ACCESS SIZE AND ORDER IN QUEUEING SYSTEMS

by

Robert N. Ayres

Thesis submitted to the Graduate Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in
Electrical Engineering

APPROVED:

R. A. Thompson, Chairman

A. W. Bennett

F. G. Gray

May, 1974

Blacksburg, Virginia

ACKNOWLEDGEMENTS

The author wishes to express his thanks to

who gave me a starting point for this thesis. I also wish to express my sincere thanks and gratitude to who was generous with both his time and thoughts in assisting me with this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

1.1 The General Queueing System

The purpose of this thesis is to examine problems associated with server access in queueing systems. In general, a queueing system consists of customers, servers, and accessors. The purpose of an accessor is to find an idle server and provide it to a customer requesting service. (See Fig. 1.1.1)

At this point several important concepts are introduced.

Definition: Access size--the maximum number of servers over which an accessor may search for an idle server. Access size will be represented by m.

Definition: Access group--the collection of accessors which search over the same servers in the same order.

1.2 The Problem

The problem consists of two parts. The first is: given an access size, determine an optimal access arrangement in terms of total system cost by altering the access order. This problem is addressed in Chapter 2. The second is to determine this optimum access size. This is addressed in Chapter 3.

1.3 Basic Concepts and Notation

The reader is assumed familiar with the basics of

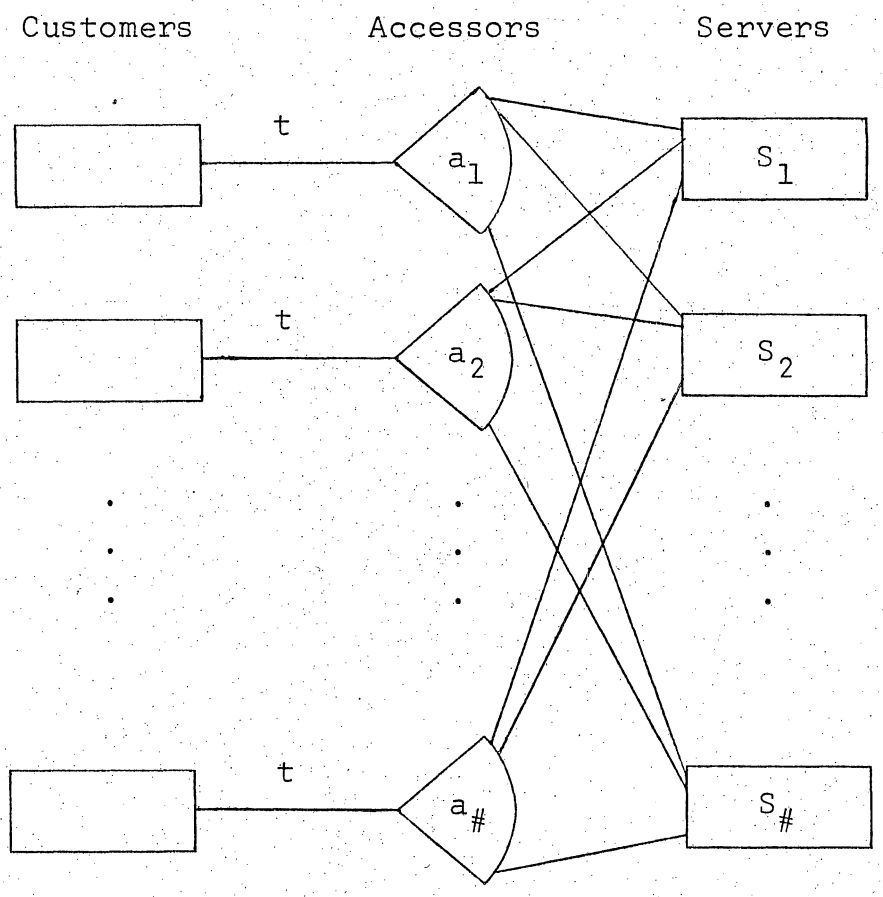


Figure 1.1.1 General Queueing System

queueing theory [3].

Definition: Offered load--the product of the arrival rate and the mean holding time denoted by t_o.

Definition: Carried load--that part of the offered load which actually receives service denoted by t_c.

Thus t_o is the mean number of arrivals that occur during an average holding time. The units of t are erlangs. Thus, a system with one erlang of traffic carries an average of one customer at any one time.

1.4 Customer Disposition

If an accessor is unable to find an idle server for an arriving customer, then one of three things are assumed to happen to the blocked customer. In the blocked customer cleared (BCC) disposition, the customer is removed from the system without receiving service. In the blocked customer held (BCH) disposition, the customer is held for some period of time independent of whether or not the customer receives service. The third disposition is blocked customers delayed (BCD). In it, customers are delayed in the system until a server is available.

The blocking probabilities P_B, for each of these customer dispositions have been calculated and plotted [3]. The following formulas were used in those calculations.

BCC
$$P_B = \frac{t_o^{S_\#}/S_\#!}{\sum_{K=0}^{S_\#} (t_o^K/K!)} \qquad 1.4.1$$

where t_o = offered traffic

$S_\#$ = number of servers

P_B = probability of blocking

BCD
$$P_B = \frac{t_o^{S_\#}/[(S_\#-1)!(S_\#-t_o)]}{\sum_{K=0}^{S_\#-1} (t_o^K/K!) + t_o^{S_\#}/[(S_\#-1)!(S_\#-t_o)]} \qquad 1.4.2$$

BCH
$$P_B = \sum_{K=S_\#}^{\infty} \frac{t_o^K}{K!} e^{-t_o} \qquad 1.4.3$$

These three formulae are known respectively as the Erlang loss formula, Erlang delay formula, and the Poisson formula. For small blocking probabilities, BCD, BCC and BCH are almost identical. This is because the few calls that are blocked will make little difference.

