PRICE SCHEDULING IN A TIME-SHARING

QUEUEING SYSTEM†

Jair Moshe Babad

TR 73 - 178

August 1973

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University for the Degree of

Doctor of Philosophy

Department of Computer Science
Cornell University
Ithaca, New York 14850

†This research was supported in part by the National Science Foundation Grant GJ-21184 and GJ-28176.

PRICE SCHEDULING IN A TIME-SHARING QUEUEING SYSTEM

Jair Moshe Babad, Ph.D.
Cornell University, 1973

This thesis deals with a time-sharing system in which users may join after paying an appropriate toll and are subjected to service and waiting charges. The form of an optimal joining policy that minimizes the expected loss of users who may join the system at some of its queues is derived, and is shown to be a control-limit policy with a single control number for every possible entry queue; a newly arriving user will join the minimal-priority entry queue that is not filled up to its control number. Tolls and charges that maximize the average expected revenue of the system, as well as the control numbers, are determined for a round-robin time-sharing system. A multi-entrance foreground-background time-sharing system with random entrance is analyzed and the characteristics of the system, such as expected waiting times and expected number of waiting users in each queue, are given. It is shown that when the entry into this system is based on tolls and charges, there exists a set of prices (and associated control numbers) that maximizes the discounted expected revenues of the system.

BIOGRAPHICAL SKETCH

The author was born in Gedera, Israel, on January 12, 1941. He entered the Hebrew University in Jerusalem in 1958, and after serving in the Israeli Army obtained a B.Sc. degree with a major in Mathematics in 1965. The same year he enrolled in an Actuarial Program at the Hebrew University in Jerusalem, and received in 1968 an Actuarial Diplome. During the years 1966-69 the author worked in Israel as an actuary of a pension fund and as a teaching assistant in the Statistics Department at Tel-Aviv University. The author came to the United Sates in 1969 to pursue a graduate career in Operations Research at Cornell University, where he obtained a M.S. degree in Operations Research in May 1972.

The author is a member of the Operations Research Society of America, the Institute of Management Science, the Association for Computing Machinery and the Institute of Mathematical Statistics.

To

Noami, Galit and Orna

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor W. L. Maxwell, my thesis advisor and Special Committee Chairman, for his unfailing assistance, encouragement and interest throughout this research. I am equally indebted to Professor S. Stidham, Jr., whose commitment to the success of this research was most beneficial and greatly appreciated. I am thankful to Professors H. M. Taylor, 3d and R. W. Conway who are presently serving as members of my Special Committee, and to Professors J. C. Kiefer, L. I. Weiss and W. F. Lucas who served in the past on my Special Committee.

For their fine joint effort in my education, I express my gratitude and appreciation to Professor R. E. Bechhofer and the other members of the Departments of Operations Research and Computer Science at Cornell University.

During my first year at Cornell I was supported by a Lanchester Fellowship; during my second through fourth year I was supported by a National Science Foundation grant No. GP-21184 and by a National Science Foundation grant No. GS-28176. This financial aid is truly appreciated.

Finally, I would like to thank Mrs. Deborah Cagey for her patience and diligence in the typing of this thesis.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.a. Problem Description and Objectives

During the last decade interest in time-sharing has grown at an increasing rate. The relative slowness of the human thinking process and responses, compared to the logical and arithmetical capabilities of modern computers, has motivated the sharing of computers' time and resources among users. While this time-sharing process is intended to appear to the human user as concurrent usage of the computer by many users, in reality it is handled through time-slicing -- a process in which each user's time requirements are broken into segments which are interleaved with the segments of other users. During a given segment, only one user has exclusive usage of the computer. Time-sharing has several advantages from the point of view of the user. During program debugging and editing, turn-around time can be essentially instantaneous, and productivity is enhanced by providing a large degree of interaction between the user and his programs and work. Due to the management and scheduling operations of the time-slicing mechanism the overheads which are associated with time-sharing are, however, usually larger than in other computer operating systems.

In a time-sharing system, the sharing of computer resources requires a priority rule, which determines the next user to be served. Greenberger [11] describes three different and usually conflicting objectives that can influence a priority decision and affect the design of a priority rule. They are:

a. Reduce average response time, and number of users waiting for response,

b. Acknowledge user importance and the urgency of his request,

c. Serve in fair order and limit length of wait.

Usually, such priority rules aim at fulfilling the first and third objectives stated above, through a scheme which maintains one or more queues of decreasing priority, and which transfers a user whose time slice was completed to the end of one of these queues -- usually to a lower priority queue than the one he was in before getting service. New users are usually scheduled to the end of a queue specified by the system. The queues are served in accordance with their priority, and a lower priority queue is not examined as long as higher ones are not empty. Each queue is usually serviced in FIFO order. Sometimes, a compensating feature is employed so that a lower priority user returns to a higher priority queue if he has waited for a long time without getting service or finishing his service requirements.

Such schemes are fair, in the sense that user's response time is directly related to his service requirements; at the same time, short requests receive enhanced service at the expense of degraded service for longer requests. Thus importance and urgency is, in effect, assumed to be equivalent to a short service requirement.

As Singer [22] notes, "the difficulty with rules of this sort is that an implicit assumption must be made about the value placed on computer time by each user...they are equivalent to prices under the special assumption that all users place the same values on computer time and experience the same cost of waiting. These conditions are of course most unlikely to be satisfied in practice..."

This work is concerned with a priority scheme that acknowledges the importance and urgency a user associates with his request. The proposed model maintains the usual structure of a time-sharing system, except for the following modifications: A newly arriving user has access to all the information about the state of the system -- the number of users in each queue and the distributions of their remaining processing requirements, the priority of the user being served at the moment of arrival and his residual service's distribution. A cost structure is incorporated into the model: The higher the priority of the initial queue joined, the higher the toll to join it and the higher the cost of waiting in the system. On the other hand, the higher the priority of the initial queue joined, the shorter the expected time in the system. The newly arrived user has to decide which priority queue to join in order to minimize his expected cost which includes the toll, the cost of service and the cost of waiting, while taking into account his urgency. The user may also decide to balk and join a competing system, if his expected cost seems to be too high. It is very important to note that the joining decision is irrevocable; once made, the user joins the queue he has selected and is then subjected to automatic advancement through the system as described earlier. There is no potential for a user whose request is not completely satisfied to take any subsequent action which will change his relative position in the system.

The revenues of such a system consist of the collected tolls and charges for service and time spent in the system by the users, and its expenses consist of equipment purchase and operating costs. Its aim is to maximize the average expected income per time unit. The system can affect the decisions of new users by changing its parameters: tolls, service charge, waiting charges, and the "time slices" which are assigned to the various queues.

The aims of analyzing such a system are to find a characterization and procedures for an optimal joining, or priority purchasing, policy for new users, and optimal parameters for the system. These two aims are interrelated, since system's optimal parameters affect and are affected by the user's behavior.

1.b. Related Research

Time-sharing systems have been extensively studied, from an analytical as well as from an experimental and practical points of view. An excellent survey of analytical study of time-sharing systems is given by McKinney [17]. In most studies a single-entry system is considered, in which all users join the system with the same priority, that might be later changed by the scheduling algorithm of the system. Several studies, though, deal with multi-entrance systems, e.g. Kleinrock [12], Coffman and Kleinrock [8], Wolff [25], Van Den Heever [23] and Adiri [1]. Most of the studies concentrate on the descriptive characteristics of time-sharing systems, and especially on the response time, i.e. the time spent in the system from joining it until all service requirements are fulfilled, because this time may be considered as a measure of the quality of service rendered by the system. For ease of analysis it is usually assumed that the arrival process is a Poisson process; for the same reason many studies also assume exponential service requirements. In all these studies an infinite queue size and infinite population of potential users are assumed, though the case of a finite population of users has been considered, e.g. by Adiri and Avi-Itzhak [4].

In many batch computing facilities priorities are assigned according to a pricing mechanism, whereby the higher the priority chosen by a user, the higher the charges for service and the better the service. Users of such systems choose their priority according to their personal utility function. Several authors have formalized and analyzed aspects of this situation. Kleinrock [13] considers an M/G/1 queueing system in which the priority assigned to a user is determined by the size of a bribe the user pays to the system; his work is extended to a round-robin time-sharing system by Mehta [18]. Marchand [16] introduces a utility function model, in which priorities are assigned with the objective of maximizing a linear combination of individual utility functions subject to constraint on the first and second moments of the waiting time. Greenberger [11] imposes a cost structure on a round-robin time-sharing system in order to determine the optimal quantum size; his work is extended by Rasch [20] who considered an exponential cost function.

Several authors, in addition to the inclusion of a cost structure, allow balking as one of the alternatives to newly arriving users; this alternative is chosen if the cost of joining the system is too high, as compared with the cost of balking. As a result, the capacity of such systems is finite, and the optimal joining policy for newly arriving users may be characterized as a control-limit type policy, in which a user consider for joining only queues that are not filled to a predetermined control value. Such systems are analyzed by Naor [19], Yechiali [26,27], Adiri and Yechiali [5], Adiri [2] and Adler and Naor [6].

In queueing systems with costs, two different objectives of minimizing costs to users may be considered: the individual user's minimization problem, and the welfare of the totality of users as measured by discounted or average expected cost. Balachandran [7] considers a system in which users pay for their waiting time, in addition to a bribe for a position in the queue; he shows that a policy from which no user would like to deviate may differ from a policy that minimizes the average cost of a job. When balking is included as a possible decision for newly arriving users it might be expected that individual users' decisions will differ from the totality of users (social) decision, because it may be advantageous to an individual to join the system, although it increases the average cost of processing a job by the system. That this is indeed the case, was shown for an M/M/1 system by Naor [19], an M/D/1 system by Adler and Naor [6], an GI/M/s by Yechiali [26,27] and for an M/M/s system by Knudsen [14].

1.c. Summary of the Thesis

The proposed model along with notation is presented in Chapter 2. The form of an user-optimal joining policy is investigated in this chapter, and it is shown that under mild restrictions this policy is a stationary non-randomized control-limit type policy; that is, a control number may be associated with each priority queue, and a new user will not join any queue in which the number of waiting users exceeds the control number, nor any lower priority queue. Chapter 3 deals with the special case of a single-entrance round-robin time-sharing system, and with the determination of optimal set of system's parameters for such a system. This chapter contains also a fairly complete treatment of a finite room round-robin M/M/1 queueing system with set-up times, including expected response times and steady-state probabilities of system's states and number of users in the system. The remaining chapters are devoted to the proposed multi-entrance foreground-background time-sharing system. This system is analyzed in Chapter 4 under the assumption of random arrivals; the results, though, are descriptive, and are not used in any optimization problem. Chapter 5 deals with the same system, under the assumption that newly arrived users make their decisions concerning joining the system, according to the prices set by the system. Chapter 6 summarizes the results and analytical work of the thesis, indicates how these results can be used in practical situations, and suggests avenues for further research on the problem.

CHAPTER 2

A MULTI-ENTRANCE, TIME-SHARING QUEUEING MODEL

2.a. Introduction

In this section two existing types of time-sharing systems as well as the proposed system will be described, in order to highlight the similarities and differences between these systems.

The conceptually simplest system is a "round-robin" system, denoted RR, as shown in Figure 2.1.

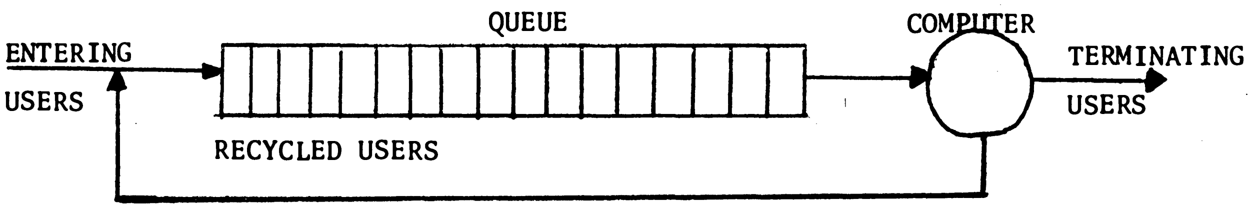

Figure 2.1: A RR Time-Sharing System

In this system, a newly arriving user joins the end of a single queue. This queue is served by a single "service facility" according to a FIFO discipline. The maximum interval of service time which is given to any user is of limited length, and is called the time slice; if the user's request is fulfilled during this time slice he leaves the system. Otherwise, he joins the end of the queue, behind any recycled users and newly arriving users since the most recent time which he entered the queue.

A different type of system is the "foreground-background" model with n queues, denoted FB_n, which is represented schematically in Figure 2.2.

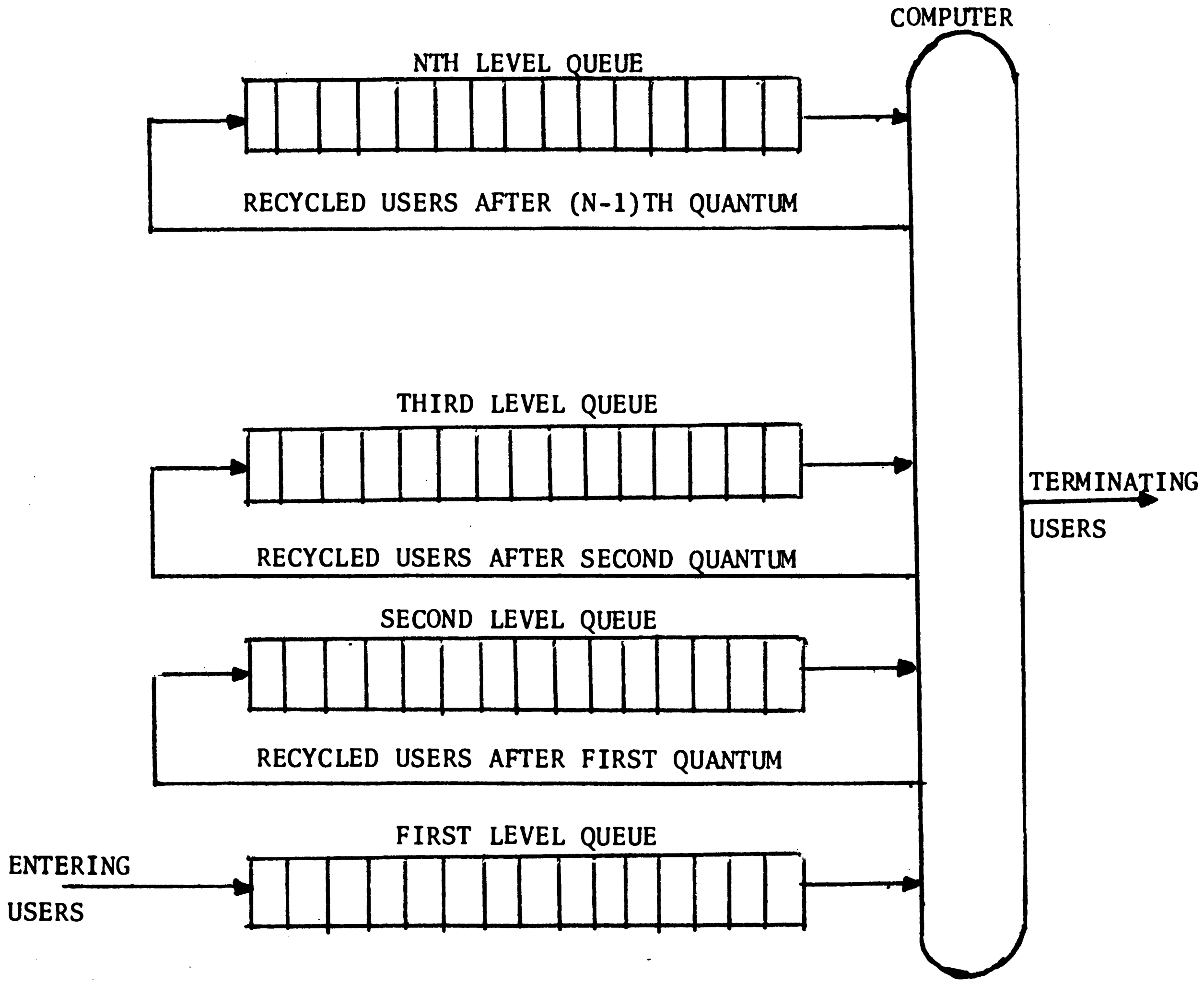

Figure 2.2: A FB_n Time-Sharing System

In this system, a newly arriving user joins the end of the level 1 queue. Associated with each queue is a time slice of limited length, except possibly the nth level queue which is assigned an infinite time slice. When the service facility terminates any time slice it determines the next queue to be served by scanning the queues starting at level 1 and stopping at the lowest level queue which is not empty. The users in each queue are served according to a FIFO discipline. The user at the head of the chosen queue gets the time slice corresponding to his level; if his service requirements are fulfilled during this time slice he leaves the system, otherwise he is transferred at the end of the time slice to the end of the next higher level queue. Systems of this type may differ in the way they treat the last level queue, which might be either a RR queue, or a (simple) queue whose users might or might not be pre-empted by lower level users.

In comparison to a RR system the FB_n system tends to improve the service to "short service requirement" users at the expense of "long service requirement" users. This improvement is also an essential part of the proposed system (Figure 2.3). In addition, the proposed system allows users to choose the "level of service" they want, by allowing them to enter at different levels of the system.

Each user has to pay a toll when he joins the system, and is subjected to service time charges and waiting time charges; the tolls and charges might differ for different possible joining levels. Each user is characterized by his service requirements and his urgency requirements (a function of his impatience and need for fast results). The decision of a user whether to join the system, and if so, at what level, is determined by his service requiremtns, urgency requirements and expected total cost.

Clearly the toll must increase as the priority increases; otherwise some levels would never be used by rational users and could be effectively ignored in any analysis. Thus, a first level user will pay a higher toll to join the system than a second level user, and so on. In addition, there is a preference pattern within each queueing level, in contrast with the FB_n system. In the FB_n system, once a first level user reaches the second level, no distinction is made between him and a user who joined the system at the second level, if such a joining procedure had been allowed.

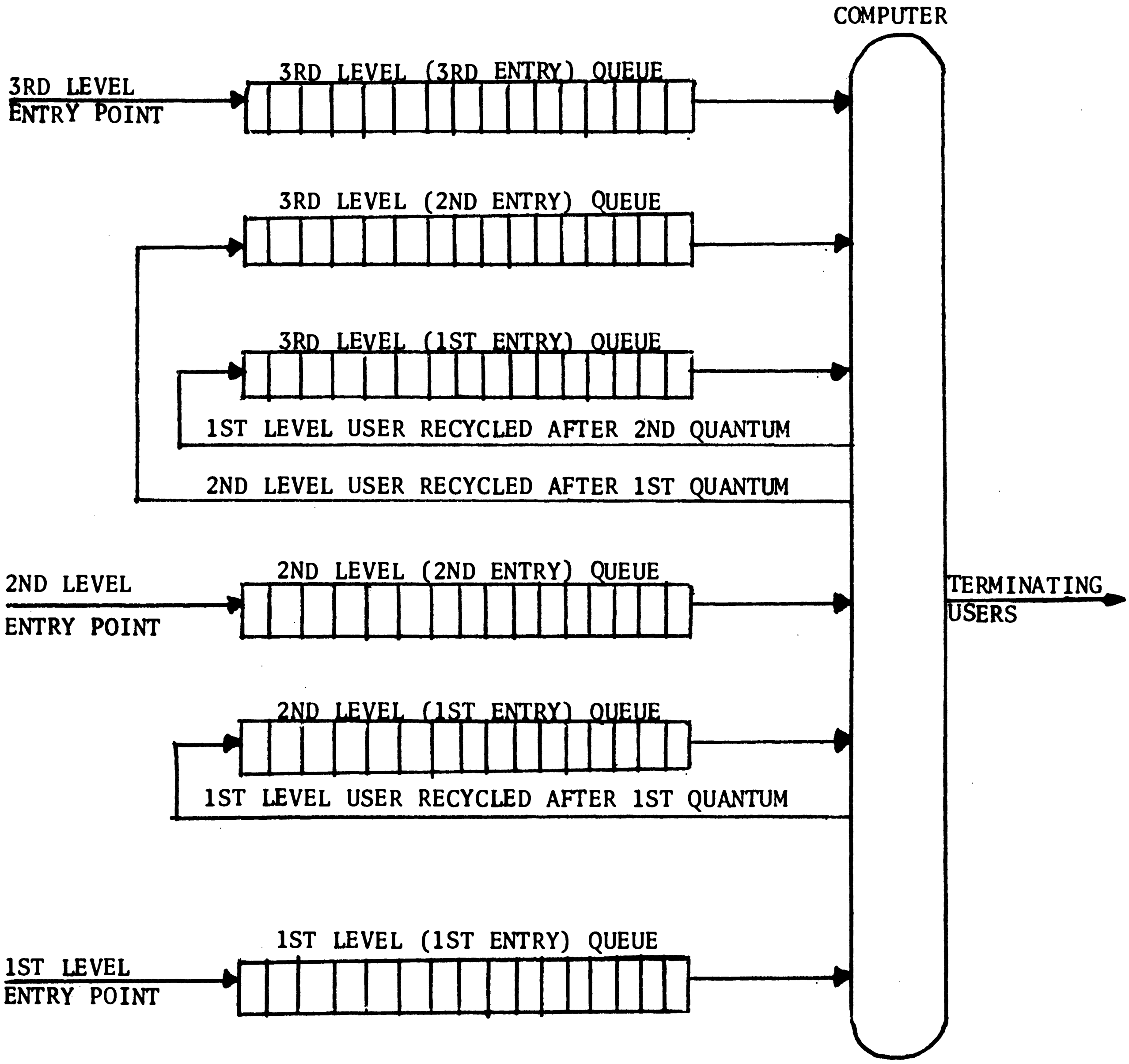

Figure 2.3: The Proposed Time-Sharing System (for n = 3)

In the proposed system, a first level user will always be preferred to a second level one, at any level in which they might both be queued at the same time. When the service facility terminates a time slice, it begins scanning the queues from the bottom up as illustrated in Figure 2.3. That is, it will check whether the first level queue is empty; if not, a user from this queue will be served. If it is empty, the system will scan the second level queues; if there are no first level users, the service facility will check the second level users' queue, and so on. This preference pattern is also reflected in the cost structure: first level users value their waiting time more than second level users, etc.

In all other respects, the proposed system is similar to a FB_n system. An exact definition of the proposed system is presented in the next section.

2.b. Description and Notation

The proposed time sharing system maintains a (finite or infinite) sequence of N queues. Each queue is identified by a pair of integers: the first denotes the level of the queue, and the second identifies the level at which the users in the queue joined the system; for example, users in the queue (4,2) joined the system in the second level, received service in the second and third levels, and are waiting to get service in the fourth level. Users may join the system at any level, up to a finite level $\underline{N}$.

Specifically, at level n, new users join the system through the (n,n) queue. The service to users is given in time slices: a user in the (n,k) queue, where $k = 1,\ldots,n$, receives a time slice which is distributed according to the distribution $S(\cdot,n,k)$. If his service requirements are not fulfilled during this time slice he will be transferred either to the end of the (n+1,k) queue (feedback mode), or to the end of the (n,k) queue (round-robin mode). The system has a queue indicator vector $\underline{S}(n,k)$; if its (n,k) element is 1 the user in the (n,k) queue will be transferred (when his time slice is completed) to the end of the (n+1,k) queue; otherwise, he will be transferred to the end of the (n,k) queue. Thus, a queue with a queue indicator of 1 corresponds to a feedback queue.

The queues are served according to their priority, and a user in a low priority queue will not be served as long as there are users waiting in higher priority queues. The higher the level, the lower the priority, and at each level, the higher the joining level the lower the priority. Thus the queue (n,k) has higher priority than the queue (i,j) if either $n < i$ or if $n = i$ and $k < j$. Within the same queue service is FIFO to users in the queue.

For simplicity of notation, the (n,k) queue will sometimes be identified by the single integer $k+\underline{n}$ where $\underline{n}$ is the sum of the integers from 1 to n-1. This mapping is unique and retains the same priority ordering implied by the pair (n,k), where $1 \leq k \leq n$.

In order to understand the operation of the system, suppose that at a given moment no one is being served. The system checks the queues, and locates the highest priority queue in which users are waiting. It takes the user at the head of this queue, and determines the time slice which will be given to him according to the time slice distribution associated with this queue. If the service requirements of the chosen user are fulfilled during this time slice, he leaves the system upon completion of his service requirements, and the time slice is terminated; otherwise, he moves to his next queue when the time slice is finished. In either case, prior to commencing service, the system requires overhead time for such things as accounting procedures for departing users, scheduling of non-departing users, queue searching for the next user, and, possibly, set-up for the next user to be served; this overhead is distributed according to a distribution $\theta(\cdot)$. When the overhead time is completed, the system starts serving the chosen user.

New users arrive to the system according to an interarrival time (between arrivals of consecutive users) distribution, denoted $A(\cdot)$. The urgency requirements of arriving users who consider whether to join the system might differ; each user has an associated urgency value. The higher the urgency value of a user, the more impatient he is to get an immediate service. It is assumed that the urgency values are positive variables with distribution $U(\cdot)$. Each user is assumed to know his urgency value, as well as the distribution $U(\cdot)$. This thesis will mostly deal, though, with a degenerate urgency distribution.

The service requirements of users are assumed to be distributed according to $H(\cdot)$. The exact value of the service requirement is unknown to the user and to the system; hence, all the decisions to be made can be based only on expected values.

The charges that users have to pay to the system are of the following types. First, a customer who joins the ith queue has to pay a toll of t_i. In addition, he has to pay c_i for each time unit that he is in the system. This charge reflects a possible connect time charge as encountered in current time-sharing systems, and compensates the system for the time the user uses its resources like telephone ports, channels, queues, etc. This waiting charge is determined according to the joining queue, and not according to the queues through which the user is advancing. In a doubly-numbered queueing system like the proposed one, the management of such charges will not constitute a heavy burden on the system. Finally, the user has to pay a fixed charge per time unit of service rendered, which will be denoted by c. We assume that the charges are positive and finite, and decrease with the initial priority selected; $t_1 \geq t_2 \geq \cdots \geq 0$ and $c_1 \geq c_2 \geq \cdots \geq 0$.

