CONGESTION PROBLEMS IN COMPUTING SYSTEMS

I.Mitrani

Ph. D. Thesis

November, 1972.

UNIVERSITY OF NEWCASTLE UPON TYNE

Acknowledgements

I would like to thank Professor E.S.Page for directing and encouraging me throughout the preparation of this thesis and for the many valuable comments and suggestions he made.

I have also benefitted from discussing some aspects of my work with Professor E.G.Coffman.

Certain properties of one of the models were pointed out to me by a referee.

My wife helped me with advice on English grammar and spelling.

During the period of this research I was on the staff of the University of Newcastle upon Tyne.

Abstract.

The subject of this dissertation is the modeling and analysis of multiprogramming computing systems.

Several cyclic queuing models are studied. The systems which they approximate have one central processor and one or more peripheral processors; queues are served in order of arrival or according to priority disciplines. Except in the simplest case of 'one central and one peripheral processor, FIFO queuing and exponential service times at both processors', all models are analysed in the steady-state.

Expressions for the central processor utilisation factor, the rate of departures from the system, the average residence time and, in the case mentioned above, the Laplace transforms of the interarrival interval and of the residence time are obtained.

Table of Contents

Page

Chapter 1

Chapter 2

Chapter 3

Chapter 4

CHAPTER 1.

Introduction.

A possible way of describing a computing system is to say that it is a 'black box' which accepts programs, or series of instructions, and executes them. A mathematical model of the system, consistent with this point of view, is a single-server queuing process, where the server, the customers and the queuing discipline represent the computer, the programs and the program scheduling strategy respectively. The general form of such a model (allowing for interruptions of service before completion) is pictured in figure 1.1 .

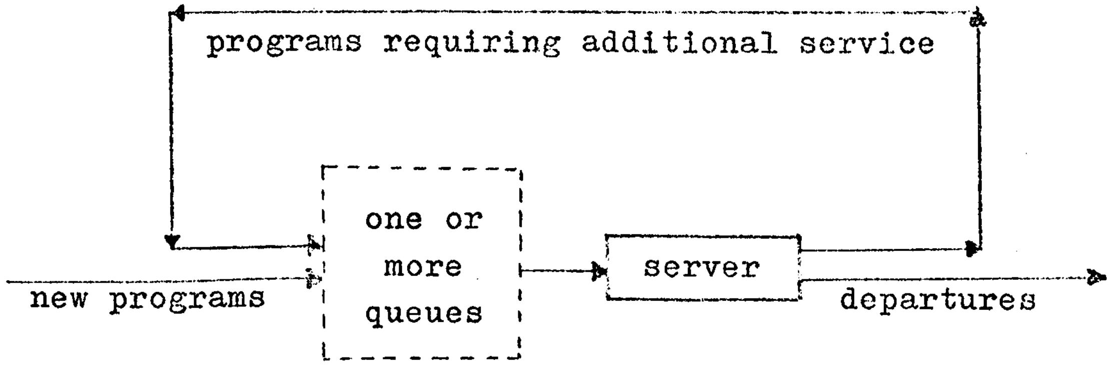


Figure 1.1

Time-sharing strategies (those which give each program in turn a quantum of service and if that service is not sufficient, return the program to an appropriate queue) in particular, have been extensively studied by means of single-server models. A survey of research in this field can be found in McKinney [20]. More recently, O'Donovan [19] developed a general method for finding average waiting times in time-sharing and priority systems.

While adequate in many cases, single-server models fail to capture some important aspects of present-day computing systems. Chief among these is the ability of different components of the system to perform different functions in parallel. To account for this ability, we shal regard computing systems as collections of 'black boxes' , or processors; the collection, as a whole, accepts programs and executes them but different processors execute different portions of the programs.

One or more of the processors are designed to perform 'calculations' ; they are usually called 'central processors' or 'Central Processing Units' (CPUs). Other processors are used for transmission of information between main storage and different types of secondary storage; these are sometimes called 'channels', or 'Data Transmission Units' (DTUs) , or Input/Output units (I/O units); we shall call them 'peripheral processors'.

Most modern computing systems are operated in 'multiprogramming' mode, executing a number of programs (or jobs, as they are usually called) concurrently. In general, multiprogramming involves the formation of queues within the system : for example, in a computing system consisting of m processors and multiprogrammed at degree n $(n>m)$, at most m jobs can be in execution at any one time and at least n-m jobs must be waiting in various queues. Furthermore, computer equipment being expensive, computing systems are usually utilised to their full capacity. This, in practical terms, means that there are always jobs outside the system, ready to take the place of any job which has completed its execution, i.e. the number of jobs in the system is maintained at nearly constant level.

It seems reasonable, therefore, that mathematical models of computing systems be based on many-server queuing networks in which a constant number of customers circulate between the servers. It is with the definition and analysis of such models that this dissertation is concerned.

Survey of existing work.

Gaver [10] (1967) was perhaps the first to study a many-server model of a multiprogramming system. His model consists of one central processor and a number of identical input/output units; a constant number of jobs are being multiprogrammed, each joining the central processor queue and the input/output queue alternately , until its demand is satisfied. The central processor utilisation factor is calculated, for several different distributions of the central processor service times (the input/output service times are assumed to be distributed exponentially).

Wallace and Mason [11] (1969) solved numerically a model with one central processor and one input/output unit.

A special feature of the model is that each job execution begins with a burst of demand for input/output. The results of the analysis are displayed in a series of graphs showing the central processor utilisation factor as a function of the degree of multiprogramming, the service times averages and the average number of input/output requests per job.

Several cyclic queuing models of multiprogramming systems with one central and one peripheral processor have been studied by Chen and Shedler [12] (1969) , Shedler [13] (1970) , Lewis and Shedler [3] (1971) and Adiri, Hofri and Yadin [14] (1971) :

[12] and [13] deal with paging systems; the basic assumptions of the models are as in [10], except that here the central processor is the one with generally distributed service times;

[3] is concerned with the effect of supervisor overhead on the central processor utilisation factor. The supervisor functions are represented by additional processors in the calculating-input/output cycle;

[14] deals with a case where the the number of jobs in the system is not fixed, an infinite queue can form in front of one or both processors. The service times of both processors are assumed to be distributed exponentially.

A model of a different kind was analysed recently by Omahen and Schrage [15] (1972) . They assumed that the resources which a job needs in order to execute are allocated to it simultaneously, rather than in sequence. Thus, in a system consisting of three processors, a job which requires two processors can be executed in parallel with a job which requires one processor, but not with one which requires two or three processors. Conditions for non-saturation are derived, for different service disciplines.

The contents of subsequent chapters.

A cyclic queuing model of a two-processor computing system multiprogrammed at a fixed degree, with exponentially distributed service times for both processors, is defined in chapter 2 (this model is a special case of the one used by Koenigsberg [16] to study coal mine operations). The transient distribution of the number of jobs in the central processor queue is obtained.

The same model is analysed in the steady-state in chapter 3 . The central processor utilisation factor, the rate of departures from the system and the average residence-in-the-system time are found. Also, the distributions of queue sizes at specific points of time are obtained and are used to find the Laplace transforms of the interarrival interval and of the residence-in-the-system time. The definition of the model, part of the steady-state analysis and section 4.6 appeared in [17].

An attempt to validate the model, using a real-life computing system for comparison, is made in chapter 4 . The model is then analysed under the assumption that the peripheral processor service times have general distribution. Finally, the effect of the degree of multiprogramming on job turnaround is discussed.

Chapter 5 deals with a multiprogramming system consisting of one central and many peripheral processors. The peripheral processors are not assumed to be equivalent, as they are in 10 ; a separate queue forms in front of each of them. Part of this chapter appeared in [18].

Chapter 6 is conserned with a two-processor priority multiprogramming system. The jobs in the system are assumed to have different characteristics and to be assigned distinct priorities. Preemptive and non-preemptive queuing disciplines at the peripheral processor are considered.

CHAPTER 2.

2.0 Summary.

We shall introduce here a mathematical model of a fixed-number-of-tasks multiprogramming system with one central and one peripheral processor. The behaviour of the system will be represented by a continuous parameter stochastic process which, under the assumptions of the model will have the Markov 'memoryless' property.

The rest of the chapter will be devoted to finding the transient distribution of that Markov process. A system of linear differential equations of first order for the distribution functions will be solved by converting it first to a system of linear algebraic equations for their Laplace transforms. Finally, it will be shown that the queue under consideration is equivalent to a M/M/1 queue with limited waiting room.

2.1 The model.

The system that we wish to analyse is pictured in fig. (2.1). At any time there are exactly N $(N \geqslant 1)$ customers (jobs, from now on) in the system; N is usually referred to as 'the degree of multiprogramming'. The central and the peripheral processors are represented by two servers which we denote by S_0 and S_1 respectively. When more than one job requires a server, they are selected for service in order of arrival. We shall denote by $Q_0(t)$ and $Q_1(t)$ the sizes of the S_0 and the S_1 queues at time t. Obviously $Q_0(t) + Q_1(t) = N$ and we can therefore take $Q_0(t)$, for example, to describe the state of the system at time t.

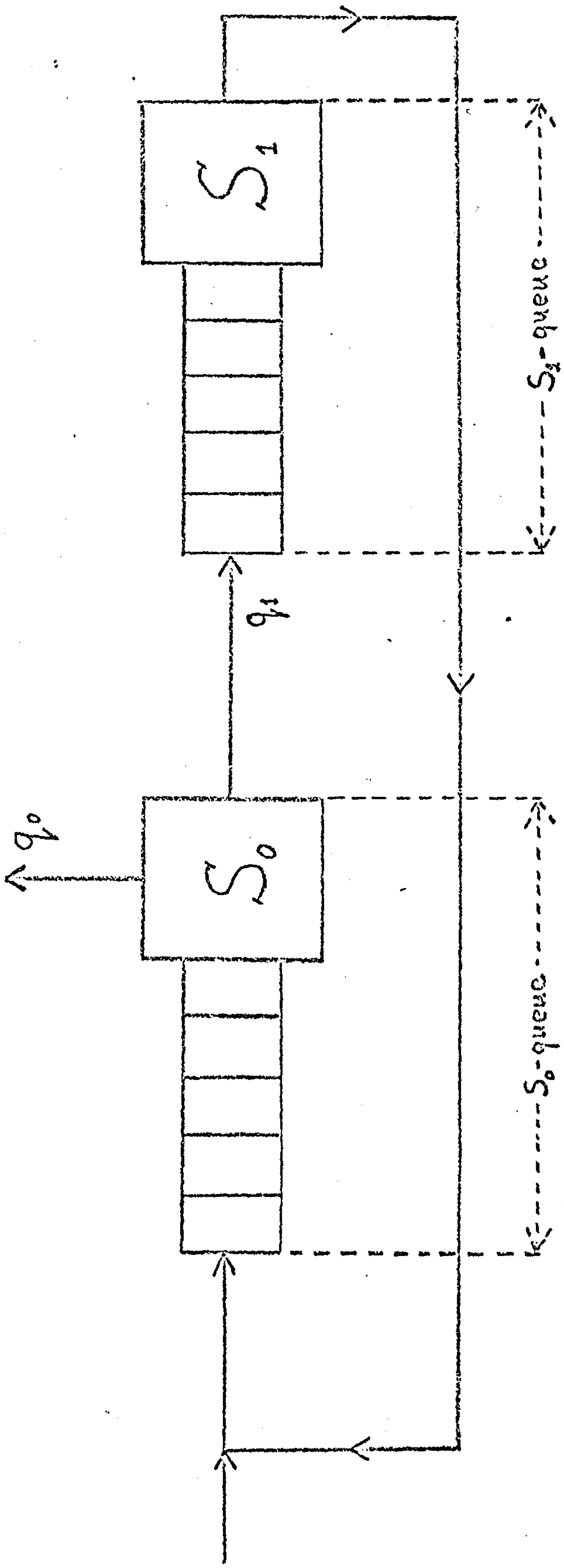


Figure 2.1

(Unless specified otherwise, when we talk about a queue it will be understood to include the job, if any, receiving service. Similarly, when we talk about a 'wait at a server' it will be understood to include the service.)

Upon entry into the system, jobs join at the end of the S_0-queue. After being served by S_0, a job either leaves the system, or joins at the end of the S_1-queue; the former takes place with probability q_0 $(q_0 \in (0,1))$ and the latter with probability $q_1 = 1 - q_0$. After being served by S_1, jobs join again at the end of the S_0-queue. The above implies that K – the number of S_0-services required by a job – is a geometrically distributed random variable:

$$P(K = k) = q_0 q_1^{k-1} \quad ; \quad k=1,2,\ldots$$

A sequence of 'waiting at S_0 – waiting at S_1' will be called a 'cycle'. Thus the residence time of a job (the time between admission to and departure from the system) consists of $K-1$ cycles followed by a wait at S_0 $(K=1,2,\ldots)$.

To maintain the number of jobs in the system constant, we assume that when a job leaves the system, a new one is admitted and joins at the end of the S_0-queue, the replacement being instantaneous.

Expressed in computing terms, the queuing discipline described above reflects the fact that the execution of a job consists of alternative CPU and Input/Output intervals. It also reflects the fact that the system is working under heavy demand conditions: there is a pool of jobs outside the system at all times, waiting and ready to be admitted for execution.

It remains to make specific assumptions concerning the distribution of S_0 and S_1 service times. We shall assume that consecutive S_0-service times are independent, identically distributed random variables with distribution function $F_0(x)$ and that consecutive S_1-service times are independent, identically distributed random variables with distribution function $F_1(x)$. For the present, both $F_0(x)$ and $F_1(x)$ will be assumed exponential distribution functions, with parameters m_0 and m_1 respectively;

$$F_0(x) = 1 - e^{-m_0 x} \quad ; \quad F_1(x) = 1 - e^{-m_1 x}$$

Most of the results of practical interest can be obtained under a less restrictive assumption, namely that only one of the distribution functions is exponential. This will be shown in chapter 4.

2.2 Transient behaviour of the process.

The family of random variables $Q_0(t)$, with t running through the set of the non-negative real numbers, is a (continuous parameter) stochastic process which can be in the finite set of states $0,1,\ldots,N$. Furthermore, the exponential form of the service times distributions and the geometrical distribution of the number of S_0-services required by a job, ensure that $\{Q_0(t),\ 0 \leq t < \infty\}$ is a Markov process.

We are interested in finding the probabilities

$$p_i(t) = P(Q_0(t) = i) \ ; \ i=0,1,\ldots,N$$

These probabilities will be referred to as 'the transient distribution of the process' or 'the time-dependent distribution of the process'. They satisfy a system of linear differential equations of first order - the Kolmogorov's forward equations:

$$
\begin{aligned}
p_0'(t) &= - m_1 p_0(t) + q_1 m_0 p_1(t) \\
p_1'(t) &= - (q_1 m_0 + m_1) p_1(t) + q_1 m_0 p_2(t) + m_1 p_0(t) \\
&- \qquad (2.1) \\
p_{N-1}'(t) &= - (q_1 m_0 + m_1) p_{N-1}(t) + q_1 m_0 p_N(t) + m_1 p_{N-2}(t) \\
p_N'(t) &= - q_1 m_0 p_N(t) + m_1 p_{N-1}(t)
\end{aligned}
$$

Equations (2.1) together with a set of initial conditions - $p_N(0)=1$; $p_i(0)=0$; $i=0,1,\ldots,N-1$ for example - yield the transient distribution of the $Q_0(t)$ process.

(Note that since the initial conditions represent a probability distribution, they must satisfy the normalising equation $p_0(0)+p_1(0)+\ldots+p_N(0)=1$. It can then be seen, by adding all equations in (2.1) , that the normalising equation is satisfied for all t .)

There are standard methods for solving a system of linear differential equations with constant coefficients; these usually involve finding the eigenvalues, and then the eigenvectors of the coefficient matrix. We shall save a great deal of the work by taking Laplace transforms of both sides of the equations in (2.1) .

The Laplace transform of a function $f(x)$ is denoted by $f^*(s)$ and is defined as

$$
f^*(s) = \int_0^\infty e^{-sx} f(x)\,dx \quad ; \quad s > 0
$$

provided that the integral in the right-hand side converges.

There is a simple relationship between the Laplace transform of a function and the Laplace transform of the derivative of the function:

$$f'^*(s) = -f(0) + s.f^*(s) \qquad (2.2)$$

whenever $f^*(s)$ exists.

Using (2.2) we can transform (2.1) into a system of linear algebraic equations for the Laplace transforms $p_i^*(s)$ ($p_i^*(s)$ exist for all positive values of s because $0 \leqslant p_i(t) \leqslant 1$) :

$$\begin{aligned}
-p_0(0) + sp_0^*(s) &= -m_1p_0^*(s) + q_1m_0p_1^*(s) \\
-p_1(0) + sp_1^*(s) &= -(q_1m_0+m_1)p_1^*(s) + \\
&\quad + q_1m_0p_2^*(s) + m_1p_0^*(s) \\
&- \qquad (2.3) \\
-p_{N-1}(0) + sp_{N-1}^*(s) &= -(q_1m_0 + m_1)p_{N-1}^*(s) + \\
&\quad + q_1m_0p_N^*(s) + m_1p_{N-2}^*(s) \\
-p_N(0) + sp_N^*(s) &= -q_1m_0p_N^*(s) + m_1p_{N-1}^*(s)
\end{aligned}$$

Let us denote the vector $(p_0^*(s), p_1^*(s), \dots, p_N^*(s))$ by $\underline{p^*(s)}$, the vector $(p_0(0), p_1(0), \dots, p_N(0))$ by $\underline{p(0)}$, and let $A(s)$ be the matrix (dimensions $(N+1 \times N+1)$)

$$\begin{bmatrix}
s+m_1 & -q_1m_0 & 0 & \dots & 0 & 0 \\
-m_1 & s+m_1+q_1m_0 & -q_1m_0 & \dots & 0 & 0 \\
0 & -m_1 & s+m_1+q_1m_0 & \dots & 0 & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
0 & 0 & 0 & \dots & s+q_1m_0+m_1 & -q_1m_0 \\
0 & 0 & 0 & \dots & -m_1 & s+q_1m_0
\end{bmatrix}$$

Now (2.3) can be written in matrix form as

$$A(s)\underline{p^*(s)} = \underline{p(0)} \tag{2.4}$$

The solution of (2.4) is given, according to Cramer's rule, by

$$p_{i-1}^*(s) = \frac{|A^{(i)}(s)|}{|A(s)|} \quad ; \; i=1,2,\ldots,N+1 \tag{2.5}$$

where $A^{(i)}(s)$ is the matrix formed from $A(s)$ by substituting $\underline{p(0)}$ for its i-th column and $|A|$ denotes the determinant of matrix A .

2.3 Explicit solution for the Laplace transforms.

It is not difficult to write an expression for $|A(s)|$. Consider the sequence of matrices A_k ; $k=1,2,\ldots$ where A_k is given by

$$A_k = \begin{bmatrix} a & b & 0 & 0 & \ldots & 0 & 0 \\ c & a & b & 0 & \ldots & 0 & 0 \\ 0 & c & a & b & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & a & b \\ 0 & 0 & 0 & 0 & \ldots & c & a \end{bmatrix}$$

(dimensions $(k\times k)$), with $a=s+q_1m_0+m_1$; $b=-q_1m_0$; $c=-m_1$.