II. ACCESS ORDER

2.1 Introduction

This chapter focuses on the problem of obtaining the maximum average load per server with a given access size. Four methods will be investigated. The first is straight multiples. Second is graded multiples as developed by Molina and Wilkinson [1]. The third is slipped multiples, and the fourth is random multiples.

2.2 Straight Multiples

This basic method assumes that the servers are divided into groups of size m, where m is the access size; and that accessors in the same group search for idle servers in the same order. Each such group of servers is given enough servers so that the maximum acceptable blocking probability is just barely reached.

Example: Suppose m = 10, P_B = .01

From the Erlang loss curves (BCC), it can be determined that 4.5 erlangs of traffic can be handled at the .01 blocking probability.

At this point it is convenient to define server occupancy as load carried per server.

$$p = t_c/S_\# \qquad 2.2.1$$

This will be used to determine the relative efficiencies of different access arrangements for the same access size.

For the above example $p = \frac{(1 - .01)\ 4.5}{10} = .445$ erlangs/ server. Obviously if this figure can be increased while maintaining the required grade of service, fewer servers will be needed.

2.3 Graded Multiples

This access arrangement, discussed in detail in references [1] and [2], is dependent on the fact that the last several servers in an access group are under-utilized and attempts to share these servers with other access groups. In this arrangement, the access size is x + y. There are y servers common to all access groups and x servers dedicated to each access group. There are g groups, so the total number of servers is g x + y. The traffic is assumed to be divided equally among the g groups.

Curves published by Wilkinson in 1933 [1] give figures for percentage increase in server occupancy over straight multiples of the same access size. Unfortunately, in actual trials which were conducted, this full gain was not achieved. In fact, only half the predicted gain was attained in actual practice. This is due to the fact that Molina assumed "No-holes-in-the-multiple." In other words, customers were transferred from shared servers to dedicated servers as soon as a dedicated server became available. This was a poor assumption and resulted in over-optimistic results.

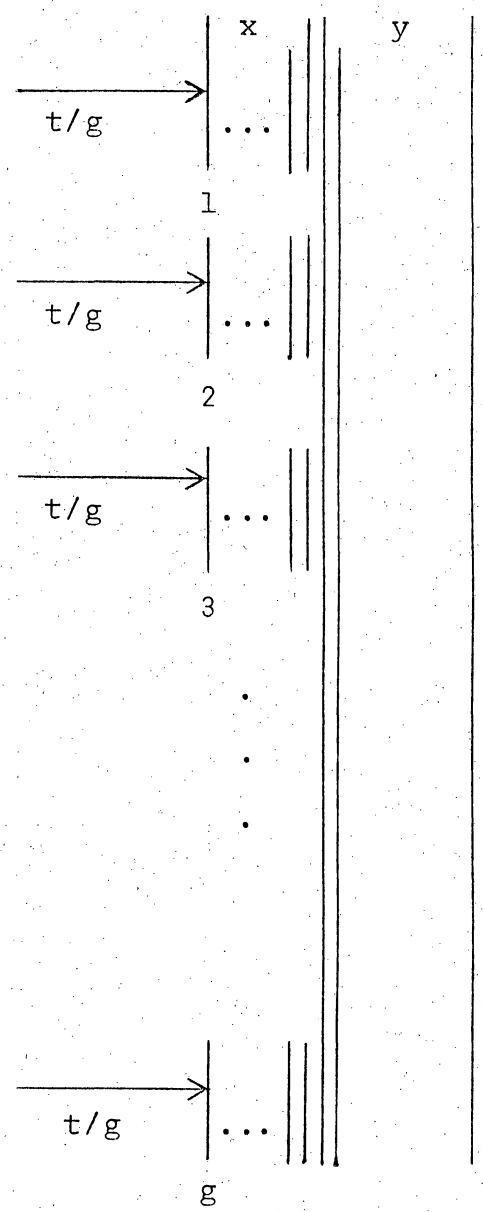


Figure 2.3.1 Generalized Graded Multiple Access Arrangement

This is still a considerable improvement over straight multipling. Unfortunately, this gain is also dependent on the total number of groups which share servers.

Example: Let $x + y = 10$, $y = x = 5$, $g = 4$, $P_B = .01$

Wilkinson's curves predict a half gain increase of 13% over the .445 erlangs/server found in the previous section for straight multipling. This 13% gain gives an occupancy of .504 erlangs/server. In order to check his results a computer simulation was made. This program is the same one used in the next section to generate slipped multiple curves. The simulation supported Wilkinson's results with a server occupancy of .502 erlangs/server.

2.4 Slipped Multiples

This access arrangement provides for the traffic load to be initially spread equally over all servers. Basically it works as follows.

If there are $a_{\#}$ accessors, $S_{\#}$ servers, and access size is m,

Accessor group 1 searches servers 1 through m

Accessor group 2 searches servers 2 through m + 1

.
.
.

$S_{\#}$ - m through $S_{\#}$

$S_{\#}$ - m + 1 through $S_{\#}$

and 1

$S_{\#}$ - m + 2 through $S_{\#}$ and
1 through 2

.
.
.

Accessor group $S_{\#}$ searches servers $S_{\#}$ and 1 through m - 1

Thus, there are always $S_{\#}$ groups with the accessors divided equally among the accessor groups. So, each server has first priority for one access group, second priority for another, . . . and m^{th} priority for the last. The only requirement is that the total number of servers is at least m + 1 (or m trivially). This is necessary in order to have the slipped multiple arrangement.

In order to determine blocking probabilities for this access arrangement, a computer simulation program was written and run for various access sizes and traffic volumes. A flow chart for this program is shown in Figure 2.4.1.