The state of the system at any given moment is given by the vector $(k,s,n_1,n_2,\ldots,n_i,\ldots)$ $(1 \leq i \leq N)$, where n_i is the number of users waiting in the ith queue at that moment, k is the queue of the user being served at that moment, and s is the elapsed time since the beginning of the overhead and time slice assigned to this user. Since the distributions of the time slices as well as the distribution of the service requirements are known, and since the queues through which each user has passed are known due to the identification of the queues, the residual service distributions of all the users in the system are also known, in principle at least. In order that this statement will hold also for any RR queue, the state of such a queue must include, in addition to the number in the queue, the number of times that each user in it joined it. We assume that a user who considers joining the system knows the state of the system at the moment of possible joining, as well as the cost structure of the system, the time slices and overhead distributions, the interarrival times distribution, the service requirements distribution and the urgency distribution, all of which are assumed to be mutually independent.

All users are assumed to have the same individual loss function. A user with urgency value u and service requirement h, who joins the system at the ith queue when the system state is T, remains in steady state $W_i(T)$ units of time in the system. Such a user incurs a (utility) loss that is affected by his urgency, the time spent in the system $(h$ and

$W_i(T))$ and his monetary loss $(t_i, ch, c_iW_i(T))$, of

$$L(t_i, u, h, ch, W_i(T), c_iW_i(T)) \tag{2.1}$$

where $L(\cdot)$ is assumed to be a positive strictly increasing unbounded function of all its arguments. Since, as mentioned earlier, only expectations are considered, the loss is

$$L_i(u,T) = E_h(E_{W_i(T)}(L(t_i, u, h, ch, W_i(T), c_iW_i(T)))) \ . \tag{2.2}$$

All of the expectations exist and this equation is meaningful since all the terms involved are non-negative.

Finally, a description of the loss that the user incurs when he balks and joins a competing system is necessary. It is assumed that the competing system is either a batch or a conventional time-sharing system in which the loss is related only to the service requirements and the waiting time in it, and the user cannot affect the service scheduling of it. The expected loss in this system is denoted L_0 and is assumed to be finite. Balking is equivalent to joining a dummy queue 0.

L_0 may be interpreted as the reward a user gets for completed service, measured on the same scale as $L(\cdot)$; this reward is assumed to be the same for all users. A user will clearly balk if the expected loss from joining the system exceeds the reward he is going to get.

The notation which has been introduced in this section is summarized below:

The physical system:

(n,k) = identification of a queue; n is the level of the queue, k is the joining level.

N = number of queues in the system (may be infinite).

$\underline{N}$ = last level which users may join.

$\theta(\cdot)$ = distribution of overhead time.

$S(\cdot,n,k)$ = distribution of time slice allocated to queue (n,k).

$\underline{S}(n,k)$ = queue indicator of the (n,k) queue; 1 for a feedback queue, 0 for a RR queue.

The users:

$A(\cdot)$ = interarrival time distribution.

$H(\cdot)$ = service requirement distribution.

$U(\cdot)$ = urgency distribution.

Charges:

c = cost of service time.

c_i = cost of waiting time, for a user who joined the system through queue i.

t_i = toll for joining queue i.

System's state:

$T = (k,s,n_1,\ldots,n_j,\ldots)$ where k is the queue at which the user now in service entered service, s is the time passed since the beginning of the current overhead and time slice, n_j is the number of users waiting in queue j. If the service facility is idle, k is zero.

$W_i(T)$ = waiting time, in steady state, for a user who joined the ith queue in state T.

Loss:

$L(t_i,u,h,ch,W_i(T),c_iW_i(T))$ = loss of a user with urgency u, service requirements h, who joined the ith queue in state T, and will wait $W_i(T)$.

$L_i(u,T)$ = expected loss to such a user.

L_0 = expected loss in competing system, or reward for completed service.

Random variables will be denoted by small letters, while their distributions will be denoted by the associated capital letters.

2.c. Form of an Optimal Joining Policy

As mentioned in Chapter 1, each arriving user has to make an irrevocable decision: whether to join the system, and if so, which queue to join. The arriving user has to base his decision on his urgency and the state of the system at his arrival. We will call the rule by which he determines his decision a joining policy.

Suppose that a user arrives at some time epoch t, when the state of the system is T, and the past history of the system is $V(t)$ which includes all the occurrences in the time prior to t, with the time epochs at which they occurred. The joining policy space is a set of rules $\{p^r(i,T,V(t))\}$, where $p^r(i,T,V(t))$ is the probability of joining the ith queue according to the rth rule when the state of the system is T and the past history is $V(t)$. Note that T completely summarizes the present and expected future states of the system.

The arriving user will base his decision on his expected loss, choosing the queue to be joined so as to minimize this loss. His expected waiting time in the system will be higher if he decides to join lower priority queues, since the waiting time in such a case will be composed of the service time of users in higher priority queues, as well as the busy periods initiated during their wait and service, i.e. the service and waiting time of users who will join higher priority queues during their stay in the system. The expected waiting time is clearly a function of the parameters of the system, which determine at each state how a new user will behave.

The problem that the newly arrived user faces may be formulated as follows:

Find a joining policy that minimizes

$$\sum_{i=0}^{i=N} p^r(i,T,V(t)) \times L_i(u,T) \tag{2.3}$$

subject to the constraints

$$\begin{aligned} &p^r(i,T,V(t)) \geq 0 \quad \text{for} \quad 0 \leq i \leq N \\ &\sum_{i=0}^{i=N} p^r(i,T,V(t)) = 1 \ . \end{aligned} \tag{2.4}$$

This is a linear programming problem in the variables $p^r(i,T,V(t))$, for which an optimal solution is:

> Let $J(u,T) = \{i\colon L_i(u,T) \leq L_j(u,T) \quad \text{for} \quad 0 \leq j \leq N\}$.
> Choose $i_1,\ldots,i_k \in J(u,T)$, and assign (arbitrarily) (2.5)
> $p^r(i_j,.,.) \geq 0$ for $1 \leq j \leq k$ such that $\sum_{j=1}^{j=k} p^r(i_j,.,.) = 1$;
> assign $p^r(i,.,.) = 0$ for $i \notin J(u,T)$, and for $i \in J(u,T)$ but $i \neq i_1,\ldots,i_k$. That is: join any one of the queues whose expected loss is minimal.

This rule does not depend on $V(t)$, and hence is stationary; it is randomized only within the $J(\cdot)$ set, and this randomization reflects only the indifference of the user among the possible queues that are contained in the set, and is not essential for the optimality. Any one of the queues in this set may be chosen without affecting the (expected) loss; hence, it seems justified to call it a non-randomized policy.

(2.5) may be interpreted in somewhat different way. There are some queues that are filled to capacity, and they will not be joined by the arriving user; only these queues that are not full are considered for joining, and they belong to the joining set $J(u,T)$. Thus, to each state T, urgency u and queue i, one can associate a number $N(i,u,T)$; if the number of users in the ith queue exceeds $N(i,u,T)$, newly arriving user with urgency u who arrives at state T will not join it.

The number $N(i,u,T)$ determines a critical capacity of the ith queue at state T for an arriving user who has urgency u. It will be called the control number corresponding to the ith queue, urgency u and state T. The optimal joining policy may thus be characterized as a stationary non-randomized control-limit type policy.

Consider now the case when all the users have the same urgency value. Suppose that the number of arrivals exceeds the number of terminating users due to random fluctuations, or due to traffic intensity exceeding one, so that some or all of the queues maintained by the system are filled up to a limit determined by the loss function, if such limit exists. As the system becomes more and more filled, the waiting time in the various queues increases accordingly, with stochastically faster increase in the lower priority queues that are not served as long as higher priority queues are not empty. The loss function will also increase, and eventually the loss in each queue will, due to the properties of the loss function, exceed the loss from joining the competing system. Users will then balk from such filled up queues and join only the non-filled up queues, as is implied by (2.5). Due to the finite value of L_0, only a finite number of users will join the system, thus determining a critical capacity for the system; newly arrived users will balk if the system is filled up to this capacity.

The system is therefore a finite room queueing system, when one urgency value is considered. For each urgency value there will be a value of critical capacity; when the system is filled to this critical capacity, newly arrived users with this urgency value will balk. The critical capacity is clearly a decreasing function of the urgency value, due to the monotonicity of the loss function. The system will therefore be filled up to at most the critical capacity corresponding to the lowest urgency value, which is finite.

The discussion of this section is thus summarized in

Theorem 2.1: The optimal joining policy is a stationary non-randomized control type policy, and the system is a finite room queueing system. To each state, urgency value and queue correspond to a control number, and a newly arrived user with this urgency will not join the queue at this state if it filled up to, or more than, this control number.

When the system contains only one queue, as is the case in a round-robin time-sharing system, users have only two possible actions: to join the queue or to balk. To each urgency value and system state there corresponds one critical capacity; users with this urgency will join the system at this state as long as it is not filled to this capacity.

Suppose now that the information in a round-robin time-sharing system is restricted to the display of the number of users waiting in the queue, and the time devoted by the system in the current quantum to the user currently being served. A newly arriving user has then to assume that all the users in the system have the same residual service time distribution, and his decision whether to join the system or balk is thus affected only by the number of users in the system. To each urgency value there corresponds one critical capacity, which is independent of the state of the queue (and the system). Note that the full and restricted information displays coincide for exponential service requirement distribution. The display of such restricted information is much easier and manageable than the display of full information; hence it is safe to assume that restricted information will be the rule, rather than the exception, in most round-robin time-sharing systems.

CHAPTER 3

A SINGLE-ENTRANCE, ROUND-ROBIN, TIME-SHARING SYSTEM, WITH ENTRANCE BASED ON PRICE

3.a. Introduction

In this chapter we apply the ideas discussed earlier to a round-robin time-sharing system. Due to the cost structure the system will have a finite room queue as discussed in Section 2.c. Display of restricted information is assumed, and thus to each urgency value there corresponds one control number which is independent of the system's state. In a round-robin system a newly-arriving user must either join or balk, and hence a toll is not essential for discrimination between entry points to the system.

Schematically, the system is described below:

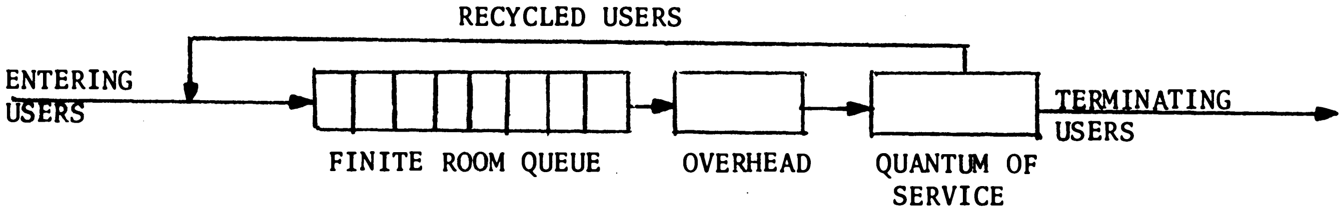

Figure 3.1: A Round-Robin System

In Sections 3.b through 3.e a simplified system is analyzed, in order to emphasize the ideas and reasoning involved in the analysis. The simplified system is characterized by:

- exponential interarrival times,
- exponential service requirements,
- fixed length quantum,
- fixed length overhead,
- degenerate urgency distribution, whose mass is concentrated at one urgency value,
- cost structure consisting of service and waiting charges only.

In Section 3.f the simplifications introduced by these assumptions are discussed, and some of them are shown to be non-critical.

Exact expression for the expected response time is derived in Section 3.b, and an algorithm for the determination of the control number is presented. Section 3.c is devoted to the case of a linear loss function, and exact values of the control number are derived for a minimax type expected response time. Sections 3.d and 3.e are devoted to the system manager's problem. In Section 3.d several steady state distributions are derived; specifically, the distribution of number of users in the system and the distribution of the states of the system. In the derivation process the classical "telephone exchange" Erlangian model is extended to the case where each user requires service in two phases. The results of Section 3.d are then used in Section 3.e to determine the optimal set of system's parameters that maximizes the average expected revenue per time unit.

3.b. Determination of the Control Number

For a given set of system's parameters, the control number is a function of the loss function, which is itself a function of the (expected) response time in the system for a newly arriving user, which in turn is a function of the control number. This section will therefore start with the determination of the expected response time, assuming the control number is given. An algorithm for the evaluation of the control number is then presented, followed by a qualitative discussion about the dependence of the control number on system's parameters.

Suppose thus that a set of system's parameters is given: a quantum of size q, overhead θ, service charge c, waiting charge c_1, and rates λ and α for interarrival times and service requirements, respectively. Due to the degeneracy of the urgency distribution only one control number exists, which in turn determines a system's capacity, i.e. a maximal value for the number of users who might be concurrently in the system; let k denote this capacity. The state of the system at any time epoch is given by a pair (i,t) where i is the number of users in the system at that time epoch and t is the service time already given in the current quantum to the user in service at that time epoch. Finally let $\beta = 1-e^{-\alpha q}$ be the probability that a user starting service in a quantum will leave the system during the quantum; due to the exponential distribution of the service requirement β is independent of the number of recycles the user had prior to commencing service.

In order to determine the expected response time, consider the Markov process generated by the time epochs in which a "tagged" user joins the system, is recycled, or leaves the system. In this process the states are the number of users in the system excluding the "tagged" user, and we are interested in the expected response time, i.e. waiting plus service time, between consecutive such time epochs. Let:

P_{ij} = P(state at a time epoch is $j |$ state at last time epoch was i)

$(0 \leq i, \; j \leq k-1)$

W_{ij} = E(response time in a cycle$|$cycle starts at state i, ends at j).

Due to the exponential service requirements W_{ij} and P_{ij} are independent of the number of cycles already completed and the number of cycles to be completed in the future, with two exceptions: the first cycle in which there might be a user getting service when the "tagged" user arrives, and the last cycle in which the "tagged" user might leave the system before completing his quantum. Thus denote by upper-bar and under-bar the quantities corresponding to the first and last cycles, respectively. For example, $\overline{W}_{i_0j}$ is the expected response time in the first cycle which the user joined where there were i_0 users in the system; the user will not terminate service in this cycle, but rather will rejoin with j users in the system.

If the "tagged" user terminates his service within the first quantum allocated to him, his expected total response time will be $\sum_{j=0}^{j=k-1} \underline{\overline{P}}_{i_0j} \times \underline{\overline{W}}_{i_0j}$ where (i_0,t) is the state of the system when he first joins it (to simplify the notation, the dependence of $\underline{\overline{P}}$ and $\underline{\overline{W}}$ on t is suppressed). If he terminates his service in the second quantum allocated to him, his expected total response will be

$$\sum_{j=0}^{j=k-1} \overline{P}_{i_0j} \times (\overline{W}_{i_0j} + \sum_{n=0}^{n=k-1} \underline{P}_{jn} \times \underline{W}_{jn}) =$$

$$= \sum_{j=0}^{j=k-1} \overline{P}_{i_0j} \times \overline{W}_{i_0j} + \sum_{j=0}^{j=k-1} \sum_{n=0}^{n=k-1} \overline{P}_{i_0j} \times \underline{P}_{jn} \times \underline{W}_{jn} .$$

Similarly, if he terminates his service in the mth quantum $(m \geq 3)$ his expected total response time will be

$$\sum_{n_1=0}^{n_1=k-1} \overline{P}_{i_0 n_1} \times \overline{W}_{i_0 n_1} + \sum_{n_1=0}^{n_1=k-1} \sum_{n_2=0}^{n_2=k-1} \overline{P}_{i_0 n_1} \times P_{n_1 n_2} \times W_{n_1 n_2} +$$

$$+ \cdots + \sum_{n_1=0}^{n_1=k-1} \cdots \sum_{n_j=0}^{n_j=k-1} \overline{P}_{i_0 n_1} \times P_{n_1 n_2} \times \cdots \times P_{n_{j-1} n_j} \times W_{n_{j-1} n_j} +$$

$$+ \cdots + \sum_{n_1=0}^{n_1=k-1} \cdots \sum_{n_m=0}^{n_m=k-1} \overline{P}_{i_0 n_1} \times P_{n_1 n_2} \times \cdots \times P_{n_{m-2} n_{m-1}} \times$$

$$\times \underline{P}_{n_{m-1} n_m} \times \underline{W}_{n_{m-1} n_m} \, .$$

Denoting by P the matrix of the P_{ij}'s, by W the matrix of the $(P_{ij} \times W_{ij})$'s, by e_i the ith unit row vector and by e' the column vector all of whose elements are 1, and using matrix notation, these expressions become:

- For a user who terminates service in the first quantum:

$$e_{i_0} \times \underline{\overline{W}} \times e' \, .$$

- For a user who terminates service in the second quantum:

$$e_{i_0} \times (\overline{W} + \overline{P} \times \underline{W}) \times e' \, . \tag{3.1}$$

- For a user who terminates service in the mth quantum $(m \geq 3)$:

$$e_{i_0} \times (\overline{W} + \overline{P} \times W + \ldots + \overline{P} \times P^{j-2} \times W + \ldots + \overline{P} \times P^{m-2} \times \underline{W}) \times e' \, .$$

Since the probability that a user will terminate his service in the ith quantum $(i \geq 1)$ is $\beta \times (1-\beta)^{i-1}$, the expected total response time is

$$e_{i_0} \times (\beta \times \underline{\overline{W}} + \beta \times (1-\beta) \times (\overline{W} + \overline{P} \times \underline{W}) + \sum_{j=3}^{j=\infty} \beta \times (1-\beta)^{j-1} \times$$

$$\times (\overline{W} + \sum_{n=0}^{n=j-3} \overline{P} \times P^n \times W + \overline{P} \times P^{j-2} \times \underline{W})) \times e' =$$

$$= e_{i_0} \times (\beta \times \underline{\overline{W}} + \beta \times (1-\beta) \times \sum_{j=0}^{j=\infty} (1-\beta)^j \times \overline{W} + \beta \times (1-\beta)^2 \times \sum_{j=0}^{j=\infty} (1-\beta)^j \times$$

$$\times \overline{P} \times \sum_{j=0}^{j=\infty} ((1-\beta) \times P)^j \times W + \beta \times (1-\beta) \times \overline{P} \times$$

$$\times \sum_{j=0}^{j=\infty} ((1-\beta) \times P)^j \times \underline{W}) \times e' .$$

But β is positive and less than 1 and P is a stochastic matrix; hence all the infinite sums converge, and one gets:

Theorem 3.1: The expected total response time for a newly arrived user who joins the system when its state is (i_0, t) is

$$W(i_0,t) = e_{i_0} \times (\beta \times \underline{\overline{W}} + (1-\beta) \times \overline{W} + (1-\beta)^2 \times \overline{P} \times (I - (1-\beta) \times P)^{-1} \times W +$$

$$+ \beta \times (1-\beta) \times \overline{P} \times (I - (1-\beta) \times P)^{-1} \times \underline{W}) \times e' .$$

The result of Theorem 3.1 has a very intuitive interpretation; rearranging the expression for the expected total response time one has:

$$e_{i_0} \times (\beta \times \underline{\overline{W}} + (1-\beta) \times (\overline{W} + \overline{P} \times (I - (1-\beta) \times P)^{-1} \times$$

$$\times (\beta \times \underline{W} + (1-\beta) \times W))) \times e' .$$

The expected total response time if only one quantum is used, an event which has probability β, is $\underline{\overline{W}}$. If more than one quantum is used, an event which occurs with probability $1-\beta$, the user has always to wait during the first cycle in the system $(\overline{W})$. In addition, in this case he is subjected in each cycle to the transition matrix $(1-\beta) \times P$, while his expected response time in each cycle is $\underline{W}$ if his service terminates at this cycle (event occurring with probability β), and W with probability $1-\beta$ otherwise.

The expressions for P_{ij} and W_{ij} are cumbersome, and are not needed in the sequel. Their development is, therefore, relegated to Appendix A.

The expected total response time for a user who knows his service requirement was derived in the development leading to Theorem 3.1. A user with service requirement of s will receive service for $m = [s/q] + 1$ quanta; his expected response time is then given by (3.1).

Consider now the problem of determining the control number. All newly arriving users face the same set of system parameters and have the same expected service time and urgency value; their loss if they join the system is thus affected only by their expected response time. A newly arrived user will join the system as long as the loss L_0 incurred by the competing system, does not exceed the loss of joining the system with expected response time $W(i,t,k)$, i.e., as long as $L_0 \geq L(W(i,t,k))$, where all superfluous parameters in the loss $L(\cdot)$ notation have been suppressed, and the capacity k has been included in the expected response time. $W(i,t,k)$ clearly increases when i increases, and decreases when t increases; for a given capacity k the maximal expected response time, and maximal loss, will thus be achieved for $i = k-1$ and $t = 0$. Moreover, $W(k-1,0,k)$ increases with k; due to the monotonicity of $L(W(\cdot))$, the control number will correspond to a capacity k' such that

$L(W(k'-1,0,k')) \leq L_0 < L(W(k',0,k'+1))$. Thus Algorithm 3.1 for the determination of the control number suggests itself:

a. Set $k = 1$.

b. Compute $W(k-1,0,k)$ according to Theorem 3.1.

c. If $L_0 \geq L(W(k-1,0,k))$ set $k \leftarrow k+1$ and return to step b.

d. Otherwise set the control number to $k-1$ and terminate.

This algorithm clearly converges, due to the strict monotonicity of $L(W(.,.,k))$ as a function of k, and the existence of the upper bound L_0.

Due to the complexity of the expression appearing in Theorem 3.1, and the general form of the loss function, it is impossible to make quantitative statements concerning the dependence of the control number on system's parameters. Some qualitative statements may be made though.

- Increasing the quantum size will result in fewer cycles and decreased amount of total overhead. We thus can expect $W(i,t,k)$ to decrease as a function of q, for any given triplet (i,t,k). As a result we would expect that the control number will increase as q increases.
- Increasing the waiting and/or service charges will increase the loss for any triplet (i,t,k). As a result, we would expect that the control number will decrease as the charges per unit time increase.
- Similarly, we would expect that the control number will decrease as the urgency value increases, due to the monotonicity of the loss with respect to the urgency values.

The analysis presented above was based upon the assumption that a newly arriving user does not know his exact service requirements, and bases his decision whether to join the system or balk on the expected service requirement. Once this assumption is violated, the analysis above fails. Instead, each newly arrived user would calculate his expected response time using a result similar to (3.1), and would decide accordingly whether to join or balk. To each state and service requirement there would thus correspond a different control number. Since the number of cycles and the expected response time both increase as the service requirement increases, these control numbers would be a decreasing function of the service requirement for any state. As a result, for any state $T = (i,t)$ there would correspond a number s_T such that users with service requirement s would join the system only if $s \leq s_T$. In other words: The service requirement distribution of arriving users would become a function of the system's state. A detailed examination of the analysis which led to Theorem 3.1 reveals, though, that Theorem 3.1 still holds; modification of W_{ij} and P_{ij} are required though. This modification is discussed in Appendix A, where the W_{ij}'s and P_{ij}'s are calculated.

In the latter case, Algorithm 3.1 is not valid any more, since more than one control number exist for the system. In this respect, this case is similar to the case when the urgency distribution is not degenerate; both cases will be further discussed in Section 3.f.

3.c. The Control Number for a Linear Loss Function

When the loss function is linear, it can be written as

$$L(u,h,ch,W(T),c_1W(T)) = n_1u + n_2h + n_3ch + n_4W(T) + n_5c_1W(T)$$

where the linear combination coefficients $n_1, n_2, n_3, n_4, n_5 \geq 0$ due to the monotonicity of the loss function with respects to its arguments, and where u is the urgency value, h is the expected service time, c is the service charge, c_1 is the waiting charge and $W(T)$ is the expected response for a user who joins the system at state T. Both ch and $c_1W(T)$ are expressed in monetary units; therefore we can expect n_5 to be equal to n_3, both being equal to the value of one monetary unit to the user. Similarly, we would expect that $n_2 = n_4$, since both represent the value placed by the user on the time spent in the system.

Following the discussion in Section 3.b the control number will correspond to the capacity k' such that

$$W(k'-1,0,k') \leq (L_0 - n_1u - n_2h - n_3ch)/(n_2 + n_3c_1) < W(k',0,k'+1)$$

if $n_2 + n_3c_1 > 0$, and may be any integer otherwise, since the loss is then independent of the waiting time. Moreover, if $L_0 - n_1u - n_2h - n_3ch \leq 0$ no one will join the system.

Restricting ourselves to one urgency value, and assuming that the user suffers real loss from monetary expenses and thus $n_3 > 0$, we may rewrite the last condition as

$$W(k'-1,0,k') \leq (A - ch)/(B + c_1) < W(k',0,k'+1) \qquad (3.2)$$

where $A = (L_0 - n_1u - n_2h)/n_3$ and $B = n_2/n_3$. (3.2) is the most suitable form for analysis of the control number's dependence on system's charges; such an analysis is carried on in Section 3.e, where system's charges are determined.

The expected response time as given by Theorem 3.1 is cumbersome and hard to calculate. An upper bound to $W(k-1,0,k)$ can be given though, which might be considered as the pessimist's bound, or a minimax bound. In this approach, a newly arriving user considers the worst possible case: that he will find the system full every time he is recycled.

The expected response time when the system is full, i.e., when it contains k users, including the newly arriving user, is for one cycle

$$R = k \times (\theta + (1-\beta) \times q + \beta \times \int_0^q (s \times \alpha \times e^{-\alpha q}/\beta)ds = k \times (\theta+\beta/\alpha)$$

where $(1-\beta) \times q$ is the contribution of a user whose service isn't terminated during a quantum, θ is the overhead given to each user, and the remaining term is the contribution to the expectation of users who terminate their service during the quantum they receive.