By manipulating $A(s)$ one can see that

$$|A(s)| = s|A_N| \tag{2.6}$$

(Add the first row of $A(s)$ to the second; the resulting second row to the third, etc. Then subtract the second column of the resulting matrix from the first; the third from the second, etc. This produces an $(N+1\times N+1)$ matrix, with the main $(N\times N)$ minor equal to A_N and the last row consisting of N zeros followed by an s .)

To find $|A_k|$ we note that expansion along its first row yields, for $k=2,3,\ldots$

$$|A_k| = a|A_{k-1}| - bc|A_{k-2}| \tag{2.7}$$

where $|A_1| = a$ and, by definition, $|A_0| = 1$ (the verification of (2.7) for $k=2$ is straightforward).

The characteristic equation of the simple recurrence relation (2.7) is

$$x^2 - ax + bc = 0$$

and its two roots are given by

$$\begin{aligned} x_{1,2} &= \tfrac{1}{2}\left[a \pm (a^2 - 4bc)^{\frac{1}{2}}\right] = \\ &= \tfrac{1}{2}\left[s+q_1m_0+m_1 \pm ((s+q_1m_0+m_1)^2 - 4q_1m_0m_1)^{\frac{1}{2}}\right] \end{aligned} \tag{2.8}$$

The general solution of (2.7) is of the form

$$|A_k| = \begin{cases} (\tfrac{1}{2}a)^k(C_1 + kC_2) & \text{when} \quad x_1=x_2=\tfrac{1}{2}a \\ C_1x_1{}^k + C_2x_2{}^k & \text{when} \quad x_1 \neq x_2 \end{cases}$$

where C_1 and C_2 are arbitrary constants. To satisfy our initial conditions they must be equal to

$$C_1 = \begin{cases} 1 & \text{when} \quad x_1 = x_2 \\ \dfrac{x_2-a}{x_2-x_1} & \text{when} \quad x_1 \neq x_2 \end{cases}$$

$$C_2 = \begin{cases} 1 & \text{when} \quad x_1 = x_2 \\ (a-x_1)/(x_2-x_1) & \text{when} \quad x_1 \neq x_2 \end{cases}$$

Substitution in (2.6) now yields

$$|A(s)| = s(x_1{}^N \frac{x_2 - a}{x_2 - x_1} + x_2{}^N \frac{a - x_1}{x_2 - x_1}) = \tag{2.9}$$
$$= \frac{s}{(a^2 - 4q_1m_0m_1)^{\frac{1}{2}}} (x_1{}^{N+1} - x_2{}^{N+1}) \quad ; \quad x_1 \neq x_2$$

and

$$|A(s)| = (N+1)s(\tfrac{1}{2}a)^N \quad ; \quad x_1 = x_2 = \tfrac{1}{2}a \tag{2.9a}$$

Formulae for $|A^{(i)}(s)|$ - the numerators in (2.5) - can be obtained by expanding $|A^{(i)}(s)|$ along its i-th column; and although the expansion is not difficult to perform in the general case, the resulting expressions are very cumbersome. We shall give the results for the case when $\underline{p(0)}$, the right-hand side in (2.4), is defined as

$$\underline{p(0)} = (0,0,\ldots,0,1) \tag{2.10}$$

i.e. when at $t=0$, $Q_0(0)=N$. $|A^{(i)}(s)|$ is now equal to the determinant left after crossing the i-th column and the last row out of $|A(s)|$, multiplied by $(-1)^{N+i+1}$. It becomes the product of two determinants - one triangular with $(-q_1m_0)$ on the main diagonal (columns $i+1, i+2, \ldots, N+1$ of $|A(s)|$) - and one closely resembling $|A_{i-1}|$, the only difference being that the element in the top left-hand corner is $s+m_1$ instead of $a = s + q_1m_0 + m_1$. By adding to and subtracting from it q_1m_0 we get

$$|A^{(i)}(s)| = (-1)^{N+i+1}(-q_1m_0)^{N-i+1}(|A_{i-1}| - q_1m_0|A_{i-2}|) = \tag{2.11}$$
$$= \frac{(q_1m_0)^{N-i+1}}{(a^2 - 4q_1m_0m_1)^{\frac{1}{2}}} (x_1{}^i - x_2{}^i - q_1m_0(x_1{}^{i-1} - x_2{}^{i-1}))$$

when $x_1 \neq x_2$. If $x_1 = x_2 = \frac{1}{2}a$, then

$$|A^{(i)}(s)| = (q_1 m_0)^{N+1-i}(i(\tfrac{1}{2}a)^{i-1} - q_1 m_0 (i-1)(\tfrac{1}{2}a)^{i-2}) \qquad (2.11a)$$

(We point out that (2.11) is true for $i=1,2,\ldots,N+1$, although its derivation is only valid for $i \geqslant 2$. The same observation applies to (2.11a) . Strictly speaking, (2.11a) is irrelevant for the purpose of finding the Laplace transforms because, as it is easy to see, $x_1 \neq x_2$ when $s > 0$. We have included it for completeness, and also because it will be mentioned later.)

Substituting (2.9) and (2.11) into (2.5) yields, for $i=0,1,\ldots,N$

$$p_i^*(s) = (q_1 m_0)^{N-i} \frac{x_1^{\,i+1} - x_2^{\,i+1} - q_1 m_0 (x_1^{\,i} - x_2^{\,i})}{s(x_1^{\,N+1} - x_2^{\,N+1})} \qquad (2.12)$$

thus determining the Laplace transforms of the transient distribution functions $p_i(t)$; $i=0,1,\ldots,N$ for the initial conditions (2.10) .

2.4 Inversion of the Laplace transforms.

Finding a function when its Laplace transform is given, is in general a laborious and costly process - mainly due to the complexity of the inversion formulae (one such formula, for example, is given on page 230 in Feller [1b]).

In our case, however, the inversion can be performed quite easily because we know already the general form of the functions $p_i(t)$. Going back to the system of linear differential equations (2.1) we note that if the coefficient matrix has eigenvalues which are different real numbers -

denote them by $v_1, v_2, \dots, v_{N+1}$ - then the solution of the system is a linear combination of exponential functions, i.e. it is of the form

$$p_i(t) = \sum_{j=1}^{N+1} a_{i,j}\, e^{v_j t} \quad ; \quad i=0,1,\dots,N \qquad (2.13)$$

If that is the case, then the Laplace transforms $p_i{*}(s)$ for which we already have one expression, must be equal to

$$p_i{*}(s) = \sum_{j=1}^{N+1} \frac{a_{i,j}}{s - v_j} \quad ; \quad i=0,1,\dots,N \qquad (2.14)$$

and by comparing the right-hand sides of (2.14) and (2.12) we can find the unknown coefficients $a_{i,j}$.

We shall now prove that v_j ; $j=1,2,\dots,N+1$ are distinct real numbers.

If v is subtracted from the main diagonal of the (2.1) coefficient matrix, the result is precisely $(-1)A(v)$, where $A(s)$ is the matrix in (2.4). v_j ; $j=1,2,\dots,N+1$ are, therefore, the roots of

$$|A(v)| = 0 \qquad (2.15)$$

From (2.9) we see that one of the v_j is zero, and the rest satisfy $x_1^{N+1} = x_2^{N+1}$; $x_1 \neq x_2$ (it is readily seen that the value of v for which $x_1 = x_2 \neq 0$, is not a solution of (2.15)). We can thus write $v_{N+1} = 0$ and, for $j=1,2,\dots,N$

$$\frac{x_1(v_j)}{x_2(v_j)} = \cos(j\,\frac{2\pi}{N+1}) + i.\sin(j\,\frac{2\pi}{N+1}) \qquad (2.16)$$

(the right-hand side of (2.16) is the trigonometrical representation of the j-th value of the N+1-st root of 1 . i is the imaginary unit and $x_1(v_j)$ and $x_2(v_j)$ are given by (2.8) with s replaced by v_j).

In order to determine from (2.16) the nature of v_j ; $j=1,2,\dots,N$ we shall use a geometrical argument:

Let

$$u_j = \tfrac{1}{2}(v_j + q_1m_0 + m_1)$$

and

$$y_j = \tfrac{1}{2}\left[(v_j + q_1m_0 + m_1)^2 - 4q_1m_0m_1\right]^{\frac{1}{2}} \quad ;$$

then $x_1(v_j) = u_j + y_j$ and $x_2(v_j) = u_j - y_j$. Since, according to (2.16) , $x_1(v_j)$ and $x_2(v_j)$ are equal in modulus, then the vectors representing the complex numbers u_j and y_j must be perpendicular (of all parallelograms only the rectangle has its two diagonals of equal size). This means that $y_j = i.c.u_j$ for some real c . We know from the quadratic equation for x_1 and x_2 that $x_1(v_j)x_2(v_j) = q_1m_0m_1$, which is a real number. On the other hand,

$$x_1(v_j)x_2(v_j) = (u_j + y_j)(u_j - y_j) = u_j^2 - y_j^2 =$$
$$=u_j^2 + c^2y_j^2 = u_j^2(1 + c^2)$$

This implies that either u_j is real, and therefore y_j is imaginary, or u_j is imaginary and y_j is real. The second alternative should be discarded because if u_j is imaginary, i.e. $v_j = i.d -(q_1m_0 + m_1)$ for some real d , then y_j would also be imaginary, which is impossible.

Since the above argument is valid for all $j=1,2,\dots,N$, it proves that all u_j , and therefore all v_j , are real numbers.

Let $x_1(v_j) = r_j\exp(i.f_j)$ and $x_2(v_j) = r_j\exp(i.(-f_j))$ with $0<f_j<\pi$; $r_j = q_1m_0m_1$. (2.16) can be rewritten as

$$\frac{x_1(v_j)}{x_2(v_j)} = e^{2i.f_j}$$

and by comparing this form of (2.16) with the old one, we find that $f_j = j\frac{\pi}{N+1}$; $j=1,2,\dots,N$. Since u_j is the real component of $x_1(v_j)$, we have

$$\cos(f_j) = \frac{u_j}{r_j} = \frac{v_j + q_1m_0 + m_1}{2(q_1m_0m_1)^{\frac{1}{2}}} = \cos(j\frac{\pi}{N+1})$$

which gives, for $j=1,2,\dots,N$

$$v_j = 2(q_1m_0m_1)^{\frac{1}{2}}\cos(j\frac{\pi}{N+1}) - (q_1m_0 + m_1) \qquad (2.17)$$

Obviously v_j ; $j=1,2,\dots,N$ are distinct and also distinct from $v_{N+1} = 0$. We have now proved that (2.13) and (2.14) hold.

Turning our attention to (2.12) we note that after reducing it by the factor $x_1 - x_2$, the fraction defining $p_i^*(s)$ becomes a rational function of s and the power of the polynomial in the numerator is lower than the power of the polynomial in the denominator. Furthermore, the roots of the denominator are precisely $v_1, v_2, \dots, v_{N+1}$. Therefore the reduced fraction can be represented as a linear combination of elementary fractions of the type (2.14) in a unique way. Denoting for short

$$G_i(s) = (q_1m_0)^{N-i}(x_1^{i+1} - x_2^{i+1} - q_1m_0(x_1^{i} - x_2^{i}))$$

$$H_i(s) = s(x_1^{N+1} - x_2^{N+1}) = H(s)$$

we can write

$$a_{i,j} = \lim_{s\to v_j}\frac{(s - v_j)G_i(s)}{H(s)} \qquad (2.18)$$

The undeterminacy in the right-hand side of (2.18) can be resolved according to L'Hospital's rule:

$$a_{i,j} = \frac{G_i(v_j)}{H'(v_j)} \quad ; \; i=0,1,\dots,N \; ; \; j=1,2,\dots,N+1 \qquad (2.19)$$

((2.19) does not hold when $j=N+1$ and $q_1m_0=m_1$; we shall deal with that case later).

Substitution of (2.19) into (2.13) now yields explicit expressions for the transient probability distribution functions of the stochastic process $\{Q_0(t) \;,\; t \geqslant 0\}$, given the initial conditions (2.10) :

$$p_i(t) = \sum_{j=1}^{N} \frac{G_i(v_j)}{H'(v_j)} e^{v_j t} + \frac{G_i(0)}{H'(0)} \quad ; \; i=0,1,\dots,N \qquad (2.20)$$

where v_j ; $j=1,2,\dots,N$ are given by (2.17) .

2.5 Alternative initial conditions.

It is perhaps worth pointing out that while the eigenvalues $v_1, v_2, \dots , v_{N+1}$ do not depend on the initial conditions, the coefficients $a_{i,j}$, in general, do. In the general case, the polynomials $(x_1 - x_2)|A^{(i+1)}(s)|$ should be substituted for $G_i(s)$ in (2.19) , where $|A^{(i+1)}(s)|$ are the determinants in the numerator in (2.5). However, one important property of the process is that $a_{i,N+1}$; $i=0,1,\dots,N$ do not depend on the vector of initial conditions $\underline{p(0)}$ (as long as the sum of its elements is 1). We shall prove this, and also find the values of $a_{i,N+1}$, because they play an important role in the long-run distribution of the $Q_0(t)$ process.

Let us set $s=0$ in $|A^{(i)}(s)|$. We have

$$|A^{(i)}(0)| = \begin{vmatrix} m_1 & -q_1m_0 & 0 & \dots & p_0(0) & \dots & 0 & 0 \\ -m_1 & q_1m_0+m_1 & -q_1m_0 & \dots & p_1(0) & \dots & 0 & 0 \\ 0 & -m_1 & q_1m_0+m_1 & \dots & p_2(0) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & p_{N-1}(0) & \dots & q_1m_0+m_1 & -q_1m_0 \\ 0 & 0 & 0 & \dots & p_N(0) & \dots & -m_1 & q_1m_0 \end{vmatrix}$$

If all rows of $|A^{(i)}(0)|$ are added to the last one, then all elements of the last row will become zeros, except the i-th element, which will be equal to $p_0(0)+p_1(0)+\dots+p_N(0) = 1$. Expansion of $|A^{(i)}(0)|$ by its last row will then eliminate the i-th column and therefore the result will be independent of the vector $\underline{p(0)}$: (2.19) can thus be used to determine $a_{i,N+1}$ whatever the initial conditions. It gives

$$a_{i,N+1} = (q_1m_0)^{N-i} \frac{x_1^{i+1}(0) - x_2^{i+1}(0) - q_1m_0(x_1^{i}(0) - x_2^{i}(0))}{x_1^{N+1}(0) - x_2^{N+1}(0)}$$

for $i=0,1,\dots,N$. These can be simplified because

$$x_1(0) = q_1m_0 \quad ; \quad x_2(0) = m_1$$

Now the expressions become

$$a_{i,N+1} = (q_1m_0)^{N-i} \frac{q_1m_0m_1^{\,i} - m_1^{\,i+1}}{(q_1m_0)^{N+1} - m_1^{\,N+1}} \; ; \; i=0,1,\dots,N \qquad (2.21)$$

The above derivation is valid only when $x_1(0) \neq x_2(0)$. To find $a_{i,N+1}$ when $q_1m_0=m_1$ we can either go back to (2.18) and use (2.9a) and (2.11a) for $G_i(s)$ and $H(s)$ respectively, or we can divide the numerator and the denominator in the right-hand side of (2.21) by $(q_1m_0-m_1)$ and let $q_1 m_0 \to m_1$. In both cases we arrive at

$$a_{i,N+1} = \frac{1}{N+1} \; ; \; i=0,1,\ldots,N \; ; \; q_1 m_0 = m_1 \qquad (2.21a)$$

2.6 Comparison with an M/M/1 queue.

This completes our analysis of the transient behaviour of the $Q_0(t)$ process. Before proceeding with the long-run analysis, we would like to make the following observation:

Consider a one-server queuing system with Poisson arrivals and exponentially distributed service times, where the arrival rate is m_1 and the average service time is $\frac{1}{q_1 m_0}$. Add the restriction that there cannot be more than N customers in the system at any one time. Now, if $Q(t)$ is the number of customers in this system at time t and $p_i(t) = P(Q(t)=i)$; $i=0,1,\ldots,N$, then the functions $p_i(t)$ satisfy precisely the system of differential equations (2.1). Therefore, the process $\{Q(t) \; , \; t \geqslant 0\}$ is equivalent to the process $\{Q_0(t) \; , \; t \geqslant 0\}$ which we are studying.

CHAPTER 3.

3.0 Summary.

We shall study the steady-state behaviour of the model defined in the last chapter and find quantities of practical interest, such as average queue sizes, central processor utilisation factor, rate of departures, average stay-in-the-system time. The latter can be found in two ways, one of which involves Little's theorem and the other - three embedded Markov chains. We shall follow the second way and prove three lemmas about the equilibrium distribution of the embedded Markov chains. Lemma 1 will enable us to find the Laplace transforms of the interarrival interval and of the residence time steady-state distributions.

3.1 Steady-state distribution and average of $Q_0(t)$.

By long-run, or steady-state behaviour of a stochastic process which depends on parameter $t \geqslant 0$, we mean its limiting behaviour as $t \to \infty$. In particular, the steady-state distribution of the stochastic process $\{Q_0(t) , t \geqslant 0\}$ is defined by the limits

$$p_i = \lim_{t \to \infty} p_i(t) \; ; \; i=0,1,\ldots,N$$

where $p_i(t)$ are its transient distribution functions.

The obvious question one asks in this connection, is whether the limits p_i exsist and if they do, whether they depend on the initial conditions of the process. Using the theory of stochastic processes (see, for example, Parzen [2]), we could say that since $Q_0(t)$ is a finite-state, irreducible Markov process, the limits p_i exist and are independent of the initial state. Then, to find them, we would have to solve the system of steady-state balance equations (obtained from (2.1) by substituting zeros for the left-hand sides and adding the equation $p_0+p_1+\ldots+p_N = 1$). However, we shall arrive at the same conclusion and results by considering the transient distribution $p_i(t)$; $i=0,\ldots,N$, and letting $t\to\infty$.

First, from (2.17) it follows that

$$v_j < 2(q_1m_0m_1)^{\frac{1}{2}} - (q_1m_0 + m_1) \leqslant 0 \; ; \; j=1,2,\ldots,N$$

(because of the arithmetic - geometric mean inequality).