The program was written with the assumption that the customer arrivals were Poisson in nature. The probability of an arrival was assumed to be p = .05 and the number of arrivals simulated for each traffic level was n = 10,000. Using Theorem 6.3 [5] we can be (1 - a) 100% confident that the error will be less than $Z \sqrt{p\ q/n}$.

where p = probability of an arrival

q = 1 - p

n = number of samples = 20 x 10,000

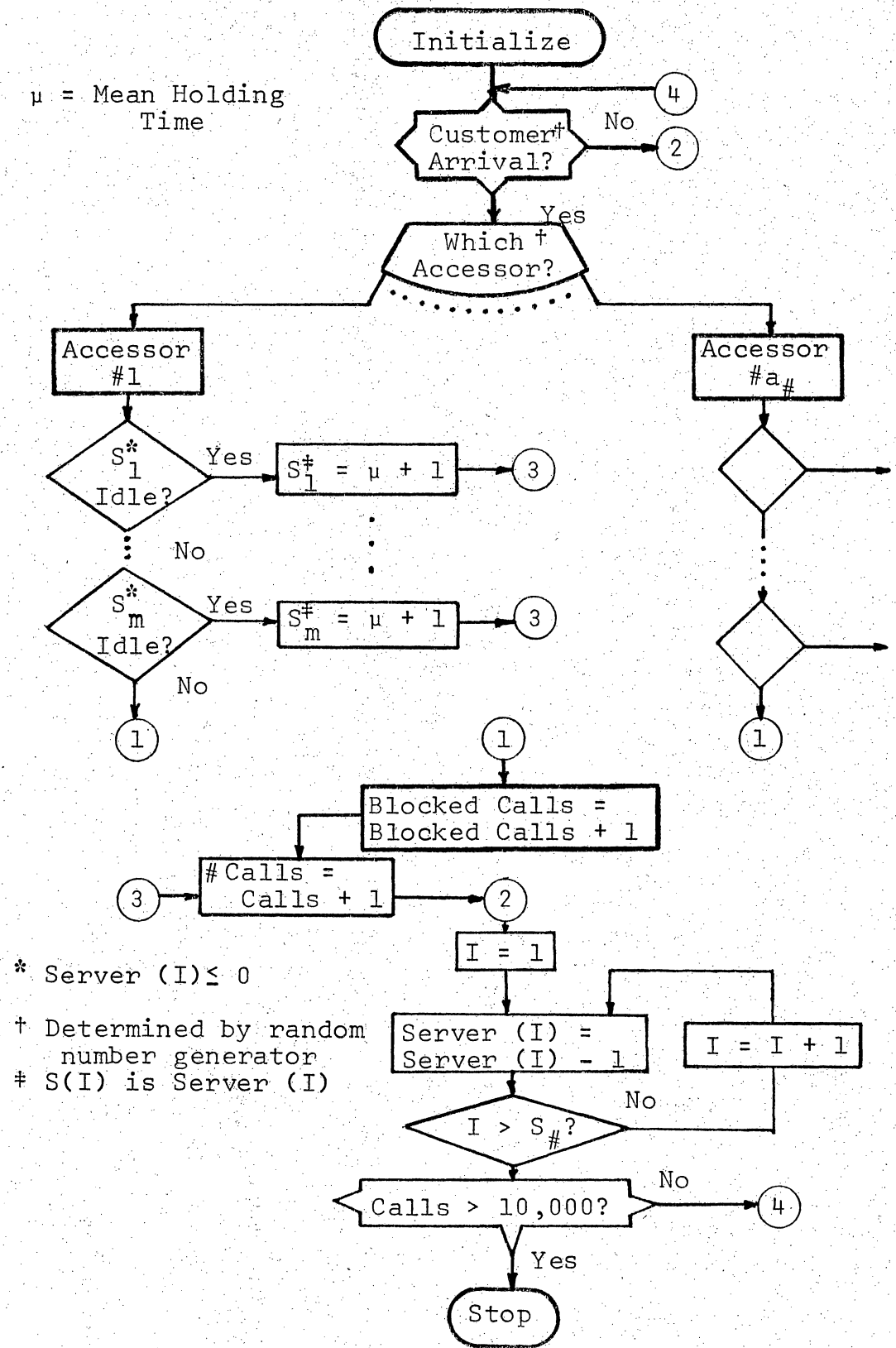


Figure 2.4.1 Simulation Flow Chart

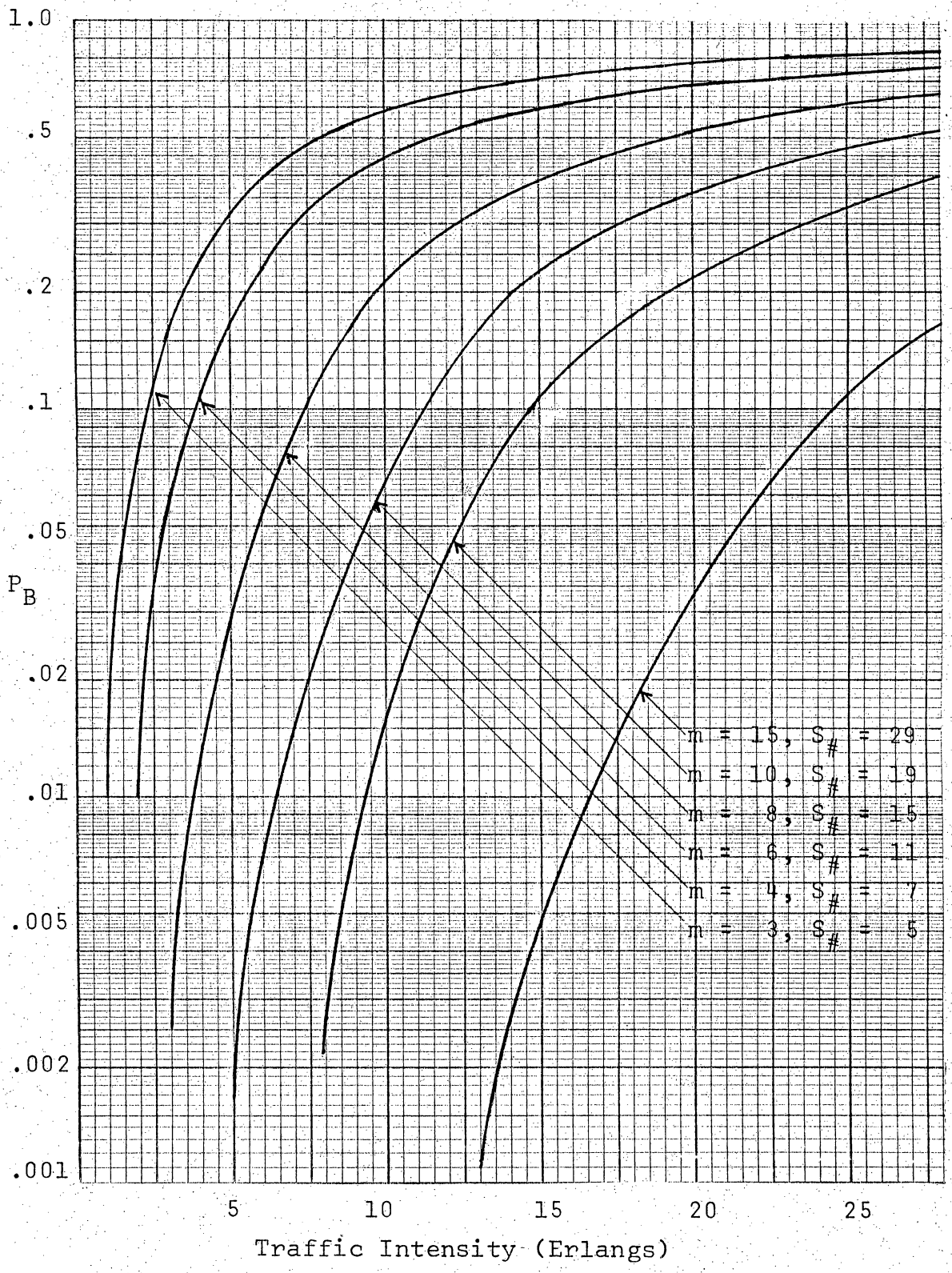


Figure 2.4.2 Slipped Multiple Blocking Probabilities

Z = value of the standard normal curve leaving an area of a/2 to the right

Doing the calculations, we find that we can be 99% sure that the error of p was p ± .00125 or ± 2.5%. In addition to this fact, samples are taken for different traffic volumes and plotted together in Figure 2.4.2.

Continuing with the example described in the previous two sections, for an access size m = 10 and P_B = .01, the new curves predict that if 19 servers are available, then 9.5 erlangs of traffic can be carried. Thus P = 9.5/19 = .5 erlangs/server.

2.5 Random Access Multiples

Unfortunately, with slipped multiples, once a server is accessed and in use there is an unbalancing effect on the rest of the servers. That is, if the K^{th} server is in use, the $K + 1^{th}$ server is more likely to be accessed and the $K - 1^{th}$ server becomes less likely than average to be accessed. This can be overcome by using random access multipling. In this, the accessor hunts for an idle server by randomly choosing servers.

This method may or may not be practical. If the accessor were a software subroutine, it would simply generate a random number to determine which server to examine next. In a hardware case, if it were a phone system, then the wiring could be randomly connected among the servers.

This would be unsatisfactory in that any randomly favored servers would always be favored.

III. OPTIMUM ACCESS SIZE

3.1 Introduction

In the first chapter, three customer dispositions were discussed. Regardless of which assumption is made, examination of the curves generated by the two Erlang formulae and the Poisson formula reveals that decreasing access size causes the required number of servers to increase in order to carry the same amount of traffic at the required grade of service.