Due to the exponential service requirement distribution, and in accordance with the pessimist's point of view, the newly arriving user is expected to be in the system R time units during the first cycle. He then either terminates his service, with probability β; or is recycled and his expected response is the same as it was at the beginning of his last cycle, an event which occurs with probability $1-\beta$. His expected waiting response time is thus

$$W(k-1,0,k) = R + (1-\beta) \times W(k-1,0,k)$$

which gives

$$W(k-1,0,k) = R/\beta = k \times (\theta/\beta+1/\alpha) \; . \qquad (3.3)$$

Combining the minimax bound (3.3) with the linear loss condition (3.2), yields $k' \times (\theta/\beta+1/\alpha) \leq (A - ch)/(B + c_1) < (k'+1) \times (\theta/\beta+1/\alpha)$. Noting that $h = 1/\alpha$, the last inequality yields as the control number's lower bound

$$k' = \lfloor(\alpha \times A - c)/((B + c_1) \times (1 + \alpha \times \theta/\beta))\rfloor \ . \tag{3.4}$$

Notice that as q increases, k' also increases. In the limiting case of unbounded quantum, in which a user is served to completion, $\beta = 1$; and the control number becomes

$$k'' = \lfloor(\alpha \times A - c)/((B + c_1) \times (1 + \alpha \times \theta))\rfloor \ .$$

Moreover, k'' is exact and not a lower bound, since in the case of unlimited quantum no one is recycled. k'' is thus the control number for a regular M/M/1 queueing system with setup, or overhead time. It is also of interest to note that for the limiting case Theorem 3.1 is no longer true, since the infinite sums which were used in the proof do not converge; instead, (3.3) with $\beta = 1$ has to be used.

3.d. System's Manager Problem - Steady State Probabilities

The system's manager wishes to maximize his average expected revenues per time unit; the control "tools" he might use in achieving the maximization are the quantum size and the charges for service and waiting, which in turn determine the capacity of the system. The average revenue per time unit in steady state is $c \times P(k) + c_1 \times Q(k)$, where $P(k)$ is the proportion of time the system is engaged in giving service, and $Q(k)$ the average number of users in the system, assuming system's capacity to be k. In this section $P(k)$ and $Q(k)$ will be developed. During their derivation the steady state distributions of system's states and number of users in the system will be computed, and an extension of the "telephone exchange" model will be presented.

$P(k)$ and $Q(k)$ depend only on the workload in the system, which is independent in steady state of the discipline used for serving the users, as long as this discipline is independent of their service requirements. Therefore we will consider the following discipline for the evaluation of $P(k)$ and $Q(k)$: When a user is selected for processing, he first gets all the service required to fulfill his service requirements, and then all the overhead times resulting from his service requirements; the selected user then leaves the system, and the next user in FIFO order is selected for processing. Under this scheme the analyzed system becomes a finite room M/G/1 system, with capacity k, in which service to each user is given in two phases - the exponential "processing" phase and the "overhead" phase. If the overhead phase is eliminated, the classical "telephone exchange" Erlangian model is achieved; the results below thus are an extension of this classical model.

Let P_i be the steady state probability of having i users in the system and having the user being served in service phase, and $P_{\underline{i}}$ be the steady state probability of having i users in the system and having the user being served in overhead phase $(0 \leq i \leq k)$; $P_{\underline{0}}$ has no meaning though, and will not be used. Then $P(k)$ and $Q(k)$ are given by

$$P(k) = \sum_{i=1}^{i=k} P_i \qquad \text{and} \qquad Q(k) = \sum_{i=1}^{i=k} i \times (P_i + P_{\underline{i}}) \; . \tag{3.5}$$

The steady state probabilities clearly exist, since the system has a finite state space.

Noting that the modified system is a semi-Markov process with states $\{0,1,\underline{1},\ldots,i,\underline{i},\ldots,k,\underline{k}\}$ in which transitions occur at time epochs of phase termination, and following Ross [21], define:

$Q_{ij}(t)$ = probability that after making a transition into state i, the process next makes a transition into state j, in an amount of time not exceeding t.

The underlying probabilities consist of exponential interarrival distribution with rate λ, exponential service phase distribution with rate α, and overhead phase consisting of several periods each of length θ whose number is distributed geometrically with parameter β. The overhead phase distribution is the result of allocating service in the original system in quanta, with probability β of terminating service in each quantum. Thus:

$$\left.\begin{aligned}
Q_{01}(t) &= 1 - e^{-\lambda t} \\
Q_{i,\underline{i}}(t) &= \alpha \times (1 - e^{-(\alpha+\lambda)t})/(\alpha+\lambda) \\
Q_{i,i+1}(t) &= \lambda \times (1 - e^{-(\alpha+\lambda)t})/(\alpha+\lambda) \\
Q_{\underline{i},i-1}(t) &= \textstyle\sum_{j=1}^{j=[t/\theta]} \beta \times (1-\beta)^{j-1} \times e^{-\lambda_j \theta} \\
Q_{\underline{i},\underline{i+1}}(t) &= \textstyle\sum_{j=1}^{j=[t/\theta]} \beta \times (1-\beta)^{j-1} \times (1 - e^{-\lambda_j \theta}) + \\
&\qquad + (1-\beta)^{[t/\theta]} \times (1 - e^{-\lambda t}) \\
Q_{k,\underline{k}}(t) &= 1 - e^{-\alpha t} \\
Q_{\underline{k},k-1}(t) &= \textstyle\sum_{j=1}^{j=[t/\theta]} \beta \times (1-\beta)^{j-1} \\
Q_{ij}(t) &= 0 \text{ for all other combinations of } i \text{ and } j \\
&\qquad (1 \leq i,\ j \leq k).
\end{aligned}\right\} \qquad (3.6)$$

Denote:

$\mu_i = \int_0^\infty t \, d[\sum_{j=0}^{j=k} (Q_{ij}(t) + Q_{i\underline{j}}(t))]$ = expected amount of time spent in state i during each visit ($i = 0,\dots,k;\ \underline{1},\dots,\underline{k}$).

π_i = steady state probability that the embedded Markov chain generated by phase termination epochs is in state i $(i = 0,\ldots,k;\ \underline{1},\ldots,\underline{k})$.

These steady state probabilities clearly exist, since the embedded Markov chain is a finite state chain. Note that in the summation above we used implicitly $Q_{i\underline{0}}(t)$ which has no meaning for the modified system, and will be assumed to be identically zero.

Clearly there exists a value $t_0 > 0$ such that $\sum_{j=0}^{j=k} (Q_{ij}(t_0) + Q_{i\underline{j}}(t_0)) > \varepsilon > 0$ for all possible states i; the process is therefore regular. This semi-Markov process constitutes of one communicating class. If the traffic intensity of the modified system is less than one, i.e., if $\lambda \times (\theta/\beta+1/\alpha) < 1$, then each state is positive recurrent, and in this case it is known that for each state:

$$P_i = (\pi_i \times \mu_i)/\sum_{j=0}^{j=k} (\pi_j \times \mu_j + \pi_{\underline{j}} \times \mu_{\underline{j}}) \ . \tag{3.7}$$

Thus, in order to give closed form expressions for $P(k)$ and $Q(k)$, the π_i's and μ_i's have to be found. Starting with the μ_i's, and denoting

$$C = \lambda/(\alpha+\lambda) \quad \text{and} \quad D = (1 - e^{-\lambda\theta})/(1 - (1-\beta) \times e^{-\lambda\theta})$$

it is easily found from (3.6) that

$$\left.\begin{array}{l} \mu_0 = 1/\lambda \\ \left.\begin{array}{l} \mu_i = C/\lambda \\ \mu_{\underline{i}} = D/\lambda \end{array}\right\} \ (i,\underline{i} = 1,\ldots,k-1) \\ \mu_k = 1/\alpha \\ \mu_{\underline{k}} = \theta/\beta \end{array}\right\} \tag{3.8}$$

Letting $\underline{\pi}$ be the vector of π_i's and Q the matrix $(Q_{ij}(\infty))$, the π_i's are the solution of the set of linear equations given by $\underline{\pi} = \underline{\pi} \times Q$. Specifying these equations explicitly, they become:

$$
\begin{array}{rlcrl}
(1-D) \times \pi_{\underline{1}} & = \pi_0 & & & \\
\pi_0 + (1-D) \times \pi_{\underline{2}} & = \pi_1 & & (1-C) \times \pi_1 & = \pi_{\underline{1}} \\
C \times \pi_{i-1} + (1-D) \times \pi_{\underline{i+1}} & = \pi_i & & (1-C) \times \pi_i + D \times \pi_{\underline{i-1}} & = \pi_{\underline{i}} \\
(i = 2,\ldots,k-2) & & & (i = 2,\ldots,k-1) & \\
C \times \pi_{k=2} + \pi_{\underline{k}} & = \pi_{k-1} & & & \\
C \times \pi_{k-1} & = \pi_k & & \pi_k + D \times \pi_{\underline{k-1}} & = \pi_{\underline{k}}
\end{array}
$$

for the case $k \geq 3$, with the special cases: (3.9)

For $k = 1$: $\pi_0 = \pi_1 = \pi_{\underline{1}}$

For $k = 2$:

$$
\begin{array}{rlcrl}
(1-D) \times \pi_{\underline{1}} & = \pi_0 & & & \\
\pi_0 + \pi_{\underline{2}} & = \pi_1 & & (1-C) \times \pi_1 & = \pi_{\underline{1}} \\
C \times \pi_1 & = \pi_2 & & \pi_2 + D \times \pi_{\underline{1}} & = \pi_{\underline{2}}
\end{array}
$$

In (3.9) π_i is the steady state probability that the embedded Markov chain is in state i, i.e., in the state where there are i users in the system while the user being served is in service phase. Similarly, the state $\underline{i}$ correspond to the state in which there are i users in the system, and the one being served is in overhead phase. It is worth noting that $C = Q_{i,i+1}(\infty)$ and $D = Q_{\underline{i},\underline{i+1}}(\infty)$ for any $1 \leq i \leq k-1$; therefore both C and D are positive. We assume that the traffic intensity of the modified system is less than one; in this case both C and D are strictly less than one.

The set of equations for the π_i's, as given by (3.9), is easily solved recursively. Setting π_0 as parameter and solving for the π_i's as functions of π_0, one uses alternatively equations from the left and right columns of (3.9), getting $\pi_{\underline{1}}, \pi_1, \pi_{\underline{2}}, \pi_2, \ldots$ in this order. π_0 is then used to normalize the sum of the π_i's to 1. An explicit solution can be given though, and is presented in a sequence of lemmas.

Lemma 3.1: Let γ and δ be the largest and smallest roots of the quadratic equation

$$f(Z) = (1-C) \times (1-D) \times Z^2 + (C \times D - C - D) \times Z + C \times D = 0 .$$

Then γ and δ are real and positive, with

$$0 < C \times D/(C+D) < \delta < \min(C,D) < 1 \quad \text{and}$$

$$\delta < \max(C/(1-C), D/(1-D)) < \gamma < C/(1-C) + D/(1-D) + CD/((1-C)(1-D)) < \infty .$$

Proof: $(1/\gamma)$ and $(1/\delta)$ are roots of

$$g(Z) = C \times D \times Z^2 + (C \times D - C - D) \times Z + (1-C) \times (1-D) = 0 .$$

But

$$\begin{aligned} g(Z) &= (CZ - 1 + C) \times (DZ - 1) - D \times (1 - C) \\ &= (DZ - 1 + D) \times (CZ - 1) - C \times (1 - D) \\ &= (CDZ - C - D)(Z + 1) + 1 - CD . \end{aligned}$$

Thus $g(0)$ and $g(1/C + 1/D)$ are both positive, while $g(1/C)$, $g(1/D)$, $g(1/C - 1)$ and $g(1/D - 1)$ are all negative. Since

$$0 < \min(1/C, 1/D) - 1 < \max(1/C, 1/D) < 1/C + 1/D \quad \text{and} \quad 1/\gamma < 1/\delta ,$$

all the inequalities concerning δ and the lower bound on γ are established. To get the upper bound on γ, one solves the quadratic equation, getting

$$\gamma = (1-(1-C)(1-D) + ((1-(1-C)(1-D))^2 - 4CD(1-C)(1-D))^{\frac{1}{2}})/(2(1-C)(1-D)) <$$

$$< (1-(1-C)(1-D))/((1-C)(1-D)) = C/(1-C) + D/(1-D) + CD/((1-C)(1-D)) .$$

Q.E.D.

Define now:

$$E_i = \gamma^{i+1} - \delta^{i+1} \quad \text{for} \quad i \geq -1. \tag{3.10}$$

Note that $E_{-1} = 0$.

Lemma 3.2: For $i \geq 0$ the numbers E_i:

a. are all strictly positive.

b. are strictly increasing for $\gamma \geq 1$.

c. are strictly increasing for $\gamma < 1$, for all i such that $(\delta/\gamma)^i > (1-\gamma)/(1-\delta)$.

d. obey the recursive relation

$(1-C)(1-D)E_{i+1} + (CD - C - D)E_i + CDE_{i-1} = 0.$

e. fulfill the relation $E_i - DE_{i-1}/(1-D) > 0$.

Proof: Immediate from Lemma 3.1 and (3.10).

The properties of the E_i's as presented above, and direct substitution into (3.9), establish

Lemma 3.3: The steady state probabilities of the embedded Markov chain, up to a constant, are given by:

For $k = 1$: $\pi_0 = \pi_1 = \pi_{\underline{1}} = 1/3$

For $k \geq 2$: $\pi_0 = E_0$

$$\left.\begin{array}{l} \pi_i = (E_i - DE_{i-1}/(1-D))/C \\ \pi_{\underline{i}} = E_{i-1}/(1-D) \end{array}\right\} (i = 1,\ldots,k-1)$$

$$\pi_k = E_{k-1} - DE_{k-2}/(1-D)$$

$$\pi_{\underline{k}} = E_{k-1} \ .$$

The value of the constant which normalizes the sum of the π_i's to 1 is immaterial, since the π_i's will be used both in the denominator and numerator of the P_i's, the steady state probabilities of the modified system. Note also that the π_i's are all strictly positive, in light of sections a and e of Lemma 3.2.

Seemingly, one could have used γ^{i+1} (or δ^{i+1}) instead of E_i, thus achieving a remarkable simplification in the form of the π_i's. This is indeed possible for all the π_i's, except for π_1; hence the need for the more complicated form.

Denote by ρ the traffic intensity of the modified system. Then

$$\rho = \lambda \times (1/\alpha + \theta \times \sum_{i=1}^{i=\infty} i \times \beta \times (1-\beta)^{i-1}) = \lambda \times (1/\alpha + \theta/\beta) < 1 \ . \quad (3.11)$$

As can be seen from Lemma 3.3, the value assumed by γ is important. Specifically, if $\gamma \geq 1$ the probability mass of the steady state probabilities of the embedded Markov chain is shifted to the right, and this distribution does not exist for the limiting case, i.e. infinite k. Therefore we have to imporve the upper limit of γ which was given in Lemma 3.1.

<u>Lemma 3.4</u>: $\gamma < \min(C/(1-C), D/(1-D)) + CD/((1-C)(1-D))$.

<u>Proof</u>: $f(Z)$ may be written as

$$\begin{aligned} f(Z) &= ((1-D)(1-C)Z - C)(Z+1) + CZ - Z + C + CD \\ &= ((1-C)(1-D)Z - D)(Z+1) + DZ - Z + D + CD . \end{aligned}$$

Thus $f(C/((1-C)(1-D))) = C + CD + C/(1-D) > 0$, and similarly $f(D/((1-C)(1-D)))$ is positive. Since $f(\max(C/(1-C), D/(1-D))) < 0$ as was shown in the proof of Lemma 3.1, and $\min(C/(1-C), D/(1-D)) + CD/((1-C)(1-D)) = \min(C,D)/((1-C)(1-D))$, the result follows.

Q.E.D.

<u>Lemma 3.5</u>: If $\rho < 1$ (as we assume) then also $\gamma < 1$.

<u>Proof</u>: By the definition of D, we have:

$$\begin{aligned} D/(1-D) &= (1 - e^{-\lambda\theta})/(\beta e^{-\lambda\theta}) = (e^{\lambda\theta} - 1)/\beta = (1/\beta)\sum_{i=1}^{i=\infty} (\lambda\theta)^i/i! < \\ &< \sum_{i=1}^{i=\infty} (\lambda\theta)^i/\beta^i = (\lambda\theta/\beta)/(1 - \lambda\theta/\beta) \end{aligned}$$

since $\beta < 1$ and $\lambda\theta/\beta < \rho < 1$ (by (3.11)).

By Lemma 3.4, and since $C/(1-C) = \lambda/\alpha$, we thus get:

$$\gamma < C/(1-C) + (C/(1-C))(D/(1-D)) < \lambda/\alpha + (\lambda\theta/\beta)(\lambda/\alpha)/(1 - \lambda\theta/\beta) .$$

But $\lambda/\alpha + \lambda\theta/\beta < 1$ by (3.11) and thus $(\lambda/\alpha)/(1 - \lambda\theta/\beta) < 1$; therefore $\gamma < \lambda/\alpha + \lambda\theta/\beta < 1$.

Q.E.D.

As a result of Lemmas 3.1 and 3.5, $D/(1-D) < \gamma < 1$.

Lemma 3.6: Let

$$S(k) = \gamma(1-\gamma^k)/(1-\gamma) - \delta(1-\delta^k)/(1-\delta) + \rho(\gamma^k - \delta^k) - CD(\gamma^{k-1} - \delta^{k-1})/((1-C)(1-D)).$$

Then

$$\sum_{j=0}^{j=k} (\pi_j\mu_j + \pi_{\underline{j}}\mu_{\underline{j}}) = S(1)/(3\lambda E_0) \quad \text{for} \quad k = 1$$

$$= S(k)/\lambda \quad \text{for} \quad k \geq 2 .$$

Proof: The proof for $k = 1$ is trivial. As for $k \geq 2$, the result follows from (3.8), (3.11) and Lemma 3.3 by direct summation.

Q.E.D.

Lemma 3.7: The steady state probabilities of the modified system are given for all $k \geq 1$ by:

$$\left.\begin{aligned} P_i &= (E_i - DE_{i-1}/(1-D))/S(k) \\ P_{\underline{i}} &= (DE_{i-1}/(1-D))/S(k) \end{aligned}\right\} (i = 0,\dots,k-1)$$

$$P_k = (C/(1-C)) \times (E_{k-1} - DE_{k-2}/(1-D))/S(k)$$

$$P_{\underline{k}} = \lambda\theta E_{k-1}/\beta .$$

Proof: Immediate from Lemma 3.3, Lemma 3.6, (3.7) and (3.8).

Lemma 3.7 enables us to give the steady state distribution of the number of users in the system. This result is of interest by itself for system's design, since the knowledge of this distribution enables the designer to determine the load on system's resources, like storage devices.

Lemma 3.8: The steady state distribution of the number of users in the system is given by:

$$P(i \text{ users in system of capacity } k) = E_i/S(k) \qquad (i = 0,\dots,k-1)$$

$$P(\text{system filled to capacity } k) = (\rho E_{k-1} - CDE_{k-2}/((1-C)(1-D)))/S(k).$$

Finally, the desired $P(k)$ and $Q(k)$ can be evaluated using Lemma 3.7 and (3.5):

Theorem 3.2: The average number of users in the system at steady state is

$$Q(k) = k - ((\gamma/(1-\gamma))(k-\gamma(1-\gamma^k)/(1-\gamma)) -$$
$$- (\delta/(1-\delta))(k-\delta(1-\delta^k)/(1-\delta)))/S(k)$$

and the proportion of time the system is engaged in giving service at steady state is given by

$$P(k) = ((1-D/(1-D))(\gamma(1-\gamma^k)/(1-\gamma) - \delta(1-\delta^k)/(1-\delta)) + (\gamma^{k+1} - \delta^{k+1}) -$$
$$- (\gamma-\delta) - CD(\gamma^k - \delta^k)/((1-C)(1-D)))/S(k) .$$

We will be interested in the behavior of $P(k)$ and $Q(k)$ as k is changed. The three following lemmas establish this behavior.

Lemma 3.9: $S(k)$ is a strictly increasing function of k.

Proof: Using Lemma (3.6), one gets for $k \geq 1$:

$$S(k+1) - S(k) = \gamma^{k+1} - \delta^{k+1} - \rho\gamma^k(1-\gamma) + \rho\delta^k(1-\delta) +$$

$$+ CD\gamma^{k-1}(1-\gamma)/((1-C)(1-D)) - CD\delta^{k-1}(1-\delta)/((1-C)(1-D))$$

$$= r(\gamma) - r(\delta),$$

where

$$r(z) = z^{k+1} - \rho z^k(1-z) + CDz^{k-1}(1-z)/((1-C)(1-D))$$

$$= z^{k-1}(z^2 - \rho z(1-z) + CD(1-z)/((1-C)(1-D)))$$

$$= z^{k-1}(z(C+D-2CD)/((1-C)(1-D)) - \rho z(1-z))$$

$$= z^k((C/(1-C) + D/(1-D)) - \rho(1-z)),$$

since z represents γ and δ, which are both roots of the quadratic equation of Lemma 3.1.

But, as was already shown in the proof of Lemma 3.5,

$$D/(1-D) = (e^{\lambda\theta} - 1)/\beta = \lambda\theta/\beta + (1/\beta)\sum_{i=2}^{i=\infty} (\lambda\theta)^i/i!$$

$$= \lambda\theta/\beta + (C/(1-C) + D/(1-D))M$$

where

$$M = ((1-C)(1-D)/(C+D-2CD))\sum_{i\geq 2}(\lambda\theta)^i/i! > 0 .$$

Hence

$$\rho = \lambda/\alpha + \lambda\theta/\beta = (C/(1-C) + D/(1-D))(1-M) > 0 .$$

Therefore $r(z) = z^k(C/(1-C) + D/(1-D))(M(1-z) + z) > 0$ for $0 < z < 1$, and is an increasing function of z within this range. Since both γ and δ are positive and less than 1 by Lemmas 3.1 and 3.5, the proof is completed.

Q.E.D.

Lemma 3.10: $Q(k)$ is a strictly increasing function of k, and

$$Q(k) \to ((\gamma/(1-\gamma))^2 - (\delta/(1-\delta))^2)/(\gamma/(1-\gamma) - \delta/(1-\delta)) \quad \text{as} \quad k \to \infty.$$

Proof:

$$\begin{aligned} Q(k+1) - Q(k) = {} & (\gamma(k-\gamma(1-\gamma^k)/(1-\gamma))/(1-\gamma) - \delta(k-\delta(1-\delta^k)/(1-\delta))/(1-\delta))/S(k) \\ & - (\gamma(k-\gamma(1-\gamma^k)/(1-\gamma))/(1-\gamma) - \delta(k-\delta(1-\delta^k)/(1-\delta))/(1-\delta) \\ & - (\gamma^k(\rho\gamma-CD/((1-C)(1-D))) - \delta^k(\rho\delta-CD/((1-C)(1-D)))))/S(k+1). \end{aligned}$$

From the proof of Lemma 3.5 we learn that $\rho\gamma - CD/((1-C)(1-D)) > \gamma^2 - CD/((1-C)(1-D)) > 0$ by Lemma 3.1. Hence the numerator of the second fraction is decreased, while its denominator $S(k+1)$ is increased by Lemma 3.9, with respect to the first fraction. Therefore the second fraction is less than the first, and $Q(k+1) - Q(k) > 0$.

Sending $k \to \infty$ establishes the limit mentioned above.

Q.E.D.

Lemma 3.11: $P(k)$ is a strictly decreasing function of k, and

$$P(k) \to 1 - D/(1-D) - (\gamma-\delta)/(\gamma/(1-\gamma) - \delta/(1-\delta)) \quad \text{as} \quad k \to \infty.$$

Proof: Let A_i be the difference between the numerators of $P(i+1)$ and $P(i)$, and let B_i be the difference between the denominators of $P(i+1)$ and $P(i)$. Then for all $i \geq 1$:

$$A_i = (1 - D/(1-D))(\gamma^{k+1} - \delta^{k+1}) + (\gamma^{k+2} - \gamma^{k+1}) - (\delta^{k+2} - \delta^{k+1}) -$$

$$- CD(\gamma^{k+1} - \gamma^k)/((1-C)(1-D)) + CD(\delta^{k+1} - \delta^k)/((1-C)(1-D))$$

$$= \gamma^k(\gamma^2 - \gamma(D/(1-D) + CD/((1-C)(1-D))) + CD/((1-C)(1-D))) -$$

$$- \delta^k(\delta^2 - \delta(D/(1-D) + CD/((1-C)(1-D))) + CD/((1-C)(1-D)))$$

= after simplification, and using Lemma 3.1

$$= C(\gamma^{k+1} - \delta^{k+1})/(1-C) > 0$$

B_i = using the proof of Lemma 3.9

$$= \gamma^k(C/(1-C) + D/(1-D) - \rho(1-\gamma)) -$$

$$- \delta^k(C/(1-C) + D/(1-D) - \rho(1-\delta)) > 0 \quad \text{by Lemma 3.9.}$$

It is easy to check that $A_i < B_i$. In addition:

$$A_{i+1}/B_{i+1} < A_i/B_i \iff \delta(C/(1-C) + D/(1-D) - \rho(1-\gamma)) < \gamma(C/(1-C) + D/(1-D) - \rho(1-\delta))$$

$$\iff \delta < \gamma \quad \text{which is the case.}$$

These properties of the A_i's and B_i's establish the validity of the lemma. The passage to the limit is then straightforward.

Q.E.D.

3.e. Determination of System's Parameters

In this section the maximization problem facing the system's manager is presented, and the case of linear loss function is studied in detail. The system considered here is still the simple system that was presented in Section 3.a, and the results obtained in the former sections are extensively used.

