will be used throughout this thesis
when testing the usefulness of the adaptive routing methods.

It can be seen that routing based on the shortest
expected time is superior to the other schemes and is in the
same "ball-park" as the ideal observer method. However, for
low utilization the fixed routing method is good and for high
utilization the smallest queue length method provides good
results.

Although this simple example gives us hope that our
approach might be useful, it is not very realistic. To
TABLE I

SIMULATION RESULTS FOR TWO PARALLEL SERVERS

\( \mu_1 = 0.05, \mu_2 = 0.012 \)

<table>
<thead>
<tr>
<th>ARRIVAL RATE</th>
<th>QUEUE LENGTH</th>
<th>ROUTING METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1)</td>
<td>(2)</td>
</tr>
<tr>
<td>0.01</td>
<td>30.1 ± 0.687</td>
<td>25.3 ± 0.777</td>
</tr>
<tr>
<td>0.02</td>
<td>38.2 ± 0.232</td>
<td>32.4 ± 0.350</td>
</tr>
<tr>
<td>0.03</td>
<td>48.3 ± 2.053</td>
<td>42.6 ± 1.930</td>
</tr>
<tr>
<td>0.04</td>
<td>70.0 ± 1.988</td>
<td>62.2 ± 2.543</td>
</tr>
<tr>
<td>0.05</td>
<td>108.8 ± 1.874</td>
<td>100.8 ± 1.766</td>
</tr>
</tbody>
</table>

(1) AVERAGE MESSAGE DELAY
(2) 95% CONFIDENCE INTERVAL
analyze more realistic systems, we need to examine the problem of estimating the time in tandem queues.
CHAPTER 2
THE EXPECTED TIME IN TANDEM QUEUES

Messages travelling between a source and destination can be thought of as travelling through a series of tandem queues (Ref. 3). If we assume, as Casteau and Pujolle have done in Reference 7, that cross-traffic does not disturb the traffic on the main path too much, then we have a path modelled as a series of simple tandem queues with infinite queueing space (Fig. 4). We wish to compute the expected time in the system given knowledge of the queue lengths on arrival and the service rates at each of the nodes.

\[ \mu_1 \quad \mu_2 \quad \mu_3 \quad \cdots \quad \mu_k \]

**Figure 4: A Path Modelled as Tandem Queues**

Stanford (Ref. 8) considered this problem and suggested a Markov Chain approach in which the state space is defined as \( \bar{n} = (n_1, \ldots, n_k) \), where \( n_i \) is the number of tokens in queue \( i \) (including the one in service). Assuming independent exponential servers, a transition is defined as any service completion and is thus dependent only on the current state of the system. If we wish to estimate the time in system for a particular (marked) token who sees state \( \bar{n}_0 \) (including
himself) on arrival, then we can assume that subsequent arrivals do not occur since they cannot effect the time in system for the marked token. The expected time the marked token will spend in the system is equal to the expected time the Markov Chain will take to reach the absorbing state \( \bar{0} = (0, \ldots, 0) \) given that it started in state \( \bar{n}_0 \).

Stanford suggests a method of ordering the states to ensure that the transition probability matrix is upper triangular (Ref. 8, p. 6-14). Using this ordering, we can enumerate the states from 0 to \( j \) where 0 denotes state vector \( \bar{0} \) and \( j \) denotes state vector \( \bar{n}_0 \).

If we define \( m_j \) as the mean time to absorption for the Markov Chain with initial transient state \( j \), then

\[
m_j = t_j + \sum_{k \in T_j} P_{jk} m_k
\]  

(1)

where

- \( t_j \) is the mean time to transition out of state \( j \),
- \( P_{jk} \) is the probability of transition from state \( j \) to transient state \( k \), and
- \( T_j \) is the set of all transient states that can be reached from state \( j \).

Then we can express equation (1) above in matrix notation

\[
\bar{m} = \bar{c} + Q \cdot \bar{m}
\]
where
\[
\bar{m} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_j \end{bmatrix}, \quad \bar{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_j \end{bmatrix}
\]
and
\[
Q = \begin{bmatrix} P_{11} & \cdots & P_{1j} \\ P_{21} & \cdots & P_{2j} \\ \vdots & \vdots & \vdots \\ P_{j1} & \cdots & P_{jj} \end{bmatrix}
\]

Thus
\[(I - Q)\bar{m} = \bar{t}.
\]

This coincides with Parzen's notation in which the full transition matrix is written
\[
P = \begin{bmatrix} A & B \\ A & 0 \end{bmatrix}
\]

where
\[A \text{ is the set of absorbing states and}
B \text{ is the set of transient states (see Ref 9, p. 246).}
\]

Since we are assuming exponentially distributed service times, the mean time to next transition from state \(j\) is just
\[t_j = 1.0/(\text{sum of the transition rates out of state } j)
\]

and the probability of going to state \(k\) from state \(j\) is
\[
P_{jk} = \frac{\text{transition rate from state } j \text{ to state } k}{\text{sum of the transition rates out of state } j}
\]

For any particular state \(\bar{i} = (i_1, \ldots, i_k)\), if \(i_m > 0, m < k\), then there is a possible transition from \(\bar{i}\) to \((\bar{i} - \bar{e}_m + \bar{e}_{m+1})\) with rate \(\nu_m \) where \(\bar{e}_m = (0, \ldots, 0, 1, 0, \ldots, 0)\).
the \( m \)th unit vector. If \( i_k > 0 \) then there is a possible transition to \((\bar{I} - \bar{e}_k)\) with rate \( \mu_k \).

Although equation (1) represents a system of linear equations, Stanford noted that with an appropriate ordering of the states the matrix \((I - Q)\) is triangular and thus can be solved by back-substitution (Ref. 6, p. 6-14).

The variance in the time to absorption can also be estimated. Defining \( m_j^{(2)} \) as the second moment of the time to absorption and \( t_j^{(2)} \) as the second moment of the time to transition, it can be shown that

\[
m_j^{(2)} = t_j^{(2)} + 2t_j \sum_{k \in T} p_{jk} m_k + \sum_{k \in T} p_{jk} m_k^{(2)}
\]

(See Annex A). Now \( t_j^{(2)} = 2(t_j)^2 \) since the time to next transition is exponentially distributed. Then

\[
t_j^{(2)} + 2t_j \sum_{k \in T} p_{jk} m_k = 2t_j (t_j + \sum_{k \in T} p_{jk} m_k) = 2t_j m_j
\]

and

\[
m_j^{(2)} = 2t_j m_j + \sum_{k \in T} p_{jk} m_k^{(2)}
\]

which is again a system of linear equations and can be solved by back-substitution. The variance associated with the mean time to absorption is

\[
v_j = m_j^{(2)} - (m_j)^2.
\]
This variance is an important consideration if we wish to predict the time in system based on our calculation of mean time to absorption. Figure 5a shows a sample curve of the mean time in system for two tandem queues. For clarity, the expected time in system $E(T/n_1, n_2)$ is plotted against $(n_1 + n_2)$. $E(T/n_1, n_2) = m_j$ where $j$ denotes the state $(n_1, n_2)$ in the Markov Chain. Although only the points are valid, curves have been drawn to show the value of $E(T/n_1, n_2)$ when $n_1$ is fixed and $n_2$ is allowed to vary. The standard deviation is plotted in a similar manner in Figure 5b.

To gain further insight into the usefulness of the expected time estimation in adaptive routing, we will examine another simple network (Fig. 6). In this case, we will evaluate the average message delay for the following routing methods:

a. Fixed routing through node 1;
b. Random routing with proportions based on the minimum steady-state time through system;
c. Adaptive routing based on shortest queue lengths where $n_1$ is compared to $(n_2 + n_3)$ to determine the route; and
d. Adaptive routing based on the shortest expected time.

The minimum random proportions are obtained as before by minimizing the function

$$T_1 = (\lambda_1/\lambda)T_1 + [(\lambda_1-\lambda)/\lambda]T_2$$

-15-
FIGURE 5a: MEAN TIME IN SYSTEM OF TWO TANDEM QUEUES

$$(\mu_1 = 0.035, \mu_2 = 0.08)$$
FIGURE 5b: STANDARD DEVIATION ASSOCIATED WITH MEAN TIME IN TWO TANDEM QUEUES ($\mu_1 = 0.035, \mu_2 = 0.08$)
FIGURE 6: NETWORK WITH TWO PATHS AND THREE NODES
However in this case,

$$T_2 = \frac{1.0}{\mu_2 - \lambda + \lambda_1} + \frac{1.0}{\mu_3 - \lambda + \lambda_1}$$

In this case, $T'(\lambda_1) = 0$ must be solved using numerical methods.

The analytic results obtained for the fixed and random routing methods are shown in Figure 7 along with the simulation results obtained for the adaptive routing methods. In this case, we have used $\mu_1 = 0.05$, $\mu_2 = 0.035$ and $\mu_3 = 0.08$. It can be seen that in this example both of the adaptive routing methods performed better than the fixed or random routing methods for all values of $\lambda$. Notice that the queue length routing method performed better than the expected time routing method! Although the difference was small, it was significant at the 95% level (see Table II).