Example: Consider a queueing system with m = 12, P_B = .01 and BCH disposition is used. The Poisson curves allow 5.4 erlangs to be carried by this system. However, if m is reduced to 6, then a group of 6 servers can carry only 1.8 erlangs of traffic. So, to carry the full system load of 5.4 erlangs, 3 groups of 6 servers must be provided for a total of 18 servers. This is an increase by 50% in the number of servers over the original system. It is important to note that this does not necessarily lead to an increase in the total system cost because of the decrease in accessor cost and the possibly large number of accessors.

The purpose of this chapter is to find the optimal access size with regard to total system cost.

3.2 Cost of Access

Definition: Cost of access--is that cost associated

with providing a customer with access to servers.

In this thesis the cost of access will be assumed to consist of two parts. The first is a fixed cost associated with providing access. The second is a variable cost which increases linearly with the size of access. So, the total cost of access would be:

$$C_a = f + m\,V \qquad 3.2.1$$

where C_a = cost of access

f = fixed cost

V = variable cost per server searched

m = access size

This assumption seems reasonable when one considers that there would be an initial cost for providing access plus an incremental cost associated with each additional server over which the accessor can search.

If the accessor were some type of software subroutine, then there would be an initial cost associated with setting up the search program plus a time cost for each server it must examine. If the accessor is some type of hardware device such as a Strowger switch in a Step-by-Step phone system, then there is the basic cost of the switch plus a set of contacts and wiring for each server over which the switch searches.

3.3 Normalized Cost per Erlang--Straight Multiples

In a queueing system the total cost is

$$C = A_{\#}\, A_{\$} + S_{\#}\, S_{\$} \qquad 3.3.1$$

where $A_{\#}$ = total number of accessors

$A_{\$}$ = cost per accessor

$S_{\#}$ = total number of servers

$S_{\$}$ = cost per server

This cost function is to be minimized.