Now, writing again (2.20) in the form

$$p_i(t) = a_{i,N+1} + \sum_{j=1}^{N} a_{i,j}e^{v_jt} \qquad ; \; i=0,1,\ldots,N$$

we see that $p_i(t)\to a_{i,N+1}$ when $t\to\infty$; $i=0,1,\ldots,N$, therefore the steady-state distribution exists. Furthermore, the coefficients $a_{i,N+1}$ are independent of the initial state of the process, as we showed in the last chapter. The distribution p_i ; $i=0,1,\ldots,N$ is thus given by (2.21) or (2.21a) depending on whether or not $q_1m_0 \neq m_1$. We shall introduce the quantity $r = m_1/(q_1m_0)$ and rewrite the expressions as

$$p_i = \begin{cases} r^i \dfrac{1 - r}{1 - r^{N+1}} & ; \; r \neq 1 \\ \dfrac{1}{N+1} & ; \; r = 1 \end{cases} \qquad ; \; i=0,1,\ldots,N \qquad (3.1)$$

We notice that when $N\to\infty$, $p_i\to r^i(1-r)$ provided that $r<1$, otherwise $p_i\to 0$; $i=0,1,\ldots$. That this should be so is obvious, since when $N \to \infty$, $Q_0(t)$ approaches an M/M/1 queuing process with traffic intensity r. It can also be seen that if $r < 1$, the expression (3.1) for p_i coincides with the steady-state conditional probability that the M/M/1 process is in state i, given that it is not in any of the states $N+1, N+2, \ldots$.

The steady-state expected value of Q_0 is given by

$$E(Q_0) = \sum_{i=1}^{N} ip_i = \begin{cases} \frac{r}{1-r} \, \frac{1-(N+1)r^N + Nr^{N+1}}{1-r^{N+1}} & ; \ r \neq 1 \\ \frac{1}{2}N \quad ; \quad r = 1 \end{cases}$$

The steady-state expectation of Q_1 is, of course, equal to $N - E(Q_0)$. We have

$$\lim_{N \to \infty} E(Q_0) = \begin{cases} \frac{r}{1-r} & ; \ r < 1 \\ \infty & ; \ r \geqslant 1 \end{cases} ; \lim_{N \to \infty} E(Q_1) = \begin{cases} \infty & ; \ r < 1 \\ \frac{(1/r)}{1-(1/r)} & ; \ r \geqslant 1 \end{cases}$$

This, again, is what one would expect.

3.2 Processor utilisation, rate of departures and average residence time.

According to the interpretation we gave to our model, $Q_0(t)$ is the number of jobs waiting at and/or being served by the central processor at time t. We called N the degree of multiprogramming. Since p_0 is the steady-state probability that the central processor is idle,

$$U = 1 - p_0 = \begin{cases} r \, \frac{1-r^N}{1-r^{N+1}} & ; \ r \neq 1 \\ \frac{N}{N+1} & ; \ r = 1 \end{cases} \tag{3.2}$$

is the steady-state probability of the central processor being busy, in other words, the steady-state utilisation factor of the central processor (similarly, $1 - p_N$ is the steady-state utilisation factor of the peripheral processor).

In most computing instalations, the CPU (central processing unit) utilisation factor is usually taken as a measure of the system effectiveness. This is mainly because, as we shall see, the throughput of jobs is directly proportional to it. Under the assumptions of our model, U is a monotone increasing function of the degree of multiprogramming, for any fixed value of r. If $r<1$ i.e. if, when both the central and the peripheral processors are working, requests for input/output occur at a higher rate than completions of input/output operations, then $U \rightarrow r$ when $N \rightarrow \infty$. Thus when the central processor is, in the above sense, faster than the peripheral processor, its utilisation factor is limited by the speed of the peripheral processor. If $r \geqslant 1$, then $U \rightarrow 1$ when $N \rightarrow \infty$.

In real-life computing systems the CPU utilisation factor is not always a monotone increasing function of the degree of multiprogramming. Usually, an increase of the degree of multiprogramming is accompanied by an increase of supervisor overhead which tends, after a certain optimum is reached, to reduce the utilisation factor. This was demonstrated by Lewis and Shedler [3] in their model of system overhead. Also, in time-sharing systems for example, a high degree of multiprogramming can lead to highly increased demand for input/output due to paging and that, in turn, can lower the CPU utilisation factor. The latter phenomenon is sometimes called 'thrashing' ; we shall dwell briefly on it in the fifth chapter.

Let us find now the steady state rate of job departures from the system (the throughput). We know that S_0 (the central processor) is busy for a proportion U of the time, i.e. for a fraction U per unit time, in the steady-state. While S_0 is busy, jobs are served by it at a rate of m_0 per unit time. Of these, a proportion q_1 join the S_1-queue and a proportion $q_0=1-q_1$ leave the system. Thus, on the average, q_0m_0U jobs leave the system per unit time. Since arrivals occur exactly at moments of departures, the steady-state rate of arrivals, L, is also given by

$$L = q_0m_0U \tag{3.3}$$

Of course $1/L$ is the average length of the interarrival (and interdeparture) intervals.

Our next task is to find the average residence time of a job (the time between its admission to, and departure from the system) in the steady-state. This quantity can be obtained in two different ways. The simpler one is by using Little's theorem which states that in the steady-state, the number of customers in a queuing system (or sub-system) is, on the average, equal to the product of the rate of arrivals and the average residence time of a customer (J.D.C.Little, [4]) . In our case, the number of jobs in the system is constant, N, so that Little's theorem together with (3.2) and (3.3) gives the following expression for the average residence time W :

$$W = \frac{N}{L} = \begin{cases} \dfrac{N}{q_0m_0r}\,\dfrac{1-r^{N+1}}{1-r^N} & ;\quad r \neq 1 \\ (N+1)/q_0m_0 & ;\quad r = 1 \end{cases} \tag{3.4}$$

(The behaviour of W as a function of N and its connection with job turnaround will be examined in chapter four.)

In order to use Little's theorem we have to show that the conditions under which it was proved, namely that the arrival process is metrically transitive (ergodic) and that the expectations involved are finite, are satisfied in our case. This can be done without a great deal of difficulty. However, we shall arrive at the same result in a different, although more complicated way. In doing so, we shall obtain additional information about the steady-state distribution of queue sizes at selected points of time. Some of this information will be used later.

3.3 Embedded Markov chains.

Until now, we have concentrated our attention on the continuous parameter stochastic process $\{Q_0(t), t \geqslant 0\}$. We shall introduce now three discrete parameter stochastic processes

$$\begin{aligned} M1 &= \{Q_0(t1_k) \;, \; k=1,2,\ldots\} \\ M2 &= \{Q_0(t2_k^+) \;, \; k=1,2,\ldots\} \\ M3 &= \{Q_0(t3_k^+) \;, \; k=1,2,\ldots\} \end{aligned}$$

where $t1_k$ is the moment when the $(N+k)$-th job enters the system (and the $(N+k-1)$-st leaves it) ; $t2_k$ is the moment when, for the k-th time, a job leaves the S_0-queue and joins the S_1-queue ; $t3_k$ is the moment when, for the k-th time, a job leaves the S_1-queue and joins the S_0-queue. (In the second and third instances, the points are of successive departures, not of successive departures by a particular job.)

Note that while $Q_0(t)$ is continuous at points $t1_k$, it is a step function at points $t2_k$ and at points $t3_k$; that is why the limits from the right were used in the definitions of M2 and M3. The possible states of M1 and M3 are $1,2,\dots,N$; the possible states of M2 are $0,1,\dots,N-1$.

Under the assumptions of our model, M1, M2 and M3 are Markov chains. Furthermore, they are finite, irreducible and aperiodic and therefore possess unique equilibrium distributions which are, at the same time, steady-state distributions. Denote by $\underline{p1}=(p1_1,p1_2,\dots,p1_N)$, $\underline{p2}=(p2_0,p2_1,\dots p2_{N-1})$ and $\underline{p3}=(p3_1,p3_2,\dots,p3_N)$ the steady-state distributions of M1, M2 and M3 respectively. We shall prove the following three lemmas :

Lemma 1: The vector $\underline{p1}$ is given by

$$p1_i = \frac{p_i}{1-p_0} = \frac{1-r}{1-r^N}\, r^{i-1} \quad ; \; r\neq 1 \;, \; i=1,2,\dots,N \qquad (3.5)$$

Lemma 2; The vector $\underline{p2}$ is given by

$$p2_i = \frac{p_i}{1-p_N} = \frac{1-r}{1-r^N}\, r^i \quad ; \; r\neq 1 \;, \; i=0,1,\dots,N-1 \qquad (3.6)$$

Lemma 3: The vectors $\underline{p3}$ and $\underline{p1}$ are identical:

$$p3_i = p1_i \quad ; \quad i=1,2,\dots,N \qquad (3.7)$$

When $r=1$ the elements of all three vectors are equal to $\frac{1}{N}$.

Assuming that lemmas 1, 2, 3 are true, we can proceed in the following way:

From lemmas 1 and 3 it follows that, in the steady-state, every time a job joins the S_0-queue, there are on the average N1 jobs there (including itself) , where N1 is given by

$$N1 = \sum_{i=1}^{N} ip1_i = \begin{cases} \frac{1}{1-r} - \frac{Nr^N}{1-r^N} & ; \quad r \neq 1 \\ \frac{1}{2}(N+1) & ; \quad r = 1 \end{cases} \tag{3.8}$$

From lemma 2 it follows that, in the steady-state, every time a job joins the S_1-queue, there are on the average N2 jobs there (including itself) , where N2 is given by

$$N2 = \sum_{i=0}^{N-1} (N-i)p2_i = \begin{cases} \frac{N}{1-r^N} - \frac{r}{1-r} & ; \quad r \neq 1 \\ \frac{1}{2}(N+1) & ; \quad r = 1 \end{cases} \tag{3.9}$$

(The apparent discrepancy $N1 + N2 \neq N$ is due to the fact that the S_0-queue and the S_1-queue are observed at different points of time.)

When we described our model we mentioned that the residence time of a job requiring k services at S_0 , k=1,2,... , consists of k-1 cycles followed by one wait at S_0 . The average number of S_0-services required by a job is $1/q_0$ (from the geometric distribution) ; therefore the average number of cycles it goes through is equal to $(1/q_0)-1 = q_1/q_0$. The average duration of an S_0-service is $1/m_0$ and of an S_1-service - $1/m_1$; hence the average length of a cycle is $N1/m_0 + N2/m_1$ and the average duration of a S_0-wait is $N1/m_0$. Thus we can write the following expression for the average residence time :

$$W = \frac{q_1}{q_0}\left(\frac{N1}{m_0} + \frac{N2}{m_1}\right) + \frac{N1}{m_0} = \frac{1}{q_0 m_0 r}(rN1 + N2) \tag{3.10}$$

Substituting (3.8) and (3.9) into (3.10) we now obtain the same value for W as the one given by (3.4) .

Before going on to prove the lemmas, it is perhaps worth pointing out that they are similar in nature to Khinchine's result (Saaty [5]) , in the sense that they show the steady-state distribution of queue sizes at selected moments of time to be equal to the time-average steady-state distribution (conditioned upon the appropriate queue not being empty) . Khinchine's result cannot be applied directly to this model because the Poisson input assumption is not satisfied here.

We shall now give the proofs of lemmas 2,3 and 1, in that order. All three proofs are empirical and consist of verifying that the probability distribution vectors defined by (3.5), (3.6) and (3.7) satisfy the steady-state balance equations of the Markov chains M1, M2 and M3 respectively. This method of proof can be applied because the Markov chains have unique equilibrium distributions and therefore the balance equations have unique solutions.

Proof of lemma 2 :

Let $V2 = (v2_{i,j})_{i,j=0}^{N-1}$ denote the matrix of transition probabilities of M2 :

$$v2_{i,j} = P[Q_0(t2_{k+1}{}^{+})=j \mid Q_0(t2_k{}^{+})=i] \quad ; \; i,j=0,1,\ldots,N-1$$

where $t2_k$ are the moments when successive jobs join the S_1-queue. We notice first that

$$v2_{0,j} = v2_{1,j} \qquad j=0,1,\ldots N-1$$

due to the fact that if $Q_0(t2_k{}^{+}) = 0$ then, inevitably, a job will join the S_0-queue before $t2_{k+1}$, hence the distribution of $Q_0(t2_{k+1}{}^+)$ will be the same as it would be if $Q_0(t2_k{}^+) = 1$.

Furthermore, for $i=2,3,\dots,N-1$ we have

$$v2_{i,i-2} = v2_{i,i-3} = \dots = v2_{i,0} = 0$$

because, according to the definition of $t2_k$, $Q_0(t)$ cannot decrease by more than 1 between $t2_k{}^+$ and $t2_{k+1}{}^+$. (In the interval $(t2_k,t2_{k+1}]$ there can be many departures from the S_0-queue, provided that they are also departures from the system, in which case $Q_0(t)$ does not decrease.)

We are going to need the fact that, if $Q_0(t2_k{}^+) \neq 0$ then the time between $t2_k$ and $t2_{k+1}$ is distributed exponentially with parameter q_1m_0 . This can be established as follows: the number of S_0-services between $t2_k$ and $t2_{k+1}$ is distributed geometrically, with probability generating function

$$G(z) = \frac{q_1z}{1 - q_0z} \quad ;$$

each S_0-service is distributed exponentially, with Laplace transform

$$f_0{}^*(s) = \frac{m_0}{s + m_0} \quad ;$$

if $Q\ (t2_k{}^+) \neq 0$ then in the interval $(t2_k,t2_{k+1})$ there is no S_0-idle period, therefore the Laplace transform of the distribution of the interval is equal to

$$G(f_0{}^*(s)) = \frac{q_1f_0{}^*(s)}{1 - q_1f_0{}^*(s)} = \frac{q_1m_0}{s + q_1m_0}$$

which is the Laplace transform of an exponential distribution with parameter $q_1 m_0$.

If two activities , A_1 and A_2 , with durations distributed exponentially with parameters e_1 and e_2 respectively, are in progress at a given moment of time, then the probability that A_1 will terminate before A_2 , is equal to $e_1/(e_1+e_2)$. Taking A_1 to be the succession of S_0-services terminating at $t2_{k+1}$ and A_2 to be an S_1-service, we get, for $i=1,2,\ldots,N-1$

$$v2_{i,i-1} = g$$

where g is defined by

$$g = \frac{q_1 m_0}{q_1 m_0 + m_1} = \frac{1}{1+r} \qquad (3.11)$$

For a transition from state i $(i=1,2,\ldots,N-2)$ to state j $(j=i,i+1,\ldots,N-2)$ to occur, there must be exactly $j-i+1$ S_1-service completions in the interval $(t2_k,t2_{k+1})$. Because of the memoryless property of the exponential distribution we can write, for i and j in the above mentioned boundaries

$$v2_{i,j} = g(1-g)^{j-i+1}$$

where g is given by (3.11) .

For a transition from state i $(i=1,2,\ldots,N-1)$ to state $N-1$ to occur, all $N-i$ jobs in the S_1-queue must be served by S_1 before $t2_{k+1}$. Thus

$$v2_{i,N-1} = (1-g)^{N-i}$$

All elements of V2 are now known and the matrix can be written in full :

$$V2 = \begin{bmatrix} g & g(1-g) & g(1-g)^2 & \dots & g(1-g)^{N-2} & (1-g)^{N-1} \\ g & g(1-g) & g(1-g)^2 & \dots & g(1-g)^{N-2} & (1-g)^{N-1} \\ 0 & g & g(1-g) & \dots & g(1-g)^{N-3} & (1-g)^{N-2} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & g(1-g) & (1-g)^2 \\ 0 & 0 & 0 & \dots & g & (1-g) \end{bmatrix} \qquad (3.12)$$

It remains to show that the vector $\underline{p2} = (p2_0,\dots,p2_{N-1})$ defined by (3.6) satisfies

$$\underline{p2}.V2 = \underline{p2}$$

where $\underline{p2}$ is treated as a row-vector and the product is 'row by column'. This is easily done by direct substitution. As an example, let us calculate the scalar product of $\underline{p2}$ and the second column of V2 ; it should be equal to $p2_1$:

$$\frac{1-r}{1-r^N}\left[g(1-g) + g(1-g)r + gr^2\right] =$$

$$\frac{1-r}{1-r^N}\,\frac{r}{(1+r)^2}\left[1 + r + r(1-r)\right] = \frac{1-r}{1-r^N}\,r = p2_1$$

The rest of the scalar products are equally easy to evaluate. (When $r = 1$ the calculations become trivial ; g is then equal to $\frac{1}{2}$ and all columns of V2 sum to 1 .)

<u>Proof of lemma 3 :</u>

This proof proceeds along the same lines as the one of lemma 2 . We denote by $V3 = (v3_{i,j})_{i,j=1}^{N}$ the matrix of transition probabilities for the Markov chain M3 :

$$v3_{i,j} = P\left[Q_0(t3_{k+1}{}^+)=j \middle| Q_0(t3_k{}^+)=i\right] \quad ; \; i,j=1,2,\ldots,N$$

Now we have

$$v3_{N,j} = v3_{N-1,j} \quad ; \quad j=1,2,\ldots,N$$

and, for $i=1,2,\ldots N-2$

$$v3_{i,i+2} = v3_{i,i+3} = \cdots = v3_{i,N} = 0$$

for reasons analogous to those for the similar properties of the matrix V2 . In short, an argument almost identical to the one in the last proof shows V3 to have the form

$$V3 = \begin{bmatrix} g & (1-g) & 0 & \cdots & 0 & 0 \\ g^2 & g(1-g) & (1-g) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ g^{N-1} & g^{N-2}(1-g) & g^{N-3}(1-g) & \cdots & g(1-g) & (1-g) \\ g^{N-1} & g^{N-2}(1-g) & g^{N-3}(1-g) & \cdots & g(1-g) & (1-g) \end{bmatrix} \qquad (3.13)$$

and it can be verified by direct substitution that the vector $\underline{p3} = (p3_1,p3_2,\ldots,p3_N)$ defined by (3.7) satisfies

$$\underline{p3}.V3 = \underline{p3}$$

<u>Proof of lemma 1 :</u>

Again our aim is to find the matrix $V1 = (v1_{i,j})_{i,j=1}^N$ where

$$v1_{i,j} = P\left[Q_0(t1_{k+1})=j \middle| Q_0(t1_k)=i\right] \quad ; \; i,j=1,2,\ldots,N$$

This time, however, the procedure is more complicated due to the fact that both the S_0-queue and the S_1-queue can go through many possible transitions between $t1_k$ and $t1_{k+1}$ (moments of successive arrivals into the system) . To get around this difficulty, we divide the interval $(t1_k, t1_{k+1})$ into S_0-steps, where an S_0-step is defined as the span between two consecutive S_0-services. Thus an S_0-step consists of either one S_0-service (if after the service is completed $Q_0 \neq 0$), or one S_0-service and one S_0-idle period. Let C_n denote the event 'there are n S_0-steps in the interval $(t1_k, t1_{k+1})$' ; n=1,2,... . Obviously C_n is independent of k and of the value of Q_0 at $t1_k$. The distribution of C_n is given by

$$P(C_n) = q_0 q_1^{n-1} \quad ; \ n=1,2,\ldots$$

If, for n=1,2,... , we introduce the matrices $V_n = (v_{n;i,j})_{i,j=1}^{N}$, where

$$v_{n;i,j} = P\left[Q_0(t1_{k+1})=j \,\middle|\, Q_0(t1_k)=i, C_n\right] \quad ; \ i,j=1,2,\ldots,N \qquad (3.14)$$

we can express V1 as an infinite series

$$V1 = \sum_{n=1}^{\infty} V_n . P(C_n) = \sum_{n=1}^{\infty} q_0 q_1^{n-1} V_n \qquad (3.15)$$

Familiar considerations show that, for n = 1

$$V_1 = \begin{bmatrix} h & h(1-h) & \ldots & h(1-h)^{N-2} & (1-h)^{N-1} \\ 0 & h & \ldots & h(1-h)^{N-3} & (1-h)^{N-2} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \ldots & h & (1-h) \\ 0 & 0 & \ldots & 0 & 1 \end{bmatrix} \qquad (3.16)$$

where h is the probability that the S_0-service will complete before the S_1-service, provided that both S_0 and S_1 are working ; h is given by

$$h = \frac{m_0}{m_0 + m_1} = \frac{1}{1 + q_1 r} \qquad (3.17)$$

(V_1 is triangular because it is known that at the end of the S_0-step the departing job is replaced, therefore Q_0 cannot decrease.)