The system's manager aim is to maximize the average expected revenues per time unit. The expected number of users in the system, in the steady state, is $Q(k)$; each of them pays c_1 money units for any time unit he stays in the system, and the total waiting revenues per time unit are thus $c_1 \times Q(k)$. The system is rendering service to users, excluding overhead, during a proportion $P(k)$ of the time; the system gets c money units per time unit of service, and thus earns $c \times P(k)$ per time unit from service. The total revenues are the sum of the waiting and service revenues; the system's manager problem is thus

$$\text{maximize} \quad R = cP(k) + c_1Q(k) \tag{3.12}$$

over all possible quantum size q, service cost c and waiting cost c_1, where k is determined by Algorithm 3.1.

Unless the functional relation between k and the system parameters is explicitly specified, this formulation cannot be used for the determination of the parameters. This relation is in essence determined by the loss function; therefore it cannot be assured in general that an optimal set of parameters exists.

Consider therefore the linear loss case that was considered in Section 3.c. Using the notation presented there, we learn from (3.3) that the control number k' is such that

$$(B+c_1)W(k') + ch \leq A \quad \text{and} \quad (B+c_1)W(k'+1) + ch > A, \quad (A,B > 0) \tag{3.13}$$

where B is the (utility) loss of the user due to waiting, h is the expected service time, and A is the expected loss if a competing system is joined, taking into account time lost in service, cost in competing system, etc.; both A and B are measured in monetary units. $W(k)$ is the expected waiting time for a user who joins the system when he is the one filling the system to capacity, and was denoted $W(k-1,0,k)$ in Sections 3.b and 3.c.

Clearly, for any given k', the system can adjust the charges c and c_1 so that equality will be achieved in (3.13); such equality is thus assumed. Thus, the system's manager problem is

$$\left.\begin{array}{c} \text{Maximize } R = cP(k) + c_1Q(k) \\ \text{over all possible } q,\ c,\ \text{and } c_1,\ \text{and subject to} \\ (B+c_1)W(k) + ch = A\ . \end{array}\right\} \quad (3.14)$$

<u>Lemma 3.12</u>: In (3.14), $c \le A/h$ and $B+c_1 \ge 0$.

<u>Proof</u>: Incorporating the constraint into R and noting that h is strictly positive, the revenue to the system becomes

$$R = (A - BW(k))P(k)/h + c_1(Q(k) - W(k)P(k)/h)\ . \quad (3.15)$$

If $c_1+B < 0$ then users get waiting incentive; if a user should join when the system is empty, then he should join when the system is partially full. Thus the larger the number of users in the system, the more attractive it becomes to join the system, and the control number will be infinite. The worst possible case, from the user point of view, is to join the system when it is empty. His expected waiting time, W say, is less then the average expected waiting time, which is $Q(\infty)/\lambda$. Hence $Q(\infty) - W(\infty)P(\infty)/h \ge Q(\infty)(1 - P(\infty)/\lambda h)$. But the last term is positive, as can be easily checked from Lemma 3.11, while $c_1 < -B < 0$; the revenues of the system are then less than those it would have received if it had set $c_1 = 0$ and $c = A/h$. Therefore $c_1+B \geq 0$; the bound for c follows immediately.

Q.E.D.

Remark: In the case $-B \leq c_1 \leq 0$ the system is paying waiting incentive, but increased waiting still incurs an increasing loss to the user, and the argument above fails -- unless it can be shown that $Q(k) - W(k)P(k)/h \geq 0$; because then again the system would have profited by putting $c_1 = 0$.

(3.15) gives us a new insight into the system's manager problem. Under this formulation, the manager has c_1 and k as decision parameters, and changes c accordingly. Clearly, if c_1 will be large, c will have to be negative and a 'service incentive' system will result.

It is seen from (3.15) that the optimal set of parameters is completely determined by $Z(k) = Q(k) - W(k)P(k)/h$. If $Z(k)$ is strictly positive for some k, then the system's manager can achieve any positive revenue he likes, by choosing for this k a positive sufficiently big waiting charge, and - if needed - by paying a service incentive. If $Z(k)$ is negative for all possible capacities k, then the system's manager will pay an incentive of B for waiting; the service charge will be A/h, and users will be indifferent to their waiting time. In such a case the revenue for k is $AP(k)/h - BQ(k)$; since $P(k)$ decreases and $Q(k)$ increases as k increases, the revenues will drop as the capacity increases. Hence, if $Z(k)$ is negative for all k, the optimal choice is $k = 1$. If $Z(k)$ is always zero, the value of c_1, and also c, is immaterial. The revenue for k will be $(A - BW(k))P(k)/h$, which decreases as k increases; therefore also in this case the optimal value of k is 1.

These conclusions will hold no more if incentives, paid by the system to the users, are not allowed. Here $0 \leq c \leq (A - BW(k))/h$ and $0 \leq c_1 \leq A/W(k) - B$. The problem becomes for each k a linear programming problem, namely

$$\left.\begin{array}{rl} \text{maximize} & R(k) = cP(k) + c_1 Q(k) \\ \text{subject to} & 0 \leq c \leq (A - BW(k))/h \\ \text{and} & 0 \leq c_1 \leq A/W(k) - B, \end{array}\right\} \qquad (3.16)$$

which is feasible only for k such that $A \geq BW(k)$.

The solution to (3.16) is:

$$\begin{array}{l} \text{Choose } c = 0, \; c_1 = A/W(k) - B \text{ with revenue } Q(k)(A/W(k) - B) \text{ if } Z(k) \geq 0 \\ \text{Choose } c = (A - BW(k))/h, \; c_1 = 0 \text{ with revenue } P(k)(A - BW(k))/h \text{ if } Z(k) \leq 0. \end{array} \qquad (3.17)$$

The manager has to check for each k the value of $A - BW(k)$ and $Z(k)$; this check terminates at the first k for which $A - BW(k)$ is negative. If all the resulting $Z(k)$'s are non-positive, the optimal solution is to choose $k = 1$. Otherwise he compares the solutions involving positive c_1 for those k for which $Z(k)$ was positive.

Now note that $W(k)$ can easily be bounded. On one hand it is clearly less than the waiting time encountered by the pessimist; on the other hand it is at least as large as the expected waiting time in one cycle, which is $k \times (\theta + \beta/\alpha)$. Thus $k \times (\theta/\beta + 1/\alpha) \times \beta \leq W(k) \leq k \times (\theta/\beta + 1/\alpha)$, and $A - BW(k)$ will be negative for any $k \geq K_1 = 1 + [A/(B(\theta/\beta + 1/\alpha)\beta)]$. Also, using Lemmas 3.10 and 3.11, one gets $Z(k) \leq Q(\infty) - \alpha\beta P(\infty)(\theta/\beta + 1/\alpha)k$; hence $Z(k)$ will be negative for any $k \geq K_2 = 1 + [Q(\infty)/(\alpha\beta P(\infty)(\theta/\beta + 1/\alpha))]$.

By the former discussion, the system's manager can always get $R(K_2)$, and will never consider any $k > K_2$. Moreover, if incentives are not allowed, the manager will consider only $k \leq \min(K_1,K_2)$.

It is easy to check, using Theorem 3.1 and Theorem 3.2, that $Z(1) = 0$. For $k = 1$ the system manager is therefore indifferent as to the form of charges to be used -- waiting or service. For $k > 1$, however, he will never use service charges; if $Z(k)$ is positive he will use waiting charges, while a negative $Z(k)$ indicates that system of capacity 1 is preferable to system of capacity k, due to the need to use service charges in such a case, and the resulting drop in revenues.

The following algorithm thus emerges:

<u>Algorithm 3.2</u>: For the determination of system's charges:

a. Set K_2.

b. Set $k = 1$, $R = (A - B(\theta/\beta+1/\alpha))/(1/\lambda+\theta/\beta+1/\alpha)$, $k' = 1$
$c = s(A - B(\theta/\beta+1/\alpha))\alpha$, $c_1 = (1-s)(A/(\theta/\beta+1/\alpha)-B)$, $(0 \leq s \leq 1)$
(this is the exact solution for $k = 1$).

c. Set $k \leftarrow k+1$. Terminate if $k > K_2$.

d. If incentives are now allowed, compute $A - BW(k)$, and terminate if it is negative.

e. Compute $Z(k) = Q(k) - W(k)P(k)\alpha$. Go to step c if negative or zero.

f. Go to step i if incentives are allowed.

g. Set $R' = Q(k)(A/W(k) - B)$. Return to step c if $R' \leq R$.

h. Set $k' \leftarrow k$, $R \leftarrow R'$, $c \leftarrow 0$, $c_1 \leftarrow A/W(k) - B$. Return to step c.

i. Choose c_1 as large as wished, compute R from (3.15) and c from the constraint of (3.14). Set $k' \leftarrow k$. Terminate.

This algorithm will terminate in a finite number of steps. Upon termination, k' is the system's capacity, R is the expected revenue, and c and c_1 are the optimal charges.

The most interesting feature of this solution is the preference of waiting charges to service charges, and, if incentives are not allowed, the usage of one type of charges only. This is a deviation from today's practice of charging for connect time as well as for service time. Seemingly the current practice does not produce the maximal expected revenue for the system; though this disadvantage might be compensated by decreased risk, as measured by the second moment of the revenues. Another provocative idea is the implementation of incentives; though they will no doubt result in a change of service requirements by users.

Finally, the discussion in this section might be used in another context. If the physical environment of the system is given, including the capacity, (3.17) (or (3.15)) might be used for optimal determination of charges. Algorithm 3.2, on the other hand, might be used when a decision has to be made about the capacity of the system.

3.f. Extensions of the Simple Model

In this section the simple model of Section 3.a is extended, and the resulting changes are discussed. Specifically, we discuss random quantum and overhead, the effect of tolls, and general service requirement distribution.

The semi-Markov process of Section 3.d will have to be changed, if the quantum or overhead is random, or if a general service requirement distribution is used. The main changes are in the $Q_{ij}(t)$ of (3.6); these will then be reflected in the μ_i's of (3.8) and the definition of C and D

that are used in (3.9). The system of equations for the steady state probabilities of the embedded Markov chain, (3.9), hold without change. Lemmas 3.1-3.4 remain true even for the extended system. The traffic intensity has a form different from (3.11); Lemma 3.5 has to be proven for the changes. The remainder of Section 3.d, which is devoted to the specific form of $P(k)$ and $Q(k)$, does not hold for the extended system. We conjecture, though, that $Q(k)$ will still be an increasing function of k, while $P(k)$ will be decreasing function of k. This conjecture stems from the very nature of the limited capacity k; as k increases, more users are able to join the system, and the number of rejected users will decrease. The probability of idle state, in which the system is empty, will then increase; as a result the proportion of time devoted to service will decrease. Since fewer users will be rejected, more users will join the system, and the average number of users in the system will increase with increased capacity. On the other hand, the average number of users in the system will converge to a finite value, as long as the traffic intensity will be strictly less than one.

The determination of optimal system's parameters in Section 3.e was based, essentially, on the above mentioned properties of $P(k)$ and $Q(k)$. Thus, all the conclusions of Section 3.e will still hold for such an extended system. The specific values of $W(k)$ are changed, as indicated in Appendix A; but the bounds that were used for $W(k)$ in Section 3.e, or similar bounds, still hold.

The inclusion of tolls in the model has different effects. From the user's point of view, the change will indeed be small; e.g. in the linear loss case, only A of (3.2) will change. The revenues from tolls are proportional to the number of users who join the system in a time unit, and determination of this number is needed. But in steady state, the flow of users into the system is equal to the flow of users out of the system; if ES is the expected total time devoted by the system to a user, including service and overhead, then $1/ES$ is the expected rate of departures, and $Q(k)/ES$ will be the expected number of departing, and joining, users. Denoting by t the toll, the revenues from tolls will thus be $tQ(k)/ES$; the toll might then be combined with the waiting charges. In the linear loss case, for example, the resulting linear programming problem (3.14) and (3.16) will be a bit more complicated; but the essence of the discussion of Section 3.e, and the form of Algorithm 3.2, will remain unchanged.

Finally, let us consider the case of different urgency values, assuming there is a finite number of these. To each urgency value, and for the same set of system's parameters, there will correspond a control number. Let $k_1,\ldots,k_m$ be the resulting capacities, and let K be the largest of them. For $P(k)$ and $Q(k)$ a system of capacity K will have to be considered, whose transition probabilities $Q_{ij}(t)$ will be much more complicated than those of (3.6). Still, $P(K)$ and $Q(K)$ can be computed for such a system, in a way similar to that of Section 3.d. The simple linear programming problem (3.14) and (3.16) of Section 3.e will be replaced by one with many constraints, one constraint for each urgency value. In this case the determination of charges will be much more complicated. Still it is conjectured that for each urgency value u there will correspond a maximal capacity k_u, and the system's manager will have to consider for urgency value u capacities not exceeding k_u. The search for optimal parameters will have to be conducted in an m-dimensional finite grid of capacities;

but it still is a finite search, in which for each point in the grid, i.e. possible configuration of capacities, there exist an optimal solution. By this procedure a set of optimal charges might be found for this case.

CHAPTER 4

A MULTI-ENTRANCE, FOREGROUND-BACKGROUND, TIME-SHARING SYSTEM, WITH RANDOM ENTRANCE

4.a. Introduction

This chapter deals with the generalized multi-entrance, time-sharing system that was proposed in Section 2.a, under the assumption of random entrance to all the levels maintained by the system. The analysis follows closely Adiri's [1] treatment of a multi-entrance FB_∞ time-sharing system. As Adiri notes in the single-entrance models "all customers upon arrival are treated in the same way. In real life situations this is not always desirable and differentiation among certain groups of customers is preferred. ...There are cases when it is desirable to couple the improvement of service to certain groups with discouragement to all jobs that have a long service requirement, the length of service requirement is not known in advance. This might be carried out by dynamically degrading the degree of customer's priority according to the length of his service requirement." The system Adiri proposes is a multi-entrance FB_∞ system, in which a j-priority customer joins the system at the jth level. In Adiri's model, though, only one queue is maintained at each level; thus it is impossible to distinguish among users in any level, and all users in each level have the same priority. The model that was proposed in Section 2.a, which is analyzed in this chapter, differs in this respect. In the proposed model the joining priority is 'carried along' with the user, so that preferred groups of users always get an improved service, though the service of all users is degraded in proportion to the length of their service requirements.

In Sections 4.b and 4.c a simplified version of this model is analyzed, in which the quanta and overhead have a fixed length, and the interarrival times and service requirements distributions are exponential. In Section 4.d these assumptions are relaxed: random quanta and overhead are discussed, and the model is extended to general service requirements distribution. Adiri's model cannot be extended to a general service requirements distribution, at least not without a major effort; this is due to the mixing of users with different joining levels in higher numbered, and lower priority, levels. The proposed model, on the other hand, can easily be extended to general service requirements distribution, because the residual service requirements distribution is always known, for any level a user might reach. In addition, several special cases of the model are discussed in Section 4.d.

4.b. Description and Notation

The system maintains a (finite or infinite) sequence of N queues. Each queue is identified by a pair of integers: the first denotes the level of the queue, and the second identifies the level at which users in the queue joined the system. The service to users is given in quanta: when a user in the (n,k) queue is chosen to be served, he first gets a fixed overhead of length θ_{nk}, and then a quantum of service of size q_{nk}. If his service requirement is fulfilled within this quantum, he leaves the system immediately upon fulfillment of his requirement; otherwise he is transferred to the end of the $(n+1,k)$ queue. If N is finite, then the quanta in the last level are assumed to be infinite.

Users arrive to the system in a Poisson process of rate λ; upon arrival a user joins the ith level with probability p_i. Thus, the arrival process to the ith level is Poisson with rate $\lambda_i = \lambda p_i$. A user joins the ith level through the (i,i) queue. The service requirements of users are distributed exponentially with rate α; service requirements and inter-arrival times are assumed to be independent.

Denote by β_{nk} the probability that a user who starts getting service in the q_{nk} quantum will terminate his service requirement during this quantum, and will leave the system. In the simplified model,

$$\beta_{nk} = 1 - e^{-\alpha q_{nk}} . \tag{4.1}$$

Denote by Q_{nk} the total amount of service given to a user in the (n,k) queue, including overhead and service. Then

$$Q_{nk} = \begin{cases} \theta_{nk} + q_{nk} & \text{with probability} \quad 1 - \beta_{nk} \\ \\ \theta_{nk} + D_{nk} & \text{with probability} \quad \beta_{nk} \end{cases} \tag{4.2}$$

where D_{nk} has the density

$$f_{D_{nk}}(x) = (1/\beta_{nk})\alpha e^{-\alpha x}, \qquad (0 \le x \le q_{nk}) . \tag{4.3}$$

Then:

$$\left.\begin{aligned} E(D_{nk}) &= (1/\alpha) - q_{nk}(1 - \beta_{nk})/\beta_{nk} \\ E(Q_{nk}) &= \theta_{nk} + \beta_{nk}/\alpha \\ E(Q^2_{nk}) &= (2\beta_{nk}(1 + \alpha\theta_{nk})/\alpha^2) + \theta^2_{nk} - (2q_{nk}(1 - \beta_{nk})/\alpha) \\ \text{for } n \ge k, \text{ and } & E(D_{nk}) = E(Q_{nk}) = E(Q^2_{nk}) = 0 \text{ for } n < k . \end{aligned}\right\} \tag{4.4}$$

Defining s_{nk} as the time supplied to a user in queues (k,k), $(k+1,k),\ldots,(n,k)$ given that the user reaches the $(n+1,k)$ queue, and γ_{nk} as the probability of this event, one has

$$s_{nk} = \sum_{i=k}^{i=n} (\theta_{ik} + q_{ik}) \quad \text{if} \quad n \geq k, \quad \text{and zero otherwise,} \tag{4.5}$$

and

$$\gamma_{nk} = \Pi_{i=k}^{i=n} (1 - \beta_{nk}) \quad \text{if} \quad n \geq k, \quad \text{and} \quad 1 \quad \text{otherwise} . \tag{4.6}$$

The arrival rate to the queue (n,k) is then

$$\lambda_{nk} = \lambda_k \gamma_{n-1,k} \quad \text{if} \quad n \geq k, \quad \text{and zero otherwise} . \tag{4.7}$$

The proportion of time the system is busy delivering service in the (n,k) queue is in steady state

$$\rho_{nk} = \lambda_{nk} E(Q_{nk}) \tag{4.8}$$

and the proportion of time the system is busy delivering service in the first n levels is

$$\rho_n = \sum_{i=1}^{i=n} \sum_{k=1}^{k=i} \rho_{ik} . \tag{4.9}$$

Thus, the whole system's nonsaturation condition is $\rho_N < 1$. Increasing ρ_N over 1 will result in gradual saturation of the queues, in descending order of their number. If $\rho_1 \geq 1$ the whole system is saturated; and if there exist an integer n such that $\rho_n < 1$ and $\rho_{n+1} \geq 1$, then the first n levels are unsaturated, and all service requirements which can be satisfied in these levels are satisfied in a finite time. As Adiri [1] notes, this gradual saturation is one of the advantages of the FB_N, as well as of the proposed system, over other scheduling disciplines of queueing systems. In the sequel, it will be assumed that

$$\rho_N < 1, \tag{4.10}$$

and that the system is in steady state.

Finally, let U_{nk} be the time the system is busy serving a user in between the (k,k) queue and the (n,k) queue; that is, in the (k,k), (k+1,k),...,(n,k) queues. Then

$$U_{nk} = \begin{cases} s_{ik} + \theta_{i+1,k} + D_{i+1,k} & \text{with probability } \gamma_{ik}\beta_{i+1,k}, \quad k-1 \leq i \leq n-1 \\ s_{nk} & \text{with probability } \gamma_{nk} \\ 0 & \text{for } n < k \end{cases} \tag{4.11}$$

Evidently U_{kk} is distributed as Q_{kk}.

<u>Lemma 4.1</u>: $\rho_n = \sum_{k=1}^{k=n} \lambda_k E(U_{nk})$ for $n \geq 1$.

<u>Proof</u>: From (4.11) one gets, using (4.4) and (4.5)

$$\begin{aligned} E(U_{nk}) = {} & \sum_{i=k-1}^{i=n-1} \gamma_{ik}\beta_{i+1,k} \sum_{j=k}^{j=i+1} \theta_{jk} + \gamma_{nk} \sum_{i=k}^{i=n} \theta_{ik} \\ & + \sum_{i=k-1}^{i=n-1} \gamma_{ik}\beta_{i+1,k} \sum_{j=k}^{j=i+1} q_{jk} + \gamma_{nk} \sum_{i=k}^{i=n} q_{ik} \\ & - \sum_{i=k-1}^{i=n-1} \gamma_{ik} q_{i+1,k} + \sum_{i=k-1}^{i=n-1} \gamma_{ik}\beta_{i+1,k}/\alpha \; . \end{aligned} \tag{4.12}$$

However, for any set of numbers $a_k,\dots,a_n$

$$\sum_{i=k-1}^{i=n-1} \gamma_{ik}\beta_{i+1,k} \sum_{j=k}^{j=i+1} a_j + \gamma_{nk} \sum_{i=k}^{i=n} a_i$$

$$= \sum_{i=k}^{i=n} a_i(\gamma_{nk} + \sum_{j=i}^{j=n} \gamma_{i-1,k}\beta_{ik}) = \text{by (4.6)} = \sum_{i=k}^{i=n} \gamma_{i-1,k}a_i \ .$$

Hence

$$E(U_{nk}) = \sum_{i=k}^{i=n} \gamma_{i-1,k}(\theta_{ik} + \beta_{ik}/\alpha) \qquad (4.13)$$

which, together with (4.4), (4.7) and (4.8) yield the desired result.

Q.E.D.

Lemma 4.1 has a very intuitive interpretation. ρ_n, as derived in (4.9), gives the proportion of time the system is busy delivering service in the first n levels, when all the users in each level are considered. In Lemma 4.1, on the other hand, the users entering each level are considered and tracked through all the levels they pass until reaching the nth level, and the proportion of time devoted to service them is recorded. It seems intuitively clear that both ways of "measuring" the service rendered by the system will give the same result.

4.c. Expected Flow Time

Define T_j to be the total flow time of a user who joins the system through the jth level; i.e., the time interval between the time epochs of joining the system and departing from it. Similarly, let W_{nk} be the time interval elapsing between the time epochs a k-user joins the nth level and his admission to service in this level, where a k-user is a user who joined the system through the kth level. Thus, a k-user whose service requirement is t time units, will have an expected flow time of

$$E(T_k|t) = \sum_{i=k}^{i=n} (E(W_{ik}) + \theta_{ik}) + t \text{ time units,} \tag{4.14}$$

where n is the highest level he will reach, and is given by

$$n = \min(j: \sum_{i=k}^{i=j} q_{ik} \geq t) . \tag{4.15}$$

The expected total flow time of a user who joins the kth level is then achieved by integrating (4.14) over all possible service requirements; mixing this result according to the probabilities of joining the various levels, yields the expected flow time of a (random) user who joins the system.

It is evident from (4.14) that the expected flow time, as well as the expected waiting time, are completely known once the $E(W_{ik})$'s are known. Accordingly, this section is concerned mainly with the $E(W_{ik})$'s; the discussion starts with $E(W_{kk})$, continues with $E(W_{k+1,k})$, and at last the general case of $E(W_{ik})$ is solved.

Suppose that at the moment a k-user joins the system, the arrival process to higher priority levels terminates immediately. W_{kk} is then the sum of two time intervals: the first one is the time needed to complete the quantum given to the user being served at the moment of arrival, and the second one is the time needed to serve all users who are waiting in higher priority queues and in the (k,k) queue at the moment of arrival. Denote these time intervals by t_1 and t_2, respectively.

Since the arrival process is Poisson, ρ_{ni} may be interpreted as the probability that an arriving user finds a non-empty (n,i) queue. Thus

$$E(t_1) = \sum_{n\geq 1} \sum_{i=1}^{i=n} \rho_{ni} E(Q_{ni}^2)/(2E(Q_{ni})) = \frac{1}{2} \sum_{n\geq 1} \sum_{i=1}^{i=n} \lambda_{ni} E(Q_{ni}^2) \qquad (4.16)$$

(see, e.g. Conway, et al. [9], p. 147).

Let N_{ni} be the number of users waiting in the (n,i) queue at a random moment of time, and let V_{nik} be the time the system is busy serving a k-user between the (i,k) queue and the (n,k) queue; that is, in the (i,k), (i+1,k),...,(n,k) queues. Then

$$V_{nik} = \begin{cases} s_{jk} - s_{i-1,k} + \theta_{j+1,k} + D_{j+1,k} & \text{with probability } \gamma_{jk}\beta_{j+1,k}/\gamma_{i-1,k} \\ & \text{for } i-1 \leq j \leq n-1 \\ s_{nk} - s_{i-1,k} & \text{with probability } \gamma_{nk}/\gamma_{i-1,k} \,. \end{cases} \qquad (4.17)$$

Clearly $V_{nkk} = U_{nk}$, and

$$E(V_{nik}) = (E(U_{nk}) - E(U_{i-1,k}))/\gamma_{i-1,k} \,. \qquad (4.18)$$

Each of the N_{ni} users joined the system at the (i,i) queue, and contributes to t_2 the service time in the queues (n,i),(n+1,i),...,(k,i) -- that is the time V_{kni}. Therefore

$$E(t_2) = \sum_{n=1}^{n=k} \sum_{i=1}^{i=n} E(N_{ni})E(V_{kni}) \qquad (4.19)$$

where

$$E(N_{ni}) = \lambda_{ni}E(W_{ni}) \qquad (4.20)$$

by Little's theorem.