Let there be $a_{\#}$ accessors in an access group, each carrying t erlangs of traffic. Then if there are n access groups,

$$A_{\#} = n\, a_{\#}$$

and total system traffic

$$T_S = n\, T_g = t\, n\, a_{\#}$$

where T_g = traffic in an access group

and $S_{\#} = n\, m$

Substituting into 3.3.1, the following is obtained

$$C = (a_{\#}\, n)(f_{\$} + m\, V_{\$}) + S_{\$}\, (n\, m)$$

Dividing by $S_{\$}$ to normalize all costs with respect to server costs

$$C/S_{\$} = (a_{\#}\, n)(f_{\$}/S_{\$} + m\, V_{\$}/S_{\$}) + (n\, m)$$

or letting $\hat{C} = C/S_{\$}$, $\hat{f}_{\$} = f_{\$}/S_{\$}$, and $\hat{V}_{\$} = V_{\$}/S_{\$}$

$$\hat{C} = (a_{\#}\, n)(\hat{f}_{\$} + m\, \hat{V}_{\$}) + (n\, m).$$

Then dividing by T_S

$$\hat{C}/T_S = (\hat{f}_{\$}/t) + (m\, \hat{V}_{\$}/t) + (m/a_{\#}\, t) \qquad 3.3.2$$

This formula gives a normalized cost per erlang of traffic. This formula is useful in a practical sense because none of the variables are dependent on system size. In fact, $\hat{f}_{\$}$ and $\hat{V}_{\$}$ are a function of design and could probably be approximated before actual designing. Also, in a given problem the specifications would determine values for t and P_B. Since $a_{\#}$ is a function of t, P_B and m; $a_{\#}$ would be chosen as large as possible so that $a_{\#}$ times t will be just less than the volume of traffic which could be carried by m servers. Thus, $a_{\#}$ could be chosen for different m from the Poisson or Erlang curves discussed in the first chapter.

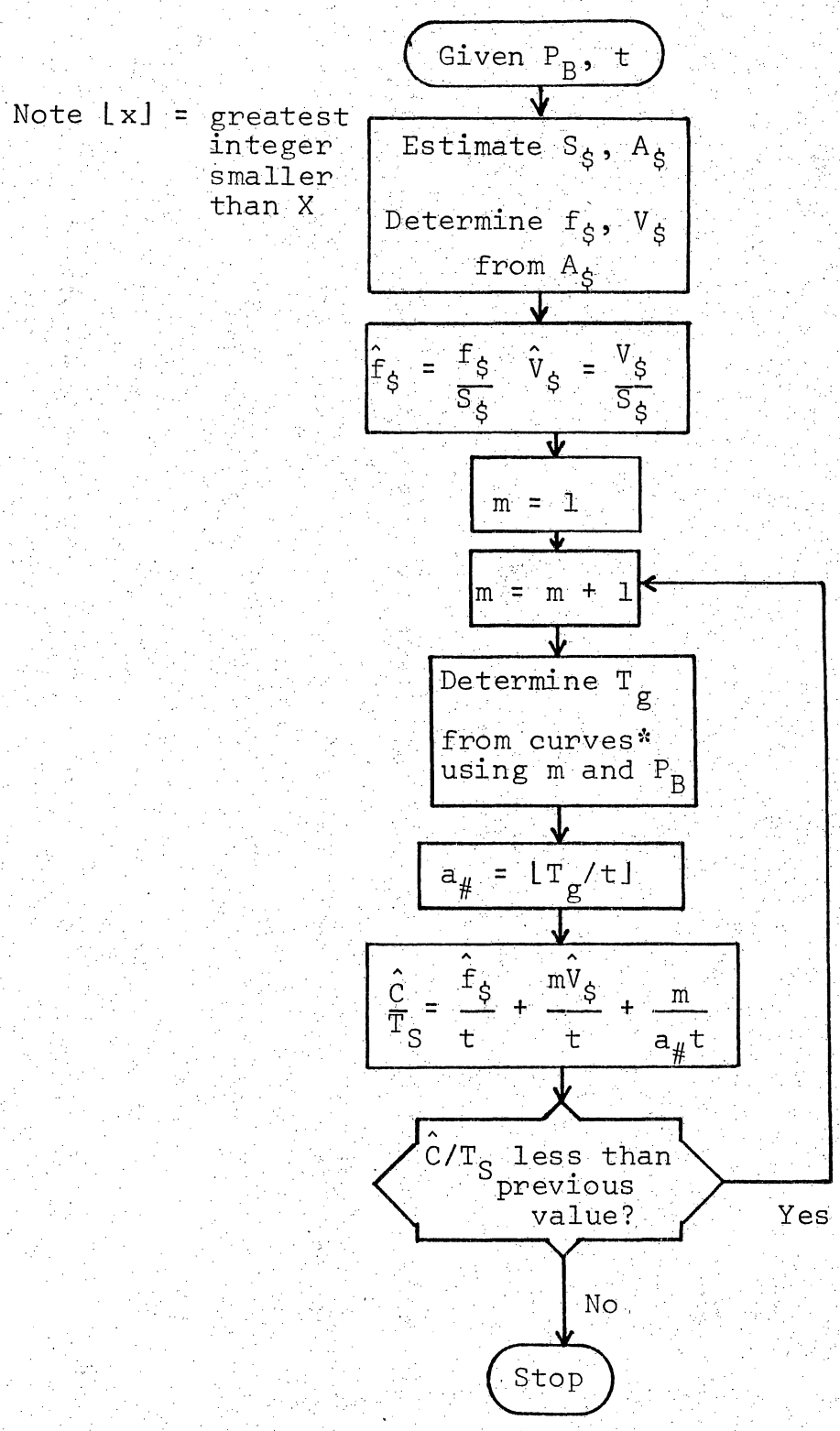
3.4 Cost Per Erlang Curves--Straight Multiples

The algorithm on the following page calculates values for $\hat{C}/T_S$ curves. On the next page are $\hat{C}/T_S$ curves plotted for arbitrarily chosen values of P_B, t, $\hat{f}_{\$}$ and $\hat{V}_{\$}$.