Suppose now that $n \geqslant 2$. Consider the moments

$$t1_k = t_0 < t_1 < \ldots < t_m < \ldots < t_n = t1_{k+1}$$

where t_m is the end of the m-th S_0-step ($m=1,2,\ldots,n$) , and/or the beginning of the m+1-st S_0-step ($m=0,1,\ldots,n-1$). We note first that, for $i,j=1,2,\ldots,N-1$ and $m=1,2,\ldots,n-1$ the transition probabilities

$$P\left[Q_0(t_m{}^+)=j \,\middle|\, Q_0(t_{m-1}{}^+)=i, C_n\right]$$

are independent of m ; this is due to the exponential distribution of S_0- and S_1-service times. However, $Q_0(t_0{}^+)$ can take the values $1,2,\ldots,N$, while $Q_0(t_1{}^+),\ldots,Q_0(t_{n-1}{}^+)$ can only take the values $1,2,\ldots,N-1$. The latter is due to the fact that an S_0-service which terminates at one of the moments t_1, t_2, ..., t_{n-1} cannot result in a departure from the system, which is the only way to have $Q_0 = N$ just after an S_0-service completion. The case $m = n$ is different again, because $Q_0(t_n{}^+)$ can take the values $1,2,\ldots,N$ and also the transition probabilities

$$P\left[Q_0(t_n{}^+)=j \,\middle|\, Q_0(t_{n-1}{}^+)=i, C_n\right] \; ; \; i=1,2,\dots,N-1 \; ; \; j=1,2,\dots,N$$

differ from those mentioned above (it is known that at t_n the job departs from the system and is replaced).

We shall introduce the three matrices $V_a = (v_{a;i,j})$, $V_b = (v_{b;i,j})$ and $V_c = (v_{c;i,j})$ of dimentions $(N{\times}N-1)$, $(N-1{\times}N-1)$ and $(N-1{\times}N)$ respectively, where

$$\begin{aligned}
v_{a;i,j} &= P\left[Q_0(t_1{}^+)=j \,\middle|\, Q_0(t_0{}^+)=i, C_n\right]; \; i=1,\dots,N \; ; \; j=1,\dots,N-1\\
v_{b;i,j} &= P\left[Q_0(t_m{}^+)=j \,\middle|\, Q_0(t_{m-1}{}^+)=i, C_n\right]; \; i,j=1,\dots,N-1; m=2,\dots,n-1\\
v_{c;i,j} &= P\left[Q_0(t_n{}^+)=j \,\middle|\, Q_0(t_{n-1}{}^+)=i, C_n\right]; \; i=1,\dots,N-1; j=1,\dots,N
\end{aligned}$$

It is clear from these definitions and the remarks preceding them, that V_a is the transition probability matrix for the first S_0-step, V_b is the transition probability matrix for each of the next $n-2$ steps and V_c is the transition probability matrix for the n-th step. We can write now

$$V_n = V_a V_b{}^{n-2} V_c \quad ; \quad n=2,3,\dots \tag{3.18}$$

(Power of 0 is assumed to yield the identity matrix.)

The elements of V_a, V_b and V_c can be easily expressed in terms of the quantity defined by (3.17). V_a has the form

$$V_a = \begin{bmatrix}
h+h(1-h) & h(1-h)^2 & \dots & h(1-h)^{N-2} & (1-h)^{N-1}\\
h & h(1-h) & \dots & h(1-h)^{N-3} & (1-h)^{N-2}\\
0 & h & \dots & h(1-h)^{N-4} & (1-h)^{N-3}\\
\vdots & \vdots & & \vdots & \vdots\\
0 & 0 & \dots & h & (1-h)\\
0 & 0 & \dots & 0 & 1
\end{bmatrix}$$

(The two terms of the top left-hand corner element of V_a arise because the probability that $Q_0(t_1^+) = 1$ given that $Q_0(t_0^+) = 1$ and C_n is equal to the probability that either the S_0-service finishes before the S_1-service (in which case the S_0-step will contain an S_0-idle period), or the S_0-service finishes after the current S_1-service but before the next one.)

V_b is equal to V_a without its last row :

$$V_b = \begin{bmatrix} h+h(1-h) & h(1-h)^2 & \dots & h(1-h)^{N-2} & (1-h)^{N-1} \\ h & h(1-h) & \dots & h(1-h)^{N-3} & (1-h)^{N-2} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & h & (1-h) \end{bmatrix}$$

V_c has the form

$$V_c = \begin{bmatrix} h & h(1-h) & \dots & h(1-h)^{N-2} & (1-h)^{N-1} \\ 0 & h & \dots & h(1-h)^{N-3} & (1-h)^{N-2} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & h & (1-h) \end{bmatrix}$$

and it can be seen that V_c is equal to V_1 without its last row.

Equation (3.18) will still hold if V_a , V_b and V_c are enlarged to become square matrices of dimentions $(N\times N)$ by appending

a) a column of zeros to V_a ,

b) an arbitrary row and a column of zeros to V_b ,

c) an arbitrary row to V_c .

With this observation in mind we form the $(N\times N)$ matrix

$$V_A = \begin{bmatrix} \cdots & \cdots & \cdots & 0 \\ \vdots & V_a & \vdots & 0 \\ \vdots & & \vdots & \vdots \\ \cdots & \cdots & \cdots & 0 \end{bmatrix} \qquad (3.19)$$

According to b) , V_A can be used as an enlarged V_b as well as an enlarged V_a . According to c) , V_1 can be used as an enlarged V_c . We can thus rewrite (3.18) as

$$V_n = V_A{}^{n-1} V_1 \quad ; \quad n=1,2,\ldots \qquad (3.20)$$

Substituting (3.20) into (3.15) we obtain

$$V1 = q_0 \left[\sum_{n=1}^{\infty} (q_1 V_A)^{n-1} \right] V_1 \qquad (3.21)$$

It is clear that the infinite series in the right-hand side of (3.21) converges, since all rows of V_A sum to 1 and $q_1 < 1$. (3.21) is therefore equivalent to

$$V1 = q_0 (I - q_1 V_A)^{-1} V_1 \qquad (3.22)$$

where I is the identity matrix of order N and power of (-1) denotes inversion (it is easy to see that $(I - q_1 V_A)$ is non-singular).

Now, to finish the proof, we have only to verify that the vector $\underline{p1} = (p1_1, p1_2, \ldots, p1_N)$ defined by (3.5) satisfies the system of balance equations $\underline{p1}.V1 = \underline{p1}$ or, in view of (3.22) , that it satisfies

$$\underline{p1}.q_0 (I - q_1 V_A)^{-1} V_1 = \underline{p1}$$

which is equivalent to

$$q_0 \underline{p1} = \underline{p1}.V_1{}^{-1} (I - q_1 V_A) \qquad (3.23)$$

After performing the necessary calculations we find that

$$V_1^{-1} = \frac{1}{h}\begin{bmatrix} 1 & -(1-h) & 0 & \dots & 0 & 0 \\ 0 & 1 & -(1-h) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -(1-h) \\ 0 & 0 & 0 & \dots & 0 & h \end{bmatrix} .$$

and that

$$V_1^{-1}(I-q_1V_A) = \frac{1}{h}\begin{bmatrix} 1-q_1h & -(1-h) & 0 & \dots & 0 & 0 \\ -q_1h & 1 & -(1-h) & \dots & 0 & 0 \\ 0 & -q_1h & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -(1-h) \\ 0 & 0 & 0 & \dots & -q_1h & h \end{bmatrix}$$

Now (3.23) can be verified directly. As an example, let us take the scalar product of <u>p1</u> and the second column of $V_1^{-1}(I-q_1V_A)$; the result should be equal to q_0p1_2 :

$$\frac{1-r}{1-r^N}\left[\frac{-(1-h)}{h} + \frac{r}{h} - q_1r^2\right] = \frac{1-r}{1-r^N}\left[-q_1r + r(1+q_1r) - q_1r^2\right]$$

$$= q_0r(1-r)/(1-r^N) = q_0p1_2$$

The above calculations are valid for $N \geqslant 3$. The case $N = 1$ is trivial. When $N = 2$ we have

$$V_A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

and (3.23) is just as easy to verify.

This completes the proofs of lemmas 1 , 2 and 3 and hence of relation (3.10) .

3.4 The Laplace transform of the interarrival interval.

The task of determining the distribution of a random variable is often more difficult than that of determining its expectation. We have not been able to find explicitly the distribution function of either the interarrival or the residence time. It is possible however, to derive expressions for the Laplace transforms of these distributions. We shall do this in detail for the interarrival interval and only outline the derivation for the residence time.

Let us consider again the embedded Markov chain M1 at the moments $t1_k$; $k=1,2,\ldots$ of successive arrivals into the system. In addition to the matrix of transition probabilities V1 , we shall associate with M1 the matrix of Laplace transforms $L1 = (l1_{i,j})_{i,j=1}^{N}$ where $l1_{i,j}$ is the Laplace transform of the distribution of the interval $(t1_k, t1_{k+1})$, given that at $t1_k$ M1 was in state i and at $t1_{k+1}$ it will be in state j . (M1 is thus treated as a semi-Markov process rather than a Markov chain.)

The Laplace transform of the interarrival time distribution given that at $t1_k$ M1 was in state i - denoted by $l1_i$ - is given by

$$l1_i = \sum_{j=1}^{N} v1_{i,j} l1_{i,j} \qquad (3.24)$$

The unconditional steady-state Laplace transform of the interarrival time distribution - denoted by l1 - is obtained from

$$l1 = \sum_{i=1}^{N} p1_i l1_i \qquad (3.25)$$

where $(p1_1, p1_2, \ldots, p1_N) = \underline{p1}'$ is the steady-state distribution of M1 , given by (3.5) .

If A and B are two $(N \times M)$ matrices with elements $a_{i,j}$ and $b_{i,j}$ respectively, then their 'element by element' product C $(c_{i,j} = a_{i,j}b_{i,j}$; $i=1,\ldots,N$; $j=1,\ldots,M)$ will be denoted by $C = (A*B)$. Using this notation and the $(N \times 1)$ matrix

$$E = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

we can combine (3.24) and (3.25) into

$$l1 = \underline{p1}(V1*L1)E \tag{3.26}$$

If a name is to be given to the matrix $(V1*L1)$, it seems appropriate to call it the 'joint transition probability-Laplace transform matrix' of M1 for the interarrival interval. We shall use similar names for matrices of this type in the future.

To find $(V1*L1)$, we shall divide the interarrival interval into S_0-steps as we did in the proof of lemma 1 . Let C_n be the event of there being exactly n S_0-steps in the interarrival interval ; let $V1_n$ be the matrix defined by (3.14) ; let $L1_n$ be the matrix of the Laplace transforms for the interval, conditioned upon C_n as well as upon the states of M1 at $t1_k$ and $t1_{k+1}$. $(V1*L1)$ can now be expressed as

$$(V1*L1) = \sum_{n=1}^{\infty} (V1_n*L1_n)P(C_n) = \sum_{n=1}^{\infty} q_0q_1^{n-1}(V1_n*L1_n) \tag{3.27}$$

There is an obvious analogy between (3.27) and (3.15), which extends into allowing the derivation of an expression for $(V1_n * L1_n)$ similar to (3.20). We associate with V_A (defined by (3.19)) the matrix of Laplace transforms $L_A = (l_{A;i,j})_{i,j=1}^{N}$, where $l_{A;i,j}$ is the Laplace transform of an S_0-step given that $Q_0 = i$ at the beginning and $Q_0 = j$ at the end of the step and the step is not the last one in the interarrival interval. The desired expression for $(V1_n * L1_n)$ is

$$(V1_n * L1_n) = (V_A * L_A)^{n-1} (V1_1 * L1_1) \; ; \; n=1,2,\ldots \qquad (3.28)$$

Here $(V_A * L_A)$ and $(V1_1 * L1_1)$ are used as (enlarged) joint transition probability-Laplace transform matrices of Q_0 for the first $n-1$ S_0-steps and for the n-th S_0-step respectively.

The derivation of (3.28) relies on the fact that the Laplace transform of the convolution of the distribution functions of two independent random variables is equal to the product of their respective Laplace transforms ; also on the exponential distribution of the S_0- and S_1-service times. The argument which leads to (3.28) is very similar to the one used in obtaining (3.20) and we shall omit it.

If both S_0 and S_1 are working, then the time until the nearest event, be it an S_0-service completion or an S_1-service completion, is distributed exponentially with parameter $m_0 + m_1$. We shall denote the Laplace transform of that distribution by $b(s)$; the Laplace transform of an S_0-service time distribution by $a(s)$; the Laplace transform of an S_1-service time distribution by $c(s)$:

$$a(s) = \frac{m_0}{s + m_0} \; ; \; b(s) = \frac{m_0 + m_1}{s + m_0 + m_1} \; ; \; c(s) = \frac{m_1}{s + m_1} \qquad (3.29)$$

It is not difficult to see that $L1_1$ is given by

$$L1_1 = \begin{bmatrix} b(s) & b^2(s) & \dots & b^{N-1}(s) & b^{N-1}(s)a(s) \\ 0 & b(s) & \dots & b^{N-2}(s) & b^{N-2}(s)a(s) \\ 0 & 0 & \dots & b^{N-3}(s) & b^{N-3}(s)a(s) \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & b(s) & b(s)a(s) \\ 0 & 0 & \dots & 0 & a(s) \end{bmatrix} \qquad (3.30)$$

and that L_A is given by

$$L_A = \begin{bmatrix} l_{A;1,1} & b^3(s) & \dots & b^{N-1}(s) & b^{N-1}(s)a(s) & 0 \\ b(s) & b^2(s) & \dots & b^{N-2}(s) & b^{N-2}(s)a(s) & 0 \\ 0 & b(s) & \dots & b^{N-3}(s) & b^{N-3}(s)a(s) & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & b(s) & b(s)a(s) & 0 \\ 0 & 0 & \dots & 0 & a(s) & 0 \end{bmatrix} \qquad (3.31)$$

where

$$l_{A;1,1} = \frac{h.b(s)c(s) + h(1-h)b^2(s)}{h + h(1-h)} \qquad (3.32)$$

(an S_0-step which results in a transition from state 1 to state 1 can contain an S_0-idle period).

All matrices in the right-hand side of (3.28) are now known. Substitution of (3.28) into (3.27) yields

$$(V1*L1) = q_0(I - q_1(V_A*L_A))^{-1}(V1_1*L1_1) \qquad (3.33)$$

in the same way as (3.22) was derived from (3.20) and (3.15) . Substituting (3.33) into (3.26) we obtain

$$l1 = q_0\underline{p1}\left[I - q_1(V_A*L_A)\right]^{-1}(V1_1*L1_1)E \qquad (3.34)$$

thus determining the Laplace transform of the steady-state interarrival time distribution.

(3.34) can be simplified a little by using the fact that all rows of $(V1_1*L1_1)$ sum to $a(s)$ (one would expect this to be so, since the last S_0-step of $(t1_k,t1_{k+1})$ consists of exactly one S_0-service). In matrix notation it means that $(V1_1*L1_1)E = a(s)E$ and (3.34) becomes

$$l1 = a(s)q_0\underline{p1}\left[I - q_1(V_A*L_A)\right]^{-1}E \qquad (3.35)$$

Expressions (3.31) and (3.32) are valid for $N \geqslant 3$. When $N = 1$ $l1$ can be found directly :

$$l1 = \sum_{n=1}^{\infty} q_0 q_1^{\,n-1}\left[a(s)c(s)\right]^{n-1}a(s) = \frac{q_0 a(s)}{1 - q_1 a(s)c(s)}$$

When $N = 2$

$$L_A = \begin{bmatrix} h.b(s)c(s)+(1-h)b(s)a(s) & 0 \\ a(s) & 0 \end{bmatrix}$$

and again, the calculations are not difficult.

In general, since the right-hand side of (3.35) is a rational function of s in which the polynomial in the numerator is of lower degree than the one in the denominator, it appears that the probability density function of the steady-state interarrival time is a linear combination of exponential functions. To prove it, one would have to establish that the roots of the denominator are distinct negative real numbers. This we have not been able to do, except for the simple cases $N = 1$ and $N = 2$.

3.5 The Laplace transform of the residence time.

The residence time of a job consists of geometrically distributed number of cycles (the span between two consecutive joinings of the S_0-queue by the same job) , followed by one wait in the S_0-queue. Let $(V_{S0}*L_{S0})$ be the joint transition probability-Laplace transform matrix of Q_0 for a wait of a job in the S_0-queue and $(V_{S1}*L_{S1})$ be the transition probability-Laplace transform matrix of Q_0 for a wait of a job in the S_1-queue. Then

$$(V_C*L_C) = (V_{S0}*L_{S0})(V_{S1}*L_{S1}) \tag{3.36}$$

is the joint transition probability-Laplace transform matrix of Q_0 for one cycle and

$$(V_R*L_R) = \sum_{n=1}^{\infty} q_0 q_1{}^{n-1}(V_C*L_C)^{n-1}(V_{S0}*L_{S0}\ ) \tag{3.37}$$

is the joint transition probability-Laplace transform matrix of Q_0 for the residence time of a job. (3.37) can also be written as

$$(V_R*L_R) = q_0\left[I - q_1(V_C*L_C)\right]^{-1}(V_{S0}*L_{S0}) \tag{3.38}$$

The steady-state distribution of Q_0 at moments when jobs enter the system is given by the vector $\underline{p1}.$, defined by (3.5) . Therefore the steady-state Laplace transform l_{RES} of the distribution of the residence time is given by

$$l_{RES} = \underline{p1}(V_R * L_R)E \qquad (3.39)$$

Our task has now been reduced to finding the two matrices $(V_{S0}*L_{S0})$ and $(V_{S1}*L_{S1})$.