This last observation can be partially explained by looking at the decision rules more closely. Using Figure 8, we can identify the situations in which the queue length method and the expected time method will give the same decision and when they will differ. Consider the situation where $n_1 = 5$ (4 currently in the system plus our marked token) then the expected time on route 1 is $E(T_1/n_1 = 5) = 5\bar{x}_1 = 5(20) = 100$. Now we can divide the graph into four quadrants using the lines $(n_2 + n_3) = n_1 = 5$ and $E(T_2) = E(T_1) = 100$. For the points in quadrants (1) and (3) both routing methods will agree. In quadrant (1), $E(T_1) < E(T_2)$ and $n_1 < (n_2 + n_3)$ so both methods will choose
FIGURE 7: AVERAGE MESSAGE DELAY IN THREE NODE SYSTEM

\( \mu_1 = 0.05, \mu_2 = 0.035, \mu_3 = 0.08 \)
TABLE II

RESULTS OF SIMULATION OF THREE QUEUE SYSTEM

\( \mu_1 = 0.05, \mu_2 = 0.035, \mu_3 = 0.08 \)

<table>
<thead>
<tr>
<th>ARRIVAL RATE</th>
<th>OBSER. PER RUN</th>
<th>QUEUE LENGTH</th>
<th>ROUTING METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(1)</td>
<td>(2)</td>
</tr>
<tr>
<td>0.01</td>
<td>20,000</td>
<td>23.60</td>
<td>± 0.2530</td>
</tr>
<tr>
<td>0.02</td>
<td>40,000</td>
<td>27.03</td>
<td>± 0.3023</td>
</tr>
<tr>
<td>0.03</td>
<td>60,000</td>
<td>30.99</td>
<td>± 0.2601</td>
</tr>
<tr>
<td>0.04</td>
<td>80,000</td>
<td>36.18</td>
<td>± 0.1500</td>
</tr>
<tr>
<td>0.05</td>
<td>100,000</td>
<td>44.22</td>
<td>± 0.2846</td>
</tr>
<tr>
<td>0.06</td>
<td>120,000</td>
<td>56.85</td>
<td>± 0.9885</td>
</tr>
</tbody>
</table>

(1) AVERAGE MESSAGE DELAY
(2) 95% CONFIDENCE INTERVAL
FIGURE 8: COMPARISON OF QUEUE LENGTH AND EXPECTED TIME

DECISION RULES FOR THREE NODE SYSTEM
($\mu_1 = 0.05, \mu_2 = 0.035, \mu_3 = 0.08$)
route 1. In quadrant (3), $E(T_1) > E(T_2)$ and $q_4 > (q_2 + q_3)$ so both methods will choose route 2. However for the points in quadrants (2) and (4), the methods will differ. In quadrant (2) $E(T_1) < E(T_2)$, thus the expected time method will choose route 1 but $n_1 > (n_2 + n_3)$ so the queue length method will choose route 2. In quadrant (4), the reverse will be the case; $E(T_1) > E(T_2)$ causing the expected time method to choose route 2 but $n_1 < (n_2 + n_3)$ so the queue length method will choose route 1. Fortunately for the queue length routing method, in this example there are relatively few points in quadrants (2) and (4) and they occur with small probability when compared to the points in quadrants (1) and (3). This suggests that the methods should produce similar results as was found in running the simulation but does not tell us why the queue length method produced better results.

Scenarios could be devised in which the queue length method does not perform better than the expected time method. For example, if in the system from Figure 6, $u_1 < \min (u_2, u_3)$ then the decision rules would differ substantially. If a figure such as Figure 8 were examined for this case one would see that there would be a significant number of points in quadrant (4). Figure 9 shows simulation results for the specific case when $u_1 = 0.035, u_2 = 0.05, u_3 = 0.08$. It can be seen that the expected time method performed slightly better than the queue length method.
FIGURE 9: AVERAGE MESSAGE DELAY FOR THREE NODE SYSTEM

\( \mu_1 = 0.035, \mu_2 = 0.05, \mu_3 = 0.08 \)
When the four node system in Figure 10 was analyzed, similar results were obtained. Here the expected time method performed slightly better than the queue length method as shown in Figure 11. However both methods produced good results.

One comment should be made concerning another heuristic routing method. It is well-known that the expected time in tandem queues is dominated by the time spent at the "bottleneck node" (see Ref. 11). If queues form in the system, they will form first at the bottleneck. If we find that node $b$ is the bottleneck in a system of $k$ tandem queues, then the tokens at nodes $(b+1)$ to $k$ will have little effect on the service time of a marked token arriving at node 1. However, the tokens seen at nodes 1 to $b$ will have a large effect on the time in system of our marked token. Possibly some weighting of the individual queue lengths could produce an efficient and general decision rule. This will be discussed further in Chapter 6.

This model of a path in a network as a series of tandem queues without cross-traffic is not very realistic. However, we have been able to predict the time to travel through this path based on the queue lengths seen on arrival. We have also gained insight into the usefulness of such a technique in an adaptive routing scheme by considering some simple networks. Ideas concerning heuristic methods have also been revealed. Rather than considering more complex examples
FIGURE 10: NETWORK WITH TWO PATHS, TWO NODES EACH
ROUTING METHODS

- Fixed: Routed to Path 1-2
- Random: Optimal Proportions
- Queue Length
- Expected Time

FIGURE 11: AVERAGE MESSAGE DELAY FOR FOUR NODE SYSTEM

\[ \mu_1 = 0.05, \mu_2 = 0.05, \mu_3 = 0.08, \mu_4 = 0.02 \]
involving tandem queues, we will instead apply this Markov Chain methodology to an extended model which will incorporate a specific type of cross-traffic.
CHAPTER 3

A MODEL OF TANDEM QUEUES WITH CROSS-TRAFFIC,
FINITE STORAGE AND FULL-SERVER BLOCKING

In the previous chapter, we assumed that cross-traffic did not disturb the traffic on the main path too much. We were then able to develop a Markov Chain model to estimate the time in a system of tandem queues. However, if a particular path between a source and destination is extracted from a network, there may be cross-traffic at each node caused by messages travelling between other source-destination pairs (see Fig. 12). Our goal again is to estimate the time to travel from source to destination given knowledge of the queue lengths seen on arrival. However in this model, we will need not only the node service rates, but also the arrival and departure rates of the cross-traffic.

We will begin by considering the three node system shown in Figure 12. Recall that our model is primarily concerned with queueing for communication channels. Cross-traffic will be defined as messages arriving from an external source and passing through one or more of the channels on our path. Message classes can be defined according to their source and destination. We will ignore messages destined for D₀ as well as the message classes (S₂, D₁) and (S₃, D₂) because they will not pass through one of the channels on our path. However, we will assume that the other messages arrive
FIGURE 12: A PATH WITH CROSS-TRAFFIC
according to a Poisson process with rate \( r_{ij} \) for \((S_i, D_j)\) where \(0 \leq i \leq 3\) and \(1 \leq j \leq 3\) and \(i \leq j\). We will also assume that the message lengths are exponentially distributed with mean \(1/\mu\). Again Kleinrock's Independence Assumption will be invoked. Thus the nodes can be modelled as exponential servers with rates \( \mu_i = \mu C_i \) where \( C_i \) is the channel capacity at node \(i\).

To convert this model to a Markov Chain, we will use our knowledge of the arrival rates and message destinations to obtain the queueing system shown in Figure 13. Here the Poisson arrival rates were obtained as follows:

\[
\lambda_1 = \lambda_{01} + \lambda_{02} + \lambda_{03} + \lambda_{11} + \lambda_{12} + \lambda_{13} \\
\lambda_2 = \lambda_{22} + \lambda_{23} \\
\lambda_3 = \lambda_{33}
\]

The departure probabilities are:

\[
P_1 = (\lambda_{01} + \lambda_{11})/\lambda_1 \\
P_2 = (\lambda_{02} + \lambda_{12} + \lambda_{22})/(\lambda_2 + (1 - P_1) \lambda_0)
\]

It should be noted that we are assuming that all of the arrivals from source \(i\) will enter node \(i\). However, departures from node \(i\) will make a random decision to depart the system (with probability \(P_1\)) or continue to the next node (with probability \((1 - P_1)\)).

Now applying the same approach as in Chapter 2, consider a "marked" token arriving at node \(1\) and seeing \(\tilde{n} = (n_1, n_2, n_3)\) tokens in the queues ahead (\(n_1\) includes himself). He cannot distinguish the token classes. Therefore, he does not know
FIGURE 13: MARKOVIAN MODEL OF THREE TANDEM QUEUES WITH CROSS-TRAFFIC
whether these tokens will depart the system or continue to the next node. However using the values $p_1$ and $p_2$, he can estimate the proportions that will depart and will continue at each node. We will assume that our marked token will travel through the entire system and arrive at his destination $D_3$. It is quite possible that tokens arriving from an external source could enter the queue of a node ahead before our marked token does. This possibility can be estimated using the arrival rates $\lambda_2$ and $\lambda_3$. Since tokens arriving at node 1 after our marked token can not affect his time in system, they can be ignored. Thus source 0 can be considered "turned off" once our marked token arrives at node 1. Similarly source 2 can be turned off when our marked token arrives at node 2; as can source 3 on arrival to node 3. Our state space is all states $\bar{n} = (n_1, n_2, n_3)$. Transitions occur at arrival instants and end-of-service instants. Thus the transitions in our model are dependent only on the present state and our model is Markovian.

To estimate the time to travel through the system given that our marked token saw $\bar{n}_0$ tokens on arrival, we must compute the mean time to reach the absorbing state $\bar{0} = (0,0,0)$ from the transient state $\bar{n}_0$ as before. In general, we must solve the system of linear equations

$$m_j = t_j + \sum_{k \neq j} p_{jk} m_k$$
for all states \( j \) that can be reached from \( \bar{r}_0 \). But since the arrival rates at the nodes ahead are unrestricted and the queues are assumed to have infinite storage capacity, there are infinitely many reachable states and thus infinitely many linear equations! The computations become intractable. A practical way of resolving this problem would be to re-examine our assumption that the queueing capacity is infinite. Indeed, this is not the case. Although the queueing capacity may be large, it must be finite and our problem is tractable if we assume finite queueing space at each node.

By introducing finite queues we have solved one problem but created two others; namely, what actions are required when the queues reach their limit and how can they be modelled? This has been the subject of much research by communication network designers and model builders (Refs. 7, 12, 13, 14). Three common practices are:

a. **Loss:** in which a message arriving to find a full queue is lost never to return;

b. **Retransmission:** in which the message is assumed to be lost if it arrives to find a full queue. It must then be retransmitted from the original source or from the last node in which it was stored (see Ref. 14); and

c. **Blocking:** in which a full queue at the node ahead may cause the server to discontinue work until space becomes available (see Ref. 7).
Although significant research in the modelling area has been carried out, most has concentrated on the steady state approximations. Our problem requires an examination of the transient behaviour of these systems.

A brief discussion of loss, retransmission and alternative blocking models is provided in Chapter 4 along with their state-transition diagrams. However, our effort will concentrate on a particular type of blocking which is similar to "full-server" blocking as described by Caseau and Pujolle (Ref. 7) and Newell (Ref. 11).

Let \( K_i \) be the maximum number of messages that can be held at node \( i \). If \( n_i = K_i \) then we will assume that messages arriving from an external source will be lost but messages arriving from node \( (i-1) \) will be blocked. A blocked message is assumed to wait at node \( (i-1) \) until a service completion at node \( i \) at which point space becomes available and it immediately enters the queue at node \( i \).