In the development above we ignored the arrival process to higher priority queues from the arrival moment on. Taking this process into account, we note that W_{kk} is distributed as a busy period commencing with a workload of $t_1 + t_2$, and terminating when the first k levels are empty. Using known results (e.g., Conway, et al. [9], p. 151) we find that $E(W_{kk}) = E(t_1 + t_2)/(1 - \rho_k)$. Combining this result with (4.16)-(4.20), we get

$$E(W_{kk}) = \Big(\sum_{n=1}^{n=k} \sum_{i=1}^{i=n} \lambda_{ni}\big(\tfrac{1}{2} E(Q_{ni}^2) + E(W_{ni})(E(U_{ki}) - E(U_{n-1,i}))/\gamma_{n-1,i}\big) \tag{4.21}$$

$$+ \tfrac{1}{2} \sum_{n>k} \sum_{i=1}^{i=n} \lambda_{ni}E(Q_{ni}^2)\Big)/\Big(1 - \sum_{n=1}^{n=k} \sum_{i=1}^{i=n} \lambda_{ni}E(Q_{ni})\Big), \quad (k > 1).$$

For $k = 1$, there are no considerations of a busy period since any arrival to the system after the tagged user is served after the tagged user receives his first quantum. Therefore

$$E(W_{11}) = \tfrac{1}{2} \sum_{n\geq 1} \sum_{i=1}^{i=n} \lambda_{ni}E(Q_{ni}^2) + \lambda_{11}E(W_{11})E(V_{111}) .$$

But $E(V_{111}) = E(U_{11}) = E(Q_{11})$, and therefore

$$E(W_{11}) = \tfrac{1}{2} \sum_{n\geq 1} \sum_{i=1}^{i=n} \lambda_{ni}E(Q_{ni}^2)/(1 - \lambda_{11}E(Q_{11})) . \tag{4.22}$$

Consider now $E(W_{k+1,k})$. At the moment a k-user's service starts after his wait in the (k,k) queue, all the first k levels are empty, except perhaps the (k,k) queue. $W_{k+1,k}$ is affected by three time intervals, and the busy periods associated with them. First, it is affected by the users who joined the (k,k) queue after the tagged k-user joined the system. There are $\lambda_{kk}E(W_{kk})$ such users, each contributing expected time of $E(Q_{kk})$; these users thus contribute

$$E(t_1) = \lambda_{kk} E(W_{kk}) E(Q_{kk}) \ . \tag{4.23}$$

Second, it is affected by the users who joined the first k queues of the $k+1$ level before the tagged user. This group of users can be partitioned into three types; the $N_{k+1,n}$ users who were in queue $(k+1,n)$ when the tagged user joined the system, the users who were in higher priority queues than the tagged user when he joined the system, and the users who joined higher priority queues while the tagged user waited in the (k,k) queue. There were N_{in} users in the (i,n) queue when the tagged user joined the system, and each of them will reach the $(k+1,n)$ queue with probability $\gamma_{kn}/\gamma_{i-1,n}$. $\lambda_{nn} E(W_{kk})$ users did join the system at the (n,n) queue while the tagged user waited in the (k,k) queue, and they will reach the $(k+1,n)$ queue with probability γ_{kn}. Thus, these users contribute

$$E(t_2) = \sum_{n=1}^{n=k} \left(E(N_{k+1,n}) + \lambda_{nn} E(W_{kk}) \gamma_{kn} + \sum_{i=n}^{i=k} E(N_{in}) \gamma_{kn}/\gamma_{i-1,n}\right) E(Q_{k+1,k})$$
$$- \lambda_{kk} \gamma_{kk} E(W_{kk}) E(Q_{k+1,k}) \ . \tag{4.24}$$

Notice that in (4.24) the contribution of the users who joined the (k,k) queue after the tagged user was eliminated, since these users will be served in the $k+1$ level after this tagged user.

Taking now the arrival process to higher priority queues into account, and noticing that such queues include the first k level queues and the first k queues of the $k+1$ level, we find that the workload contributing to $W_{k+1,k}$ consists of the busy period generated by $t_1 + t_2 + \theta_{kk} + q_{kk}$ and users who join such higher priorities queues. Thus

$$E(W_{k+1,k}) = E(t_1 + t_2 + \theta_{kk} + q_{kk})/(1 - \rho_{k+1} + \rho_{k+1,k+1}) - (\theta_{kk} + q_{kk}) .$$

Using (4.20)-(4.24), one thus gets

$$\begin{aligned} E(W_{k+1,k}) = {} & (\lambda_{kk}E(W_{kk})(E(Q_{kk}) - \gamma_{kk}E(Q_{k+1,k})) + \theta_{kk} + q_{kk} \\ & + \textstyle\sum_{n=1}^{n=k} E(Q_{k+1,n})(E(N_{k+1,n}) + \lambda_{nn}\gamma_{kn}E(W_{kk}) \qquad (4.25) \\ & + \textstyle\sum_{i=n}^{i=k} E(N_{in})\gamma_{kn}/\gamma_{i-1,n}))/(1 - \rho_{k+1} + \rho_{k+1,k+1}) - (\theta_{kk} + q_{kk}). \end{aligned}$$

In the general case of $W_{k+j+1,k}$, $(j \geq 1)$, one has again to distinguish between users who joined the $k+j+1$ level before the tagged user, and users who reached the $(k+j,k)$ queue after the tagged user. Consider first the users in the $(k+j,k)$ queue. The expected time contributed by these users is then

$$E(t_1) = \lambda_{kk}\gamma_{k+j-1,k}E(Q_{k+j,k})(s_{k+j-1,k} + \textstyle\sum_{n=k}^{n=k+j} E(W_{n,k})) .$$

In the $(k+j+1,n)$ queue, where $1 \leq n \leq k$, there are again three types of users; each of whom contributes to the expected time $E(Q_{k+j+1,n})$. There are $N_{k+j+1,n}$ users who were in this queue when the tagged user joined the system. Of the N_{in} $(n \leq i \leq k+j)$ users who were in the (i,n) queue when the tagged user joined the system, $N_{in}\gamma_{k+j,n}/\gamma_{i-1,n}$ will reach the $(k+j+1,n)$ queue; this source thus adds $\sum_{i=n}^{i=k+j} N_{in}\gamma_{k+j,n}/\gamma_{i-1,n}$ users. Lastly there are the users who joined the system while the tagged user waited, and was served, until he reached the commencement of his quantum after the $(k+j,k)$ queue. The expected number of such users who reach the $(k+j+1,n)$ queue is $\lambda_{nn}\gamma_{k+j,n}(s_{k+j-1,k} + \sum_{n=k}^{n=k+j} E(W_{n,k}))$.

Adding the contribution of the k+1 level to t_1, and continuing in the same way as led to (4.25), we thus get

$$E(W_{k+j+1,k}) = (\theta_{k+j,k+j} + q_{k+j,k+j} + \lambda_{kk}(\gamma_{k+j-1,k}E(Q_{k+j,k})$$

$$- \gamma_{k+j,k}E(Q_{k+j+1,k}))(s_{k+j-1,k} + \sum_{i=k}^{i=k+j} E(W_{ik}))$$

$$+ \sum_{n=1}^{n=k} (\lambda_{nn}\gamma_{k+j,n}(s_{k+j-1,k} + \sum_{i=k}^{i=k+j} E(W_{ik})) \qquad (4.26)$$

$$+ \sum_{i=n}^{i=k+j+1} E(N_{in})\gamma_{k+j,n}/\gamma_{i-1,n})E(Q_{k+j+1,n}))/$$

$$/(1 - \rho_{k+j} - \sum_{n=1}^{n=k} \rho_{k+j+1,n}) - (\theta_{k+j,k+j} + q_{k+j,k+j}),$$

$$(1 \leq j) .$$

Noticing that (4.25) is the special case of (4.26) with $j = 0$, and using (4.20), the results obtained above can be summarized as follows:

<u>Theorem 4.1</u>: Let $W_{k+j,k}$ for $j \geq 0$ be the time interval elapsing between the time epochs a k-user joins the (k+j)th level and his admission to service in this level. Then the expected value of $W_{k+j,k}$ is:

(1) For $j = 0$:

$$(\tfrac{1}{2} \sum_{n\geq 1} \sum_{i\geq n} \lambda_{in}E(Q_{in}^2) + (1 - \delta_{1k})(\sum_{n=1}^{n=k} \sum_{i=n}^{i=k} \lambda_{in}E(W_{in})(1/\gamma_{i-1,n}) \cdot$$

$$\cdot \sum_{m=i}^{m=k} \gamma_{m-1,n}E(Q_{mn})))/(1 - \sum_{n=1}^{n=k} \sum_{i=n}^{i=k} \lambda_{in}E(Q_{in})) .$$

(2) For $j > 0$:

$$(\theta_{k+j-1,k+j-1} + q_{k+j-1,k+j-1} + \lambda_{kk}(\gamma_{k+j-2,k}E(Q_{k+j-1,k}) - \gamma_{k+j-1,k}E(Q_{k+j,k}))\cdot$$

$$\cdot (s_{k+j-2,k} + \sum_{i=k}^{i=k+j-1} E(W_{ik})) + \sum_{n=1}^{n=k} (\lambda_{nn}\gamma_{k+j-1,n}(s_{k+j-2,k} +$$

$$+ \sum_{i=k}^{i=k+j-1} E(W_{ik})) + \sum_{i=n}^{i=k+j} E(W_{in})\lambda_{in}(\gamma_{k+j-1,n}/\gamma_{i-1,n}))E(Q_{k+j,n}))/$$

$$/(1 - \sum_{n=1}^{n=k+j-1} \sum_{i=n}^{i=k+j-1} \lambda_{in}E(Q_{in}) - \sum_{n=1}^{n=k} \lambda_{k+j,n}E(Q_{k+j,n})) \ .$$

In the derivation of $W_{k+j,k}$ for $j > 0$, we considered always the time epoch in which the k-user starts getting service in the (k+j-1,k) queue. This epoch was chosen, since at this time epoch all the first k+j-2 levels are empty, as well as the first k-1 queues of the k+j-1 level; while at the (k+j-1,k) queue all waiting users are behind the tagged user.

A direct expression for $E(W_{in})$ is not possible. Theorem 4.1, though, provides us with a way of solving them recursively; each $E(W_{in})$ is a linear combination of $E(W_{jm})$ where $1 \leq j \leq i$ and $1 \leq m \leq n$, while $E(W_{11})$ is given directly. Hence we can solve for $E(W_{21}), E(W_{22})$, $E(W_{31}), E(W_{32}), E(W_{33}), E(W_{41}), \ldots$ in this order.

4.d. Extensions and Special Cases

In this section the assumptions of the simplified model will be relaxed. First the model is extended by allowing general service requirement distribution, and then by the addition of a RR queue into which are transferred all users whose service requirement are not fulfilled in the Nth level. We then discuss the changes required in case of random quanta and/or random overhead, and, finally, present two special cases of the simplified model.

An examination of the development in Section 4.c reveals that the only places in which a specific form of the service requirement distribution was used are (4.1) and (4.4). Thus, any service requirement distribution may be used, as long as (4.4) is changed accordingly. In addition, the proof of Lemma 4.1 has to be changed, although the lemma still holds.

Similarly, only a small change is needed in case random quanta and/or random overhead are used. In this case we distinguish between a "long" quantum in which the user does not depart and is transferred to the next queue, and a "short" quantum at the end of which the user leaves the system. A long quantum is then used in (4.5), while a short quantum is used in (4.11). In addition, a long quantum replaces $\theta_{k+j,k+j} + q_{k+j,k+j}$ in (4.25), (4.26) and Theorem 4.1. The derivation of the expected length of long and short quanta is straightforward, and will be omitted (see e.g., Adiri [1]).

At last, we note that the development given above will hold even if a RR queue is added to the system, into which are transferred all the users whose service was not terminated at the Nth level. The arrival process to this queue will evidently be a Poisson process, and the expected waiting time in this queue may be easily found (Adiri & Avi-Itzhak [2]). The expected waiting time in the other queues of the system will not be changed in form, though the numerical values might be changed due to the needed change in (4.16).

Consider now the special case where each user's service is not interrupted; i.e. where each quantum has an infinite length. Then, denoting in parentheses the appropriate equation, we have:

β_{nk} is 1 for all n and k, γ_{nk} (4.6) is one if $n < k$ and zero otherwise, s_{nk} (4.5) is identically zero since the condition defining it is never fulfilled, and ((4.11) and (4.17)) $EU_{nk} = E(V_{nik}) = E(Q_{kk})$ if $n = i = k$, and zero otherwise. In this case only $E(W_{kk})$ is meaningful, while $E(W_{nk})$ for $n \neq k$ has no meaning, since users will never join any (n,k) queue where $n \neq k$. Clearly λ_{nk} (4.7) equals λ_k if $n = k$, and is zero otherwise. Thus we can ignore the second subscript, and get:

$$E(W_1) = \frac{1}{2} \sum_{n\geq 1} \lambda_n E(Q_n^2)/(1 - \lambda_1 E(Q_1)) \tag{4.27}$$

and

$$E(W_k) = (\frac{1}{2} \sum_{n\geq 1} \lambda_n E(Q_n^2) + \sum_{n=1}^{n=k} \lambda_n E(W_n) E(Q_n))/(1 - \sum_{n=1}^{n=k} \lambda_n E(Q_n)), \tag{4.28}$$

$$(k \geq 2).$$

The same results are achieved from Adiri's [1] paper under the assumption of infinite quanta.

Two special cases of (4.27) and (4.28) are the M/M/1 queueing system in which $\lambda_k = 0$ for $k > 1$, and the M/M/1 system with two priority queues and head of line discipline, in which case $\lambda_k = 0$ for $k > 2$. In both cases the overhead is identically zero. Simple calculations will reveal that the known results for these cases are achieved from (4.27) and (4.28).

It was shown that many known results in queueing theory may be derived from this model. Specifically, all the expected waiting times, and expected number of users, in the various queues of a head of the line priority queueing system may be derived from it, even for the case of a general service requirement distribution.

CHAPTER 5

A MULTI-ENTRANCE, FOREGROUND-BACKGROUND, TIME-SHARING SYSTEM, WITH ENTRANCE BASED ON PRICE

5.a. Introduction

This chapter, like its predecessor, deals with the generalized multi-entrance time-sharing system that was proposed in Section 2.a; but in contrast to its predecessor, a cost structure is imposed on the system and the optimal joining policy of Theorem 2.1 is used. A linear utility loss function is assumed throughout the chapter. In Section 5.b several consequences of this assumption are investigated. Specifically, it is shown that a newly arriving user will not join an entry level, if at the arrival epoch there is a higher priority level in which users are waiting for service. In Section 5.c upper bounds for the control numbers and the capacity of the system are derived. Sections 5.d and 5.e deal with a simplified system, in which the arrival process is a Poisson process, and service requirements, quanta and overhead periods are all exponentially distributed. The problem faced by the users, that of determining the control numbers for a given set of system's parameters, is transformed in Section 5.d into a semi-Markov decision process; this process has a finite action space and, due to the upper bounds of Section 5.c, a finite state space. The process can therefore be transformed into a linear programming problem, a task that is carried on in Section 5.d. In Section 5.e it is shown that there is at least one set of system's charges that maximizes the expected discounted revenues of the system, and that in any such set service is not charged by the system.

5.b. Control Numbers for a Linear Loss Function

When the loss function is linear, it can be written as

$$L_i(u,T) = n_0u + n_1t_i + n_2E(h) + n_3cE(h) + n_4E(W_i(T)) + n_5c_iE(W_i(T)) \quad (5.1)$$

where h is the service time, $W_i(T)$ is the response time in steady state for a user who joins the ith level at state T, u is the urgency value, c is the service charge, c_i is the waiting charge and the linear combination coefficients $n_0,\ldots,n_5$ are non-negative due to the monotonicity of the loss function.

Since n_1, n_3 and n_5 express the relation between monetary units and utility units, we expect them to be equal; similarly, we expect n_2 to be equal to n_4, because both express the relation between time units and utility units. Equation (5.1) may therefore be written as

$$L_i(u,T) = n_0u + (n_2+n_1c)E(h) + n_1t_i + (n_2+n_1c_i)E(W_i(T)). \quad (5.2)$$

If n_1t_i and $n_2+n_1c_i$ are identically zero the loss becomes $D_u = n_0u + (n_2+n_1c)E(h)$, and a user with urgency value u will always join the system if $D_u \leq L_0$, and always balk otherwise; there will not be any control numbers, and newly arriving users will be indifferent with respect to the possible entry queues. The analysis of Chapter 4 is thus appropriate for this case, and it will not be considered in the sequel.

Assuming that all newly arriving users have the same urgency value, (5.2) can be further simplified to

$$L'_i(T) = A_i + B_iE(W_i(T)), \quad (5.3)$$

where $A_i = n_1 t_i$, $B_i = n_2 + n_1 c_i$, and the loss in the competing system is $L_0' = L_0 - n_0 u - (n_2 + n_1 c)E(h)$; L_0' is assumed to be positive, because otherwise all newly arriving users will balk and the system will be always empty.

Suppose that a newly arriving tagged user finds all the levels below $\underline{i}$ empty, while the $\underline{i}$th level is not empty; specifically, assume that the $(\underline{i},\underline{i})$ queue is not empty, and/or that the user currently being served was taken from the $(\underline{i},\underline{i})$ queue. Also, let T' be the state of the system at the time the last user in the $(\underline{i},\underline{i})$ queue, denoted the i-user, joined the system, and T be the state when the tagged user arrives to the system. Finally, let $\underline{k} \geq \underline{i}$.

During the time interval in which the system advanced from T' to T, users in any level above $\underline{i}$ were not served, because the $\underline{i}$ level, and possibly lower levels, were not empty. On the other hand, users from lower levels may have been transferred to levels above $\underline{i}$. Thus the waiting time resulting from users in levels lower than $\underline{i}$ is eliminated from both $W_{\underline{i}}(T)$ and $W_k(T)$, as compared to $W_{\underline{i}}(T')$ and $W_k(T')$, while $W_k(T)$ is much more affected than $W_{\underline{i}}(T)$ by the users in levels above $\underline{i}$. Hence stochastically

$$W_k(T)/W_{\underline{i}}(T) \geq W_k(T')/W_{\underline{i}}(T') \ . \tag{5.4}$$

The i-user joined the $\underline{i}$ level because

$$L_{\underline{i}}'(T') = A_{\underline{i}} + B_{\underline{i}}E(W_{\underline{i}}(T')) < L_k'(T') = A_k + B_k E(W_k(T')) \tag{5.5}$$

and since $A_{\underline{i}} \geq A_k$, due to the decreasing monotonicity of the tolls,

$$B_{\underline{i}}E(W_{\underline{i}}(T')) < B_k E(W_k(T')) \ . \tag{5.6}$$

From (5.4) and (5.6) we have

$$B_k(E(W_k(T)) - E(W_k(T'))) > B_{\underline{i}}(E_{\underline{i}}(W_{\underline{i}}(T)) - E(W_{\underline{i}}(T'))) \tag{5.7}$$

which, when combined with (5.5), implies $L'_{\underline{i}}(T) < L'_k(T)$.

Consider next the case when all levels below $\underline{i}$ are empty, as well as the $(\underline{i},\underline{i})$ queue, but the $\underline{i}$ level is not empty due to a user in a queue $(\underline{i},j)$, or because the user currently being served was taken from the $(\underline{i},j)$ queue where $1 \leq j \leq \underline{i}$. Similar argumentation then shows that $L'_j(T) < L'_k(T)$.

The discussion above is summarized in

Lemma 5.1: When the loss function is linear, a newly arriving user will not join any level k, if a higher priority level is not empty at the time epoch of his arrival.

5.c. Upper Bounds for the Control Numbers

Upper bounds for the control numbers are derived in this section. In order to get these bounds we compute the expected response times of a newly arriving tagged user for all possible entry queues, under the assumption that the arrival stream is shut off immediately after the arrival epoch of the tagged user. The computed response times under this assumption are lower bounds for the expected response times without it, under any possible set of control numbers. The computed response times lead to lower bounds on the loss, which, as implied from Section 2.c, give upper bounds for the control numbers. Finally, we treat two special cases: a M/M/1 system with linear loss in which all quanta are identical and all the overheads are identical, and a system in which a user is served to completion. Throughout this section we use the notation of Section 4.b, and follow the development of Section 4.c.

Let N_{ni} be the number of users waiting in the (n,i) queue at the arrival epoch of the tagged user, and assume that the tagged user joins the system at the kth level. If the user being served at the arrival epoch was chosen from the (i,j) queue, then N_{ij} includes this user, and we denote by $E(t)$ the difference between his expected remaining time during the current quantum and $E(Q_{ij})$. This convention will assure that the contribution of this user to the response time of the tagged user is correctly taken in account, while he still can be treated as any other user who waits in the system at the arrival epoch.

The probability that the tagged user leaves the system after being served in the (n,k) queue, where $n \geq k$, is $\gamma_{n-1,k}\beta_{nk}$; in this case his response time includes waiting in the $(k,k),\ldots,(n,k)$ queues, overheads in these queues, and a full quantum of service in each of these queues except the last one. Thus the expected response time is

$$\begin{aligned}
E(T_k) &= E(t) + \sum_{i=k}^{i=\infty} \Big(E(W_{ik}) + \theta_{ik} + E(D_{ik}) + \sum_{j=k}^{j=i-1} (E(W_{jk}) + \theta_{jk} + q_{jk})\Big)\gamma_{i-1,k}\beta_{ik} \\
&= E(t) + \sum_{i=k}^{i=\infty} (E(W_{ik}) + \theta_{ik} + q_{ik})\sum_{j=i}^{j=\infty} \gamma_{j-1,k}\beta_{jk} + \\
&\quad + \sum_{i=k}^{i=\infty} (E(D_{ik}) - q_{ik})\gamma_{i-1,k}\beta_{ik} \,.
\end{aligned}$$

But

$$\sum_{j=i}^{j=\infty} \gamma_{j-1,k}\beta_{jk} = \gamma_{i-1,k} \tag{5.8}$$

and therefore

$$E(T_k) = E(t) + \sum_{i=k}^{i=\infty} (E(W_{ik}) + \theta_{ik} + (1-\beta_{ik})q_{ik} + \beta_{ik}E(D_{ik}))\gamma_{i-1,k}$$

which, when combined with (4.2), yields

$$E(T_k) = E(t) + \sum_{i=k}^{i=\infty} (E(W_{ik}) + E(Q_{ik}))\gamma_{i-1,k} \; . \tag{5.9}$$

In order to evaluate $E(T_k)$ we need the $E(W_{ik})$'s. We have to distinguish between two cases: $i = k$ and $i > k$. In the first case all levels, up to and including k, may contain users, while in the second case only the $i-1$ and ith level have to be considered.

In the case $i = k$, the tagged user has to wait until all the first k levels are empty. A user in the (i,j) queue, where $1 \le i \le k$ and $1 \le j \le i$, affects $E(W_{kk})$ through his service in the queues $(i,j),\ldots,(k,j)$, while the probability that he will get service in the (n,j) queue where $i \le n \le k$ is $\gamma_{n-1,j}/\gamma_{i-1,j}$. Therefore a user in the (i,j) queue contributes $\sum_{n=i}^{n=k} E(Q_{nj})\gamma_{n-1,j}/\gamma_{i-1,j}$ to $E(W_{kk})$. There are N_{ij} users waiting in the (i,j) queue at the epoch of arrival, and hence we get by summing up over all queues in the first k levels

$$E(W_{kk}) = \sum_{i=1}^{i=k} \sum_{j=1}^{j=i} \sum_{n=i}^{n=k} N_{ij}E(Q_{nj})\gamma_{n-1,j}/\gamma_{i-1,j} \, . \tag{5.10}$$

Two groups of users contribute to $E(W_{ik})$, where $i > k$: the users in the queues $(i-1,k+1),\ldots,(i-1,i-1)$, and the users in the queues $(i,1),\ldots,(i,k)$. Let (n,j) be one of these queues. When the tagged user reaches the ith level the (n,j) queue contains the N_{nj} users who were in it at the arrival epoch of the tagged user, as well as all the users that were transferred into it from lower levels. The probability that a user from level m will reach the nth level is $\gamma_{n-1,j}/\gamma_{m-1,j}$, and there are N_{mj} such users. Thus the (n,j) queue contains $\sum_{m=j}^{m=n} N_{mj}\gamma_{n-1,j}/\gamma_{m-1,j}$ users, each contributing $E(Q_{nj})$ to $E(W_{ik})$. Therefore

$$E(W_{ik}) = \sum_{j=1}^{j=k} \sum_{m=j}^{m=i} N_{mj}\gamma_{i-1,j}/\gamma_{m-1,j}E(Q_{ij}) + \\ + \sum_{j=k+1}^{j=i-1} \sum_{m=j}^{m=i-1} N_{mj}\gamma_{i-2,j}/\gamma_{m-1,j}E(Q_{i-1,j}) \ . \tag{5.11}$$

Combining (5.9), (5.10) and (5.11), using (4.6) and rearranging

$$E(T_k) = E(t) + \sum_{n=1}^{n=\infty} \sum_{j=1}^{j=n} (N_{nj}/\gamma_{n-1,j})\sum_{i=n}^{i=\infty} \gamma_{i-1,k}\gamma_{i-1,j}E(Q_{ij}) \\ + \sum_{i=k}^{i=\infty} \gamma_{i-1,k}E(Q_{ik}) - \sum_{n=k+1}^{n=\infty} \sum_{j=k+1}^{j=n} (N_{nj}/\gamma_{n-1,j})\sum_{i=n}^{i=\infty} \gamma_{i-1,k}\beta_{ik}\gamma_{i-1,j}E(Q_{ij}). \tag{5.12}$$

In the above development and in (5.12) we set $N_{nj}/\gamma_{n-1,j} = 0$ whenever $\gamma_{n-1,j} = 0$; this setting is justified by the fact that in this case $N_{nj} = 0$ and the (n,j) queue is always empty.

Equation (5.12) lends itself to a very intuitive interpretation. In its first line the response time of the tagged user is computed by considering the expected service of the tagged user and the contribution it receives from each individual queue. There are $N_{n,j}$ users in the (n,j) queue at the arrival epoch, and since we do not have to consider their transition from the (j,j) queue to the (n,j) queue, we eliminate the probability of such transition through the division by $\gamma_{n-1,j}$; i.e. we consider the conditional probability of additional transfers, given that these users already reached the (n,j) queue. These users will contribute $E(Q_{ij})$ to $E(T_k)$, if two independent events both happen: the tagged user reaches this entry queue.

d. Each queue has limited capacity, as given by (5.17); when this capacity is exceeded, a newly arriving user will balk.