If, after joining the S_0-queue, a job finds (including itself) i jobs there, then its wait in the S_0-queue consists of i S_0-services. The first $i-1$ of them are equivalent from transition probabilities-Laplace transforms point of view, but the i-th one is different because after its completion the destination of the job is known. Let $(V_{OA}*L_{OA})$ be the joint transition probability-Laplace transform matrix of Q_0 for each of the first $i-1$ S_0-services and $(V_{OB}*L_{OB})$ be the joint transition probability-Laplace transform matrix of Q_0 for the i-th S_0-service. It can be seen then, that $(V_{SO}*L_{SO})$ has the following form :

$$(V_{SO}*L_{SO}) = \begin{bmatrix} \text{1-st row of } & (V_{OB}*L_{OB}) \\ \text{2-nd row of } & (V_{OA}*L_{OA})(V_{OB}*L_{OB}) \\ & \vdots \\ \text{N-th row of } & (V_{OA}*L_{OA})^{N-1}(V_{OB}*L_{OB}) \end{bmatrix} \qquad (3.40)$$

The elements $vl_{i,j}$; $i,j=1,2,\ldots,N$ of $(V_{OA}*L_{OA})$ can be expressed in terms of h , $a(s)$ and $b(s)$:

$$\begin{aligned}
&vl_{1,j} = 0 \quad ; \quad j=1,2,\ldots,N \quad ; \\
&vl_{i,j} = 0 \quad ; \quad j \leqslant i-2 \quad ; \\
&vl_{i,i-1} = q_1 h.b(s) \ ; \ 2 \leqslant i \leqslant N-1 \ ; \ vl_{N,N-1} = q_1 a(s) \ ; \qquad (3.41) \\
&vl_{i,j} = h(1-h)^{j-i} b(s)^{j-i+1}[q_0+q_1(1-h)b(s)] \ ; \ 2 \leqslant i \leqslant j \leqslant N-2 \ ; \\
&vl_{i,N-1} = (1-h)^{N-1-i} b(s)^{N-i}[q_0 h+q_1(1-h)a(s)] \ ; \ 2 \leqslant i \leqslant N-1 \ ; \\
&vl_{i,N} = q_0(1-h)^{N-i} b(s)^{N-i} a(s) \quad ; \quad i=2,3,\ldots,N
\end{aligned}$$

($vl_{1,j}$ are defined as zeros for convenience, since they are irrelevant for the purpose of computing $(V_{SO}*L_{SO})$.)

$(V_{OB}*L_{OB})$ is the matrix

$$\begin{bmatrix} hb(s) & h(1-h)b^2(s) & \dots & h(1-h)^{N-2}b^{N-1}(s) & (1-h)^{N-1}b^{N-1}(s)a(s) \\ 0 & hb(s) & \dots & h(1-h)^{N-3}b^{N-2}(s) & (1-h)^{N-2}b^{N-2}(s)a(s) \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & hb(s) & (1-h)b(s)a(s) \\ 0 & 0 & \dots & 0 & a(s) \end{bmatrix} \tag{3.42}$$

(We note that $(V_{OB}*L_{OB})$ and $(V1_1*L1_1)$ are equal.)

$(V_{S1}*L_{S1})$ can be obtained in a similar way :

If, after joining the S_1-queue, a job leaves Q_0 in state j (therefore finds N-j jobs, including itself, in the S_1-queue) then its wait in the S_1-queue consists of N-j S_1-services. Of these, the last N-j-1 S_1-services are equivalent from transition probabilities-Laplace transforms point of view, but the first is slightly different, because Q_0 can be equal to zero at its beginning.

Let $(V_{1A}*L_{1A})$ and $(V_{1B}*L_{1B})$ be the transition probability-Laplace transform matrices of Q_0 for the first, and for the remaining S_1-services respectively. $(V_{S1}*L_{S1})$ has the following form :

$$(V_{S1}*L_{S1}) = \begin{bmatrix} \text{1-st row of } (V_{1A}*L_{1A})(V_{1B}*L_{1B})^{N-1} \\ \text{2-nd row of } (V_{1A}*L_{1A})(V_{1B}*L_{1B})^{N-2} \\ \vdots \\ \text{N-th row of } (V_{1A}*L_{1A}) \end{bmatrix} \tag{3.43}$$

$(V_{1A}*L_{1A})$ is the matrix

$$\begin{bmatrix} c(s) & 0 & 0 & \cdots & 0 \\ gd(s)c(s) & (1-g)d(s) & 0 & \cdots & 0 \\ (gd(s))^2c(s) & g(1-g)d^2(s) & (1-g)d(s) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ (gd(s))^{N-1}c(s) & g^{N-2}(1-g)d^{N-1}(s) & g^{N-3}(1-g)d^{N-2}(s) & \cdots & (1-g)d(s) \end{bmatrix} \tag{3.44}$$

where g is defined by (3.11) , $c(s)$ is defined by (3.29) and

$$d(s) = \frac{q_1m_0 + m_1}{s + q_1m_0 + m_1} \tag{3.45}$$

is a Laplace transform analogous to $b(s)$, but taking into account the fact that jobs join the S_1-queue at a rate q_1m_0 when S_0 is busy.

$(V_{1B}*L_{1B})$ is the matrix

$$\begin{bmatrix} gd(s)c(s) & (1-g)d(s) & 0 & \cdots & 0 \\ (gd(s))^2c(s) & g(1-g)d^2(s) & (1-g)d(s) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ (gd(s))^{N-1}c(s) & g^{N-2}(1-g)d^{N-1}(s) & g^{N-3}(1-g)d^{N-2}(s) & \cdots & (1-g)d(s) \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{3.46}$$

(The elements of the last row of $(V_{1B}*L_{1B})$ are defined as zeros for convenience, since they are irrelevant for the purpose of computing $(V_{S1}*L_{S1})$.)

Everything needed to find l_{RES} , for $N \geqslant 3$, has now been expressed in terms of known quantities. When $N = 1$, l_{RES} is equal to the Laplace transform of the steady-state interarrival time distribution. One would suspect, in view of relation (3.4) , that in the general case, L_{RES} will be equal to the N-th power of the Laplace transform of the steady-state interarrival time distribution. This, however, is not true for $N = 2$.

Before we leave this subject we would like to point out that all rows of $(V_{OA}*L_{OA})$ and all rows of $(V_{OB}*L_{OB})$ sum to $a(s)$, therefore the first row of $(V_{SO}*L_{SO})$ sums to $a(s)$, the second - to $a^2(s)$, ... , the N-th - to $a^N(s)$ (this is also intuitively obvious) . (3.39) , after substitution of (3.38) , can thus be simplified by replacing $(V_{SO}*L_{SO})E$ by the matrix

$$E_1 = \begin{bmatrix} a(s) \\ a^2(s) \\ \vdots \\ a^N(s) \end{bmatrix}$$

CHAPTER 4.

4.0 Summary.

We shall deal here with the following three topics:

Relation between the model and real-life computing systems. Theoretically obtained values of several quantities will be compared with their observed values.

Relaxation of the exponential services assumption. The model will be analysed in the steady-state, assuming that only one of the two processors has exponentially distributed service times.

Turnaround. Steady-state turnaround will be introduced as an alternative (to the CPU utilisation factor) measure of system efficiency and its dependence on the degree of multiprogramming will be discussed.

4.1 Validity of the assumptions.

The assumptions under which we studied our model (they were given in detail in chapter 2) were:

a. Heavy demand conditions (availability of replacement for every departing job).

b. Single peripheral processor.

c. Non-priority (FIFO) service discipline at both processors.

d. Independent, identically exponentially distributed service times at both processors.

e. Geometrical distribution of the number of CPU services required by a job.

Hardly any multiprogramming computing systems conform to assumptions (b) and (c), which are therefore the most obvious candidates for modification; this will be the subject of subsequent chapters. On the other hand, heavy demand conditions exist in the majority of computing instalations - if not all the time, then at least during peak periods. In this sense, assumption (a) is a reasonable one and will be maintained throughout this dissertation.

Very little is known about the distribution of the number of CPU (and input/output) intervals per job, although the average of that number can usually be obtained from information kept by accounting routines. In the absence of evidence against it, and since it is essential for the analysis, assumption (e) will also be maintained throughout this dissertation.

Let us consider now the nature of the central and the peripheral processors' service times. We shall not discuss the question of whether consecutive service times are independent or not; it is fairly obvious that they are, for a normal job mix; also, none of the analysis would be possible if they were not. The question of whether consecutive service times of a given processor are identically distributed can be answered in the negative, generally. It is well known that some jobs are 'CPU-bound' (long intervals of calculating, short intervals of input/output) and some are 'I/O-bound' (long I/O intervals, short CPU intervals); the average number of cycles also varies. The problem of modeling a system with several different job classes is closely related to that of modeling a system with priority servicing; a model which incorporates both is considered in chapter 6.

It is less clear whether the service times of the central and peripheral processors are distributed exponentially or not. In order to check this, and at the same time to see how well, or badly, the model approximates a real system, some statistics were collected during the normal operation of a university computing installation.

The installation has an IBM 360/67 computer which is used by a large and varied user population (students, academic and administrative staff, some industrial users). At the time of the experiment, the relevant configuration consisted of a central processor with half a million bytes of main storage, a multiple disk unit, a drum unit, a card reader/punch and a line printer. It was run under the MFT II (Multiprogramming with Fixed number of Tasks, version two) Operating System.

MFT II effects multiprogramming by dividing the main storage (that part of it which is not occupied by the supervisor) into partitions of fixed size; when a partition becomes available, the queue of jobs waiting outside (on a disk) is scanned for a job that will fit in it. A job in a higher partition has preemptive priority at the CPU, and head-of-the-line priority for I/O, over a job in a lower partition.

The number of partitions, and thus the level of multiprogramming, was equal to six. The top partition was used by the statistics-collecting program during the experiment, and by a system program called 'HASP' otherwise; the next two partitions were used by the system's reader and writer programs. There was sufficient demand to keep the three user partitions busy for long periods of time.

Our program modified the supervisor a little, so that it

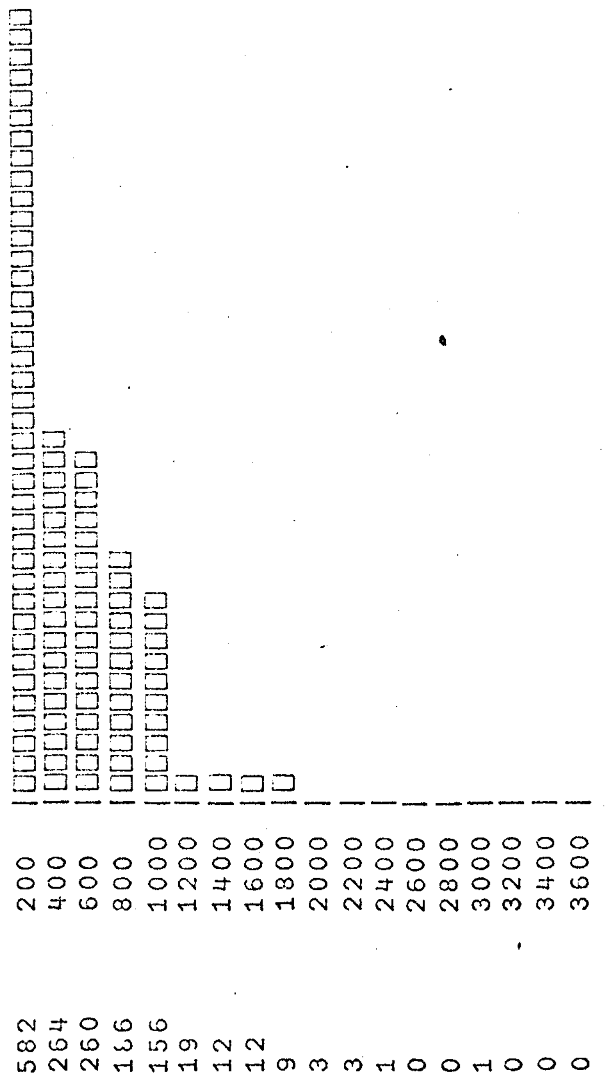
```
398      20    |□□□□□□
2642     40    |□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
1568     60    |□□□□□□□□□□□□□□□□□□□□□□□□
694      80    |□□□□□□□□□□□
313     100    |□□□□□
179     120    |□□□
155     140    |□□
77      160    |□
63      180    |□
43      200    |□
48      220    |□
45      240    |□
42      260    |□
21      280    |
20      300    |
24      320    |
34      340    |□
20      360    |
18      380    |
30      400    |
17      420    |
14      440    |
10      460    |
23      480    |
12      500    |
12      520    |
13      540    |
19      560    |
8       580    |
10      600    |
```

HISTOGRAM STEP = 20 TIMER UNITS

MEAN = 70.5

VARIANCE = 7513.0 ; ST.DEVIATION = 86.6

Fig. 4.1



HISTOGRAM STEP = 200 *TIMER UNITS*

MEAN = 414.4
VARIANCE = 124061.5 ; *ST.D.*=352.2

Fig. 4.2

received control every time a job initiated or terminated an input/output operation and every time a job began or completed a central processor service. The program calculated the lengths of all CPU and I/O services and built histograms of their distribution. The IBM timer unit was chosen as a unit of time (1 timer unit = 26 microseconds).

In the case of the I/O services, no distinction was made between the four peripheral processors which gave them. In the case of the CPU services, no distinction was made between completions due to requests for input/output and completions due to preemptions by higher priority jobs.

The program also recorded the number of jobs in the central processor queue at moments selected at random by the operator.

A typical histogram of CPU services is shown in figure 4.1 . The numbers in the top row are the upper end points of the histogram steps; those in the bottom row are the observed frequencies (e.g. 398 CPU services had lengths between 0 and 20 timer units, 2642 services had lengths between 20 and 40 timer units, etc.). Figure 4.2 shows a similar histogram of input/output services. Both histograms were obtained while running a mixed stream of short jobs (not more than 5 minutes residence time), of the sort that are usually run during the greater part of the day.

Although the coefficients of variation of the two samples are not far from unity, a χ^2 test comparing the observed distributions with exponential distributions gave negative results in both cases.

So, the real computing system did not conform to the assumptions of the model, except in the general sense of being multiprogrammed, giving CPU and I/O services in a cyclic fashion and working under heavy demand conditions.

However, the model can still be used to approximate reality with a reasonably high degree of accuracy, as we shall illustrate by an example.

4.2 Application of the theory.

We shall use the steady-state formulae of chapter 3 to obtain numerical values for the CPU utilisation factor, the average residence time of a job and the average length of the CPU queue. The values of the parameters m_0 , m_1 and q_1 will be estimated using data from the real system.

We take the value of the average CPU service time from figure 4.1 : $\frac{1}{m_0} = 70.5$ timer units. To obtain an estimate for m_1 , we divide the average length of an I/O operation (from figure 4.2) by four, since there are four input/output units : $\frac{1}{m_1} = \frac{414.4}{4} = 103.6$ timer units.

The accounting routine of our installation records, among other things, the number of input/output requests per job. Using this information, the value of q_1 was estimated as $q_1 = 0.9994$.

The traffic intensity r has the value

$$r = \frac{m_1}{q_1 m_0} = \frac{70.5}{0.9994 \times 103.6} = 0.68$$

According to (3.2) , for $N = 6$, the CPU utilisation factor should be equal to

$$U = \frac{0.68 - (0.68)^7}{1 - (0.68)^7} = 0.65$$

Independent measurements (see E.D.Barraclough [6]) showed that the actual value of this quantity, during periods of heavy demand, was $U = 0.62$.

Formula (3.4) gives the average residence time of a job as $W = \frac{N}{L}$, where the rate of departures L is given by (3.3) - $L = q_0 m_0 U$. In our case, the jobs in three of the six partitions are systems programs which never depart. Assuming that the CPU busy time is distributed equally between the partitions, the real rate of departures is half of that quantity: $L = \frac{1}{2} q_0 m_0 U$. In order to obtain the result in seconds, instead of timer units, we multiply the right-hand side by 26×10^{-6} :

$$W = \frac{6\times2\times70.5\times26}{0.0006\times0.65\times1000000} = 56.4 \text{ seconds}$$

In the real system, the average residence time of a short job (jobs of this type were used in estimating m_0 and m_1) was $W = 62.1$ secs. This average was provided by the accounting routine and was taken over all short jobs run in a period of six months.

We mentioned that the statistics-collecting program recorded the number of jobs in the CPU queue at random moments of time. After 20 such recordings, the arithmetic mean of the numbers in the queue was $Q = 2.1$. It should be pointed out that in every instance, the program itself was one of the jobs in the queue; we are dealing with a sequence of moments when the statistics-collecting program joins the CPU queue (since it has top priority, it goes straight into service instead of at the end of the queue). Therefore, the corresponding theoretical quantity is $N1$, given by (3.8) :

$$N1 = \frac{1}{1 - 0.68} - \frac{6\times(0.68)^6}{1 - (0.68)^6} = 2.47$$

These results show an agreement between theory and practice which seems close enough to justify the use of the model in approximating real systems.

4.3 Generalisation of the model.

We shall study a model identical to the one described in chapter 2 , except for assumption (d) which now has the form:

d'. Consecutive S_0-service times are independent, identically distributed random variables with distribution function $F_0(x) = 1 - \exp(-m_0x)$; consecutive S_1-service times are independent, identically distributed, positive valued random variables with a general distribution function $F(x)$ and a finite expectation.

We shall obtain the steady-state central processor utilisation factor, and thus the rate of departures from the system and the average residence time of a job, by the method of the embedded Markov chain and semi-Markov process. This method was used by Lewis and Shedler [3] in their study of supervisor overhead.

Let us consider the discrete parameter stochastic process $M = \left\{Q_0(t_k{}^+);\ k=1,2,\dots\right\}$, where t_1 ,t_2, ... are the moments of successive departures from the S_1-queue ; the possible states of M are $1,2,\dots,N$. Since the intervals between successive arrivals at the S_1-queue are distributed exponentially (with parameter q_1m_0) when S_0 is not idle (see chapter 3, proof of lemma 2) , M is a Markov chain. Denote by

$V = (v_{i,j})_{i,j=1}^{N}$ the transition probability matrix of M. The elements of V are defined as

$$v_{i,j} = P\left[Q_0(t_{k+1}{}^{+})=j \,\middle|\, Q_0(t_k{}^{+})=i\right] \quad ; \qquad (4.1)$$

to find them we need the probabilities of events of the type 'exactly i jobs arrive at the S_1-queue during one S_1-service' (the intervals (t_k,t_{k+1}) consist of either an S_1-service or an S_1-idle period followed by an S_1-service).

Suppose that there are at least i $(i=1,2,\ldots)$ jobs at the S_0-queue at time t and let $t+T_i$ be the moment of the i-th, since t, arrival at the S_1-queue. Because of the exponential distribution of the interarrival intervals, the random variable T_i has Erlangian distribution, with probability density function

$$f_i(x) = \frac{q_1m_0(q_1m_0)^{i-1}e^{-q_1m_0x}}{(i-1)!} \quad ; \; i=1,2,\ldots \qquad (4.2)$$

We shall need the quantities

$$b_i = \int_0^{\infty}\left[\int_0^{x} f_i(t)dt\right]dF(x) \quad ; \quad i=1,2,\ldots \qquad (4.3)$$

b_i is the probability that at least i jobs arrive at the S_1-queue during one S_1-service, given that there were at least i jobs at the S_0-queue at the beginning of the S_1-service. If there were exactly i jobs at the S_0-queue at the beginning of the S_1-service, then b_i is the probability that exactly i jobs arrive at the S_1-queue during the S_1-service.