In Figure 14, we present a portion of the state-transition diagram for the case when the number of tokens at node 1 is fixed and \( K_2 = 4, K_3 = 2 \). For example, we can assume \( n_1 = 1 \) and examine the transitions in the subspace \( (1,n_2,n_3) \) where \( n_2 = 0, \ldots, K_2 + 1 \) and \( n_3 = 0, \ldots, K_3 + 1 \). For \( n_1 = 2 \), the subspace \( (2,n_2,n_3) \) would be identical. Transitions involving a change in \( n_1 \) would connect these two subspaces. However these are not shown. It should be noted that some modifications to this
b. STATE-TRANSITION DIAGRAM ($q_1$ fixed, $K_2$=4, $K_3$=2)

There is no state $(3,5)$ because $n_3 = 3$ implies there is a blocked token in node 2 and only 3 spaces remaining in queue 2. State $(3,4)$ implies blocked tokens at nodes 1 and 2. Thus the total number of states is

$$N = (K_1 + 1)[(K_2 + 2)(K_3 + 2)-1]$$

FIGURE 14: THREE FINITE TANDEM QUEUES WITH FULL-SERVER BLOCKING
state-transition diagram are required if \( n_1 = 0 \). In Reference 19, a computer routine is provided which calculates the values of \( t_j \) and \( p_{jk} \) using the formulae

\[
  t_j = \text{(sum of transition rates out of state } j)^{-1},
\]

\[
  p_{jk} = \frac{\text{(transition rate from state } j \text{ to state } k)}{\text{(sum of transition rates out of state } j)}.\]

Unfortunately, the matrix \( Q \) is not triangular as was the case for tandem queues without cross-traffic. Thus the system of linear equations

\[
(I - Q)^{\vec{m}} = \vec{t}
\]

must be solved explicitly. One method that has proven efficient even for large values of \( K_1, K_2 \), and \( K_3 \) is Gauss-Seidel Iteration (see Ref. 21, p. 169 for the algorithm details). Another method of solving this system will be discussed in Chapter 5.

The second moment of the time to absorption can be obtained as before by solving

\[
m_j^{(2)} = 2 \cdot t_j m_j + \sum_{k} p_{jk} m_k^{(2)}
\]

This too can be obtained using Gauss-Seidel Iteration. Then the variance in the time to absorption is

\[
v_j = m_j^{(2)} - (m_j)^2.
\]

For the system in Figure 13, the state space would be all vectors \( \vec{n} = (n_1, n_2, n_3) \) where \( n_1 = 0, \ldots, K_1; n_2 = 0, \ldots, K_2, (K_2+1); \) and \( n_3 = 0, \ldots, K_3, (K_3+1) \). Here
\[ n_2 = (K_2 + 1) \] implies that the queue at node 2 is full and a message is blocked at node 1. Similarly, \[ n_3 = (K_3 + 1) \] implies the queue at node 3 is full and a message is blocked at node 2. This blocked message can be thought of as filling the \((K_i + 1)^{st}\) position in the queue for node \(i\). There are approximately \((K_1 + 1)(K_2 + 2)(K_3 + 2)\) states which can be enumerated from 0 to \(N\) as follows: 0 denotes the absorbing state \((0, 0, 0)\); 1 denotes \((0, 0, 1)\); and so on until \(K_3 + 1\) denotes \((0, 0, K_3 + 1)\); \(K_3 + 2\) denotes \((0, 1, 0)\); \(K_3 + 3\) denotes \((0, 1, 1)\) and so on.

Now method we calculate the mean time to absorption from state \(j\) as

\[ m_j = t_j + \sum_{k \in T} p_{jk}m_k \]

where

- \(t_j\) is the mean time to next transition from \(j\)
- \(p_{jk}\) is the probability of going to state \(k\) from state \(j\) in one step; and
- \(T\) is the set of transient states \(1, \ldots, N\).

Or in matrix form,

\[ \bar{m} = \bar{E} + Q \cdot \bar{m} \implies (I - Q)\bar{m} = \bar{E} \]

Here \(\bar{m}\) and \(\bar{E}\) are vectors of length \(N\) and \(Q\) is an \(N \times N\) matrix.

As an example consider the three queue system shown in Figure 12 with

\[
\begin{bmatrix}
0.01 & \text{if } i = 0 \text{ and } j = 3 \\
0.001 & \text{otherwise}
\end{bmatrix}
\]

\[
\mu_1 = \mu_2 = \mu_3 = 0.05.
\]
Using the formulae for $l_1$, $l_2$, $l_3$, $p_1$ and $p_2$, this system can be converted to the Markovian queueing model shown in Figure 13 where

$$
l_1 = 0.015, \quad l_2 = 0.002, \quad l_3 = 0.001
$$

$$
p_1 = 0.13333, \quad \text{and} \quad p_2 = 0.2.
$$

Assuming that the nodes have maximum size of 10 we can apply our Markov Chain method to find the mean time to absorption for all possible states. We must solve 1572 simultaneous linear equations. Using the Gauss-Seidel iteration method, this required 15 iterations to obtain a total absolute error of 0.01. Here the total absolute error is defined as

$$
\frac{\sum_{i=1}^{n} \left| \text{abs} \left( t_i - \sum_{j=1}^{n} (I - Q)_{ij} s_j \right) \right|}{\sum_{j=1}^{n} T}
$$

where

$(I - Q)_{ij}$ is the $(i,j)^{th}$ element of the matrix $(I - Q)$,

$s_i$ is the current estimate of the mean time to absorption from state $i$,

$t_i$ is the mean time to next transition from state $i$ (the right hand side of $(I - Q) \cdot \bar{m} = \bar{v}$), and

$T$ is the set of all transient states.

In Figure 15, results of the mean time calculation are presented in which $n_1$ is fixed at 1 and $n_2$ and $n_3$ are allowed to vary. It can be seen that the results have the same basic shape as the previous results for the two tandem model shown in Figure 5a except in this case the curves overlap slightly.
Figure 15: Mean time in system of three finite tandem queues with full-server blocking
In an attempt to verify these results, simulation was used to compare the predicted time in system (based on the expected time calculation from the Markov Model) and the simulated time in system for tokens travelling from source 0 to destination 3 in the three queue system of Figure 13. The values obtained for \( \lambda_1, \lambda_2, \lambda_3, p_1 \) and \( p_2 \) in the example above were also used in the simulation. However, the maximum queue lengths were specified as five for all three queues.

Figure 16a shows a histogram of the time in system observed in one simulation run. Figure 16b shows that although the average error is approximately zero, on any given estimate the actual error could be quite large. The average error in the prediction was quite small compared to the average time in system (-0.07 vs. 91.43).

A second simulation was carried out in which we did not require independent service times. The message length was determined on creation of the message and was the same at each node along the route. In this case, the mean time in system was found to be 101.31 and the average error was 7.97. The half-widths of the 95% confidence intervals obtained for these values were 0.67 and 0.46 respectively. Using Kleinrock’s Independence Assumption in this example, we underestimated the time in system and thus our predictor was in error.

We will save our investigation of the usefulness of this predictor in adaptive routing until Chapter 7. First, we
FIGURE 16a: HISTOGRAM OF TIME IN SYSTEM OF THREE FINITE QUEUES

* These values refer to the center of an interval of length 5.
FIGURE 16b: ERROR IN ESTIMATION OF TIME IN SYSTEM OF THREE FINITE QUEUES

* These values refer to the center of an interval of length 50.
will briefly discuss other approaches to retransmission and blocking (Chapter 4). Chapter 5 will be devoted to an alternative method of solving the system of linear equations in these Markov Chain models. In Chapter 6, we will consider a method of approximating the time in tandem queues using a fluid flow approach.
CHAPTER 4

ALTERNATIVE MODELS INVOLVING RETRANSMISSION AND
EMPTY SERVER BLOCKING

In the previous chapter, finite queues were introduced into our model of a store-and-forward communication network. This required some consideration of the actions that would be taken if the queues reach their capacity. We specifically considered a type of blocking described as full-server blocking. In this chapter, we will consider alternative actions that might be taken if the queues becomes full.

A pure loss system, in which messages are lost and assumed never to return if they arrive to find a full queue, is not realistic for most communication networks. Lost messages do not disappear. In the worst case, they are retransmitted from their source. However, in a store-and-forward communication network, the message is saved at each step until it has been safely transmitted to the next node on its path. If a message is transmitted but not accepted by the receiving node, the sending node will not receive an acknowledgement. After a certain delay period, it will assume the message was lost and will retransmit it. Similarly, messages from some external source will be re-entered after a waiting period if they are not accepted into the system on a first attempt. Thus a model which includes loss and retransmission when queues are full is a
more realistic way to model a store-and-forward communication network.

Unfortunately, modelling a complete retransmission system is quite difficult. Labetoulle and Pujolle (Ref. 13) assumed that the external sources interfaced with the system through infinite queues. Inside the system, they assumed that the queues were finite and messages could be lost. However, to incorporate the concept of retransmission, they modified the internal arrival rates and service rates. That is, they applied recursive formulae to adjust:

a. The node arrival rates to account for retransmission from other nodes; and

b. The node service rates to account for retransmission to other nodes.

With these adjustments, they were able to obtain results for the steady state queue lengths in a loss system and use these as an approximation to the loss-free system.

In our problem, we need to analyze the transient behaviour of a system of tandem queues with cross-traffic. This cross-traffic could be from external sources or from other nodes in the system. For simplicity, we will assume that cross-traffic arriving to find a full queue will be lost never to return. This lack of symmetry in the model of the network is unfortunate. However, similar models of cross-traffic have been presented (see Ref. 7). In our queueing model, retransmission can be modelled as repeated
service at a node. However in our original Markovian model, service at a node involved two actions:

a. An exponentially distributed delay to complete service; and

b. A random decision ("coin toss") to depart the system or enter to the queue at the next node.

A problem arises when considering this "coin tossing" operation.