Consider now a M/M/1 system with linear loss in which all quanta are identical and all overheads are identical. Then $E(Q_{ik}) = Q$, say, for all $i \geq k$, and $E(Q_{ik})$ is zero otherwise. The service requirement is exponentially distributed, and therefore the probability of completion of service in any given quantum is the same for all the queues; let it be denoted by β. Then $\gamma_{nk} = (1-\beta)^{n+k-1}$ if $n \geq k$, and 1 otherwise. In this case:

$$\frac{S_{nj}(k)}{Q} = \begin{cases} (2-\beta-(1-\beta)^{k-n+1})/(\beta(2-\beta)) & j \leq n < k \\ (1-\beta)^{n-k}/(\beta(2-\beta)) & j \leq k \leq n \\ (1-\beta)^{n-k+1}/(\beta(2-\beta)) & k < j \leq n \end{cases} \tag{5.18}$$

and

$$N_{nj}(k) = \begin{cases} N'_k/(2-\beta-(1-\beta)^{k-n+1}) & j \leq n < k \\ N'_k/(1-\beta)^{n-k} & j \leq k \leq n \\ N'_k/(1-\beta)^{n-k+1} & k < j \leq n\,. \end{cases} \tag{5.19}$$

As can be expected $S_{nj}(k)$ and $N_{nj}(k)$ are unaffected by the entrance level, except for its relation to k, due to the memoryless property of the exponential service requirement distribution. N'_k itself is given, using (5.3), by

$$N'_k = (L'_0 - A_k - (E(t)+Q)B_k)\beta(2-\beta)/(B_kQ),$$

if $B_k \neq 0$, and is undetermined otherwise; in the sequel it is assumed that the latter case does not occur. Since N'_k assumes the largest value when $E(t)$ assumes its minimal value, and because $E(t) \geq -Q$, we get

$$N'_k = (L'_0 - A_k)\beta(2-\beta)/(QB_k) \ . \tag{5.20}$$

Combining (5.17), (5.19) and (5.20)

$$N'_{nj} = (\beta(2-\beta)/Q)\max\begin{cases}(L'_0 - A_k)/(1-\beta)^{n-k+1}B_k & k = 1,\ldots,j-1\\ (L'_0 - A_k)/(1-\beta)^{n-k}B_k & k = j,\ldots,n\\ (L'_0 - A_k)/(2-\beta-(1-\beta)^{k-n+1})B_k & k = n+1,\ldots,\underline{N}\end{cases} \tag{5.21}$$

which can easily be modified to accommodate different urgency values.

Finally, we consider a system in which each user is served to completion. In this system $\gamma_{nj} = 0$ whenever $n \geq j$, $\beta_{jj} = 1$ for $1 \leq j \leq N$, entry queues only contain users, and all other queues are empty. The notation may thus be simplified from double subscripts to a single subscript. The response time of users in the kth level is not affected by users in higher levels, because users in higher levels are not served as long as there are users in the kth level.

In this case

$$S_n(k) = \begin{cases} E(Q_n) & n \leq k \\ 0 & n > k \ . \end{cases} \tag{5.22}$$

$$N_n(k) = N'_k E(Q_k)/E(Q_n) \qquad 1 \leq n \leq k, \tag{5.23}$$

and

$$E(T_k) = E(t) + \sum_{n=1}^{n=k} N_n E(Q_n) + E(Q_k) = E(t) + E(Q_k) + N_k E(Q_k) = \tag{5.24}$$
$$= E(t) + (N_k + 1)E(Q_k) \ .$$

In the special case of the M/M/1 system considered above (5.20) is simplified to

$$N_k' = (L_0' - A_k)/QB_k \tag{5.25}$$

and the maximal capacity of the kth queue is, according to (5.23), $\max\{(L_0' - A_i)/QB_i\colon\ 1 \le i \le k\}$. Due to the monotonicity of the A_i's and B_i's, the maximum is achieved for $i = k$; hence the maximal capacity of the kth queue is given by (5.25). Equation (5.25) may also be easily modified to accommodate different urgency values.

To conclude this section, let us consider qualitatively the effects of changes in system's parameters on the upper bounds for the control numbers.

a. Changing overhead in the (i,j) queue: increasing θ_{ij} to $\theta_{ij} + \delta$: This change increases $E(Q_{ij})$ by δ, and $S_{nj}(k)$ by $\gamma_{i-1,k}\gamma_{i-1,j} \cdot (1-I(j > k)\beta_{ik})\delta/\gamma_{n-1,j}$. N_k' will not be changed, except for N_j' which will decrease; but the increase in $S_{nj}(k)$ will by (5.16) and (5.17) lead to a decrease in N_{nj}'. The capacities of the various queues and the system therefore might decrease as a result of this change.

b. Adding δ to the quantum in the (i,j) queue:

This change will increase $E(Q_{ij})$ and β_{ij}. Its effect on $S_{nj}(k)$ is hard to measure, though, without more information on the service requirement distribution. Specifically, if this distribution has decreasing failure rate then β_{mj} for $m > j$ will decrease, and the change in $S_{nj}(k)$ cannot be assessed without a complete knowledge of the service requirement distribution. On the other hand, if the failure rate does not decrease, $S_{nj}(k)$ will decrease; analysis, similar to the former case, then shows that the capacity of the system and its queues will increase.

c. Increasing the toll or waiting charge for the jth entry queue:

Such a change increases the loss from joining the jth level, and thus decreases N_j'; by (5.16) and (5.17), this change will result in a decrease in the capacity of the system.

d. Increasing the service charge:

This change affects N_k' for all entry queues, by decreasing all of them. As a result, system's capacity will decrease.

5.d. Determination of the Control Numbers

For a given set of system's parameters, and for given control numbers, the expected response times can be evaluated in the same way that they were for the round-robin system of Section 3.b and Appendix A. The resulting expressions, though, are very complicated, because of the dependence of the control numbers on the state of the system. This approach is therefore infeasible for the determination of the control numbers. The control numbers can be found in an indirect way: considering the totality of users and formulating their decisions as whether to balk or to join the system, we can establish a semi-Markov decision process in which the totality of users' objective is to minimize their discounted expected loss over an infinite horizon. This decision problem has an optimal stationary policy, which specifies for each state of the system the entry queue to be joined at the state, thus leading to an optimal set of control numbers. In order to simplify the presentation only one urgency value is considered, and it is assumed that the overhead periods, the quanta and the service requirements are all exponentially distributed. The last part of the section discusses deviations from these assumptions, as well as the relations between optimal policies for an individual user and the totality of users.

The system is described by the following parameters:

α = service requirement rate,

λ = arrival rate,

θ_{ij} = overhead rate for the (i,j) queue,

q_{ij} = quantum rate for the (i,j) queue,

where service requirements, interarrival times, overheads and quanta are all exponentially distributed.

C_{ij} = upper bound on the control number in the (i,j) queue, as given by (5.17); all upper bounds are finite,

N = the number of levels in the system; it is finite, and a user in the highest level is served to completion,

$\underline{N}$ = highest level which newly arriving users may join, assumed finite.

The loss function is linear as given in (5.2); the term corresponding to urgency loss, n_0u, will be ignored since only one urgency value is considered.

The presentation of the semi-Markov decision process, which is denoted SMDP in this section, follows Ross [21]. The SMDP has the following elements:

a. State space:

The states of the SMDP are given as vectors; a state vector is

$$(S,R,(k_1,k_2),N_{ij}: \ 1 \leq i \leq N, \ 1 \leq j \leq i) \tag{5.26}$$

which is shortened to

$$(S,R,(k_1,k_2),N_{ij}) \ .$$

S describes the system, and is either 'empty', 'overhead' or 'service'. R describes the reason for the termination of the former state, and is either 'arrival', 'overhead (termination)', 'quantum (termination)' or 'service (termination)'. Of the twelve possible combinations of (S,R) pairs only six are meaningful for the SMDP, namely:

(empty, arrival), (overhead, arrival), (overhead, overhead),
(service, arrival), (service, quantum) and (service, service).

In the sequel, the various S and R entities are abbreviated to their first letter; i.e. 'a' for 'arrival', 'o' for 'overhead', etc. (k_1,k_2) denotes the queue from which the user currently served was chosen. In contrast to the convention in Section 5.c, the user being served is not included in N_{k_1,k_2}. N_{ij} specifies the number of users waiting to be served in the (i,j) queue, and does not exceed C_{ij}.

Due to the exponential distributions of service requirement, quanta and overheads, one need not consider as part of the state space the time extended in the current overhead and quantum by the system to the user being served.

b. Action space:

The users have to decide between various actions only when R is 'arrival'; in all other states only one action is possible, as specified by the scheduling algorithm of the system. The actions are denoted by A_j, where $0 \leq j \leq N$. A_0 has two meanings: when R is 'arrival' it means that the newly arriving user balks, and otherwise its meaning is 'continue according to the scheduling algorithm of the system'. A_j for $1 \leq j$ is used for R = 'arrival' only, and means that the newly arriving user is joining the jth level. In this formulation to some states correspond a single action, while to others correspond many actions; this is a deviation from the usual formulation of semi-Markov decision problems, which considers the same action space for all the states. We may thus allow the action space to be $(A_0,\ldots,A_{\underline{N}})$ independent of the states of the system, while imposing a prohibitively high cost on action-state combinations not described above, so as to eliminate such combinations from any reasonable policy. Such action-state combinations may then be ignored in the following analysis.

c. Law of motion:

If the system is in state i and action a is chosen, then the next state of the system is chosen according to the transition probability $P_{ij}(a)$. In Table 5.1 below we specify only the transition probabilities that differ from zero, and only for the action-state combinations that were specified above; for other action-state combinations we assume that $P_{ii}(a) = 1$.

d. Transition times:

Conditional on the current state i, the next state j and the chosen action a, the time until the next transition occurs is a random variable

	i	a	j	N	N_{k1} in the new state j	(k_1,k_2) in j
line	(1)	(2)	(3)	(4)	(5)	(6)
1	e-a	A_0	e-a		N_{k1}	(0,0)
2	e-a	A_n	o-a		N_{k1}	(n,n)
3	e-a	A_n	o-o		N_{k1}	(n,n)
4	o-a	A_0	o-a		N_{k1}	(k_1,k_2)
5	o-a	A_n	o-a		$N_{k1}+\delta(k1),(nn)$	(k_1,k_2)
6	o-a	A_0	o-o		N_{k1}	(k_1,k_2)
7	o-a	A_n	o-o		$N_{k1}+\delta(k1),(nn)$	(k_1,k_2)
8	o-o	A_0	s-a	no	N_{k1}	(k_1,k_2)
9	o-o	A_0	s-q	no	N_{k1}	(k_1,k_2)
10	o-o	A_0	s-s	no	N_{k1}	(k_1,k_2)
11	s-a	A_0	s-a	no	N_{k1}	(k_1,k_2)
12	s-a	A_n	s-a	no	$N_{k1}+\delta(k1),(nn)$	(k_1,k_2)
13	s-a	A_0	s-q	no	N_{k1}	(k_1,k_2)
14	s-a	A_n	s-q	no	$N_{k1}+\delta(k1),(nn)$	(k_1,k_2)

Table 5.1 - SMDP Summary

line	i (1)	a (2)	j (3)	N (4)	N_{k1} in the new state j (5)	(k_1,k_2) in j (6)
15	s-a	A_0	s-s	no	N_{k1}	(k_1,k_2)
16	s-a	A_n	s-s	no	$N_{k1}^{+\delta}(k1),(nn)$	(k_1,k_2)
17	o-o	A_0	s-a	yes	N_{k1}	(k_1,k_2)
18	o-o	A_0	s-s	yes	N_{k1}	(k_1,k_2)
19	s-a	A_0	s-a	yes	N_{k1}	(k_1,k_2)
20	s-a	A_n	s-a	yes	$N_{k1}^{+\delta}(k1),(nn)$	(k_1,k_2)
21	s-a	A_0	s-s	yes	N_{k1}	(k_1,k_2)
22	s-a	A_n	s-s	yes	$N_{k1}^{+\delta}(k1),(nn)$	(k_1,k_2)
23	s-q	A_0	o-a	no	$N_{k1}^{+\delta}(k_1+1,k_2),(k1)^{-\delta}(k1),(\underline{ij})$	$(\underline{i},\underline{j})$
24	s-q	A_0	o-o	no	$N_{k1}^{+\delta}(k_1+1,k_2),(k1)^{-\delta}(k1),(\underline{ij})$	$(\underline{i},\underline{j})$
25	s-s	A_0	e-a	system empty	N_{k1}	(0,0)
26	s-s	A_0	o-a	system not empty	$N_{k1}^{-\delta}(k1),(\underline{ij})$	$(\underline{i},\underline{j})$
27	s-s	A_0	o-o	system not empty	$N_{k1}^{-\delta}(k1),(\underline{ij})$	$(\underline{i},\underline{j})$

Table 5.1 - SMDP Summary (continued)

line	i (1)	a (2)	j (3)	n (4)	$P_{ij}(a)$ (7)	$F_{ij}(\cdot\|a)$ (8)	$D_\delta(i,j,a)$ (9)
1	e-a	A_0	e-a		1	λ	$\lambda/(\delta+\lambda)$
2	e-a	A_n	o-a		$\lambda/(\lambda+\theta_{nn})$	$\lambda+\theta_{nn}$	$\lambda/(\delta+\lambda+\theta_{nn})$
3	e-a	A_n	o-o		$\theta_{nn}/(\lambda+\theta_{nn})$	$\lambda+\theta_{nn}$	$\theta_{nn}/(\delta+\lambda+\theta_{nn})$
4	o-a	A_0	o-a		$\lambda/(\lambda+\theta_{k_1k_2})$	$\lambda+\theta_{k_1k_2}$	$\lambda/(\delta+\lambda+\theta_{k_1k_2})$
5	o-a	A_n	o-a		$\lambda/(\lambda+\theta_{k_1k_2})$	$\lambda+\theta_{k_1k_2}$	$\lambda/(\delta+\lambda+\theta_{k_1k_2})$
6	o-a	A_0	o-o		$\theta_{k_1k_2}/(\lambda+\theta_{k_1k_2})$	$\lambda+\theta_{k_1k_2}$	$\theta_{k_1k_2}/(\delta+\lambda+\theta_{k_1k_2})$
7	o-a	A_n	o-o		$\theta_{k_1k_2}/(\lambda+\theta_{k_1k_2})$	$\lambda+\theta_{k_1k_2}$	$\theta_{k_1k_2}/(\delta+\lambda+\theta_{k_1k_2})$
8	o-o	A_0	s-a	no	$\lambda/(\lambda+q_{k_1k_2}+\alpha)$	$\lambda+q_{k_1k_2}+\alpha$	$\lambda/(\delta+\lambda+q_{k_1k_2}+\alpha)$
9	o-o	A_0	s-q	no	$q_{k_1k_2}/(\lambda+q_{k_1k_2}+\alpha)$	$\lambda+q_{k_1k_2}+\alpha$	$q_{k_1k_2}/(\delta+\lambda+q_{k_1k_2}+\alpha)$
10	o-o	A_0	s-s	no	$\alpha/(\lambda+q_{k_1k_2}+\alpha)$	$\lambda+q_{k_1k_2}+\alpha$	$\alpha/(\delta+\lambda+q_{k_1k_2}+\alpha)$
11	s-a	A_0	s-a	no	$\lambda/(\lambda+q_{k_1k_2}+\alpha)$	$\lambda+q_{k_1k_2}+\alpha$	$\lambda/(\delta+\lambda+q_{k_1k_2}+\alpha)$
12	s-a	A_n	s-a	no	$\lambda/(\lambda+q_{k_1k_2}+\alpha)$	$\lambda+q_{k_1k_2}+\alpha$	$\lambda/(\delta+\lambda+q_{k_1k_2}+\alpha)$
13	s-a	A_0	s-q	no	$q_{k_1k_2}/(\lambda+q_{k_1k_2}+\alpha)$	$\lambda+q_{k_1k_2}+\alpha$	$q_{k_1k_2}/(\delta+\lambda+q_{k_1k_2}+\alpha)$
14	s-a	A_n	s-q	no	$q_{k_1k_2}/(\lambda+q_{k_1k_2}+\alpha)$	$\lambda+q_{k_1k_2}+\alpha$	$q_{k_1k_2}/(\delta+\lambda+q_{k_1k_2}+\alpha)$

Table 5.1 - SMDP Summary (continued)

line	i	a	j	n	$P_{ij}(a)$	$F_{ij}(\cdot\|a)$	$D_\delta(i,j,a)$
	(1)	(2)	(3)	(4)	(7)	(8)	(9)
15	s-a	A_0	s-s	no	$\alpha/(\lambda+q_{k_1k_2}+\alpha)$	$\lambda+q_{k_1k_2}+\alpha$	$\alpha/(\delta+\lambda+q_{k_1k_2}+\alpha)$
16	s-a	A_n	s-s	no	$\alpha/(\lambda+q_{k_1k_2}+\alpha)$	$\lambda+q_{k_1k_2}+\alpha$	$\alpha/(\delta+\lambda+q_{k_1k_2}+\alpha)$
17	o-o	A_0	s-a	yes	$\lambda/(\lambda+\alpha)$	$\lambda+\alpha$	$\lambda/(\delta+\lambda+\alpha)$
18	o-o	A_0	s-s	yes	$\alpha/(\lambda+\alpha)$	$\lambda+\alpha$	$\alpha/(\delta+\lambda+\alpha)$
19	s-a	A_0	s-a	yes	$\lambda/(\lambda+\alpha)$	$\lambda+\alpha$	$\lambda/(\delta+\lambda+\alpha)$
20	s-a	A_n	s-a	yes	$\lambda/(\lambda+\alpha)$	$\lambda+\alpha$	$\lambda/(\delta+\lambda+\alpha)$
21	s-a	A_0	s-s	yes	$\alpha/(\lambda+\alpha)$	$\lambda+\alpha$	$\alpha/(\delta+\lambda+\alpha)$
22	s-a	A_n	s-s	yes	$\alpha/(\lambda+\alpha)$	$\lambda+\alpha$	$\alpha/(\delta+\lambda+\alpha)$
23	s-q	A_0	o-a	no	$\lambda/(\lambda+\theta_{\underline{ij}})$	$\lambda+\theta_{\underline{ij}}$	$\lambda/(\delta+\lambda+\theta_{\underline{ij}})$
24	s-q	A_0	o-o	no	$\theta_{\underline{ij}}/(\lambda+\theta_{\underline{ij}})$	$\lambda+\theta_{\underline{ij}}$	$\theta_{\underline{ij}}/(\delta+\lambda+\theta_{\underline{ij}})$
25	s-s	A_0	e-a	system empty	1	λ	$\lambda/(\delta+\lambda)$
26	s-s	A_0	o-a	system not empty	$\lambda/(\lambda+\theta_{\underline{ij}})$	$\lambda+\theta_{\underline{ij}}$	$\lambda/(\delta+\lambda+\theta_{\underline{ij}})$
27	s-s	A_0	o-o	system not empty	$\theta_{\underline{ij}}/(\lambda+\theta_{\underline{ij}})$	$\lambda+\theta_{\underline{ij}}$	$\theta_{\underline{ij}}/(\delta+\lambda+\theta_{\underline{ij}})$

Table 5.1 - SMDP Summary (continued)

line	i	a	j	N	$\underline{C}(i,a)$	$c(i,a)$	$C_\delta(i,a)$
	(1)	(2)	(3)	(4)	(10)	(11)	(12)
1	e-a	A_0	e-a		L_0	0	L_0
2	e-a	A_n	o-a		n_1t_n	S_n	$n_1t_n+S_n/(\delta+\lambda+\theta_{nn})$
3	e-a	A_n	o-o		n_1t_n	S_n	$n_1t_n+S_n/(\delta+\lambda+\theta_{nn})$
4	o-a	A_0	o-a		L_0	S_{k_2}	$L_0+S_{k_2}/(\delta+\lambda+\theta_{k_1k_2})$
5	o-a	A_n	o-a		n_1t_n	S_{k_2}	$n_1t_n+S_{k_2}/(\delta+\lambda+\theta_{k_1k_2})$
6	o-a	A_0	o-o		L_0	S_{k_2}	$L_0+S_{k_2}/(\delta+\lambda+\theta_{k_1k_2})$
7	o-a	A_n	o-o		n_1t_n	S_{k_2}	$n_1t_n+S_{k_2}/(\delta+\lambda+\theta_{k_1k_2})$
8	o-o	A_0	s-a	no	0	S_0	$S_0/(\delta+\lambda+q_{k_1k_2}+\alpha)$
9	o-o	A_0	s-q	no	0	S_0	$S_0/(\delta+\lambda+q_{k_1k_2}+\alpha)$
10	o-o	A_0	s-s	no	0	S_0	$S_0/(\delta+\lambda+q_{k_1k_2}+\alpha)$
11	s-a	A_0	s-a	no	L_0	S_0	$L_0+S_0/(\delta+\lambda+q_{k_1k_2}+\alpha)$
12	s-a	A_n	s-a	no	n_1t_n	S_0	$n_1t_n+S_0/(\delta+\lambda+q_{k_1k_2}+\alpha)$
13	s-a	A_0	s-q	no	L_0	S_0	$L_0+S_0/(\delta+\lambda+q_{k_1k_2}+\alpha)$
14	s-a	A_n	s-q	no	n_1t_n	S_0	$n_1t_n+S_0/(\delta+\lambda+q_{k_1k_2}+\alpha)$

Table 5.1 - SMDP Summary (continued)

	i	a	j	N	$\underline{C}(i,a)$	$c(i,a)$	$C_\delta(i,a)$
line	(1)	(2)	(3)	(4)	(10)	(11)	(12)
15	s-a	A_0	s-s	no	L_0	S_0	$L_0+S_0/(\delta+\lambda+q_{k_1k_2}+\alpha)$
16	s-a	A_n	s-s	no	n_1t_n	S_0	$n_1t_n+S_0/(\delta+\lambda+q_{k_1k_2}+\alpha)$
17	o-o	A_0	s-a	yes	0	S_0	$S_0/(\delta+\lambda+\alpha)$
18	o-o	A_0	s-s	yes	0	S_0	$S_0/(\delta+\lambda+\alpha)$
19	s-a	A_0	s-a	yes	L_0	S_0	$L_0+S_0/(\delta+\lambda+\alpha)$
20	s-a	A_n	s-a	yes	n_1t_n	S_0	$n_1t_n+S_0/(\delta+\lambda+\alpha)$
21	s-a	A_0	s-s	yes	L_0	S_0	$L_0+S_0/(\delta+\lambda+\alpha)$
22	s-a	A_n	s-s	yes	n_1t_n	S_0	$n_1t_n+S_0/(\delta+\lambda+\alpha)$
23	s-q	A_0	o-a	no	0	$S_{\underline{j}}$	$S_{\underline{j}}/(\delta+\lambda+\theta_{\underline{ij}})$
24	s-q	A_0	o-o	no	0	$S_{\underline{j}}$	$S_{\underline{j}}/(\delta+\lambda+\theta_{\underline{ij}})$
25	s-s	A_0	e-a	system empty	0	0	0
26	s-s	A_0	o-a	system not empty	0	$S_{\underline{j}}$	$S_{\underline{j}}/(\delta+\lambda+\theta_{\underline{ij}})$
27	s-s	A_0	o-o		0	$S_{\underline{j}}$	$S_{\underline{j}}/(\delta+\lambda+\theta_{\underline{ij}})$

Table 5.1 - SMDP Summary (continued)

Remarks:

Column 1: i is the current state of the system.

Column 2: a is the chosen decision.

Column 3: j is the next state of the system.

Column 4: This column is used to distinguish between entities which correspond to the last level and other entities; this distinction is important only when 'yes' or 'no' appear in this column. If levels below N are considered, the lines with 'no' in this column have to be used; otherwise, the lines with 'yes' in this column have to be used.

Column 5: The number of users in the system's queues in the next state. N_{kl} is the number of users before a decision is made about the action to be taken. $(\underline{i},\underline{j})$ is the lowest numbered queue that is not empty, and from which the next user to be served is chosen. δ_{kl} is the Kronecker delta.

Column 6: This is the queue from which is chosen the user to be served in the next state.

Column 11: Using the expressions of columns 5 and 6, i.e. the number of users in the next state, S_j is defined as $n_2 + n_1 c_j + \sum_{n=1}^{n=N} \sum_{m=1}^{m=n} (n_2 + n_1 c_m) N_{nm}$, where $c_0 = c$ is the service charge. Notice that the number of users and the user to be served in the next state depend on i and a only; hence S_j is a function of i and a only.

Table 5.1 - SMDP summary (continued)

with probability distribution $F_{ij}(\cdot|a)$. In our case these distributions are all exponential, and their parameters are specified in Table 5.1 for all the action-state combinations listed above; for other combinations, the rate λ is assumed.

e. Costs and discounting:

Two costs are associated with each action-state combination: an immediate cost $\underline{C}(i,a)$, and a cost rate $c(i,a)$. That is, if action a is chosen in state i, and a transition occurs after t time units, then the total cost incurred until the next transition is $\underline{C}(i,a) + tc(i,a)$. It is shown by Ross [21, Theorem 7.1] that this cost structure may be replaced by an immediate cost

$$C_\delta(i,a) = \underline{C}(i,a) + \sum_{\text{(possible states } j)} P_{ij}(a)\int_0^\infty \int_0^t e^{-\delta s}c(i,a)ds\, dF_{ij}(t|a) \quad (5.27)$$

which is the expected one stage discounted cost incurred when action a is taken in state i, when the discount rate is δ. $C_\delta(i,a)$, $\underline{C}(i,a)$ and $c(i,a)$ for all the action-state combinations given above are specified in Table 5.1; the values for other combinations are assumed to be prohibitively high.