The probability that exactly i jobs arrive at the S_1-queue during one S_1-service, given that there were at least $i+1$ jobs in the S_0-queue at the beginning of the service, is equal to

$$a_i = b_i - b_{i+1} \quad ; \; i=1,2,\ldots \qquad (4.4)$$

The probability that no jobs arrive at the S_1-queue during one S_1-service, given that there was at least one job at the S_0-queue at the beginning of the service, is equal to

$$a_0 = \int_0^\infty \left[\int_0^x dF(t) \right] f_1(x)dx = 1 - b_1 \qquad (4.5)$$

We can now write the transition probability matrix V in full:

$$V = \begin{bmatrix} b_1 & a_0 & 0 & \ldots & 0 & 0 \\ b_2 & a_1 & a_0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ b_{N-1} & a_{N-2} & a_{N-3} & \ldots & a_1 & a_0 \\ b_{N-1} & a_{N-2} & a_{N-3} & \ldots & a_1 & a_0 \end{bmatrix} \qquad (4.6)$$

It can be seen from (4.3) and (4.5) that, except in the trivial case $F(x) = \begin{cases} 0 \; ; & x<0 \\ 1 \; ; & x \geqslant 0 \end{cases}$, which can be ignored, we have $b_i > 0$ $(i=1,2,\ldots)$ and $a_0 > 0$. This means that for all sufficiently large values of n, all elements of V^n are strictly positive and therefore the Markov chain M is irreducible and aperiodic. Since it is also finite, M has a unique steady-state distribution $\underline{p} = (p_1, p_2, \ldots, p_N)$. The vector $\underline{p}$ can be found by solving the system of linear equations

$$\underline{p}.V = \underline{p} \quad ; \quad p_1+p_2+\ldots+p_N = 1 \qquad (4.7)$$

(one of the first N equations is redundant).

4.4 Expressions for U , L and W.

Note that the probability p_i $(i=1,2,\ldots,N)$ cannot be interpreted as the proportion of time during which there are i jobs in the S_0-queue; only as the proportion of arrivals from the S_1-queue which find i jobs in the S_0-queue. In order to obtain quantities like the S_0 utilisation factor, for instance, we must consider the times involved in the transitions of the Markov chain M , i.e. regard M as a semi-Markov process.

Denote by $F_{i,j}(x)$ $(i,j=1,2,\ldots,N)$ the distribution function of the interval (t_k,t_{k+1}) , given that M was in state i at t_k^+ and will be in state j at t_{k+1}^+ . Then

$$F_i(x) = \sum_{j=1}^{N} v_{i,j}F_{i,j}(x) \quad ,$$

where $v_{i,j}$ are given by (4.6) , is the distribution function of the interval (t_k,t_{k+1}) given that M was in state i at t_k^+ . To find $F_i(x)$ we do not have to determine $F_{i,j}(x)$ because if $i<N$, then (t_k,t_{k+1}) consists of exactly one S_1-service and if $i=N$, then (t_k,t_{k+1}) consists of a (geometrically distributed) number of S_0-services followed by one S_1-service. Hence

$$F_i(x) = F(x) \text{ ; } i=1,2,\ldots,N-1 \text{ ; } F_N(x) = G(x)*F(x) \qquad (4.8)$$

where $G(x) = 1 - e^{-q_1m_0x}$ and $*$ denotes convolution.

If $\frac{1}{m_i}$ is the expected length of the interval (t_k, t_{k+1}) given that M was in state i at t_k^+ $(i=1,2,\ldots,N)$, and $\frac{1}{m}$ is the expected length of an S_1-service, it follows from (4.8) that

$$\frac{1}{m_i} = \frac{1}{m} \quad ; \quad i=1,2,\ldots,N-1 \quad ; \quad \frac{1}{m_N} = \frac{1}{q_1 m_0} + \frac{1}{m} \qquad (4.9)$$

We can find now the average length $t_{i,i}$ $(i=1,2,\ldots,N)$ of the interval between two consecutive moments when M enters state i (this interval is called 'first passage time of M from state i to state i'). The following expression for $t_{i,i}$ in terms of the averages $\frac{1}{m_i}$ and the steady-state distribution $\underline{p} = (p_1, p_2, \ldots, p_N)$ is a basic result in the theory of semi-Markov processes (see, for instance, Barlow and Proschan [7], p. 133) :

$$t_{i,i} = \frac{1}{p_i} \sum_{j=1}^{N} \frac{p_j}{m_j} \quad ; \quad i=1,2,\ldots,N \qquad (4.10)$$

We are interested in $t_{1,1}$ in particular ; (4.9) and (4.10) yield

$$t_{1,1} = \frac{1}{p_1}\left[\frac{1-p_N}{m} + p_N\left(\frac{1}{q_1 m_0} + \frac{1}{m}\right)\right] = \frac{1}{p_1}\left(\frac{1}{m} + \frac{p_N}{q_1 m_0}\right) \qquad (4.11)$$

Note that the first passage time of the process M from state 1 to state 1 is the time between the beginnings of two consecutive S_0-busy periods; it consists of exactly one S_0-busy period and one S_0-idle period. Denoting by b the average length of an S_0-busy period, we can express the steady-state probability that S_0 is busy (the S_0 utilisation factor) as the ratio

$$U = \frac{b}{t_{1,1}} \tag{4.12}$$

Busy periods do not depend on the order in which customers are served. We can imagine therefore, that jobs entering the system from outside go straight into service, instead of at the end of the S_0-queue. These jobs can be regarded as extentions of the ones they replace and, to that extent, ignored. We can think that jobs arrive at the S_0-queue from the S_1-queue only, their 'extended' service times being distributed exponentially with parameter $q_1 m_0$. Thus an S_0-busy period consists of the extended services of all jobs which arrive at the S_0-queue from the S_1-queue during a first passage time of M from state 1 to state 1.

We shall call the moments of arrival at the S_0-queue 't_k-moments' ; a t_k-moment such that $Q_0(t_k^+)=1$ will be called a '1-moment' . Now, the steady-state proportion of 1-moments among the t_k-moments is equal to p_1. Therefore, the steady-state average number of t_k-moments between two consecutive 1-moments is equal to $\frac{1}{p_1}$. Hence

$$b = (q_1 m_0 p_1)^{-1} \tag{4.13}$$

Substituting (4.11) and (4.13) into (4.12) we obtain

$$U = \left(\frac{q_1 m_0}{m} + p_N\right)^{-1} = \left(\frac{1}{r} + p_N\right)^{-1} \tag{4.14}$$

again denoting the traffic intensity $\frac{m}{q_1 m_0}$ by r.

Knowing U, we can find, in the same way as in chapter 3, the steady-state rate of departures from the system $(L = q_0 m_0 U)$, and the steady-state average residence time of a job

$(W = \frac{N}{q_0 m_0 U})$.

Accepting as obvious the fact that if $r \geqslant 1$ then $U \to 1$ as $N \to \infty$, and if $r < 1$ then $p_N \to 0$ as $N \to \infty$, we derive from (4.14) the following

Corollary : Irrespective of the distribution function $F(x)$,

$$\lim_{N \to \infty} p_N = 1 - \frac{1}{r} \quad ; \quad r \geqslant 1$$

$$\lim_{N \to \infty} U = r \quad ; \quad r < 1$$

(this corollary could be used, if better ways were not available, to prove that in M/G/1 and G/M/1 queuing systems with traffic intensity $r < 1$, the probability that the server is idle is equal to $1 - r$).

Remark. The method described here can be used to analyse a model in which the S_1-service times are distributed exponentially and the S_0-service times have a general distribution. One should then consider the Markov chain embedded at moments of arrival at the S_1-queue, the corresponding semi-Markov process and its first passage times from state 0 to state 0 .

4.5 Special cases.

a). When the S_1-service times are distributed exponentially, i.e. when $F(x) = 1 - e^{-mx}$, expression (4.14) should give the same value for U as (3.2) . Taking the integrals in the right-hand side of (4.3) we find that, in this case, the matrix V is equivalent to the matrix $V3$ given by (3.13) and therefore the solution of (4.7) is the vector $\underline{p3}$ given by (3.7) .

Substitution of $p3_N$ into (4.14) gives

$$U = (\frac{1}{r} + \frac{1-r}{1-r^N} r^{N-1})^{-1} = \frac{r(1-r^N)}{1-r^{N+1}}$$

which is the same as (3.2) .

b). The extreme case when the length of the S_1-service times is constant, is also of some interest and is not difficult to solve. Now we have

$$F(x) = \begin{cases} 0 & ; \quad x < 1/m \\ 1 & ; \quad x \geqslant 1/m \end{cases}$$

and (4.3) reduces to

$$b_i = \int_0^{1/m} f_i(t)dt = 1 - e^{-1/r} \sum_{k=0}^{i-1} (k!r^k)^{-1} \quad ; \ i=1,2,\ldots$$

which, together with (4.4) and (4.5) yields

$$a_i = \frac{e^{-1/r}}{i!r^i} \quad ; \quad i=0,1,\ldots$$

The system of linear equations (4.7), written in the form

$$\begin{aligned} a_0p_{N-1} &= (1-a_0)p_N \\ a_0p_{N-2} &= (1-a_1)p_{N-1} - a_1p_N \\ a_0p_{N-3} &= (1-a_1)p_{N-2} - a_2p_{N-1} - a_2p_N \\ &- - - - - - - - - - - - - - - - - - \qquad (4.15) \\ a_0p_1 &= (1-a_1)p_2 - a_2p_3 - \ldots - a_{N-2}p_{N-1} - a_{N-2}p_N \\ p_1 + p_2 &+ \ldots + p_N = 1 \end{aligned}$$

can be easily solved by elimination.

Performing the above calculations for $N = 6$, $r = 0.68$ (see the example in 4.2) we obtain

$$\underline{p} = (0.57,\ 0.249,\ 0.109,\ 0.047,\ 0.019,\ 0.006)$$

(4.14) now gives $U = 0.677$. (We had $U = 0.65$ in the case of exponentially distributed S_1-service times and $U = 0.62$ in the real system. This result seems to indicate that an assumption of constant S_1-service times is further removed from reality than that of exponentially distributed S_1-service times.)

The average number of jobs found in the S_0-queue by an arrival from the S_1-queue is equal to

$$\sum_{i=1}^{6} i.p_i = 1.71 \quad ;$$

this number was equal to $N1 = 2.47$ in the case of exponential $F(x)$ (see 4.2) . It appears, somewhat surprisingly, that an increase in the coefficient of variation of the $F(x)$ distribution leads to an increase in the average S_0-queue size (as observed by arrivals from the S_1-queue) and, at the same time, to a decrease in the S_0-utilisation factor. This phenomenon will be, perhaps, better illustrated by the following special case, where one can also give an intuitive explanation of it.

c). Going to the other extreme, we shall choose for the S_1-service times a distribution function with practically infinite coefficient of variation. Define

$$F_{@}(x) = \begin{cases} 0 & ; \quad x < 0 \\ 1 - @ & ; \quad 0 \leqslant x < \frac{1}{m@} \\ 1 & ; \quad \frac{1}{m@} \leqslant x \end{cases}$$

where $@$ is a small positive number. This distribution has a mean of $\frac{1}{m}$ and a coefficient of variance equal to $\frac{1-@}{@}$. Substituting $F_@(x)$ into (4.3) we find, for $i=1,2,\ldots$

$$b_i = @\left[1 - e^{-1/(@r)} \sum_{k=0}^{i-1} (k!@^k r^k)^{-1}\right]$$

By choosing a sufficiently small @ we can make a_0 arbitrary close to 1 and a_i $(i=1,2,\ldots)$ arbitrary close to 0 . The solution of the system of equations (4.15) then becomes, approximately,

$$p_i \sim 0 \quad ; \quad i=1,2,\ldots,N-1 \quad ; \quad p_N \sim 1$$

In this case, the S_0-utilisation factor approaches its lowest possible value, $\frac{r}{1+r}$ (for a given r) , while the average S_0-queue size (as observed by arrivals from the S_1-queue) approaches its highest possible value, N .

It is easy to see why this is so. The above distribution of S_1-service times implies that the great majority of the S_1-services are of zero length, but the rare exeptions are very, very long. This means that almost all arrivals at the S_0-queue find N jobs in it, but when a long S_1-service occurs, then S_0 is idle for a large period of time. Another way of obtaining the value of the S_0-utilisation factor would be as follows: Since there are, on the average, $\frac{1}{@}$ short S_1-services between two long ones, the average length of an S_0-busy period is, approximately, $\frac{1}{q_1 m_0 @}$. The average length of an S_0-idle period is, approximately,equal to that of a long S_1-service, i.e. $\frac{1}{m@}$. Therefore, the steady-state probability that S_0 is busy is equal to

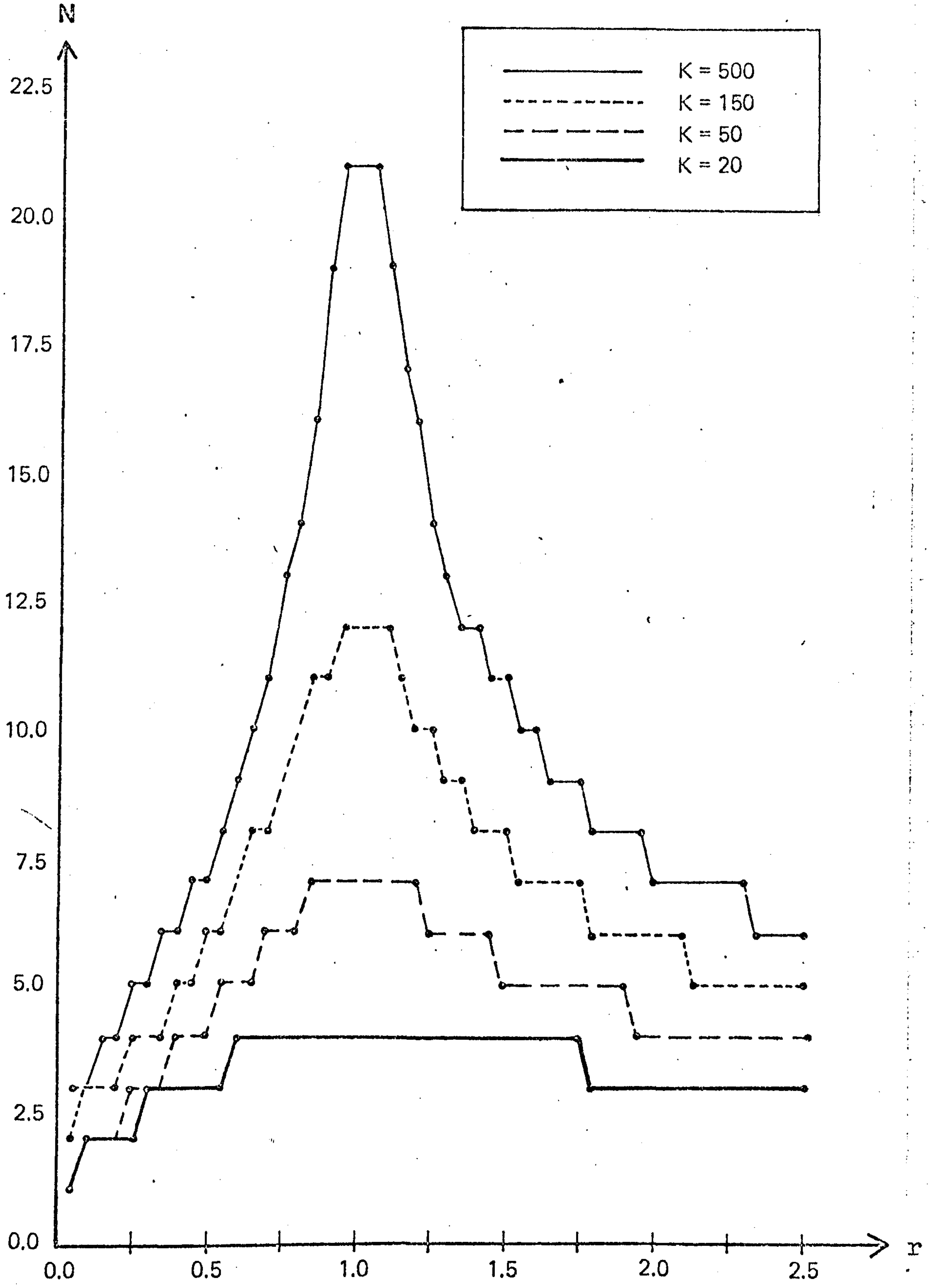
$$(q_1 m_0 @)^{-1} / \left[(q_1 m_0 @)^{-1} + (m@)^{-1}\right] = \frac{r}{1+r}$$

4.5 Job turnaround.

Apart from assuming heavy demand, we have hitherto ignored the flow of jobs outside the system. In real computing systems, there is usually a queue of jobs awaiting execution - sometimes in the form of card decks, sometimes on disk, drum or tape; we shall call it the 'outside queue'. Many programs, especially when in the stage of development, rejoin the outside queue soon after being executed. In these conditions, the turnaround time T (the time between joining the outside queue and leaving the system) becomes important.

Computing managers usually measure the efficiency of their systems by the number of jobs executed per unit time, i.e. the rate of departures from the system. This, we saw, is proportional to the central processor utilisation factor and is an increasing function of the degree of multiprogramming. Users, on the other hand, measure the efficiency of the system by the time it takes to get their programs executed, i.e. the turnaround time.

The problem of determining the average turnaround time given the parameters of the system and of the input stream of demands is a difficult one and we shall not deal with it here. Instead, we shall assume that the steady-state average size of the outside queue is known; denote it by K (in a real-life situation this quantity can be obtained empirically). We shall also assume that the value of K is relatively large and that the probability of the outside queue vanishing is negligible. Since, in the steady-state, the rate at which jobs join the outside queue is equal to the rate of departure from



Optimal valuc of N as a function of r.

for K = 20, 50, 150, 500

Figure 4.3

the system, the last assumption means that jobs join the outside queue at a rate approximately equal to $L = q_0 m_0 U$.

Little's theorem now leads to the following approximation for the steady-state expectation of T :

$$E(T) \sim \frac{K + N}{L} \tag{4.16}$$

In general, K is a function of N , as well as of the other parameters. If, for example, $K + N = \text{const}$ then $E(T)$ is inversely proportional to the CPU utilisation factor and the objects of the computer manager and the user coincide. In some cases K is independend of N (e.g. when the size of the outside queue is artificially controlled or when the demand rate rises in proportion with L). We can find then an optimal value for N which minimises the expected turnaround time.

(4.16) can be written as

$$E(T) \sim (K + N)\frac{1 - r^{N+1}}{q_0 m_0 (r - r^{N+1})} \tag{4.17}$$

Figure 4.3 shows the values of N minimising (4.17) plotted against r , for four different values of K .

CHAPTER 5.