One approach would be to repeat the complete service if the token was blocked entry to the next node. Figure 17a provides a diagram of the queueing model in this case. Figure 17b presents a portion of the Markov state-transition diagram. On completion of service at node i the token will depart the system with probability $p_i$ or attempt to enter the queue at node $(i + 1)$ with probability $(1 - p_i)$. If in the latter case, the node $(i + 1)$ is full, it will return to the head of the queue at node i, repeat service and repeat the "coin toss" to determine whether to depart the system. The probability of departing the system $p_i$ must be calculated in such a way that it includes this retransmission, otherwise more loss will be predicted in the model than would actually occur. It is not clear how this calculation could be carried out to incorporate the transient nature of the problem. The probability of retransmission is dependent on the queue lengths seen on arrival!
a. QUEUEING SYSTEM INVOLVING REPEATED SERVICE

b. MARKOV STATE-TRANSITION DIAGRAM

\( n_1 \text{ fixed, } K_2=4, K_3=2 \)

FIGURE 17: THREE FINITE TANDEM QUEUES WITH REPEATED SERVICE
An alternative approach would be to define new states to avoid the repeated "coin tossing". For example, on entering node i, a decision could be made to either depart the system or continue to node \((i + 1)\). After this decision is made, service on the token is carried out. If the decision was to go to node \((i + 1)\) and node \((i + 1)\) was found to be full then the token returns to node \(i\) and repeats service. However, this time no "coin toss" is required. This can be modelled as a more complex service mechanism as shown in Figure 18a. Notice only one token is allowed in the server at a time. This new server requires a new set of states (shown as retransmission states in Figure 18b). If none of the queues are full the state-transition diagram is identical to the previous cases. However, if a full queue causes retransmission of a token, the Markov Chain moves into the set of retransmission states. A number of transitions could occur in this subspace before the system returns to a normal state. The complexity of the Markov Chain method would be doubled because the state space has doubled.

Using the example system from Chapter 3, the time in finite tandem queues was compared for a retransmission system and a full-server blocking system. The simulation results are provided in Table III and shown in Figure 19. In these runs, the arrival rate of external messages \((\lambda_{ij}\ i \neq 0, j \neq 3)\) was allowed to vary. For low values of \(\lambda_{ij}\), the queues were full only about 1% of the time. In this case, the two
Figure 18a: Three finite tandem queues with retransmission:

The queueing model
FIGURE 18b: THREE FINITE TANDEM QUEUES WITH RETRANSMISSION:

MARKOV STATE-TRANSITION DIAGRAM

\((K_2 = 4, K_3 = 2)\)
FIGURE 19: SIMULATION OF FULL-SERVER BLOCKING AND RETRANSMISSION IN THREE FINITE QUEUES WITH CROSS-TRAFFIC
TABLE III

RESULTS OF SIMULATION COMPARING BLOCKING AND RETRANSMISSION

<table>
<thead>
<tr>
<th>EXTERNAL ARRIVAL RATE</th>
<th>FULL-SERVER BLOCKING</th>
<th>RETRANSMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1)</td>
<td>(2)</td>
</tr>
<tr>
<td>0.001</td>
<td>91.8</td>
<td>± 1.352</td>
</tr>
<tr>
<td>0.002</td>
<td>113.0</td>
<td>± 0.602</td>
</tr>
<tr>
<td>0.003</td>
<td>139.6</td>
<td>± 1.409</td>
</tr>
<tr>
<td>0.004</td>
<td>169.2</td>
<td>± 2.024</td>
</tr>
<tr>
<td>0.005</td>
<td>196.7</td>
<td>± 2.118</td>
</tr>
<tr>
<td>0.006</td>
<td>219.4</td>
<td>± 3.145</td>
</tr>
</tbody>
</table>

(1) AVERAGE MESSAGE DELAY
(2) 95% CONFIDENCE INTERVAL
methods produced similar results. For higher values of \[ i \] the percent of time that the queues were full increased to 25%. In this case, the retransmission method began to diverge from the full-server blocking results producing higher average message delay.

It is apparent that these retransmission models are quite complex which makes our full-server blocking model in Chapter 3 look more promising. Furthermore it is apparent that full-server blocking produced results comparably to retransmission if the queues are full only occasionally. However, one might ask are there other blocking models that should be considered?

Caseau and Pujolle describe two types of blocking that do not involve retransmission (Ref. 7). Let \( t \) be the time at which node \( (i + 1) \) becomes full. Let \( t' > t \) be the time at which node \( (i + 1) \) completes the transmission of the message it is serving. So at \( t' \), node \( (i + 1) \) is no longer full and may accept incoming messages. For the interval \( (t, t') \), no customer can gain entry to the queue at node \( (i + 1) \). We will assume that customers arriving from an external source in this interval will be lost. There are two possible blocking policies for node \( i \):

a. At time \( t \), node \( i \) could be blocked in an empty-server state by not allowing its customer to complete service;
b. At time t, node i may carry out service on its customer, however, if this customer wishes to enter node (i + 1) before time t' then the node becomes blocked in a full-server state.

Full-server blocking was considered in Chapter 3. We will now consider empty-server blocking.

When a node becomes full we could imagine it sending messages to other nodes telling them not to transmit any further messages. When space becomes available the previously full node would send another message to the nodes to inform them they may begin transmitting as normal. In our model of tandem queues, we will again assume external messages will be lost. However the node (i + 1) will be allowed to "turn-off" node i when it, node (i + 1), becomes full. Consider the state-transition diagram for this scenario shown in Figure 20. This is almost identical to the state-transition diagram for full-server blocking shown in Figure 14 except that:

a. The maximum queue lengths differ by one; and

b. The arrival of an external token causing node (i + 1) to become full will cause blocking at node i.

In the empty-server blocking system, messages wishing to depart the system after service at node i will be blocked if node (i + 1) becomes full. In the full-server blocking system, these messages will be allowed to complete service at
node i and only messages wishing to enter node \((i + 1)\) will be blocked. The full-server blocking situation seems more realistic. In fact, Pennotti and Schwartz used a similar model in their analysis of congestion and flow control (Ref. 15). They assumed that external tokens were serviced while tokens heading for the next node might be blocked. They allowed for two queues in each node but gave system tokens priority. We will not allow external tokens to be served while system tokens are blocked but, like Pennotti and Schwartz, we will allow a token wishing to depart the system to complete his service at node i if node \((i + 1)\) is full. Furthermore, if the next token in the queue is also an external token its service will not be impeded.

As we have seen retransmission introduces significant computational problems. Since the two types of blocking were very similar, we feel quite justified in concentrating our effort on the full-server blocking scenario. If one wished to consider empty-server blocking scenario, the absorbing state methodology would apply equally well.
FIGURE 20: MARKOV STATE-TRANSITION DIAGRAM FOR THREE FINITE TANDEM QUEUES WITH EMPTY-SERVER BLOCKING ($k_2 = 5$, $k_3 = 3$)

* $p_2 \mu_2$

** $(1-p_2) \mu_2$
CHAPTER 5

AN ALTERNATIVE METHOD OF SOLVING THE MARKOV CHAIN MODELS

We have seen that our generalized tandem queue models in Chapter 3 and Chapter 4 result in a large system of linear equations which can be solved using Gauss-Seidel iteration. The Gauss-Seidel method was chosen because it did not require the matrix \((I - Q)\) to be held in storage. A subroutine could be written to generate the \((i,j)\)th component of \((I - Q)\) when it was needed. In this chapter, we present an alternative solution method that attempts to separate the queues and solve a series of smaller problems. Although we will describe the method for the full-server blocking model described in Chapter 3, this method could be equally well applied to the models in Chapter 4. Special notation used in this chapter is listed in Table IV.

Recall that in Figure 14, we presented a portion of the state-transition matrix for the three tandem queue example with \(n_1\) fixed, \(K_2 = 4\) and \(K_3 = 2\). We can assume \(n_1\) was fixed at 1. If we define the set of states \(S_1\) as those in which \(n_1 \leq 1\) and \((n_2, n_3)\) may vary and define the set \(S'_0\) as those states in which \(n_1 = 0\), we can see that \(\emptyset \subseteq S'_0 \subseteq S_1\). In this example, \(S'_0\) contains 23 states and \(S_1\) contains 46 states. Now to get to the absorbing state \(\emptyset\) from some state in \(S_1\) with \(n_1 = 1\), we must first pass through some state in \(S'_0\) in which \(n_1 = 0\). Once the Markov Chain enters \(S'_0\), it
<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F$</td>
<td>Absorption probability matrix in which $f_{jk}$ is the probability of being absorbed into state $k'S'_{i-1}$ given that the initial state was $j'S'_i$</td>
</tr>
<tr>
<td>$\bar{R}_i$</td>
<td>Vector of mean times to absorption from all states in which $n_1 = i$ (i.e. set $S'_i$)</td>
</tr>
<tr>
<td>$N$</td>
<td>Matrix $n_{jk}$ where $n_{jk}$ is the mean number of visits to state $k'S'<em>i$ before absorption into $S'</em>{i-1}$ given that the initial state was $j'S'_i$</td>
</tr>
<tr>
<td>$P_i$</td>
<td>Transition probability matrix for set of states $S_i$</td>
</tr>
<tr>
<td>$Q_i$</td>
<td>Transition probability matrix for set $S'_i$</td>
</tr>
<tr>
<td>$R_i$</td>
<td>Probability matrix for transitions from set $S'<em>i$ to set $S'</em>{i-1}$</td>
</tr>
<tr>
<td>$S_i$</td>
<td>Set of states in which $n_1 &lt; i$</td>
</tr>
<tr>
<td>$S'_i$</td>
<td>Set of states in which $n_1 = i$</td>
</tr>
<tr>
<td>$\bar{F}_i$</td>
<td>Vector of mean time to next transition for states in set $S'_i$</td>
</tr>
<tr>
<td>$\bar{v}$</td>
<td>Vector of mean time is set $S'<em>i$ before being absorbed into $S'</em>{i-1}$</td>
</tr>
</tbody>
</table>
cannot leave this set. It will eventually be absorbed into state ₀. Thus S'₀ is an absorbing subspace for the states in S₁.

Similarly, S₂ can be defined as all states with n₁ ≤ 2 and S₁ is an absorbing subspace for the states in S₂. In our example, S₂ contains 69 states. In general Sᵢ₋₁ is an absorbing subspace in Sᵢ.

Let us assume we have solved the two queue problem on nodes 2 and 3:

\[(I - Q₀)\bar{m}₀ = \bar{τ}_₀\]

Thus we know all the values of mᵢ where j+ S'₀.