Since it is observed that the curves are unimodal, the algorithm may be terminated as soon as the values of $\hat{C}/T_S$ begin to increase. Note that only integer values of m are valid. The minimum point on the curve is the optimum access size for each set of constraints.

3.5 Observations--Straight Multiples

Equation 3.3.2 can be broken into three parts. The first term $\hat{f}_{\$}/t$ is a constant with respect to m and does



*Poisson or Erlang as appropriate

Figure 3.4.1 Cost per Erlang Algorithm--Straight Multiples

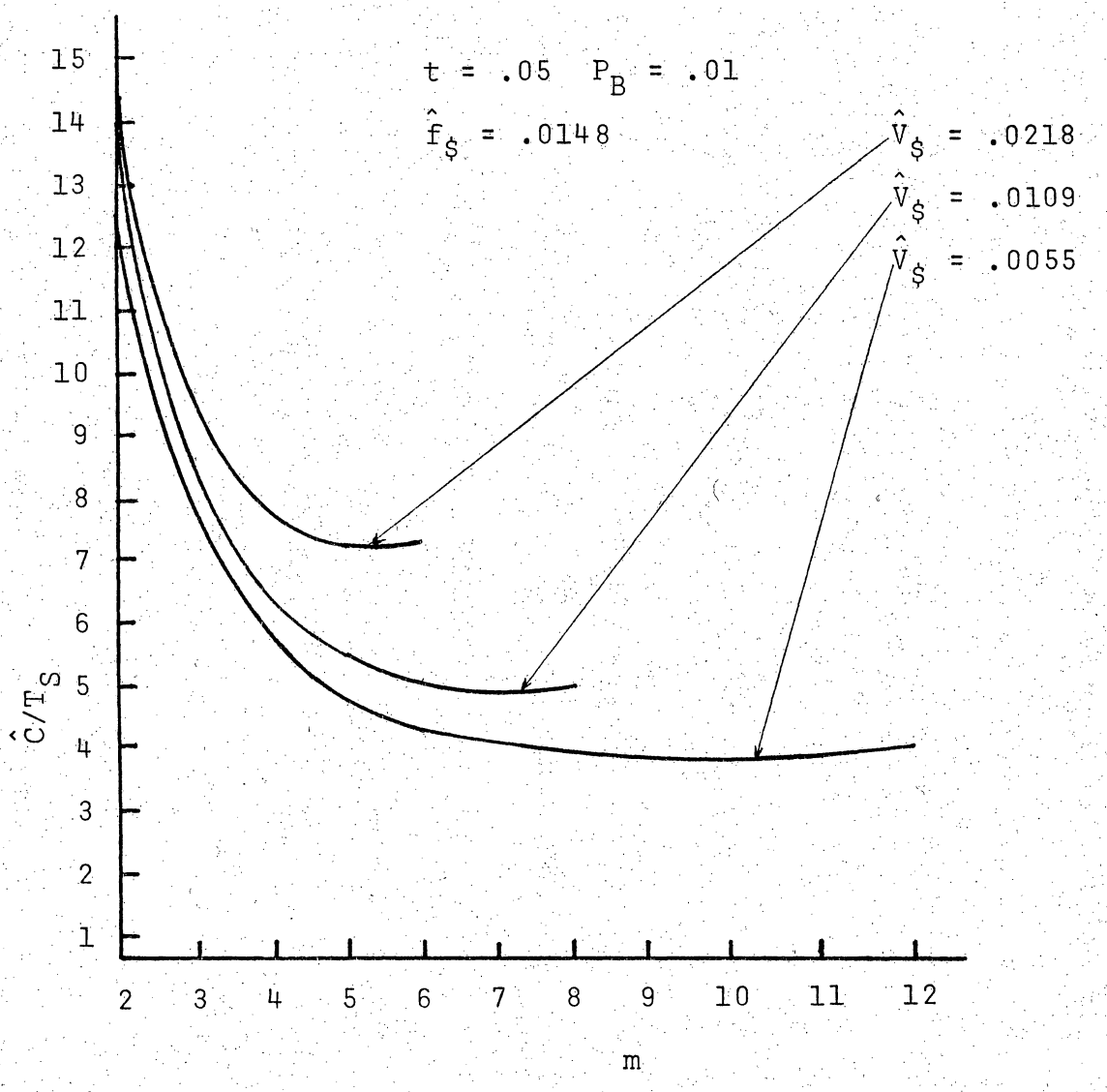


Figure 3.4.2 Sample $\hat{C}/T_S$ Curves

not affect the location of the minimum. The second term is a linearly increasing term while the third term is a non-linearly decreasing term which decreases rapidly at first and then levels out. So, for small m, the third term decreases much more rapidly than the second term increases and dominates the $\hat{C}/T_S$ function. As m increases, the third term remains relatively constant and the linearly increasing second term dominates. Thus, the curve is unimodal.

For smaller $\hat{V}_\$$, the minimum will occur for a smaller m due to the fact that the third term dominates for smaller m. Also, for either larger P_B or smaller t, $\hat{C}/T_S$ will have smaller values for small m and larger values for large m. This is due to the fact that the third term drops off more quickly due to increased blocking probability or decreased traffic.

It is observed that when log m is plotted versus $\hat{C}/T_S$, the resulting curve appears to be approximately parabolic. Further work in this area is indicated.

3.6 Normalized Cost per Erlang--Slipped Multiples

Starting with equation 3.3.1

$$C = A_\# A_\$ + S_\# S_\$$$

and observing that $A_\# = n\, a_\#$

$$C = (a_\# n)(f_\$ + m V_\$) + S_\$ S_\#$$

$$\hat{C} = (a_\# n)(\hat{f}_\$ + m \hat{V}_\$) + S_\#$$

but $S_{\#} = n$

$$\hat{C} = (a_{\#}\ n)(\hat{f}_{\$} + m\ \hat{V}_{\$}) + n$$

and since $T_S = n\ t\ a_{\#}$

$$\hat{C}/T_S = \hat{f}_{\$}/t + m\ \hat{V}_{\$}/t + 1/t\ a_{\#}$$

The algorithm for slipped multiple cost per erlang curves is similar to that for straight multiples.