In the sequel we also need the expected discounted value of one unit of money that is paid at the next transition into j when action a is chosen in state i. This discounted value will be denoted $D_\delta(i,j,a)$, and equals

$$D_\delta(i,j,a) = P_{ij}(a)\int_0^\infty e^{-\delta t}dF_{ij}(t|a) \ . \quad (5.28)$$

It is listed in Table 5.1.

The SMDP has a finite state space, a finite action space, and one communicating class. The costs for all combinations of action-state are uniformly bounded because the capacity of the system is finite, and due to the exponential distributions of transition times, only a finite number of transitions occurs during a finite time interval with probability 1. Therefore there exists a stationary optimal policy that minimizes the expected total discounted cost (Ross [21]). Denoting by $V_\delta(i)$ the expected total discounted cost that is incurred when the initial state is i and this policy is used, we know from Ross [21, Theorems 7.1 and 7.2] that $V_\delta(i)$ satisfies the optimality equation

$$V_\delta(i) = \min_a (C_\delta(i,a) + \sum_{(\text{possible states } j)} V_\delta(j) D_\delta(i,j,a)) \qquad (5.29)$$

and the optimal policy is the one that selects at state i an action which minimizes the right-hand side of (5.29)

From the formulation of the SMDP it is evident that decisions are made by the users on arrival only. Thus the states (o-o), (s-q) and (s-s) could have been eliminated from the formulation. This was not done for two reasons: First, such an elimination obscures the flow of events in the system. Second, the number of users after each transition in the modified system will be a random variable, whose distribution is quite complicated; similarly, the expressions for $C_\delta(i,a)$ become unmanageable.

A solution to (5.29) can be found in several ways, such as policy iteration and successive approximations methods. Another approach, which is pursued in this section, is to transform (5.29) into the following linear programming problem (see, e.g. Ross [21], p. 151):

$$\text{maximize} \sum_{\text{(all possible states)}} V(i) \tag{5.30}$$

$$\text{subject to } C_\delta(i,a) + \sum_{\text{(possible states } j)} V(j)D_\delta(i,j,a) \geq V(i) \quad \text{for all states } i \text{ and actions } a.$$

The solution of (5.30) is the required $V_\delta(i)$. Moreover, the constraints which are satisfied with equality for the optimal solution correspond to decisions according to the optimal policy (see, e.g. Wagner [24], p. 761).

The constraints of (5.30), after being fully specified according to the information of Table 5.1, may be somewhat simplified. This simplification eliminates some states, and reveals the relations among various $V(i)$'s. The constraints and their modifications are specified below. For clarity, each constraint is followed by the line number in Table 5.1 from which it is derived, while N_{k1} of (5.26) corresponds to the state i, prior to the transition.

$$\delta V(e,a,(0,0),0) \leq (\delta+\lambda)L_0 \qquad \text{(line 1)} \tag{5.31}$$

$$(\delta+\lambda+\theta_{nn})V(e,a,(0,0),0) - \lambda V(o,a,(n,n),0) - \theta_{nn}V(o,o,(n,n),0) \leq$$
$$\leq n_2 + n_1c_n + (\delta+\lambda+\theta_{nn})n_1t_n \qquad \text{(lines 2 \& 3).} \tag{5.32}$$

Both (5.31) and (5.32) are related to the state of an empty system at the epoch of arrival: (5.31) corresponds to balking, and (5.32) to joining level n.

$$(\delta+\theta_{k_1k_2})V(o,a,(k_1,k_2),N_{k1}) - \theta_{k_1k_2}V(o,o,(k_1,k_2),N_{k1}) \leq$$
$$\leq S_{k_2} + (\delta+\lambda+\theta_{k_1k_2})L_0 \qquad \text{(lines 4 \& 6)} \tag{5.33}$$

$$(\delta+\lambda+\theta_{k_1k_2})V(o,a,(k_1,k_2),N_{k1}) - \lambda V(o,a,(k_1,k_2),N_{k1}+\delta_{(k1),(nn)}) -$$

$$- \theta_{k_1k_2}V(o,o,(k_1,k_2),N_{k1}+\delta_{(k1),(nn)}) \leq S_{k_2} + (\delta+\lambda+\theta_{k_1k_2})n_1t_n \tag{5.34}$$

(lines 5 & 7).

These two equations correspond to the state $(o,a,(k_1,k_2),N_{k1})$ for all configurations of users waiting in the queues and in service; the first one is for the balking decision, and the second for the A_n decision. S_{k_2} is computed, as said in the remarks to Table 5.1, according to the number of waiting users after an action was taken.

$$(\delta+\lambda+q_{k_1k_2}+\alpha)V(o,o,(k_1,k_2),N_{k1}) - \lambda V(s,a,(k_1,k_2),N_{k1}) - q_{k_1k_2}V(s,q,(k_1,k_2),N_{k1}) -$$

$$- \alpha V(s,s,(k_1,k_2),N_{k1}) = S_0 \quad (k_1 < N) \tag{5.35}$$

(lines 8, 9 & 10)

$$(\delta+\lambda+\alpha)V(o,o,(N,k_2),N_{k1}) - \lambda V(s,a,(N,k_2),N_{k1}) -$$

$$- \alpha V(s,s,(N,k_2),N_{k1}) = S_0 \tag{5.36}$$

(lines 17 & 18).

These two equations correspond to the state $(o,o,(k_1,k_2),N_{k1})$ for all configurations of users waiting in the queues and in service.

$$(\delta+q_{k_1k_2}+\alpha)V(s,a,(k_1,k_2),N_{k1}) - q_{k_1k_2}V(s,q,(k_1,k_2),N_{k1}) -$$

$$- \alpha V(s,s,(k_1,k_2),N_{k1}) \leq S_0 + (\delta+\lambda+q_{k_1k_2}+\alpha)L_0 \quad (k_1 < N) \tag{5.37}$$

(lines 11, 13 & 15).

$$(\delta+\alpha)V(s,a,(N,k_2),N_{k1}) - \alpha V(s,s,(N,k_2),N_{k1}) \leq$$
$$\leq S_0 + (\delta+\lambda+\alpha)L_0 \qquad \text{(lines 19 \& 21)}. \tag{5.38}$$

$$(\delta+\lambda+q_{k_1k_2}+\alpha)V(s,a,(k_1,k_2),N_{k1}) - \lambda V(s,a,(k_1,k_2),N_{k1}+\delta_{(k1),(nn)}) -$$
$$- q_{k_1k_2}V(s,q,(k_1,k_2),N_{k1}+\delta_{(k1),(nn)}) - \alpha V(s,s,(k_1,k_2),N_{k1}+\delta_{(k1),(nn)}) \leq \tag{5.39}$$
$$\leq S_0 + (\delta+\lambda+q_{k_1k_2}+\alpha)n_1t_n \qquad (k_1 < N) \qquad \text{(lines 12, 14 \& 16)}.$$

$$(\delta+\lambda+\alpha)V(s,a,(N,k_2),N_{k1}) - \lambda V(s,a,(N,k_2),N_{k1}+\delta_{(k1),(nn)}) -$$
$$- \alpha V(s,s,(N,k_2),N_{k1}+\delta_{(k1),(nn)}) \leq S_0 + (\delta+\lambda+\alpha)n_1t_n \qquad \text{(lines 20 \& 22)}. \tag{5.40}$$

These four equations correspond to the state $(s,a,(k_1,k_2),N_{k1})$ for all possible combinations of users waiting in the queues and in service, The first two are for balking, the last two for the action A_n.

$$(\delta+\lambda+\theta_{\underline{ij}})V(s,q,(k_1,k_2),N_{k1}) - \lambda V(o,a,(\underline{i},\underline{j}),N_{k1}+\delta_{(k_1+1,k_2),(k1)}-\delta_{(k1),(\underline{i},\underline{j})}) -$$
$$- \theta_{\underline{ij}}V(o,o,(\underline{i},\underline{j}),N_{k1}+\delta_{(k_1+1,k_2),(k1)}-\delta_{(k1),(\underline{ij})}) = S_{\underline{j}} \qquad (k_1 < N) \tag{5.41}$$
$$\text{(lines 23 \& 24)}.$$

This equation corresponds to the state $(s,q,(k_1,k_2),N_{k1})$ for all combinations of users waiting in the system and being served, except that the user being served cannot be chosen, for this state, from the last level.

$$(\delta+\lambda)V(s,s,(k_1,k_2),0) = \lambda V(e,a,(0,0),0) \qquad \text{(line 25)}. \tag{5.42}$$

This corresponds to the case when the last user in the system leaves it.

$$(\delta+\lambda+\theta_{\underline{ij}})V(s,s,(k_1,k_2),N_{k1}) - \lambda V(o,a,(\underline{i},\underline{j}),N_{k1}-\delta_{(k1),(\underline{ij})}) - \\ - \theta_{\underline{ij}}V(o,o,(\underline{i},\underline{j}),N_{k1}-\delta_{(k1),(\underline{ij})}) = S_{\underline{j}} \qquad \text{(lines 26 \& 27).} \tag{5.43}$$

This corresponds to the case $(s,s,(k_1,k_2),N_{k1})$ when the system is not empty; i.e., when $\sum_{n=1}^{n=N} \sum_{m=1}^{m=n} N_{nm} > 0$. All other combinations of users waiting in the system and being served are included in (5.43).

The objective function of (5.30) and (5.31)-(5.43) constitute the linear programming problem whose solution gives the optimal policy. Some simplifications are evident. Equation (5.41) and (5.43) imply that

$$V(s,q,(k_1,k_2),N_{k1}) = V(s,s,(k_1,k_2),N_{k1}+\delta_{(k_1+1,k_2),(k1)}), \ (k_1 < N) \tag{5.44}$$

and therefore the state $(s,q,(k_1,k_2),N_{k1})$ can be eliminated from the linear program. This will affect (5.35), (5.37), (5.39) and will eliminate (5.41). The state $(e,a,(0,0),0)$ is eliminated by (5.42); (5.42) can thus be dropped, while (5.31) and (5.32) have to be modified. In addition, it is evident from (5.43) that $V(s,s,(k_1,k_2),N_{k1})$ is independent of (k_1,k_2); one specific combination of (k_1,k_2) suffices for this state. We chose the combination $(0,0)$, and the state is $(s,s,(0,0),N_{k1})$ for all combinations of users who wait in the system, but is independent of the user being served. The objective function, too, has to be changed by dropping the eliminated states from it. This change might seem strange at first, because one would have suspected that the modifications above have to be made in the objective function, in addition to making them in the appropriate equations. However, this objective function was chosen as a replacement to the objective of maximizing $V(i)$ for all possible states i, and the modification of the original objective function would have been the elimination of the states mentioned above; this modification is thus carried over to the objective in (5.30).

Equation (5.43) can be used for further simplification, since it implies

$$V(s,s,(0,0),N_{k1}+\delta_{(k1),(k_1k_2)}) = (S_{k_2} + \lambda V(o,a,(k_1,k_2),N_{k1}) + \theta_{k_1k_2}V(o,o,(k_1,k_2),N_{k1}))/(\delta+\lambda+\theta_{k_1k_2}) \tag{5.45}$$

which, in turn, may be used to simplify (5.32), (5.33) and (5.34). The state $(o,o,(k_1,k_2),N_{k1})$ is then eliminated by (5.45), a change which affects (5.35) and (5.36), as well as the objective function. Once these changes have been made, (5.43) may be discarded, and the emerging modified problem is:

$$\text{maximize} \sum_{\substack{\text{(all possible combinations)}\\ \text{(of states)}}} (V(o,a,(k_1,k_2),N_{k1}) + V(s,a,(k_1,k_2),N_{k1}) + V(s,s,(0,0),N_{k1})) \tag{5.46}$$

subject to:

$$\delta V(s,s,(0,0),0) \leq \lambda L_0 \tag{5.47}$$

$$(\delta+\lambda)V(s,s,(0,0),0) - \lambda V(s,s,(0,0),\delta_{(k1),(nn)}) \leq \lambda n_1 t_n \tag{5.48}$$

$$V(o,a,(k_1,k_2),N_{k1}) - V(s,s,(0,0),N_{k1}+\delta_{(k1),(k_1k_2)}) \leq L_0 \tag{5.49}$$

$$V(o,a,(k_1,k_2),N_{k1}) - V(s,s,(0,0),N_{k1}+\delta_{(k1),(k_1k_2)}+\delta_{(k1),(nn)}) \leq n_1 t_n \tag{5.50}$$

$$(\delta+\lambda+q_{k_1k_2}+\alpha)(\delta+\lambda+\theta_{k_1k_2})V(s,s,(0,0),N_{k1}+\delta_{(k1),(k_1k_2)}) -$$

$$- \theta_{k_1k_2}q_{k_1k_2}V(s,s,(0,0),N_{k1}+\delta_{(k1),(k_1+1,k_2)}) - \theta_{k_1k_2}\alpha V(s,s,(0,0),N_{k1}) -$$

$$- \theta_{k_1k_2}\lambda V(s,a,(k_1,k_2),N_{k1}) - \lambda(\delta+\lambda+q_{k_1k_2}+\alpha)V(o,a,(k_1,k_2),N_{k1}) = \tag{5.51}$$

$$= \theta_{k_1k_2}S_0(N_{k1}) + (\delta+\lambda+q_{k_1k_2}+\alpha)S_{k_2}(N_{k1}) \qquad (k_1 < N).$$

$$(\delta+\lambda+\alpha)(\delta+\lambda+\theta_{N,k_2})V(s,s,(0,0),N_{k1}+\delta_{(k1),(N,k_2)}) -$$

$$- \theta_{N,k_2}\alpha V(s,s,(0,0),N_{k1}) - \theta_{N,k_2}\lambda V(s,a,(N,k_2),N_{k1}) - \tag{5.52}$$

$$- \lambda(\delta+\lambda+\alpha)V(o,a,(N,k_2),N_{k1}) = \theta_{N,k_2}S_0(N_{k1}) + (\delta+\lambda+\alpha)S_{k_2}(N_{k1})$$

$$(\delta+q_{k_1k_2}+\alpha)V(s,a,(k_1,k_2),N_{k1}) - q_{k_1k_2}V(s,s,(0,0),N_{k1}+\delta_{(k_1+1,k_2),(k1)}) -$$

$$- \alpha V(s,s,(0,0),N_{k1}) \leq S_0(N_{k1}) + (\delta+\lambda+q_{k_1k_2}+\alpha)L_0 \qquad (k_1 < N) \tag{5.53}$$

$$(\delta+\alpha)V(s,a,(N,k_2),N_{k1}) - \alpha V(s,s,(0,0),N_{k1}) \leq$$

$$\leq S_0(N_{k1}) + (\delta+\lambda+\alpha)L_0 \tag{5.54}$$

$$(\delta+\lambda+q_{k_1k_2}+\alpha)V(s,a,(k_1,k_2),N_{k1}) - \lambda V(s,a,(k_1,k_2),N_{k1}+\delta_{(k1),(nn)}) -$$

$$- q_{k_1k_2}V(s,s,(0,0),N_{k1}+\delta_{(k1),(nn)}+\delta_{(k_1+1,k_2),(k1)}) - \tag{5.55}$$

$$- \alpha V(s,s,(0,0),N_{k1}+\delta_{(k1),(nn)}) \leq S_0(N_{k1}+\delta_{(k1),(nn)}) + (\delta+\lambda+q_{k_1k_2}+\alpha)n_1t_n$$

$$(k_1 < N)$$

$$(\delta+\lambda+\alpha)V(s,a,(N,k_2),N_{k1}) - \lambda V(s,a,(N,k_2),N_{k1}+\delta_{(k1),(nn)}) -$$
$$- \alpha V(s,s,(0,0),N_{k1}+\delta_{(k1),(nn)}) \le S_0(N_{k1}+\delta_{(k1),(nn)}) + (\delta+\lambda+\alpha)n_1 t_n . \qquad (5.56)$$

In the modified system all the states correspond either to service termination or to arrival events. The modified system contains considerably fewer states than the original one, but the same decisions are made in both systems. Equations (5.47) and (5.48) correspond to the decisions A_0 and A_n for an arrival to an empty system, while (5.49) and (5.50) correspond to A_0 and A_n for an arrival during an overhead period, (5.53) and (5.54) to the action A_0 for an arrival during a quantum, and (5.55) and (5.56) to the decision A_n for an arrival during a quantum. Notice that in the expressions for S_j we included the state for which S_j is computed; specifically

$$S_j(N_{k1}) = n_2 + n_1 c_j + \sum_{n=1}^{n=N} \sum_{m=1}^{m=n} (n_2 + n_1 c_m) N_{nm} \quad \text{where} \quad c_0 = c. \qquad (5.57)$$

In the modified problem (5.51) and (5.52) do not correspond to any decision to be made by the user, but rather supply information about the relations between the $V(i)$'s. Equations (5.51) and (5.52) may thus be used for the elimination of the state $(o,a,(k_1,k_2),N_{k1})$. In this case (5.49)-(5.52) and (5.46) have to be modified.

Some simplifications in the notation seem appropriate at this stage, for the presentation of the linear programming problem only. Specifically,

$q_{N,k_2} = 0 \quad \text{for} \quad 1 \le k_2 \le N$

$V(s,s,(0,0),N_{k1}+\delta_{(k1),(N+1,k_2)}) = 0$ for all k_2 and combinations of N_{k1}

$$V(s,s,(0,0),N_{k1}) = V(s,N_{k1}) \tag{5.58}$$

$V(s,a,(k_1,k_2),N_{k1}) = V(a,k_1,k_2,N_{k1})$ where $1 \le k_1 \le N$ and $1 \le k_2 \le k_1$

The linear programming problem that gives the optimal policy for the users is thus given by

$$\text{maximize} \sum_{\substack{\text{(all possible combinations)} \\ \text{(of } k_1,\ k_2 \text{ and } N_{k1} \text{)}}} (V(s,N_{k1}) + V(a,k_1,k_2,N_{k1})) \tag{5.59}$$

subject to

$$\delta V(s,0) \le \lambda L_0 \tag{5.60}$$

$$(\delta+\lambda)V(s,0) - \lambda V(s,\delta_{(k1),(nn)}) \le \lambda n_1 t_n \tag{5.61}$$

where these equations correspond to the actions A_0 and A_n for an arrival to an empty system,

$$\begin{aligned}(\delta+\lambda+q_{k_1k_2}+\alpha)(\delta + \theta_{k_1k_2})V(s,N_{k1}+\delta_{(k1),(k_1k_2)}) - \\ - \theta_{k_1k_2}q_{k_1k_2}V(s,N_{k1}+\delta_{(k1),(k_1+1,k_2)}) - \theta_{k_1k_2}\alpha V(s,N_{k1}) - \theta_{k_1k_2}\lambda V(a,k_1,k_2,N_{k1}) \\ \le \lambda(\delta+\lambda+q_{k_1k_2}+\alpha)L_0 + \theta_{k_1k_2}S_0(N_{k1}) + (\delta+\lambda+q_{k_1k_2}+\alpha)S_{k_2}(N_{k1})\end{aligned} \tag{5.62}$$

$$(\delta+\lambda+q_{k_1k_2}+\alpha)(\delta+\lambda+\theta_{k_1k_2})V(s,N_{k1}+\delta_{(k1),(k_1k_2)}) -$$
$$-\lambda(\delta+\lambda+q_{k_1k_2}+\alpha)V(s,N_{k1}+\delta_{(k1),(k_1k_2)}+\delta_{(k1),(nn)}) -$$
$$-\theta_{k_1k_2}q_{k_1k_2}V(s,N_{k1}+\delta_{(k1),(k_1+1,k_2)}) - \theta_{k_1k_2}\alpha V(s,N_{k1}) - \theta_{k_1k_2}\lambda V(a,k_1,k_2,N_{k1}) \tag{5.63}$$
$$\leq \lambda(\delta+\lambda+q_{k_1k_2}+\alpha)n_1t_n + \theta_{k_1k_2}S_0(N_{k1}) + (\delta+\lambda+q_{k_1k_2}+\alpha)S_{k_2}(N_{k1})$$

where these equations correspond to the actions A_0 and A_n for an arrival during an overhead period,

$$(\delta+q_{k_1k_2}+\alpha)V(a,k_1,k_2,N_{k1}) - q_{k_1k_2}V(s,N_{k1}+\delta_{(k_1+1,k_2),(k1)}) -$$
$$-\alpha V(s,N_{k1}) \leq S_0(N_{k1}) + (\delta+\lambda+q_{k_1k_2}+\alpha)L_0 \tag{5.64}$$

$$(\delta+\lambda+q_{k_1k_2}+\alpha)V(a,k_1,k_2,N_{k1}) - \lambda V(a,k_1,k_2,N_{k1}+\delta_{(k1),(nn)}) -$$
$$-q_{k_1k_2}V(s,N_{k1}+\delta_{(k1),(nn)}+\delta_{(k_1+1,k_2),(k1)}) - \tag{5.65}$$
$$-\alpha V(s,N_{k1}+\delta_{(k1),(nn)}) \leq S_0(N_{k1}+\delta_{(k1),(nn)}) + (\delta+\lambda+q_{k_1k_2}+\alpha)n_1t_n$$

where these equations correspond to the actions A_0 and A_n for an arrival during a quantum.

In order to find the size of this linear programming problem, we note that $\Pi_{n=1}^{n=N}\,\Pi_{m=1}^{m=n}\,(C_{nm}+1)$ combinations are possible for the N_{k1}'s. k_1 has the range from 1 to N, while k_2 has the range from 1 to k_1; N(N+1)/2 combinations of (k_1,k_2) are thus possible. There are $\underline{N}+1$ possible actions, and they have to fulfill the three pairs of equations (5.60)-(5.61), (5.62)-(5.63) and (5.64)-(5.65). Notice that (5.60)-(5.61), though, complement (5.62)-(5.63) because V(s,0) does not appear with a plus sign on the left-hand side of (5.62)-(5.63), while it appears in (5.60)-(5.61). Therefore this problem has:

$$\begin{aligned} &(N(N+1)/2+1)\Pi_{n=1}^{n=N}\ \Pi_{m=1}^{m=n}\ (C_{nm}+1) \quad \text{variables,} \\ &(\underline{N}+1)N(N+1)\Pi_{n=1}^{n=N}\ \Pi_{m=1}^{m=n}\ (C_{nm}+1) \quad \text{equations.} \end{aligned} \tag{5.66}$$

The number of variables and equations grows very rapdily, and becomes unmanageable even for small systems; e.g. for $N=\underline{N}=2$ and $C_{nm}=5$ for all queues, there are 375 variables and 2250 equations, while increasing N and $\underline{N}$ to 3 results in 109375 variables and 750000 equations. The problem (5.59)-(5.65) represents a considerable reduction of the original problem (5.30)-(5.43), but additional simplifications and approximations are clearly needed in order to make this approach practical.

A significant savings in the number of variables may be achieved by using:

$$V(a,k_1,k_2,N_{k1}) = V(s,N_{k1}+\delta_{(k1),(k_1k_2)})(\delta+\lambda+\theta_{k_1k_2})(\delta+\lambda+q_{k_1k_2}+\alpha)/(\lambda\theta_{k_1k_2}) \tag{5.67}$$

which is justified by considering the meaning of both terms. $V(s,k_1,k_2,N_{k1})$ is the value received by the users when the initial state is (k_1,k_2,N_{k1}); i.e. when N_{k1} users wait in the system and the user currently being served is from queue (k_1,k_2). Due to the exponential distributions employed in the formulation, it is equivalent to the value the users get when the initial state corresponds to service termination with $N_{k1}+\delta_{(k1),(k_1k_2)}$ users in the system, where the user to be served next is taken from the (k_1,k_2) queue, and where new arrivals have not occurred between the states that were just mentioned. The requirement that the user to be served next is taken from the (k_1,k_2) queue is taken care of by the equations in which $V(a,k_1,k_2,N_{k1})$ appears. In addition, we have to discount $V(a,k_1,k_2,N_{k1})$ back to the time epoch represented by $V(s,N_{k1}+\delta_{(k1),(k_1k_2)})$; the result is given by (5.67) above.

Setting

$$L_0 = n_1t_0 \quad \text{and} \quad V(N_{k1}) = V(s,N_{k1}), \tag{5.68}$$

the problem is simplified into:

$$\text{maximize} \sum_{(\text{all possible } N_{k1})} V(N_{k1}) \tag{5.69}$$

subject to

$$(\delta+\lambda)V(0) - \lambda V(\delta_{(k1),(nn)}) \leq \lambda n_1t_n \tag{5.70}$$

$$\begin{aligned}&-\lambda(\delta+\lambda+q_{k_1k_2}+\alpha)V(N_{k1}+\delta_{(k1),(k_1k_2)}+\delta_{(k1),(nn)}) - \theta_{k_1k_2}q_{k_1k_2}V(N_{k1}+\delta_{(k1),(k_1+1,k_2)})\\ &\quad - \theta_{k_1k_2}\alpha V(N_{k1}) \leq (\delta+\lambda+q_{k_1k_2}+\alpha)(\lambda n_1t_n + S_{k_2}(N_{k1})) + \theta_{k_1k_2}S_0(N_{k1})\end{aligned} \tag{5.71}$$

$$\begin{aligned}&(\delta+\lambda+q_{k_1k_2}+\alpha)^2(\delta+\lambda+\theta_{k_1k_2})V(N_{k1}+\delta_{(k1),(k_1k_2)}) -\\ &\quad - \lambda(\delta+\lambda+q_{k_1k_2}+\alpha)(\delta+\lambda+\theta_{k_1k_2})V(N_{k1}+\delta_{(k1),(k_1k_2)}+\delta_{(k1),(nn)}) -\\ &\quad - \lambda\theta_{k_1k_2}q_{k_1k_2}V(N_{k1}+\delta_{(k1),(k_1+1,k_2)}+\delta_{(k1),(nn)}) - \lambda\theta_{k_1k_2}\alpha V(N_{k1}+\delta_{(k1),(nn)}) \leq\\ &\quad \leq (\delta+\lambda+q_{k_1k_2}+\alpha)n_1t_n + S_0(N_{k1}+\delta_{(k1),(nn)})\end{aligned} \tag{5.72}$$

where $0 \leq n \leq N$, $1 \leq k_1 \leq N$ and $1 \leq k_2 \leq k_1$.