We know that S'₀ is an absorbing subspace in S₁. Since we know the mean time to reach state ₀ for all the states in S'₀, we can use this to calculate the mean time to reach state ₀ from any state in S₁. We can write the transition matrix on S₁ as

\[
P₁ = \begin{bmatrix}
S'₀ & S'₁ \\
Q₀ & 0 & \text{ } \\
R₁ & Q₁
\end{bmatrix}
\]

We have seen that

\[
(I - P₁) \begin{bmatrix}
\bar{m}₀ \\
\bar{m}₁
\end{bmatrix} = \begin{bmatrix}
\bar{τ}_₀ \\
\bar{τ}_₁
\end{bmatrix}
\]

which reduces to

\[(I - Q₀)\bar{m}₀ = \bar{τ}_0\]

\[(I - Q₁)\bar{m}₁ = \bar{τ}_1 + R₁\bar{m}₀\]
Notice that both $\bar{m}_0$ and $\bar{m}_1$ contain 23 elements. Since we have solved (3) by assumption, the right hand side of (4) is a known vector (since $\bar{m}_0$ is known) and we can solve to find $\bar{m}_1$ as a system of linear equations. This system is half the size of the original system in (2).

When we repeat this for $S'_2$, we see that we need only the results obtained for $S'_1$. Since

$$S'_0 \begin{bmatrix} Q_0 & 0 & 0 \\ P_2 & = & S'_1 \begin{bmatrix} R_1 & Q_1 & 0 \\ S'_2 & = & 0 & R_2 & Q_2 \end{bmatrix} \end{bmatrix}$$

$$(I - P_2) \begin{bmatrix} \bar{m}_0 \\ \bar{m}_1 \\ \bar{m}_2 \end{bmatrix} = \begin{bmatrix} \bar{e}_0 \\ \bar{e}_1 \\ \bar{e}_2 \end{bmatrix}$$

gives the equation

$$(I - Q_2) \bar{m}_2 = \bar{e}_2 + R_2 \bar{m}_1.$$ 

Furthermore, $Q_1 = Q_2$ and $\bar{t}_1 = \bar{t}_2$.

It becomes apparent that the value of $n_i$ (the subscript) does not affect the transitions except at the boundaries. We can define $Q_i = Q$, $\bar{e}_i = \bar{e}$ for all $1 \leq i \leq K_1$ and $R_i = R$ for all $2 \leq i \leq K_1$. If we let $N = (I - Q)^{-1}$, $\bar{r} = N \cdot \bar{t}$ and $F = N \cdot R$, then for $i \geq 2$ we have

$$\bar{m}_i = \bar{r} + F \cdot \bar{m}_{i-1},$$

$N = \{ n_{jk} \}$ where $n_{jk}$ can be thought of as the mean number of visits to $k \cdot S'_i$ before being absorbed into $S'_{i-1}$ given that the initial state was $j \cdot S'_i$. $\bar{r}$ is a vector in which $r_j$ is the mean time in set $S'_j$ before being absorbed into $S'_{i-1}$.
given that the initial state was $j \in S'_i$. $F$ is the absorption probability matrix. That is, $F = \{f_{jk}\}$ where $f_{jk}$ is the probability of being absorbed into state $k \in S'_{i-1}$ given that the initial state was $j \in S'_i$.

To solve the entire three queue problem using this method we must solve $(K_1 + 1)$ smaller problems in a recursive fashion. Each of these smaller problems corresponds to the situation in which $n_1$ is fixed while all values of $n_2$ and $n_3$ are considered. To clarify this approach, a numerical example is provided. For convenience, we have only considered a two queue system, shown in Figure 21. The sets of states are:

$$S'_i = \{(i,0),(i,1),(i,2),(i,3)\} \quad i \geq 0$$

and

$$S_i = \{S'_0,S'_1, \ldots, S'_i\}.$$

![Diagram](attachment:image.png)

**FIGURE 21**: EXAMPLE OF TWO FINITE TANDEM QUEUES WITH CROSS-TRAFFIC
\[(I - Q) = \begin{bmatrix}
1.0 & -0.333 & 0.0 & 0.0 \\
-0.333 & 1.0 & -0.222 & 0.0 \\
0.0 & -0.429 & 1.0 & 0.0 \\
0.0 & 0.0 & -1.0 & 1.0
\end{bmatrix}\]

\[R_1 = \begin{bmatrix}
0.0 & 0.667 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.444 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.571 \\
0.0 & 0.0 & 0.0 & 0.0
\end{bmatrix}\]

\[\bar{t} = \begin{bmatrix}
1.667 \\
1.111 \\
1.429 \\
3.333
\end{bmatrix}\]

and \[\bar{m}_0 = \begin{bmatrix}
0.000 \\
0.333 \\
0.667 \\
1.000
\end{bmatrix}\]

Thus we can compute

\[N = (I - Q)^{-1} = \begin{bmatrix}
1.140 & 0.419 & 0.093 & 0.000 \\
0.419 & 1.260 & 0.280 & 0.000 \\
0.180 & 0.540 & 1.120 & 0.000 \\
0.180 & 0.540 & 1.120 & 1.000
\end{bmatrix}\]

\[F_1 = N \cdot R_1 = \begin{bmatrix}
0.0 & 0.760 & 0.187 & 0.053 \\
0.0 & 0.280 & 0.560 & 0.160 \\
0.0 & 0.120 & 0.240 & 0.640 \\
0.0 & 0.120 & 0.240 & 0.640
\end{bmatrix}\]

\[\bar{r} = N \cdot \bar{t} = \begin{bmatrix}
2.500 \\
2.500 \\
2.500 \\
5.833
\end{bmatrix}\]

-59-
and

\[ \bar{m}_1 = \bar{F} + \mathbf{F}_1 \cdot \bar{m}_0 = \begin{bmatrix} 2.500 \\ 2.500 \\ 2.500 \\ 5.833 \end{bmatrix} + \begin{bmatrix} 4.310 \\ 6.267 \\ 8.400 \\ 8.400 \end{bmatrix} = \begin{bmatrix} 6.810 \\ 8.767 \\ 10.900 \\ 14.233 \end{bmatrix} \]

To compute \( \bar{m}_2 \) we notice that

\[ R_2 = \begin{bmatrix} 0.333 & 0.333 & 0.0 & 0.0 \\ 0.0 & 0.222 & 0.222 & 0.0 \\ 0.0 & 0.0 & 0.286 & 0.286 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \]

thus

\[ F = NR_2 = \begin{bmatrix} 0.380 & 0.473 & 0.120 & 0.027 \\ 0.140 & 0.420 & 0.360 & 0.080 \\ 0.060 & 0.180 & 0.440 & 0.320 \\ 0.060 & 0.180 & 0.440 & 0.320 \end{bmatrix} \]

and

\[ \bar{m}_2 = \bar{F} + \mathbf{F}_1 \cdot \bar{m}_1 = \begin{bmatrix} 2.500 \\ 2.500 \\ 2.500 \\ 5.833 \end{bmatrix} + \begin{bmatrix} 8.427 \\ 9.698 \\ 11.337 \\ 11.337 \end{bmatrix} = \begin{bmatrix} 10.927 \\ 12.198 \\ 13.837 \\ 17.170 \end{bmatrix} \]

We can carry on by applying the formula

\[ \bar{m}_n = \bar{F} + \mathbf{F}_1 \cdot \bar{m}_{n-1} \]

Thus

\[ \bar{m}_3 = \begin{bmatrix} 14.546 \\ 15.508 \\ 16.934 \\ 20.267 \end{bmatrix}, \quad \bar{m}_4 = \begin{bmatrix} 17.942 \\ 18.767 \\ 20.101 \\ 23.433 \end{bmatrix} \]
Now if a marked token arrived to this system and saw the state \((4, 2)\), its expected time in system can be found in \(M_4\) and in this case is 20.101.

If we consider a general three queue example, we see that \(S_1\) has \(\{(K_2 + 2)(K_3 + 2)\}\) states. Thus inversion must be carried out on a \(\{(K_2 + 2)(K_3 + 2)\}\) square matrix. If \(K_2\) and \(K_3\) are large this would not be desirable. Similarly for \(k\) tandem queues we would have approximately \(\prod_{i=2}^{k} (K_i + 2)\) states. Even though we have reduced the problem of \(k\) queues to an easier problem involving \((k - 1)\) queues, the computational and storage problems are not resolved.

We have identified one way to separate the queues and maintain the accuracy of the absorbing state method. However, it is apparent that we should examine approximation methods if this type of approach is to become practical.
the solution of a system of ordinary differential equations of the form:
\[ \dot{Q}_j(t) = \dot{A}_j(t) + \sum \frac{b_{ij}u_i(t)}{i} - u_j(t) \]
where
- $\dot{Q}_j(t)$ is the time derivative of the backlog function,
- $\dot{A}_j(t)$ is the time derivative of the external arrival function,
- $b_{ij}$ is the fraction of flow out of node $i$ that goes to node $j$, and
- $u_j(t)$ is the service rate at node $j$ as a function of time.

There is only one equation for each node in the network! Thus it is quite conceivable that the entire communication network could be modelled and thus a more realistic cross-traffic model could be examined. In fact since the flows are deterministic our adaptive routing problem reduces to an optimal control problem (Ref. 1, p. 225).
<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_j(t)$</td>
<td>Total rate of arrivals to node $j$</td>
</tr>
<tr>
<td>$b_{ij}$</td>
<td>Fraction of flow out of node $i$ that goes to node $j$</td>
</tr>
<tr>
<td>$D_j(t)$</td>
<td>Cumulative departures from node $j$ as a function of time</td>
</tr>
<tr>
<td>$(r)$</td>
<td>New arrival rate after $r$ slope changes in backlog function</td>
</tr>
<tr>
<td>$Q_j(t)$</td>
<td>Backlog at node $j$ at time $t$</td>
</tr>
<tr>
<td>$s^{(r)}$</td>
<td>$r$th slope change in the backlog function</td>
</tr>
<tr>
<td>$t_i$</td>
<td>Time at which node $i$ empties</td>
</tr>
<tr>
<td>$t^{(r)}$</td>
<td>Time of $r$th slope change in backlog function</td>
</tr>
</tbody>
</table>
FIGURE 22: EXAMPLE OF FIVE TANDEM QUEUES
tokens are replaced by fluids and the flow out of nodes is continuous and deterministic, we may plot the backlog of work \( Q_j \) at each node as a function of time (Fig. 23). Here \( t_i \) is the time at which the backlog is completed (i.e. \( Q_i(t_i) = 0 \)). Notice that the backlog increases at node 2 during the period \((0, t_1)\) because work is flowing into node 2 faster than it is flowing out (\( u_1 = (3/2) > u_2 = 1 \)). Also notice that once the backlog reduces to zero it remains there. The fluid flow model would suggest that the time to pass through the system would be \( t_3 \). However, since this ignores the service time at nodes 4 and 5, we may wish to add some function of the mean service times.