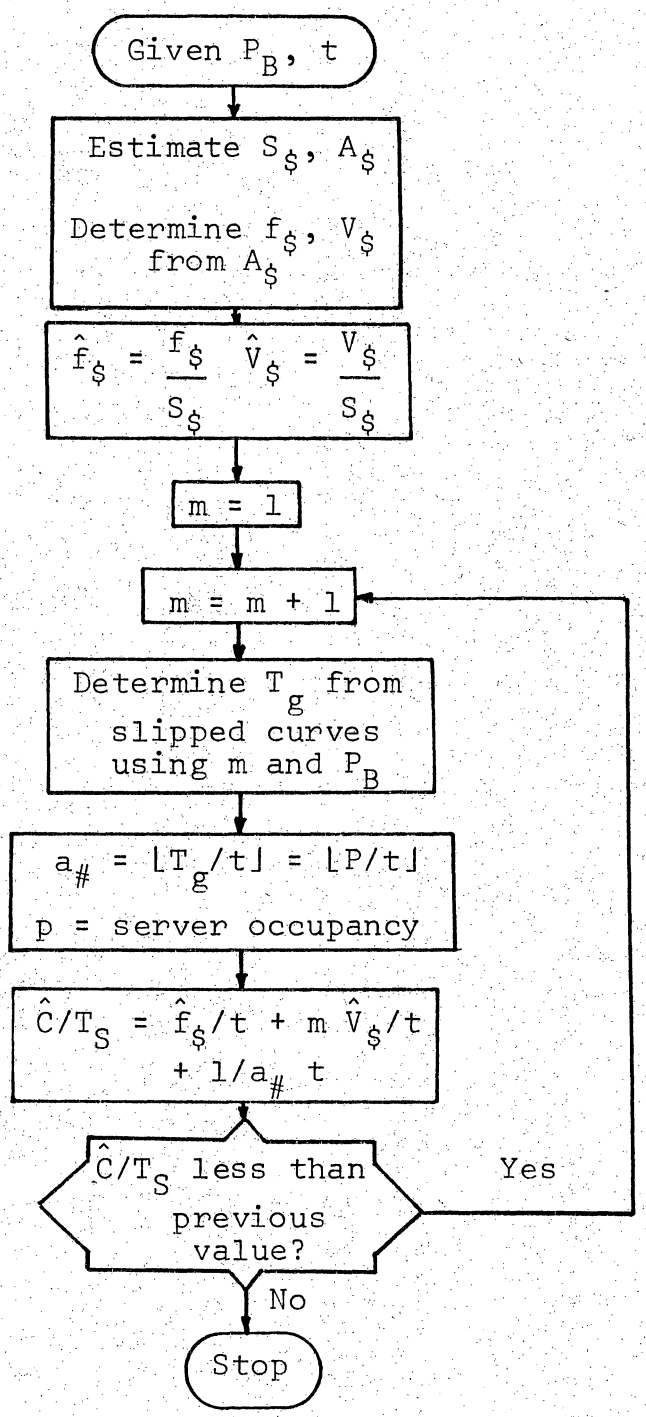


Figure 3.6.1 Cost per Erlang Algorithm--Slipped Multiples.

IV. EXAMPLE--PROGRESSIVE TELEPHONE SWITCHING SYSTEM

Consider a portion of a progressive telephone system. Associated with each line circuit there is a hunter of access size m. It searches for an idle link circuit when a caller offhooks. Preliminary studies show that each link circuit will cost \$127.00. They also show a fixed cost of \$2.00 and a variable cost of \$1.11 associated with the hunters. Let each line circuit originate .05 erlangs of traffic and have a blocking probability of .01 for this stage of the telephone system.

So,

$$\hat{f}_{\$} = 2/127 = .0157$$

$$\hat{V}_{\$} = 1.11/127 = .0087$$

$$\hat{C}/T_S = \hat{f}_{\$}/t + m\,\hat{V}_{\$}/t + 1/t\; a_{\#}$$

if m = 3, 1 erlang of traffic can be carried by 5 servers

$$p = 1/5 = .2$$

1 server can carry $\lfloor .2/.05 \rfloor = 4$ lines $= a_{\#}$

$$\hat{C}/T_S = (.0157)/(.05) + 3(.0087)/(.05) + 1/(.05)(4)$$
$$= 5.836$$

m = 4, 2 erlangs of traffic can be carried by 7 servers

$$p = 2/7 = .286$$

1 server can carry $\lfloor .286/.05 \rfloor = 5$ lines $= a_{\#}$

$$\hat{C}/T_S = 5.010$$

m = 6, 3.9 erlangs of traffic can be carried by 13 servers

$p = 3.9/11 = .35$

1 server can carry $\lfloor .35/.05 \rfloor = 7$ lines $= a_{\#}$

$\hat{C}/T_S = 4.218$

m = 8, 6.75 erlangs can be carried by 15 servers

$p = 6.75/15 = .45$

1 server can carry $\lfloor .45/.05 \rfloor = 9$ lines $= a_{\#}$

$\hat{C}/T_S = 3.926$

m = 10, 9.5 erlangs can be carried by 19 servers

$p = 9.5/19 = .5$

1 server can carry $\lfloor .5/.05 \rfloor = 10$ lines $= a_{\#}$

$\hat{C}/T_S = 4.054$

m = 15, 16.8 erlangs can be carried by 29 servers

$p = 16.8/29 = .58$

1 server can carry $\lfloor .58/.05 \rfloor = 11$ lines $= a_{\#}$

$\hat{C}/T_S = 4.744$

These values are plotted in figure 4.1.1.