5.0 Summary.

We shall define and analyse in the steady-state a model of a multiprogramming system with one central and several peripheral processors - an extension of the model defined in chapter 2. Two cases will be considered:

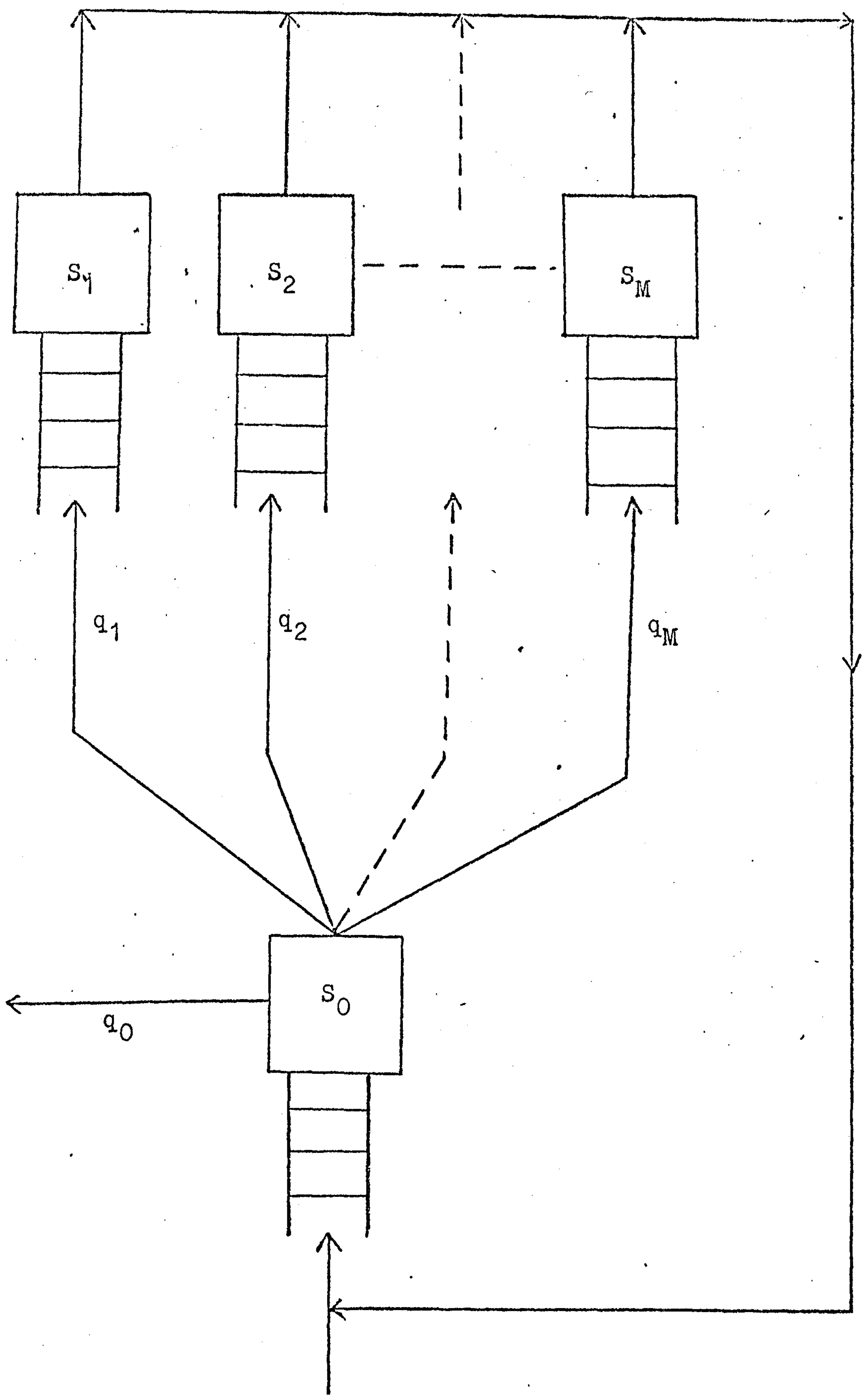
1. The service times of all processors are distributed exponentially. In this case, explicit formulae will be obtained for the joint distribution of queue sizes, the central processor utilisation factor, the rate of departures from the system and the average residence time of a job.

2. The central processor service times have general distribution; all others are distributed exponentially. Now an embedded Markov chain and a semi-Markov process can be used to find the above quantities. We shall give an outline of the derivation.

5.1 The model.

The system that we are going to study is pictured in figure 5.1 . It consists of $M+1$ servers $S_0, S_1, \ldots, S_M$ $(M \geqslant 1)$ working in parallel; S_0 represents the central processor and $S_1, S_2, \ldots, S_M$ represent the M peripheral processors. There are exactly N $(N \geqslant 1)$ customers (jobs) in the system at any one time; this means that when a job departs from the system it is replaced instantaneously by a new job from 'outside'.

Each server serves a separate queue of jobs; denote the



The flow of jobs in the system

Figure 5.1

size of the S_i-queue at time t by $Q_i(t)$ $(i=0,1,\dots,M)$ (as before, jobs receiving service are included in their respective queues). Since $Q_0(t)+Q_1(t)+\dots+Q_M(t) = N$, we can take the vector $\underline{Q(t)} = [Q_1(t),Q_2(t),\dots,Q_M(t)]$ for instance, to represent the state of the system at time t .

Thus a state of the system is an integer valued vector with M elements $\underline{n} = (n_1,n_2,\dots,n_M)$. The set s of possible states is defined as

$$s = \left\{ \underline{n} \;\middle|\; n_i \geqslant 0 \ ; \ i=1,2,\dots,M \ ; \ \sum_{i=1}^{M} n_i \leqslant N \right\}$$

It can be shown by induction on M that there are

$$C_M^{N+M} = \frac{(N+M)!}{M!N!}$$

state-vectors in s .

All queues are served in order of arrival and independently of each other. When a job enters the system, it joins at the end of the S_0-queue. After receiving an S_0-service, jobs either leave the system or join at the end of the S_i-queue $(i=1,2,\dots,M)$; the former occurs with probability q_0 $(0<q_0<1)$ and the latter - with probability q_i $(i=1,2,\dots,M \ ; \ 0<q_i<1 \ ; \ q_1+q_2+\dots+q_M = 1-q_0)$. Jobs departing from the S_i-queue $(i=1,2,\dots,M)$ join at the end of the S_0-queue.

The above cyclic queuing discipline implies, in terms of job structure, that

a) jobs consist of alternative central processor and input/output intervals, there being M types of input/output intervals ;

b) if K_i is the number of I/O intervals of type i $(i=1,2,\ldots,M)$ required by a job and $g(k_1,k_2,\ldots,k_M) = P(K_1=k_1,K_2=k_2,\ldots,K_M=k_M)$ then

$$g(k_1,k_2,\ldots,k_M) = q_0\frac{(k_1+k_2+\ldots+k_M)!}{k_1!k_2!\ldots k_M!}q_1^{k_1}q_2^{k_2}\ldots q_M^{k_M} \; ; \qquad k_i=0,1,\ldots \; ; \; i=1,2,\ldots M \tag{5.1}$$

i.e. the joint distribution of the number of input/output requests of type $1,2,\ldots,M$ per job is 'M-dimensional geometric'.

(If a die with $M+1$ facets numbered $0,1,\ldots,M$ is thrown repeatedly and if, at the n-th throw, the probability of the i-th facet coming up is q_i $(i=0,1,\ldots,M)$ then (5.1) gives the probability that facets $1,2,\ldots,M$ will come up $k_1,k_2,\ldots,k_M$ times respectively, before the first coming up of facet 0.)

The total number of input/output requests, and thus the number of cycles a job goes through, is distributed geometrically with parameter q_0 :

$$P(K_1+K_2+\ldots+K_M = k) = q_0(1-q_0)^k \; ; \; k=0,1,\ldots$$

This can be seen either directly or by summing (5.1) over all $k_1,k_2,\ldots,k_M$ such that $k_1+k_2+\ldots+k_M = k$

Consecutive service times of the processor S_i are assumed to be independent, identically distributed random variables with distribution function $F_i(x)$; $i=0,1,\ldots,M$.

5.2 Exponentially distributed service times.

We shall assume first that the service times of all processors are distributed exponentially : $F_i(x) = 1 - e^{-m_i x}$; $i=0,1,\ldots,M$. Now the stochastic process $\{\underline{Q(t)}\ ,\ t \geqslant 0\}$ is a finite, irreducible Markov chain and therefore its steady-state distribution exists and is independent of the initial distribution $\underline{Q(0)}$.

(The exponential services assumption makes our model a special case of the model studied by Gordon and Newell [8] : they considered a queuing system in which a fixed number of customers are served in stages, with several servers in each stage and constant probabilities of going to stage j after leaving stage i . The analysis is easier in our case. The system of equations for the joint steady-state distribution of queue sizes is simpler than the similar system in Gordon and Newell's paper; this will allow us to solve it directly and to obtain the solution in closed form.)

We are interested in the steady-state distribution

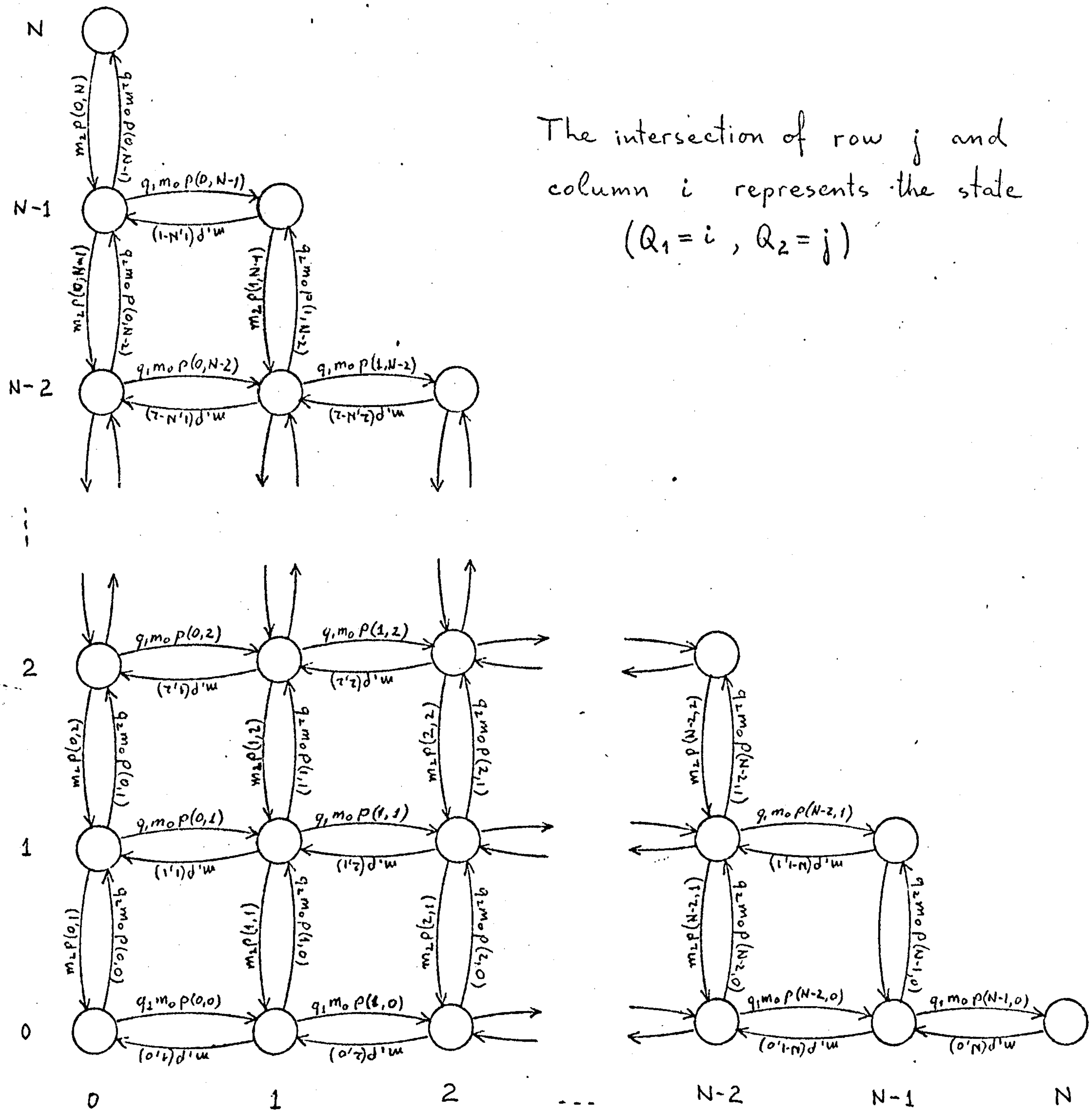
$$p(\underline{n}) = \lim_{t \to \infty} P\left[\underline{Q(t)} = \underline{n}\right] \quad ; \quad \underline{n} \in s \qquad (5.2)$$

where $\underline{n} = (n_1, n_2, \ldots, n_M)$, $\underline{Q(t)} = \left[Q_1(t), Q_2(t), \ldots, Q_M(t)\right]$ and s is the set of possible states for the process. The probabilities (5.2) satisfy the following linear system of balance equations :

$$\sum_{i=1}^{M}\left[q_i m_0 \delta(n_0) + \delta(n_i) m_i\right] p(n_1, n_2, \ldots, n_M) =$$

$$= \sum_{i=1}^{M} \delta(n_0) m_i p(n_1, \ldots, n_i + 1, \ldots, n_M) + \qquad (5.3)$$

$$+ \sum_{i=1}^{M} q_i m_0 \delta(n_i) p(n_1, \ldots, n_i - 1, \ldots, n_M) \quad ; \quad \underline{n} \in s$$



The intersection of row j and column i represents the state $(Q_1 = i\ ,\ Q_2 = j)$

Steady-state balance diagram for M=2

Figure 5.2

where $n_0 = N - (n_1+n_2+\ldots+n_M)$ and $\delta(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n > 0 \end{cases}$.

Equations (5.3) can be obtained either by letting $t \to \infty$ in the time-dependent differential equations of the process $\underline{Q}(t)$ or by considering its steady-state balance diagram (shown, for $M = 2$, in figure 5.2). Balance diagrams have the property that, if a region of the diagram is enclosed by an imaginary line (or surface), then the sum of the transition intensities going out of the region is equal to the sum of the transition intensities coming into the region. Enclosing the points in the diagram one at a time we obtain equations (5.3).

The system (5.3) together with the normalising equation

$$\sum_{\underline{n} \in S} p(\underline{n}) = 1 \qquad (5.4)$$

determines the unknown probabilities uniquely. One can guess the form of the solution by applying an intuitive argument:

Let $r_1, r_2, \ldots, r_M$ be the traffic intensities

$$r_i = \frac{q_i m_0}{m_i} \quad ; \quad i=1,2,\ldots,M$$

Imagine M independent M/M/1 queuing systems with traffic intensities r_1, r_2, ..., r_M respectively. The joint steady-state distribution of the number of customers in them is given by

$$P(Q_1=n_1, Q_2=n_2, \ldots, Q_M=n_M) = \prod_{i=1}^{M} (1 - r_i) r_i^{n_i} \qquad (5.5)$$

provided that $r_i < 1$; $i=1,2,\ldots,M$.

Now, if we add the restriction that the total number of customers in the M systems must not exceed N, then the balance equations which describe the resultant 'restricted' system will be precisely equations (5.3) (this remark is an extension of the similar one at the end of chapter 2). Since the adding of the restriction does not affect the equations in (5.3) for which $n_0 > 0$, we know that at least those equations are satisfied by the probabilities (5.5). Direct substitution shows that the others are satisfied too.

Thus the general solution of (5.3) is given (since it is a homogeneous system) by

$$p(\underline{n}) = A\prod_{i=1}^{M} r_i^{\,n_i} \quad ; \quad \underline{n} \in s \tag{5.6}$$

where A is an arbitrary constant. Substitution of (5.6) into (5.4) yields

$$A = p(0,0,\dots,0) = \left[\sum_{\underline{n}\in s}\left(\prod_{i=1}^{M} r_i^{\,n_i}\right)\right]^{-1} \tag{5.7}$$

(Note that (5.6) and (5.7) are valid for all positive values of $r_1, r_2, \dots, r_M$; the steady-state of the restricted system always exists. When the steady-state of the unrestricted system exists, i.e. when $r_i < 1$; $i=1,2,\dots,M$, then (5.6) and (5.7) imply that

$$P_{res}(\underline{Q} = \underline{n}) = P_{unres}(\underline{Q} = \underline{n} \,|\, \underline{n} \in s) \quad ; \quad \underline{n} \in s$$

where the subscripts 'res' and 'unres' mean 'in the restricted system' and 'in the unrestricted system'.)

It remains to evaluate the expression in the right-hand side of (5.7). We shall write (5.7) in the form

$$A = G_N(\underline{r}_M)^{-1} \tag{5.8}$$

where $G_N(\underline{r}_M)$ is the function

$$G_N(r_1,r_2,\dots,r_M) = \sum_{i=0}^{N} r_M^{\,i} \sum_{j=0}^{N-i} r_{M-1}^{\,j} \dots \sum_{k=0}^{N-(i+j+\dots)} r_1^{\,k} \tag{5.9}$$

The number of elements in the vector is indicated explicitly by a subscript in order to enable it to be a variable. The notation

$$_k\underline{x}_n = (x_1,\dots,x_{k-1},x_{k+1},\dots,x_n) \;;\; k=2,\dots,n-1 \;;$$

$$_1\underline{x}_n = (x_2,\dots,x_n) \;;\; {}_n\underline{x}_n = \underline{x}_{n-1} = (x_1,\dots,x_{n-1})$$

will also be used.

Let $T(\underline{x}_n)$, $Y(\underline{x}_n)$ and $Z(\underline{x}_n)$ be the functions

$$\begin{gathered} Y(\underline{x}_n) = \prod_{i=2}^{n} \prod_{j=1}^{i-1} (x_i - x_j) \;;\; n \geqslant 2 \;;\; Y(\underline{x}_1) = 1 \\ T(\underline{x}_n) = \prod_{i=1}^{n} (1 - x_i) \;;\; Z(\underline{x}_n) = T(\underline{x}_n) Y(\underline{x}_n) \end{gathered} \tag{5.10}$$

We shall prove the following formula :

$$G_N(\underline{r}_M) = \frac{1}{T(\underline{r}_M)} \left[1 - \frac{1}{Y(\underline{r}_M)} \sum_{i=1}^{M} (-1)^{M+i} r_i^{\,N+M} Z({}_i\underline{r}_M) \right] \tag{5.11}$$

(The right-hand side of (5.11) is defined only when $r_i \neq 1$; $r_i \neq r_j$; $i,j=1,2,\dots,M$, $i \neq j$. Cases when this is not so should be treated individually, either by applying L'Hospital's rule or by direct summation. For example, if $r_i=1$; $i=1,2,\dots,M$ then $G_N(\underline{r}_M) = C_M^{N+M} = \frac{(N+M)!}{N!M!}$.)