Instead of plotting the backlog of work explicitly, Newell examined the departure processes by defining \( D_j(t) \) as the cumulative departures from node \( j \) during the period \((0, t)\). \( D_j(0) \) was defined as the tokens seen ahead of node \( j \) at time of arrival. For example, if our marked token arrives and sees \( \bar{n} = (n_1, n_2, ..., n_k) \) then \( D_k(0) = 0 \) and

\[
D_j(0) = \sum_{i=j+1}^{k} n_i \quad \text{for} \quad 0 < j < k - 1.
\]

Node 0 will be assumed to have a zero service rate and \( D_0(t) = D_0(0) \) will be used in the formula for \( D_1(t) \).

If we assume that these departure processes are continuous and deterministic then a simple expression can be obtained for \( D_j(t) \):

-64-
$Q_1(t) = 3 - (3/2)t$ for $0 \leq t \leq 2$

$Q_2(t) = \begin{cases} 
4 + (1/2)t & \text{for } 0 \leq t \leq 2 \\
5 - (t-2) & \text{for } 2 < t \leq 7 
\end{cases}$

$Q_3(t) = \begin{cases} 
3 + (1/2)t & \text{for } 0 \leq t < 7 \\
(13/2) - (1/2)(t-7) & \text{for } 7 \leq t \leq 20 
\end{cases}$

$Q_4(t) = 3 - (3/4)t$

$Q_5(t) = \begin{cases} 
4 + (1/4)t & \text{for } 0 \leq t \leq 4 \\
5 - (1/2)(t-4) & \text{for } 4 < t \leq 14 
\end{cases}$

**Figure 23:** Backlog at nodes as a function of time: five tandem queue example
\[ D_j(t) = \min \{ D_j(0) + u_j t, D_{j-1}(t) \} \]

These departure processes can now be plotted on a single graph as shown in Figure 24. The backlog of work at node \( j \) can be read from this graph as

\[ Q_j(t) = D_{j-1}(t) - D_j(t) \quad \text{for} \quad 1 \leq j \leq k \]

and the backlog disappears when \( D_j(t) \) and \( D_{j-1}(t) \) coincide.

The time for a token to obtain service at a node can be read from the graph as the horizontal distance between \( D_{j-1} \) and \( D_j \). Figure 24 shows the departure processes for our five node example. The travel time for our marked token is shown as the horizontal dotted line. It is readily apparent that

\[ t_1 = \frac{D_0(0) - D_1(0)}{u_1} = 3 \frac{2}{3} = 3 \]

\[ t_2 = \frac{D_0(0) - D_2(0)}{u_2} = 7 \frac{1}{2} = 7 \quad \text{and} \quad t_3 = \frac{D_0(0) - D_3(0)}{u_3} = 10 \frac{2}{3} = 20 \]

In this case, the time in system is determined by the number of tokens seen between our marked token and node 3. Furthermore \( D_3(t) \) remains undisturbed by the other departure processes for the interval \((0, t_3)\) whereas the other functions \( D_j(t) \) all have one point at which they change slope.

Newell considered finite tandem queues with blocking. The methodology is basically the same except with \( k_j \) defined as the maximum storage capacity of node \( j \). The expression for \( D_j(t) \) becomes:

\[ D_j(t) = \min \{ D_j(0) + u_j t, D_{j-1}(t), D_{j+1}(t) + k_{j+1} \} \]

If the queue (backlog) at node \((j+1)\) reaches its capacity then the departure process at node \( j \) is disturbed. In fact,
FIGURE 24: DEPARTURE PROCESSES FOR FIVE TANDEM QUEUE EXAMPLE
the cumulative departure function $D_j(t)$ runs parallel to $D_{j+1}(t)$ but is shifted up by $K_{j+1}$. Node $j$ may also reach its capacity affecting the departure process at node $(j-1)$ and so on. Thus $D_j(t)$ may be constrained by a "downstream bottleneck" node $b$ causing the all storage space between node $j$ and node $b$ to fill to capacity. Or it may be constrained by an "upstream bottleneck" node $m$ causing all the queues to empty between node $(m+1)$ and node $j$.

In the graphical procedure, these storage capacity constraints cause additional slope changes in the $D_j$ functions.

This fluid flow approximation method can be extended to consider tandem queues with cross-traffic. Consider the four tandem queue example shown in Figure 25. In the fluid flow model, these random departures are modelled as a splitting of the flow into fractions corresponding to the probabilities. Thus $p_j$ is the fraction of the flow out of node $j$ that departs the system and $(1-p_j)$ is the fraction of the flow out that continues to node $(j+1)$. The departure processes in this case become complicated because of this flow splitting. So instead we will consider the backlog functions $Q_j(t)$. Figure 26 shows the functions that would be obtained if our marked token arrived to find the state $(6,4,4,3)$.

Obviously, $Q_j(t)$ must be non-negative. Furthermore, if $Q_j(t) = 0$ for $t > 0$ then $Q_j(t') = 0$ for all $t' > t$. That is, once the backlog at a node becomes zero it will remain zero.
FIGURE 25: EXAMPLE OF FOUR TANDEM QUEUES WITH CROSS-TRAFFIC
Figure 26: Backlog Functions for Four Tandem Queue Example
As long as \( Q_1(t) > 0 \),
\[
Q_1(t) = n_1 - \mu_1 t
\]
and node 1 will empty at time \( t_1 = n_1/\mu_1 \).

Notice that during the period \((0, t_1)\) work is flowing from node 1 to node 2 at rate \((1 - p_1)\mu_1\). The total rate of flow into node 2 is
\[
\lambda'_2 = (1 - p_1)\mu_1 + \lambda_2.
\]
Thus for \( Q_1(t) > 0 \) and \( Q_2(t) > 0 \),
\[
Q_2(t) = n_2 + (\lambda'_2 - \mu_2)t.
\]
If \((\lambda'_2 - \mu_2) > 0\) the backlog increases but if \((\lambda'_2 - \mu_2) < 0\) the backlog decreases.

The flow rate into node 2 will change if node 1 empties before node 2. \( Q_1(t_1) = 0 \) implies our marked token has arrived at node 2 in which case the external arrivals may be "turned off". Thus our new function for \( Q_2(t) \) is
\[
Q_2(t) = Q_2(t_1) - \mu_2 t
\]
which will empty at \( t_2 = t_1 + Q_2(t_1)/\mu_2 \). In this case, the rate of flow from node 2 into node 3 is undisturbed during the period \((0, t_2)\) and is equal to \((1 - p_2)\mu_2\).

On the other hand, if node 2 empties at some time \( t_2 < t_1 \) then \( Q_2(t_1) = 0 \) and the rate of flow into node 3 will be reduced to \((1 - p_2)\lambda'_2\) during the period \((t_2, t_1)\).

Thus we have \( Q_1(t) \) a straight line function and \( Q_2(t) \) a continuous function which can have at most one slope change before it reaches zero. We can analyze \( Q_3(t) \) in a similar manner and it becomes apparent that it can have at most two
slope changes before reaching zero as shown in Figure 26 when 
\( t_2 < t_1 < t_3 \).

Consider node \( j \) in a system of \( k \) tandem queues with 
cross-traffic. If node \( j \) is non-empty then \( Q_j \) can be 
expressed as a linear function of time:
\[
Q_j(t) = Q_j(t^{(r)}) + s^{(r)}(t-t^{(r)})
\]
where 
\( t^{(r)} \) is the time of the last slope change, and 
\( s^{(r)} \) is the slope.

The slope can be calculated as the difference between the 
flow into the node and the flow out of the node. Denote the 
closest non-empty node ahead of node \( j \) as node \( i \). Then the 
flow into node \( j \) would be
\[
\lambda_j^{(r)} = \sum_{m=i}^{j-1} \left( 1 - P_m \right) \mu_i + \sum_{n=i+1}^{j-1} \left( 1 - P_m \right) \lambda_n + \lambda_j \tag{2}
\]
The flow out of node \( j \) would be \( \mu_j \) since node \( j \) is non-empty.

Thus the slope of the function at time \( t \) would be 
\[
s^{(r)} = \lambda_j^{(r)} - \mu_j.
\]
The slope will change if node \( i \) empties before node \( j \) in 
which case either:

a. all the nodes ahead of node \( j \) will be empty, our 
   marked token will go directly to node \( j \) and the new 
   slope will be \( s^{(r+1)} = -\mu_j \); or

b. there is some closest non-empty node \( h \) ahead of 
   node \( j \) and the new slope will be \( s^{(r+1)} = \lambda_j^{(r+1)} - \mu_j \) 
   where \( \lambda_j^{(r+1)} \) is obtained using formula (2) with \( i \) 
   replaced by \( h \).
Another way of looking at this would be to drop nodes from the system when they become empty and combine their arrival rates as appropriate.

Thus we can work quickly through the system changing slopes, dropping empty nodes and combining arrival streams to get just one non-empty node at some time $t'$. When this node empties at time $t^*$, this is our fluid flow estimate of the time to pass through the system. This estimate was obtained with considerably less computation than the Markov Chain method.

Newell suggests that these approximations will produce good results when one of the queues is large. Three finite tandem queues with cross-traffic were simulated and the fluid approximation was compared to the observed time in system. The external arrival rates were 0.003, the service rates were 0.05 and the maximum queue length at each node was 5. Table VI shows the average estimation error as a percent of the average time in system for varying values of the system arrival rate. A definite improvement was noticed as the system became more congested. However, the fluid approximation was found to underestimate the time in system.

In an attempt to improve the approximation, the service times were incorporated into the estimate. That is, if the fluid model predicted that our marked token would arrive at node $(j+1)$ at time $t_j$ and to find all the nodes ahead empty then the average service times at these nodes was added to $t_j$. 