If one wishes to determine actual cost of a 100 line telephone system, $T_S = 100(.05) = 5$ erlangs

Real cost = (5)(127)(3.926) = $2500

This is the cost of 100 hunters plus 11 link circuits. This can be done for any access size and for any system size.

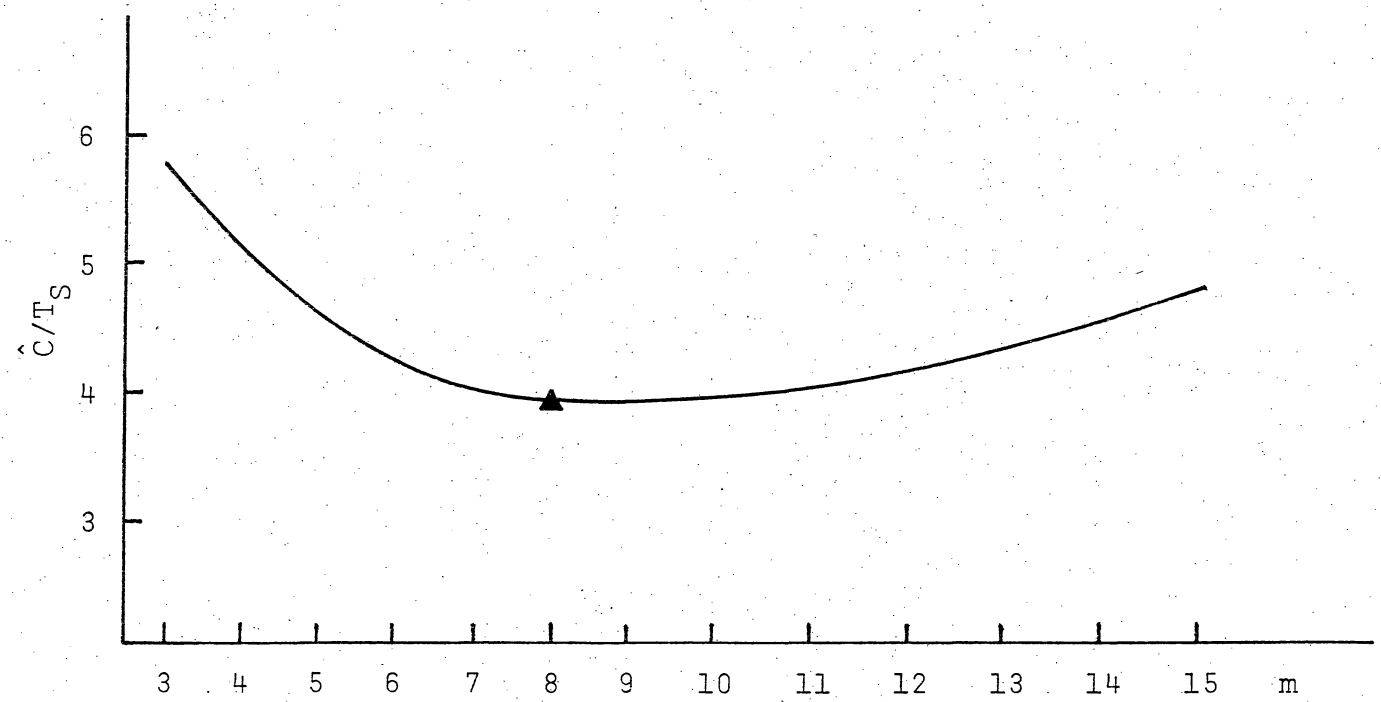


Figure 4.1.1 Normalized Cost Per Erlang--Example

BIBLIOGRAPHY

1. Wilkinson, R. I., "The Interconnection of Telephone Systems--Graded Multiples," Bell System Technical Journal, 10, No. 4, pp. 531-564, October 1931.

2. Buchner, M. M. Jr., and S. R. Neal, "Inherent Load-Balancing in Step-By-Step Switching Systems," Bell System Technical Journal, 50, No. 1, pp. 135-165, January 1971.

3. Cooper, R. B., Introduction to Queueing Theory. The Macmillan Company, New York, 1972.

4. Saaty, T. L., Elements of Queueing Theory with Applications. McGraw-Hill, New York, 1961.

5. Walpole, R. E., and R. H. Myers, Probability and Statistics for Engineers and Scientists. The Macmillan Company, New York, 1972.

6. Breipohl, A. M., Probabilistic Systems Analysis. John Wiley and Sons, New York, 1970.

7. Morse, P. M., Queues, Inventories and Maintenance. Publications in Operations Research No. 1, John Wiley and Sons, New York, 1958.

8. Haight, F. A., Handbook of the Poisson Distribution. Publications in Operations Research No. 11, John Wiley and Sons, New York, 1967.

9. Parzen, E., Modern Probability Theory and Its Applications. John Wiley and Sons, New York, 1960.

10. Talley, D., Basic Telephone Switching Systems. Hayden Book Company, New York, 1969.

11. Ley, B. J., Computer Aided Analysis and Design for Electrical Engineers. Holt, Rinehart and Winston, New York, 1970.

12. McCraken, D. D., A Guide to Fortran IV Programming. John Wiley and Sons, New York, 1965.

The vita has been removed from the scanned document

ACCESS SIZE AND ORDER IN QUEUEING SYSTEMS

by

Robert N. Ayres

(ABSTRACT)

General queueing systems are discussed in order to familiarize the reader with the problem. Various access arrangements including straight, graded, slipped, and random access multiples are discussed and compared. Blocking probability curves for slipped multiple access order are generated. Cost of access is discussed. Cost per erlang curves, from which optimum access size is determined, are generated. Algorithms for generating these curves are developed. A stage in a progressive telephone switching system is designed as an example.