Another approximation, which results in significant savings in the number of equations, may be derived from practical considerations. The set of equations that is given by (5.71) correspond to decisions of users arriving during an overhead period, while (5.72) corresponds to decisions during a quantum. All the non-service operations of the service mechanism are confined to the overhead period, and in particular the display of information. Thus, from practical point of view, the information given to an arriving user is confined to the information displayed during the overhead period, and arrivals during the quantum period have the same information as arrivals during an overhead period. As a result, both have the same decisions; one set of equations may be eliminated, (5.71) say. The linear programming problem consists then of (5.69), (5.70) and (5.72), and contains

$$\begin{aligned} &\Pi_{n=1}^{n=N} \Pi_{m=1}^{m=n} (C_{nm} + 1) \quad \text{variables,} \\ &(\underline{N}+1)(N(N+1)/2 + 1)\Pi_{n=1}^{n=N} \Pi_{m=1}^{m=n} (C_{nm} + 1) \quad \text{equations.} \end{aligned} \tag{5.73}$$

Additional savings in the number of equations can be achieved through the utilization of Lemma 5.1, which restricts the number of possible actions at any given state.

The linear programming problem (5.69)-(5.72) has been considerably reduced in size, as compared to the original problem (5.30)-(5.43). It is still of considerable size even for relatively small systems, as is evident from (5.73). The modified program, though, has a very low density of non-zero elements in each row, and the dual of it has a special structure (see, e.g., Wagner [24], p. 761). Special algorithms were devised for the solution of such problems (Lasdon [15], Sections 4.2 and 4.3), and as a result the approach adopted in this section for the determination of the control numbers is a feasible one.

When more than one urgency value is allowed, the approach given above is still valid. In this case the event of arrival has to be broken into a set of events, each of which having the form 'an arrival with a specific urgency value'. The number of equations and variables grows linearly with the number of urgency values, though the form of the equations is modified only slightly.

Deviations from the exponential distributions assumptions have a more profound effect. If the Markovian nature of the problem is to be retained, an arrival during an overhead period or a quantum needs to know the residual service during this quantum. The problem is then transformed into a continuous state problem, and a different solution approach is needed. The problem can be approximated by a finite state problem if it is assumed that newly arriving users base their decisions on the state of the system as it was at the beginning of the current service period. Such an approximation tends to increase the expected response times by ignoring the service already given to the user currently served, and therefore underestimates the control numbers.

In the formulation of the SMDP the totality of users' point of view was taken into account, and the solution of this problem is thus a social optimal solution. The individual user, though, needs an individual optimal solution. It has been demonstrated by other researchers (e.g., Knudsen [14], Naor [19], Adler and Naor [6] and Yechiali [26,27]) that the individual solution might differ from the social solution; specifically, a user who balks according to the social solution might still join the system according to the individual solution. We expect the same situation to exist in our system; if an individual suffers loss from joining then also the totality of users lose from his joining, and therefore under both solutions joining the system is not done if it results in a loss to the individual user. On the other hand, the social solution might prevent a user from joining though no loss is incurred by him, in order not to crowd the queues and to provide for lower costs for users who will arrive later. The social solution thus gives lower bounds on the control numbers. The effect of future arrivals on the social solution is more marked as the discount rate decreases; therefore a high discount rate is preferred in order to get a good approximation to the individual solution.

5.e. The Control Numbers and System's Parameters

This section deals with the system that was presented in the last section, but from a different point of view. In the former section the users' problem was solved for a given set of system's parameters, while in this section we consider the problem the system's manager faces. We show that there is at least one set of system's charges that is optimal for the system's manager, and that in such a set the service charge is zero.

We first establish the relations between the discounted expected revenues of the system and the discounted expected cost that is incurred by the users. In this discussion, the following notation is used:

$K(i,a)$ = reciprocal of the denominator of $D_\delta(i,j,a)$, as given in column 9 of Table 5.1; this value depends on i and a only.

$\underline{S}_j = c_j + \sum_{m=1}^{m=N} c_m \sum_{n=m}^{n=N} N_{nm}$

$S'_j = n_2(1 + \sum_{n=1}^{n=N} \sum_{m=1}^{m=n} N_{nm})$

$V(i)$ = discounted expected cost incurred by the users when the initial state of the system is i.

$U(i)$ = discounted expected revenues of the system when the initial state is i.

Thus S_j as defined in the remarks to Table 5.1 is equal to $n_1\underline{S}_j + S'_j$, while $C_\delta(i,a)$ of column 12 in this table is equal to $\underline{C}(i,a)$ + $K(i,a)(n_1\underline{S}(i,a) + S'(i,a))$, where $\underline{S}(i,a)$ and $S'(i,a)$ are the $\underline{S}$ and S' that correspond to the pair (i,a) by Table 5.1

In the previous section the constraints of the linear programming problem identify the optimal policy; specifically, constraints that are satisfied in equality for the optimal solution correspond to decisions according to the optimal policy. Using such constraints only, and denoting by D the resulting matrix whose elements are the appropriate $D_\delta(i,j,a)$ and by C the column vector of the appropriate $C_\delta(i,a)$, equation (5.29) is reduced to $V = C + DV$. From (5.28) it is seen that $\sum_{(\text{all } j)} D_\delta(i,j,a) <$ $\sum_{(\text{all } j)} P_{ij}(a) = 1$, and therefore $I - D$ is invertible; thus $V = (I - D)^{-1}C$.

Denoting by $\underline{C}$ the column vector of the $\underline{C}(i,a)$ that correspond to the optimal decisions, by $\underline{C}_1$ a modification of $\underline{C}$ in which all the elements that correspond to decisions A_n $(n > 0)$ are set to zero, and setting $\underline{C}_2 = (\underline{C} - \underline{C}_1)/n_1$, we can write the last equation as

$$V = (I - D)^{-1}(\underline{C}_1 + KS') + n_1(I - D)^{-1}(\underline{C}_2 + K\underline{S}) \qquad (5.74)$$

where K, $\underline{S}$ and S' are constructed by choosing the K(i,a)'s, $\underline{S}$(i,a)'s and S'(i,a)'s that correspond to optimal decisions.

In equation (5.74), the discounted expected payments to the system are represented by $(I - D)^{-1}(\underline{C}_2 + K\underline{S})$; they therefore correspond to U, and (5.74) is reduced to

$$V = (I - D)^{-1}(\underline{C}_1 + KS') + n_1U = R + n_1U \qquad (5.75)$$

which is the desired relation between the discounted expected revenues of the system and the discounted expected costs that are incurred by the users. The vector R is, however, a function of the chosen actions; the same is true of U and V, and in order to stress this dependence we rewrite (5.75) as

$$V(a) = R(a) + n_1U(a) \ . \qquad (5.76)$$

From Table 5.1 it can be seen that the service charge c is independent of the decisions that are made by the users; in other words, if a service charge is included in $C_\delta(i,a)$, then it is charged always in state i, for whatever decision a that is made by the users. This is explained by (5.2), from which we find that the charge for service is the same for all users, and is independent of the state of the system. We may therefore modify the cost structure that is imposed on the system, by subtracting the charge for service from L_0, and eliminating it from the loss function (5.2). This modification decreases the right-hand side of the constraints of the linear programming problem, and as a result of (5.76) it also decreases the discounted revenues of the system. Thus the revenues of the system are a decreasing function of the service charge c, and the system's manager who wants to maximize the revenues of the system will choose the minimal possible service charge; according to our assumptions about the cost structure, he will choose $c = 0$. In other words: in an optimal set of system parameters that maximizes the revenues of the system service will never be charged; the system will derive its revenues from tolls and waiting charges only. This is quite an astonishing result, but it is in accordance with the similar result that was derived in Section 3.e for a round-robin time-sharing system.

It is known from linear programming theory that the objective function of (5.30) is an increasing concave piecewise linear function of the constraints' vector of (5.31)-(5.43), i.e., of the vector whose elements are the terms that appear on the right-hand side of these equations. This vector, in turn, is an increasing function of the system's charges. The objective function is therefore an increasing concave piecewise linear function of the system's charges. But as the charges increase, the number of newly arriving users who join the system decreases; eventually, for high enough charges, all newly arriving users will balk. The system will then always be empty, $V(\text{state } (e\text{-}a)) = (\delta+\lambda)L_0/\delta$, which is also the bound on the objective function. In this case $U = 0$, as is evident from (5.74), as well as from the physical description of the system. From Table 5.1 we find that this is the case when

$$L_0 \leq n_1 t_n + K(i,A_n)(n_2 + n_1 c_n) \quad \text{for all states } i \text{ and decisions } A_n.$$

Alternatively, this is the case when

$$t_n + K(i,A_n)c_n \geq (L_0 - K(i,A_n)n_2)/n_1 \qquad (5.77)$$

for all states i and decisions A_n. Thus system's charges are effectively bounded, and

$$0 \leq c_n \leq \max_{(\text{states } i)} (L_0 - K(i,A_n)n_2)/(K(i,A_n)n_1)$$

$$(n = 1,\ldots,N) \qquad (5.78)$$

$$0 \leq t_n \leq \max_{(\text{states } i)} (L_0 - K(i,A_n)n_2)/n_1 \;.$$

Since the objective function of (5.30) and R are both bounded, it is evident from (5.75) that U is bounded also. It might be positive only for system's charges in the ranges given by (5.78), i.e., for system's charges within a compact set. Moreover, for any given set of decisions it is continuous, because of the continuity of V. Thus U has a finite number of discontinuities only, and therefore attains its maximal value on this compact set of system's charges. In other words, there is at least one set of system's charges which is optimal for the system.

CHAPTER 6

SUMMARY AND SUGGESTIONS FOR FURTHER RESEARCH

This thesis introduces a new scheduling algorithm for time-sharing systems. A system that operate under this scheduling algorithm maintains many levels of queues, all of which are accessible to newly arriving users. Users that wait for service are subjected to a priority scheme that always prefers users who joined the system through a high priority entry queue to those who joined it through a low priority queue. Under this scheme levels are served according to a head-of-the-line priority scheme across levels, and the queues in each level are served according to head-of-the-line priority scheme between the queues in the level. Users who join the system have to pay an appropriate toll, and are subjected to waiting and service charges. These tolls and charges vary according to the entry level, with higher charges for higher priority entry levels.

Newly arriving users decide upon arrival whether to balk, or to join the system; in the latter case they also decide about the entry level they are going to join. This decision is based on the objective of minimizing their expected loss. It is shown that the optimal joining policy is a control-limit type policy, which results in a finite capacity system. This conclusion is derived under mild restrictions, and holds for almost all queueing systems in which balking is allowed and joining decisions are based on loss's minimization.

An M/M/1 round-robin time-sharing system with a finite-room queue is then investigated in detail, and the characteristics of the queue are determined; such characteristics include the expected number of waiting users in the queue, the expected waiting time, etc. Such a finite-room system is the result of imposing charges for the use of the resources of the system. A procedure is proposed for the determination of optimal charges, that maximize the average expected revenue of the system, and the resulting capacity of the system is determined. These results may be used in various ways by existing time-sharing systems that uses a round-robin scheduling algorithm. They may be used for physical design of such systems, and especially the allocation of storage devices. They may be used to derive information about the utilization and throughput of these systems, and also for the derivation of charges that either limit the capacity of the system, or maximize the average expected revenues.

The proposed time-sharing system is investigated under two different assumptions about entry into the system. It is first assumed that users join the entry queues in a random fashion, and descriptive characteristics of the system are derived. A pricing scheme is then incorporated as described above into the system, and the existence of optimal set of charges is proved. A procedure for the determination of an optimal joining policy is then given, as well as upper bounds on the capacity of the system and the charges imposed by the system. The proposed procedure is of theoretical interest mainly, because it requires the solution of a very large linear programming problem. The upper bounds are of practical value, though, and might be used for system's designing and charges determination.

The results of this dissertation may be extended in several directions. All the systems that are analyzed in this thesis are M/M/1 queueing systems; it is of interest to investigate M/G/1 systems, M/M/s systems, and M/G/s systems. Such an analysis will probably be very difficult, due to the finite capacity of these systems. Another extension is to price scheduled batch systems; this is achieved through the allocation of infinite quanta in the models described above. Simpler results and procedures may be expected in batch systems, because they lack much of the interrelations among different queues that characterize the models presented in this thesis. Partial results in this direction can be found in a Master's thesis that is currently developed by M. Modiano at the Operations Research Department of Cornell University. Additional research has to be done in the proposed multi-entrance foreground-background time-sharing system: the determination of optimal charges, and solution to the individual user's optimization problem. Finally, further research is required for the extension of the results brought here to the case of many classes of users, where each class is characterized by its urgency value, loss function, and arrival and service distributions.

REFERENCES

[1] Adiri, I. "A Dynamic Time-Sharing Priority Queue". Journal of the Association for Computing Machinery, Vol. 18, No. 4, October 1971, pp. 603-610.

[2] Adiri, I. "Optimal Decision Policies in a Non-Monopoly Service Station". Technical Report No. RC 3718, IBM, Yorktown Heights, N.Y., February 1972.

[3] Adiri, I. and Avi-Itzhak, B. "A Time-Sharing Queue". Management Science, Vol. 15, No. 11, July 1969, pp. 639-657.

[4] Adiri, I. and Avi-Itzhak, B. "A Time-Sharing Queue with a Finite Number of Customers". Journal of the Association for Computing Machinery, Vol. 16, No. 2, April 1969, pp. 315-323.

[5] Adiri, I. and Yechiali, U. "Optimal Pricing and Priority Purchasing Policies in Queues". Technical Report No. RC 3581, IBM, Yorktown Heights, N.Y., September 1971.

[6] Adler, I. and Naor, P. "Social Optimization versus Self-Optimization in Waiting Lines". Operations Research, Statistics and Economics Mimeograph Series No. 56, Department of Industrial and Management Engineering, Technion, Haifa, Israel, October 1969.

[7] Balachandran, K. R. "Purchasing Priorities in Queues". Management Science, Vol. 18, No. 5, January 1972, pp. 319-326.

[8] Coffman, E. G. and Kleinrock, L. "Feedback Queueing Models for Time-Shared Systems". Journal of the Association for Computing Machinery, Vol. 15, No. 4, October 1968, pp. 549-576.

[9] Conway, R. W., Maxwell, W. L. and Miller, L. W. "Theory of Scheduling". Addison-Wesley Publishing Co., 1967.

[10] Feller, W. "An Introduction to Probability Theory and its Applications". Vol. II, John Wiley & Sons, Inc., 1966.

[11] Greenberger, M. "The Priority Problem and Computer Time-Sharing". Management Science, Vol. 12, No. 11, July 1966, pp. 888-906.

[12] Kleinrock, L. "Time-Shared Systems: A Theoretical Treatment". Journal of the Association for Computing Machinery, Vol. 14, No. 2, April 1967, pp. 242-261.

[13] Kleinrock, L. "Optimum Bribing for Queue Position". Operations Research, Vol. 15, 1967, pp. 304-318.

[14] Knudsen, N. C. "Individual versus Social Optimization in Queueing Systems". Technical Report No. 108, Department of Operations Research, Cornell University, Ithaca, N.Y., April 1970.

[15] Lasdon, L. S. "Optimization Theory for Large Systems". The MacMillan Co., New York, N.Y., 1970.

[16] Marchand, M. "Priority Pricing with Application to Time-Shared Computers". Proceedings of Fall Joint Computer Conference, 1968, pp. 511-519.

[17] McKinney, J. M. "A Survey of Analytical Time-Sharing Models". Computing Surveys, Vol. 1, No. 2, June 1969, pp. 105-116.

[18] Mehta, C. R. "Time-Sharing with an Explicit Queueing Discipline". Technical Report No. 54, O.R. Center, MIT, Cambridge, Massachusetts, July 1970.

[19] Naor, P. "The Regulation of Queue Size by Levying Tolls". Econometrica, Vol. 37, 1969, pp. 15-23.

[20] Rasch, P. J. "A Queueing Theory Study of Round-Robin Scheduling of Time-Shared Computer Systems". Journal of the Association for Computing Machinery, Vol. 17, No. 1, January 1970, pp. 131-145.

[21] Ross, S. M. "Applied Probability Models with Optimization Techniques". Holden-Day, San Francisco, 1970.

[22] Singer, N. M., Kanter, H. and Moore, A. "Price and the Allocation of Computer Time". Proceedings of Fall Joint Computer Conference, Vol. 33, No. 1, 1968, pp. 493-498.

[23] Van Der Heever, R. J. "Computer Time-Sharing Priority Systems". Technical Report No. ORC 69-22, Operations Research Center, University of California, Berkeley, California, August 1969.

[24] Wagner, H. M. "Principles of Operations Research". Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969.

[25] Wolff, R. W. "Time-Sharing with Priorities". Technical Report No. ORC 68-13, Operations Research Center, University of California, Berkeley, California, June 1968.

[26] Yechiali, U. "On Optimal Balking Rules and Toll Charges in the GI/M/1 Queueing Process". Operations Research, Vol. 19, No. 2, March-April 1971, pp. 349-370.

[27] Yechiali, U. "Customers' Optimal Joining Rules for the GI/M/s Queue". Management Science, Vol. 18, No. 7, March 1972, pp. 434-443.

APPENDIX A

DERIVATION OF W_{ij} AND P_{ij} FOR A SINGLE-ENTRANCE,
ROUND-ROBIN, TIME-SHARING SYSTEM

This appendix supplements the analysis which was carried on in Chapter 3. In that chapter a round-robin, time-sharing system was considered, the essential elements of which were described in Section 3.a and the beginning of Section 3.b. Within the framework of this system, we consider the Markov process generated by the time epochs in which a tagged user joins the system, is recycled, or leaves the system. The states of the process are the number of users excluding the tagged one, and the expected response time between consecutive time epochs is sought. In this appendix we develop the expressions for

P_{ij} = P(state at a time epoch is j|state at last time epoch was i)

W_{ij} = E(response time in a cycle|cycle starts at state i, ends at state j),

(i,j = 0,...,k-1).

The first part of the discussion will concentrate on the simple system that was described in Section 3.a. In addition to the analysis of W_{ij} and P_{ij}, the expressions for $\underline{W}_{ij}$, $\overline{W}_{ij}$, $\overline{\underline{W}}_{ij}$ and the corresponding P's will be presented, where $\overline{W}_{ij}$ is the expected response time in the first cycle, $\underline{W}_{ij}$ is the expected response time in the last cycle (both conditioned on the tagged user requesting more than one cycle), and $\overline{\underline{W}}_{ij}$ is the expected response time for a user who stays in the system for exactly one cycle. We then indicate the needed changes in these expressions for more complicated systems. Specifically, we consider two deviations from the assumptions of the simple system: random overhead periods, and general service requirements distribution. Throughout this appendix the notation of Sections 3.a and 3.b is used.

To find W_{ij}, we condition on the number of users who will leave the system during the considered cycle. Let us denote this number by n; n clearly may not exceed i, since at most all the users in the system, i at all, will leave. In addition, if j is less than i, at least $i-j$ users have left the system. The probability that a user leaves the system during the considered cycle is β; the number of users leaving the system is therefore distributed binomially, with parameters i and β. Each of the users who remains in the system adds one quantum and one overhead period to the response time of the tagged user; the same contribution is given by the tagged user, since he will not leave the system during this cycle. Thus $i-n+1$ users contribute each $q + \theta$ to the response time of the tagged user. On the other hand, a user who leaves the system, contributes an overhead period and his expected service time during the quantum he received; this expected time is $(1/\beta)\int_0^q t\alpha e^{-\alpha t}dt = 1/\alpha - q(1-\beta)/\beta$, and therefore the n users who leave the system contribute each $\theta + 1/\alpha - q(1-\beta)/\beta$. Thus, the expected response time in the cycle will be

$$\begin{aligned} W_{ij} &= \sum_{n=\max(0,i-j)}^{n=i} \binom{i}{n}\beta^n(1-\beta)^{i-n}((i-n+1)(\theta+q) + n(\theta + 1/\alpha - q(1-\beta)/\beta)) \\ &= \sum_{n=\max(0,i-j)}^{n=i} \binom{i}{n}\beta^n(1-\beta)^{i-n}((i+1)(\theta+q) + n(1/\alpha - q/\beta)) \ . \end{aligned} \tag{A.1}$$

$\underline{W}_{ij}$ is found in a similar way. In this case the tagged user contributes not a whole quanta, but rather as a user who leaves the system. Thus

(A.1) may still be used, with $(n+1)(1/\alpha - q/\beta)$ replacing $n(1/\alpha - q/\beta)$.

As for $\overline{W}_{ij}$, the expected response time in the first cycle, we find a somewhat different situation. In this case, the tagged user has information about the time, say t, already given to the user now in service. There are two possibilities. If $t \leq \theta$, the user in service is still getting his overhead; in this case $(i+1)(\theta+q)$ in (A.1) is replaced by $(i+1)(\theta+q) - t$. Otherwise, the user in service is considered separately of the other users. He may stay in the system, contribute $q-t$, while the other users contribute $W_{i-1,j-1}$; this will happen with probability $p = \int_{t-\theta}^{q} e^{-\alpha t}dt$. Or he will leave the system, contributing $(1/p)\int_{t-\theta}^{q} t\alpha e^{-\alpha t}dt$, while the other users contribute $W_{i-1,j}$; this will happen with probability $1-p$. The expected response time $\overline{W}_{ij}$ is then computed in a straightforward way. Finally, for $\underline{\overline{W}}_{ij}$, both types of modifications are required.

P_{ij} is developed in a similar way by conditioning on the number n of users who will leave the system during the cycle. Let S_{in} be the random time of service given to those users. S_{in} is the sum of n random variables, each distributed as truncated exponential; let $P_{in}(t) = P(S_{in} \leq t)$, and let $F(t) = P(\text{service of one user} \leq t)$. Then

$$dF(t) = \begin{cases} \alpha e^{-\alpha t}/\beta & 0 \leq t \leq q \\ 0 & q < t \end{cases}$$

and $P_{in}(t) = (F^{*n})(t) = (\text{the nth convolution of } F)(t)$. Thus, the requested probability is given by

$$P_{ij} = \sum_{n=\max(0,i-j)}^{n=i} \binom{i}{n} \beta^n (1-\beta)^{i-n} \int_0^{nq} e^{-\lambda((i-n+1)(\theta+q)+n\theta+t)} \times$$
$$\times (\lambda((i-n+1)(\theta+q)+n\theta+t))^{j-(i-n)}/(j-(i-n))!\,dP_{in}(t)\ . \tag{A.2}$$

The integrand is the probability, according to a Poisson arrival process, of filling the queue up to j users after it had been depleted by n departures, during the time interval of the cycle. Note that for $j = k-1$ we have to take care of balking users, since in this case the system is filled to capacity with the tagged user, and any additional arrivals will balk. In this case, we have to replace 'integrand for k-1' by $\sum_{j\geq k-1}$ 'integrand for j'. This modification is straightforward but complications in its details will not be considered here and will be ignored in the sequel.

The integrand in (A.2) may be simplified to

$$(\lambda^{j-(i-n)} e^{-\lambda((i-n+1)(\theta+q)+n\theta)}/(j-(i-n))!) \times$$
$$\times \sum_{m=0}^{m=j-(i-n)} \binom{j-(i-n)}{m} ((i-n+1)(\theta+q)+n\theta)^{j-(i-n)-m} \int_0^{nq} e^{-\lambda t}\, t^m\, dP_{in}(t). \tag{A.3}$$

The integral in (A.3) is the Laplace transform of the mth moment of a convolution, and using well-known results for such a transform (see, e.g., Feller [10], pp. 411-412), we may replace it by an appropriate sum. The value of such a replacement stems from the complicated form of $P_{in}(\cdot)$, which makes the integration quite hard. A tedious, though straightforward, calculation shows the integral in (A.3) to be equal to

$$(\alpha/\beta)^n (m!/(\alpha+\lambda)^{m+n}) \sum_{\substack{(a_1+\cdots+a_n=m)\\(a_1\geq 0,\ldots,a_n\geq 0)}} \prod_{s=1}^{s=n} (1- e^{-(\alpha+\lambda)q} \sum_{r=0}^{r=a_s} ((\alpha+\lambda)q)^r/r!) \tag{A.4}$$

Combining (A.2), (A.3) and (A.4) together will yield the desired form of P_{ij}. This form is well suited for computer programming, but impractical for hand calculations.

As for $\underline{P}_{ij}$, $\overline{P}_{ij}$ and $\overline{\underline{P}}_{ij}$: the same modifications which were indicated for the corresponding W's are required. The resulting expressions are even more complicated, and will not be brought here.

The solution developed above can be extended to a system in which our simplifying assumptions do not hold. First, we may interpret k as the capacity of the specific state i, rather than as the capacity of the system. Thus, situations in which the capacity is a function of the system's state are easily handled. This observation will be useful for a foreground-background model. Second, the assumption of a fixed overhead is far from crucial. If this is not the case, we have to add in (A.2) an additional integration, performed on the convolution of the i+1 overheads. The result will resemble (A.3) and (A.4), but will be much more complicated. Similar treatment is needed for random quantum size. Third, it is quite easy, in principle, to take into account different urgency values. In this case, to each urgency value corresponds a different capacity, while the Poisson stream of arrivals is broken down into many streams, each corresponding to a different urgency value. The integral in (A.2) will then be replaced by a sum over all combinations of arrivals from these streams, while for each stream an expression similar to (A.2) will emerge.

Finally, a more general service distribution can be accommodated. In this case, we have to know for each user in the system the number of quanta he already received. Each user will therefore have a different probability of leaving the system, and instead of the binomial distribution of (A.2) and (A.3) a summation over all possibilities of departing users has to be used.