-73-
### TABLE VI

THE ACCURACY OF FLUID APPROXIMATIONS

<table>
<thead>
<tr>
<th>SYSTEM ARRIVAL RATE</th>
<th>AVERAGE TIME IN SYSTEM</th>
<th>ESTIMATION ERROR OF FLUID MODEL</th>
<th>ESTIMATION ERROR WITH ADDED SERVICE TIMES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1)</td>
<td>(2)</td>
<td>(1)</td>
</tr>
<tr>
<td>0.01</td>
<td>106.8</td>
<td>41.42</td>
<td>0.39</td>
</tr>
<tr>
<td>0.02</td>
<td>139.98</td>
<td>36.19</td>
<td>0.26</td>
</tr>
<tr>
<td>0.03</td>
<td>175.36</td>
<td>32.09</td>
<td>0.18</td>
</tr>
</tbody>
</table>

(1) AVERAGE ESTIMATION ERROR  
(2) AVERAGE ESTIMATION ERROR/AVERAGE TIME IN SYSTEM
to obtain an estimate of his time in the system. For example in Figure 27, we see that at time \( t_2 \) our marked token will complete service at node 2 and node 3 will be empty. His arrival to node 3 is assumed to cause an instantaneous unit increase in the backlog function. Thus his estimated time in system would be \( t_2 + 1/\mu_3 \). Kleinrock presented a similar concept for his single queue situation (Ref. 16, p. 95).

The accuracy of this new method was analyzed using the same three queue system discussed for the pure fluid flow method. Some improvement was found in the estimation accuracy as shown in Table VI. However, this method also underestimated the time in system.

Although fluid flow methods appear quite crude, they do correctly weight the importance of the tokens seen on arrival according to their contribution to the delay of our marked token. The potential use of the fluid flow model with added service times in adaptive routing will be investigated in the next chapter.

It should be noted that a fluid approximation was developed for the infinite tandem queues with cross-traffic. However the fluid approach could be much more practical than the Markov Chain model in other situations. For example, Vandergraft describes how networks of queues can be modelled using a fluid flow approach (Ref. 17). His method requires
FIGURE 27: FLUID FLOW MODEL WITH ADDED SERVICE TIMES
the solution of a system of ordinary differential equations of the form:

\[ \dot{Q}_j(t) = \dot{A}_j(t) + \sum_i b_{ij} \mu_i(t) - \mu_j(t) \]

where

- \( \dot{Q}_j(t) \) is the time derivative of the backlog function,
- \( \dot{A}_j(t) \) is the time derivative of the external arrival function,
- \( b_{ij} \) is the fraction of flow out of node \( i \) that goes to node \( j \), and
- \( \mu_j(t) \) is the service rate at node \( j \) as a function of time.

There is only one equation for each node in the network! Thus it is quite conceivable that the entire communication network could be modelled and thus a more realistic cross-traffic model could be examined. In fact since the flows are deterministic our adaptive routing problem reduces to an optimal control problem (Ref. 1, p. 225).
CHAPTER 7

A SIMULATION EXPERIMENT ON ADAPTIVE ROUTING IN A NETWORK
OF FINITE QUEUES WITH CROSS-TRAFFIC

We have discussed, in detail, methods of estimating the
time in tandem queues. We have seen the potential benefit of
using estimates for adaptive routing in networks consisting
of paths without cross-traffic. Our goal now is to examine
the usefulness of these estimation methods when paths may be
effected by cross-traffic.

We will consider the two path network in Figure 28.
Path 1 will be used to denote the path with two tandem
queues. The three tandem queue path will be denoted as
Path 2. The message length will be exponentially distributed
and Kleinrock's independence assumption will be used. Thus
we have independent exponential servers. We will assume the
queues have finite storage capacity. Full-server blocking
will be used for tokens wishing to enter a full queue from
another node in the system. Tokens arriving from an external
source to find a full queue will be lost. The arrival
streams between the various source and destination pairs will
again be independent Poisson processes with parameter $\lambda_{ij}$
for $(S_i, D_j)$ $1 \leq i \leq j \leq 2$ and $3 \leq i \leq j \leq 5$. Thus we have
two independent paths. Our effort will concentrate on
routing system traffic (i.e. traffic between source 0 and
destination 6) based on the queue lengths seen on arrival and
knowledge of the service rates at the nodes and the arrival rates for all source/destination pairs.

By combining the arrival streams as discussed in Chapter 3, we can estimate the total rate at which traffic will arrive at each node from an external source:

\[
\begin{align*}
\lambda_1 &= \lambda'_{11} + \lambda'_{12} \\
\lambda_2 &= \lambda'_{22} \\
\lambda_3 &= \lambda'_{33} + \lambda'_{34} + \lambda'_{35} \\
\lambda_4 &= \lambda'_{44} + \lambda'_{45} \\
\lambda_5 &= \lambda'_{55}
\end{align*}
\]

However estimating the departure probabilities, using the formulae in Chapter 3, is not so easy because we require prior knowledge of the amount of system traffic that will be routed on each path. This is exactly the aspect of the system that we wish to vary using adaptive routing.

Thus we must somehow estimate the proportions of system traffic that will be routed to each path before we apply the Markov Model. One approach would be to calculate the optimal fixed proportions based on the steady-state time in a similar system involving infinite queues and random routing. Then use this result as an estimate of the rate at which system arrivals will be routed by the adaptive schemes. That is, assuming \( \lambda'_{06} \) can be split into \( \lambda^{(1)}_{06} \) and \( \lambda^{(2)}_{06} \) corresponding to the optimal fixed proportions sent to path 1 and 2 respectively then we can estimate the departure probabilities in our Markov model as
\[ p_1 = \frac{\lambda'_{11}}{\lambda_{06} + \lambda^{(1)}_{06}} \]
\[ p_3 = \frac{\lambda'_{33}}{\lambda_{06} + \lambda^{(2)}_{06}} \]
\[ p_4 = \frac{\lambda'_{34} + \lambda'_{44}}{\lambda_4 + (1 - p_3)(\lambda_3 + \lambda^{(2)}_{06})} \]

As shown in Chapter 3, the values of \( p_2 \) and \( p_5 \) are not required in the estimation techniques.

Recall our method of calculating the optimal fixed proportions. For convenience we will denote \( \lambda'_{06} \) as \( \lambda \) and \( \lambda^{(1)}, \lambda^{(2)} \) as \( \lambda^{(1)}, \lambda^{(2)} \) appropriately. We must solve

\[ T = \min \left( \frac{\lambda^{(1)}}{\lambda} T_1 + \left( \frac{(\lambda - \lambda^{(1)})}{\lambda} \right) T_2 \right) \text{ for } 0 \leq \lambda^{(1)} \leq \lambda \]

Here \( T_1 \) and \( T_2 \) are the steady-state times to pass through the paths given that system arrivals to path 1 have rate \( \lambda^{(1)} \) and to path 2 have rate \( (\lambda - \lambda^{(1)}) \). Numerical methods were used to solve \( T'(\lambda^{(1)}) = 0 \) and then obtain the value of \( \lambda^{(1)} \) that provides the minimum \( T \).

We will consider five routing schemes.

a. **Expected Time method**: Based on the Markov Chain model with the parameters estimated as described above, we will route to path 1 if \( E(T_1/n_1, n_2) \leq E(T_2/n_3, n_4, n_5) \) and to path 2 if \( E(T_1/n_1, n_2) > E(T_2/n_3, n_4, n_5) \).

b. **Total Queue Length method**: We will route to path 1 if \( (n_1 + n_2) \leq (n_3 + n_4 + n_5) \) and to path 2 if \( (n_1 + n_2) > (n_3 + n_4 + n_5) \).

c. **Fluid Approximation method**: Based on the fluid flow model with parameters as estimated above, we will
route to path 1 if \( F(n_1, n_2) \leq F(n_3, n_4, n_5) \) and to path 2 if \( F(n_1, n_2) > F(n_3, n_4, n_5) \). Here \( -F(n) \) represents the fluid flow approximation of the time on the path as described in Chapter 6.

d. **Random Routing:** Using the values of \( \lambda^{(1)} \) and \( \lambda^{(2)} \), we will route to path 1 and 2 randomly with probabilities \( r_1 = \lambda^{(1)}/\lambda \) and \( r_2 = \lambda^{(2)}/\lambda \). Notice that \( \lambda^{(1)} \) and \( \lambda^{(2)} \) were obtained by assuming infinite queues. Thus they are not expected to provide the result \( T \) when applied to the finite queue system.

e. **Fixed Routing:** All tokens will be routed to path 1 unless node 1 is full in which case they will be routed to path 2.

It should be noted that in all of the methods the routing scheme will be applied if both node 1 and node 3 have storage space available. If one of these nodes are full the token will be sent on the other path. If both nodes are full, the token is lost. Thus even with the fixed routing method some tokens may be sent on path 2. It should also be noted that we have limited the scope of this work to comparing adaptive routing schemes based on prediction versus random routing and fixed routing. Other schemes such as quadratic routing or the Arpanet adaptive routing schemes (Ref. 18) have been left out of this study.
Simulation was used to analyze all five methods for the following sample system. We assumed
\[ \lambda'_{11} = \lambda'_{12} = \lambda'_{22} = 0.003 \]
\[ \lambda'_{33} = \lambda'_{34} = \lambda'_{35} = \lambda'_{44} = \lambda'_{45} = \lambda'_{55} = 0.002 \]
\[ \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = 0.05 \]
\[ K_1 = K_2 = K_3 = K_4 = K_5 = 5 \]
and allowed the system arrival rate \( \lambda \) to vary from 0.01 to 0.07 in increments of 0.01. Five independent replications were made for each routing method and each value of \( \lambda \). The average of these five observations are shown in Table VII along with the 95% confidence intervals. These estimates of average message delay are also plotted in Figure 29.

The difference between the best and worst method for each value of \( \lambda \) was about 20% of the average message delay. However as can be seen in Figure 29, the curves were found to criss-cross. The fixed and queue length routing methods performed best when the system arrival rate was low. As the arrival rate increased, the performance of the fixed routing method deteriorated relative to the queue length method. However, the fluid approximation method improved and surpassed the queue length method. The random routing and expected time methods performed poorly compared to the other methods for all values of \( \lambda \).

The poor performance of the expected time method might be partially explained by examining the Markov parameters. Table VIII shows the percent of system traffic routed to path 2.
### Table VII

Average Message Delay Found in Simulation of Adaptive Routing Systems

<table>
<thead>
<tr>
<th>System Arrival Rate</th>
<th>Expected Time</th>
<th>Queue Length</th>
<th>Fluid Approx.</th>
<th>Random</th>
<th>Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>55.92 (0.71)</td>
<td>47.93 (0.55)</td>
<td>53.94 (0.63)</td>
<td>58.53 (0.48)</td>
<td>48.61 (0.33)</td>
</tr>
<tr>
<td>0.02</td>
<td>66.62 (0.72)</td>
<td>57.95 (0.88)</td>
<td>60.95 (0.62)</td>
<td>69.75 (0.28)</td>
<td>62.36 (0.74)</td>
</tr>
<tr>
<td>0.03</td>
<td>80.21 (0.90)</td>
<td>72.56 (1.40)</td>
<td>69.93 (1.02)</td>
<td>81.78 (0.70)</td>
<td>81.0 (1.21)</td>
</tr>
<tr>
<td>0.04</td>
<td>98.70 (0.73)</td>
<td>89.71 (1.33)</td>
<td>83.34 (0.50)</td>
<td>97.12 (0.54)</td>
<td>99.97 (0.58)</td>
</tr>
<tr>
<td>0.05</td>
<td>118.7 (1.02)</td>
<td>106.1 (0.80)</td>
<td>101.8 (1.11)</td>
<td>115.1 (0.76)</td>
<td>115.5 (0.68)</td>
</tr>
<tr>
<td>0.06</td>
<td>139.0 (1.15)</td>
<td>122.12 (1.37)</td>
<td>124.8 (0.75)</td>
<td>136.7 (1.03)</td>
<td>127.8 (1.21)</td>
</tr>
<tr>
<td>0.07</td>
<td>158.9 (1.48)</td>
<td>136.84 (0.98)</td>
<td>158.6 (0.78)</td>
<td>148.58 (1.39)</td>
<td>141.1 (0.65)</td>
</tr>
</tbody>
</table>

(The values in brackets are the half-widths of the 95% confidence intervals)
FIGURE 29: AVERAGE MESSAGE DELAY FOR FIVE NODE SYSTEM WITH EXTERNAL TRAFFIC
### TABLE VIII
COMPARISON OF PARAMETER ESTIMATES AND OBSERVED VALUES FOR ADAPTIVE ROUTING METHODS

<table>
<thead>
<tr>
<th>SYSTEM ARRIVAL RATE</th>
<th>PERCENT OF SYSTEM TRAFFIC ROUTED TO PATH 2, ESTIMATED*</th>
<th>QUEUE LENGTH</th>
<th>EXPECTED TIME</th>
<th>FLUID APPROX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>7.46</td>
<td>3.58</td>
<td>15.04</td>
<td>30.39</td>
</tr>
<tr>
<td>0.02</td>
<td>31.24</td>
<td>7.60</td>
<td>17.77</td>
<td>33.15</td>
</tr>
<tr>
<td>0.03</td>
<td>39.17</td>
<td>13.17</td>
<td>21.84</td>
<td>36.42</td>
</tr>
<tr>
<td>0.04</td>
<td>43.13</td>
<td>20.32</td>
<td>28.32</td>
<td>39.63</td>
</tr>
<tr>
<td>0.05</td>
<td>45.50</td>
<td>28.75</td>
<td>35.49</td>
<td>42.11</td>
</tr>
<tr>
<td>0.06</td>
<td>47.07</td>
<td>35.54</td>
<td>42.24</td>
<td>44.08</td>
</tr>
<tr>
<td>0.07</td>
<td>48.18</td>
<td>40.63</td>
<td>45.92</td>
<td>45.46</td>
</tr>
</tbody>
</table>

*These values were obtained by assuming infinite queues and calculating the fixed proportions that should be randomly routed to path 2 to minimize the steady-state average message delay.*
that was used in the parameter estimation and the values that were observed in the simulation runs. Fewer messages were routed to path 2 by the expected time method than estimated. Recall that these estimates were used to calculate the departure probabilities in the Markov Chain model. Thus the departure probabilities would be overestimated on path 1 and underestimated on path 2. This, in turn, would cause the expected time method to underestimate the travel time on path 1 and overestimate the travel time on path 2 and thus make inappropriate routing decisions.

In previous chapters we have discussed the computational requirements of the Markov Chain method. We have now seen the problems in estimating the Markov parameters and applying the method to do adaptive routing. For this example system, the results were not worth the effort. Improvements could be made in the methodology. However, when the potential benefits are compared to the considerable costs, the usefulness of the Markov Chain method in adaptive routing is seriously questioned.

Fortunately, adaptive routing based on the queue length and fluid approximation did show good results. The queue length method produced good results for all values of \( \lambda \) and is very easy to apply given knowledge of the queue lengths on the path (see Ref. 6). The fluid approximation method is also easy to implement if additional information about service rates and traffic patterns is available. Table VIII
shows that the parameter values used in the fluid method were quite close to the actual observed values when \( \lambda \) was moderately large. The simulation results in Table VII demonstrate that the fluid approximation method, if used in an adaptive routing scheme, could potentially reduce the average message delay in a moderately loaded system. The fluid approximation method looks most promising.
CHAPTER 8

CONCLUDING MATERIAL

We began this thesis with an optimistic view of the possibility of modelling communications networks using Markov Chain methods. It was hoped that a mathematical result concerning the expected time in tandem queues could be extended and applied to adaptive routing of messages through a network. Unfortunately when external Poisson arrivals and random departures were incorporated in the Markov Chain model, the complexity of the computations was greatly increased. Finite queues also created modelling difficulties. Furthermore, the Markovian assumption in the estimation method broke down when it was applied in an adaptive routing scheme. In simulation experiments, this Markov Chain method performed poorly compared to other simpler routing methods. This was thought to be caused in part by the difficulty in estimating the model parameters. In light of the computational problems involved in this method, its incorporation into an adaptive routing scheme does not seem feasible.

A fluid approximation of the time in tandem queues with cross-traffic was developed. Although, the estimates were found to be crude, this approach was able to correctly weight the queue lengths seen on arrival as to their importance in creating delays. The fluid approximation was found to be
quite easy to automate and simulation results indicated that its use in an adaptive routing scheme could reduce the average message delay in a moderately loaded system. This method looks promising.

Further work might develop the fluid model in more detail and investigate the possibility of its application in a communication network. The following problems have not been considered in this thesis but should be considered in future studies:

a. **Non-stationary traffic patterns:** It is believed that adaptive routing is of most benefit when the system is heavily loaded and message traffic involves surges (see Ref. 18). The fluid model might be easily extended to consider these problems.

b. **Non-independent service times:** We have noted a potential problem in the use of Kleinrock's independence assumption when applied to a particular system of tandem queues. This should be examined more fully.

c. **Overtaking:** The models in this thesis have been carefully constructed to avoid problems caused by messages from a source overtaking other messages on route to the same destination. By using adaptive routing in a communication network, overtaking is a real possibility. The impact of this overtaking on message delay should be investigated.
d. **Stale Data on Queue Lengths:** We have assumed that the adaptive routing is based on perfect knowledge of the queues on the possible paths. In real systems, this is not the case. Information on the queue length at a node is periodically distributed to other nodes and routing tables are updated based on this data. The use of similarly "stale data" in an expected time calculation and its effect on prediction accuracy should be examined.

e. **Non-independent cross-traffic:** The message traffic in a communication network is quite interdependent. As discussed briefly in Chapter 6, the fluid flow model could be extended to consider more complex types of cross-traffic interference.

It should be noted that this investigation was restricted to examining the potential benefit of a mathematical result for adaptive routing in networks of queues. The practical application of this type of approach to communications networks remains as future work. Individuals wishing to undertake this research might wish to examine the computer programs developed during this study (Ref. 19). Reference 20 on simulating communications networks might also be helpful if a more realistic model is desired. Also reference 18 provides a good survey of the types of adaptive routing methods currently in use.
REFERENCES


ANNEX A

DERIVATION OF SECOND MOMENT OF TIME TO ABSORPTION

The following derivation was provided by Stanford (Ref. 22):

Let $X_j$ be the random variable denoting the time to absorption from state $j$ and let $Y_j$ be the random variable denoting the time to transition out of state $j$.

Then let

$F_j(t) = \text{Prob}[X_j \leq t]$ and $G_j(t) = \text{Prob}[Y_j \leq t]$.

Thus

$F_j(t) = (1 - \sum_{k \in T} p_{jk} G_j(t)) + \sum_{k \in T} p_{jk} \text{Prob}[Y_j + X_k \leq t]$.

Taking the Laplace Transforms, we have

$\Phi_j(s) = (1 - \sum_{k \in T} p_{jk} \Theta_j(s)) + \sum_{k \in T} p_{jk} \Theta_j(s) \Phi_k(s)$ \hspace{1cm} (A1)

where

$\Phi_j(s) = \int_0^\infty e^{-st} dF_j(t)$ and

$\Theta_j(s) = \int_0^\infty e^{-st} dG_j(t)$.

Taking the first derivative we get

$\Phi'_j(s) = (1 - \sum_{k \in T} p_{jk} \Theta'_j(s))$

$+ \sum_{k \in T} p_{jk} [\Theta_j(s) \Phi'_k(s) + \Theta'_j(s) \Phi_k(s)]$

-95-
Evaluating at $s = 0$ and changing sign, we get

$$m_j = (1 - \sum_{k \in T} p_{jk}) t_j + \sum_{k \in T} p_{jk} (m_k + t_j)$$

which reduces to the well-known formula

$$m_j = t_j + \sum_{k \in T} p_{jk} m_k.$$

To obtain the second moment we must take the second derivative of equation (A1):

$$\phi''_j (s) = (1 - \sum_{k \in T} p_{jk}) \phi''_j (s) + \sum_{k \in T} p_{jk} [\theta_j (s) \phi''_k (s) + 2 \theta'_j (s) \phi'_k (s) + \theta''_j (s) \phi_k (s)]$$

Evaluating this at $s = 0$ gives

$$m_j^{(2)} = (1 - \sum_{k \in T} p_{jk}) t_j^{(2)} + \sum_{k \in T} p_{jk} (m_k^{(2)} + 2 t_j m_k + t_j^{(2)})$$

this reduces to the desired formula

$$m_j^{(2)} = t_j^{(2)} + 2 t_j \sum_{k \in T} p_{jk} m_k + \sum_{k \in T} p_{jk} m_k^{(2)}$$

which in turn gives

$$m_j^{(2)} = 2 t_j m_j + \sum_{k \in T} p_{jk} m_k^{(2)}$$