Proof of (5.11) :

Expression (5.9) which defines $G_N(\underline{r}_M)$ can be rewritten as

$$G_N(\underline{r}_M) = \sum_{i=0}^{N} r_M{}^i G_{N-i}(\underline{r}_{M-1}) \qquad (5.12)$$

thus providing a recursive relationship very suitable for induction on M .

When $M = 1$ we have

$$G_N(\underline{r}_1) = \sum_{i=0}^{N} r_1{}^i = \frac{1}{1 - r_1}(1 - r_1{}^{N+1})$$

which agrees with (5.11) because $Y(\underline{r}_1) = 1$ by definition.

Suppose that (5.11) is true for $M = n$. Substituting it into (5.12) we obtain, for $M = n+1$,

$$G_N(\underline{r}_{n+1}) = \sum_{i=0}^{N} r_{n+1}{}^i \left\{\frac{1}{T(\underline{r}_n)}\left[1 - \frac{1}{Y(\underline{r}_n)}\sum_{j=1}^{n}(-1)^{n+j} r_j{}^{N-i+n} Z({}_j\underline{r}_n)\right]\right\}$$

$$= \frac{1}{T(\underline{r}_n)}\left[\frac{1-r_{n+1}{}^{N+1}}{1-r_{n+1}} - \frac{1}{Y(\underline{r}_n)}\sum_{j=1}^{n}(-1)^{n+j} r_j{}^{N+n} Z({}_j\underline{r}_n)\frac{1-(r_{n+1}/r_j)^{N+1}}{1-(r_{n+1}/r_j)}\right]$$

$$= \frac{1}{T(\underline{r}_{n+1})}\left[1-r_{n+1}{}^{N+1} - \frac{1-r_{n+1}}{Y(\underline{r}_n)}\sum_{j=1}^{n}(-1)^{n+1+j} r_j{}^{N+n+1} Z({}_j\underline{r}_n)\frac{1-(r_{n+1}/r_j)^{N+1}}{r_{n+1} - r_j}\right]$$

(common denominator)

$$= \frac{1}{T(\underline{r}_{n+1})}\left\{1-r_{n+1}{}^{N+1} - \frac{1}{Y(\underline{r}_{n+1})}\sum_{j=1}^{n}(-1)^{n+1+j} r_j{}^{N+n+1} Z({}_j\underline{r}_{n+1})\left[1-\left(\frac{r_{n+1}}{r_j}\right)^{N+1}\right]\right\}$$

$$= \frac{1}{T(\underline{r}_{n+1})}\left\{1 - \frac{1}{Y(\underline{r}_{n+1})}\left[r_{n+1}{}^{N+1} R(\underline{r}_{n+1}) + \sum_{j=1}^{n}(-1)^{n+1+j} r_j{}^{N+n+1} Z({}_j\underline{r}_{n+1})\right]\right\}$$

where

$$R(\underline{r}_{n+1}) = Y(\underline{r}_{n+1}) + \sum_{j=1}^{n}(-1)^{n+j}r_j{}^{n}Z({}_j\underline{r}_{n+1}) \qquad (5.13)$$

To complete the proof we have to show that

$$R(\underline{r}_{n+1}) = r_{n+1}{}^{n}Z({}_{n+1}\underline{r}_{n+1})$$

Note first, that if $r_j = 1$; $j=1,2,\ldots,n$, then

$$Y(\underline{r}_{n+1}) = (-1)^{n+1-j}Z({}_j\underline{r}_{n+1}) \; ; \; Z({}_k\underline{r}_{n+1}) = 0 \; ; \; k\neq j \qquad (5.14)$$

Also, if $r_i = r_j$; $i<j$; $i=1,2,\ldots,n-1$; $j=2,3,\ldots,n$, then $Y(\underline{r}_{n+1}) = 0$ and

$$Z({}_j\underline{r}_{n+1}) = (-1)^{j-i-1}Z({}_i\underline{r}_{n+1}) \; ; \; Z({}_k\underline{r}_{n+1}) = 0 \; ; \; k\neq i,j \qquad (5.15)$$

(5.14) and (5.15) are direct consequences of the definitions (5.10) .

It follows from (5.14) and (5.15) that $R(\underline{r}_{n+1}) = 0$ for $r_j = 1$ $(j=1,2,\ldots,n)$ and for $r_i = r_j$ $(i=1,2,\ldots,n-1$; $j=2,3,\ldots,n$; $i<j)$. This means that $R(\underline{r}_{n+1})$ is divisible by $Z({}_{n+1}\underline{r}_{n+1})$, i.e.

$$R(\underline{r}_{n+1}) = Z({}_{n+1}\underline{r}_{n+1})R_1$$

Since $R(\underline{r}_{n+1})$ is a polynomial of degree n in all its arguments and since $r_1,r_2,\ldots,r_n$ appear in $Z({}_{n+1}\underline{r}_{n+1})$ in power n , R_1 is a polynomial (of degree not greater than n) only in r_{n+1} . Substitution into (5.13) shows that $R_1 = r_{n+1}{}^{n}$ at the $n+1$ points $1,r_1,r_2,\ldots,r_n$, hence R_1 is identically equal to $r_{n+1}{}^{n}$. Q.E.D.

5.3 Quantities of interest.

The steady-state central processor utilisation factor is given by

$$U = P(Q_0 > 0) = \sum_{n_1+n_2+\ldots+n_M \leq N-1} p(\underline{n}) \quad .$$

which, according to (5.6) , (5.8) and (5.9) can be written as

$$U = \frac{G_{N-1}(\underline{r}_M)}{G_N(\underline{r}_M)} \tag{5.16}$$

In order to examine the behaviour of U with the increase of N we shall use (5.11) and rewrite (5.16) as

$$U = \frac{Y(\underline{r}_M) - \sum_{j=1}^{M}(-1)^{M+j} r_j^{\,N-1+M} Z({}_j\underline{r}_M)}{Y(\underline{r}_M) - \sum_{j=1}^{M}(-1)^{M+j} r_j^{\,N+M} Z({}_j\underline{r}_M)} \tag{5.17}$$

Let $r_k = \max(r_1, r_2, \ldots, r_M)$. It will be seen that if $r_k < 1$ then $U \rightarrow 1$ when $N \rightarrow \infty$. This case is perhaps not very interesting in the context of computing systems because it means that, for all i , the rate of input/output requests for S_i is lower than S_i's rate of service, i.e. that no peripheral processor can be a bottleneck.

Suppose then, that $r_k \geqslant 1$. Divide the numerator and the denominator in the right-hand side of (5.17) by r_k^{N+M-1} and let $N \rightarrow \infty$. Now $U \rightarrow 1/r_k$. This agrees with the intuitively obvious fact that the efficiency of a computing system is limited by the peripheral processor which is slowest in relation to the rate of I/O requests for it.

When the central processor is not idle, jobs depart from the system at a rate $q_0 m_0$. In the steady-state, the central processor is busy for a proportion U of the time, therefore the unconditional steady-state rate of departures from the system is given by

$$L = q_0 m_0 U \tag{5.18}$$

To find the average residence time of a job we use Little's theorem. Since jobs enter the system at exactly the moments when others leave it, (5.18) gives also the steady-state rate of arrivals into the system. The number of jobs in the system is equal to N at all times. Provided that its assumptions are satisfied, Little's theorem yields

$$W = \frac{N}{L} = \frac{N}{q_0 m_0 U} \tag{5.19}$$

for the steady-state average residence time of a job, W . The assumptions we have to verify are a) the arrival process is metrically transitive (ergodic) and b) the residence time has a finite expectation. a) follows from theorem 1.2 on page 460 in Doob [9] because the intervals between successive arrivals into the system are independent and, in the steady-state, identically distributed random variables. b) follows from the fact that the residence time of a job consists of, on the average, $(1 - q_0)/q_0$ full cycles followed by one wait at S_0 and the cycles have finitely bounded expectations — they do not exceed $N/m_0 + N/\min(m_1,m_2,\ldots,m_M)$.

It would be interesting to substitute parameters from a real-life computing system into our formulae and to compare the observed performance of the system with that predicted by the model.

When we tested the one-peripheral-processor model on a computing system with four peripheral processors (see 4.2) we did not distinguish between the four different types of I/O requests in estimating the values of m_1 and q_1, i.e. we assumed that the four different peripherals were equivalent and combined them into one. If we now take

$$\frac{1}{m_0} = 70.5 \quad ; \quad \frac{1}{m_1} = \frac{1}{m_2} = \frac{1}{m_3} = \frac{1}{m_4} = 414.4 \quad \text{and}$$

$q_1 = q_2 = q_3 = q_4 = 0.9994/4 = 0.2498$ (the data from 4.2), i.e $r_1 = r_2 = r_3 = r_4 = 1.47$, then (5.16) yields $U = 0.43$ which is a worse approximation of the observed value of U (0.62) than that given by the simpler model.

The explanation of this result probably lies in the fact that this model is more sensitive to errors in the estimates of the parameters than the previous one. Unfortunately, we have no reliable information from which to obtain more accurate estimates and thus cannot put the model to a more rigorous test.

5.4 General CPU service times.

Suppose now, that the central processor service times have a general distribution $F_0(x)$ with finite mean $1/m_0$, while the service times of all peripheral processors are distributed exponentially, $F_i(x) = 1 - \exp(-m_i x)$; $i=1,2,\ldots,M$.

The method of analysis which was used in 4.3 and 4.4 is applicable, with slight modifications, to the present model. Since the stochastic process $\underline{Q(t)}$ no longer possesses the Markov 'memoryless' property, we shall consider an embedded Markov chain.

Let $t_1, t_2, \ldots$ be the moments when successive jobs leave the S_0-queue to join one of the other queues. When they do not begin with an S_0-idle period, the intervals (t_k, t_{k+1}) consist of several (possibly none) S_0-services resulting in departures from the system, followed by one S_0-service resulting in a request for input/output. The distribution function of (t_k, t_{k+1}) is then given by

$$\tilde{F}_0(x) = \sum_{i=1}^{\infty} q_0^{i-1}(1-q_0)F_0^{(i)}(x) \qquad (5.20)$$

where $F_0^{(i)}(x)$ denotes i-fold convolution of $F_0(x)$. The expectation of $\tilde{F}_0(x)$ is equal to $\frac{1}{(1-q_0)m_0}$.

The stochastic process $\left\{\underline{Q(t_k^+)}\ ,\ k=1,2,\ldots\right\}$ is a finite, irreducible and aperiodic Markov chain which has a unique steady-state distribution. The set of possible states of this Markov chain is equal to s minus the state $(0,0,\ldots,0)$. We shall denote that set again by s ; it now has $C_M^{N+M} - 1$ elements.

The steady-state distribution of $\left\{\underline{Q(t_k^+)}\ ,\ k=1,2,\ldots\right\}$ will be denoted by $\tilde{p}(\underline{n})$; $\underline{n} \in s$, to distinguish it from the time-average steady-state distribution of $\left\{\underline{Q(t)}\ ,\ t \geq 0\right\}$. To find $\tilde{p}(\underline{n})$; $\underline{n} \in s$, we need the transition probabilities

$$v(\underline{n}', \underline{n}'') = P\left[\underline{Q(t_{k+1}^+)} = \underline{n}'' \middle| \underline{Q(t_k^+)} = \underline{n}'\right] \; ; \qquad (5.21)$$

for all $\underline{n}', \underline{n}'' \in s$. (It is more convenient to denote the states of the Markov chain by vectors. They could also be numbered from 1 to $C_M^{N+M} - 1$; then the probabilities (5.21) would form a $(C_M^{N+M} - 1)\times(C_M^{N+M} -1)$ matrix.)

The derivation of $v(\underline{n}',\underline{n}'')$ will be brief because it is similar to that of $v_{i,j}$ in 4.3 . Denote

$$f_{i,j}(x) = \frac{m_i(m_i x)^{j-1} e^{-m_i x}}{(j-1)!} \quad ; \; i=1,2,\ldots,M \; ; \; j=1,2,\ldots$$

then

$$b_{i,j} = \int_0^\infty \left[\int_0^x f_{i,j}(t)dt\right] dF_0(x) \; ; \; i=1,2,..,M \; ; \; j=1,2,\ldots$$

is the probability that at least j jobs arrive at the S_0-queue from the S_i-queue during the interval (t_k,t_{k+1}) , provided that at t_k^+ there were at least j jobs in the S_i-queue and S_0 was not idle. If there were exactly j jobs in the S_i-queue and S_0 was not idle at t_k^+ , then $b_{i,j}$ is the probability that exactly j jobs join the S_0-queue before t_{k+1}^+ .

$$a_{i,j} = b_{i,j} - b_{i,j+1} \; ; \; i=1,2,\ldots,M \; ; \; j=1,2,\ldots$$

is the probability that exactly j jobs arrive at the S_0-queue from the S_i-queue during (t_k,t_{k+1}) given that at t_k^+ there were at least $j+1$ jobs in the S_i-queue and S_0 was not idle.

$$a_{i,0} = 1 - b_{i,1} \; ; \; i=1,2,\ldots,M$$

is the probability that no jobs arrive at the S_0-queue from the S_i-queue during (t_k,t_{k+1}) given that at t_k^+ there was at least one job at the S_i-queue and S_0 was not idle.

The probabilities (5.21) can now be expressed in terms of $a_{i,j}$ and $b_{i,j}$. We shall divide the set of possible states s into the two disjoint subsets

$$s_0 = \left\{ \underline{n} \middle| n_1+n_2+\ldots+n_M = N \right\} \; ; \; s_1 = \left\{ \underline{n} \middle| n_1+n_2+\ldots+n_M < N \right\}$$

and deal separately with each :

a) $\underline{n}' = (n_1', n_2', n_3', \ldots, n_M') \in s_1$.

The transition probabilities have non-zero values for the following $\underline{n}''$ vectors:

1. If $n_i'' = n_i' + 1$ for some i $(i=1,2,\ldots,M)$ and $n_k'' = n_k'$ for all $k \neq i$, then

$v(\underline{n}',\underline{n}'')$ = P(next I/O request is for S_i)×P(no I/O operations are completed before then) =

$$= \frac{q_i}{1-q_0} \prod_{k=1}^{M} \left[\delta(n_k') a_{k,0} + 1 - \delta(n_k') \right] \quad ; \quad \delta(n) = \begin{cases} 1 & \text{if } n > 0 \\ 0 & \text{if } n = 0 \end{cases}$$

(The probability of no service completions at S_k is either $a_{k,0}$ or 1 , depending on whether $n_k' \neq 0$ or $n_k' = 0$.)

2. If $n_i'' = n_i' + 1$ for some i $(i=1,2,\ldots,M)$ and $n_k'' = n_k' - j_k$ $(j_k=0,1,\ldots,n_k')$; $k=1,..,i-1,i+1,..,M$, then

$v(\underline{n}',\underline{n}'')$ = P(next I/O request is for S_i)×P(no service completions at S_i)×P(j_k service completions at S_k , for $k=1,..,i-1,i+1,..,M$) =

$$= \frac{q_i}{1-q_0} \left[\delta(n_i') a_{i,0} + 1 - \delta(n_i') \right] \times$$

$$\prod_{\substack{k=1 \\ k \neq i}}^{M} \left\{ \delta(n_k' - j_k) a_{k,j_k} + \delta(n_k') \left[1 - \delta(n_k' - j_k) \right] b_{k,j_k} + 1 - \delta(n_k') \right\}$$

(The probability of j_k service completions at S_k is either a_{k,j_k} or b_{k,j_k} or 1, depending on whether $n_k > j_k$, $n_k = j_k \neq 0$ or $n_k = j_k = 0$.)

3. If $n_i'' = n_i' - j_i$ $(j_i=0,1,\dots,n_i')$ for $i=1,2,\dots,M$, then

$v(\underline{n}',\underline{n}'')$ = P(next I/O request is for one of those S_i, where there are at least j_i+1 jobs)$\times$P(j_i+1 service completions at that S_i)$\times$P(j_k service completions at S_k, for $k=1,..,i-1,i+1,..,M$) =

$$= \sum_{i=1}^{M} \frac{q_i \delta(n_i'-j_i)}{1-q_0} \left\{ \delta(n_i'-j_i-1) a_{i,j_i+1} + \left[1-\delta(n_i-j_i-1)\right] b_{i,j_i+1} \right\} \times$$

$$\times \prod_{\substack{k=1\\k\neq i}}^{M} \left\{ \delta(n_k'-j_k) a_{k,j_k} + \delta(n_k')\left[1-\delta(n_k'-j_k)\right] b_{k,j_k} + 1-\delta(n_k') \right\}$$

For all other $\underline{n}''$, $v(\underline{n}',\underline{n}'') = 0$.

b) $\underline{n}' = (n_1',n_2',\dots,n_M') \in s_0$.

Now, since S_0 was idle at t_k^+, an arrival from one of the other queues must occur before t_{k+1}^+. The probability of that arrival being from the S_i-queue is equal to

$$\frac{\delta(n_i')m_i}{\delta(n_1')m_1+\delta(n_2')m_2+\dots+\delta(n_M')m_M}$$

Therefore, for all $\underline{n}'' \in s$,

$$v(\underline{n}',\underline{n}'') = \sum_{i=1}^{M} \delta(n_i')m_i v\left[(n_1',..,n_i'-1,..,n_M'),\underline{n}''\right] / \sum_{j=1}^{M} \delta(n_j')m_j$$

where $(n_1',..,n_i'-1,..,n_M') \in s_1$.

Knowing the transition probabilities $v(\underline{n}',\underline{n}'')$ we can find the steady-state distribution $\tilde{p}(\underline{n})$; $\underline{n} \in s$ from the system of balance equations

$$\tilde{p}(\underline{n}) = \sum_{\underline{n}' \in s} \tilde{p}(\underline{n}')v(\underline{n}',\underline{n}) \quad ; \underline{n} \in s \; ; \quad \sum_{\underline{n} \in s} \tilde{p}(\underline{n}) = 1 \qquad (5.22)$$

5.5 Expressions for U , L and W .

(The derivation which follows is very similar to that in 4.4 and some of the explanations will be omitted).

Let $F_{\underline{n}}(x)$ be the distribution function of the interval (t_k, t_{k+1}) , given that at its beginning $\underline{Q} = \underline{n}$ $(\underline{n} \in s)$. If $\underline{n} \in s_1$, i.e. if S_0 was not idle at t_k^+ , we have

$$F_{\underline{n}}(x) = \tilde{F}_0(x) \qquad (5.23)$$

with $\tilde{F}_0(x)$ given by (5.20) . If $\underline{n} \in s_0$, then

$$F_{\underline{n}}(x) = G_{\underline{n}}(x) * F_0(x) \qquad (5.24)$$

where $G_{\underline{n}}(x)$ is the distribution function of an S_0-idle period at the beginning of which $\underline{Q} = \underline{n}$ and $*$ denotes convolution. $G_{\underline{n}}(x)$ is given by

$$G_{\underline{n}}(x) = 1 - \exp\left\{-\left[\sum_{i=1}^{M} \delta(n_i)m_i\right]x\right\} \qquad (5.25)$$

Denote by $1/m_{\underline{n}}$ the average length of (t_k, t_{k+1}) , given that $\underline{Q(t_k^+)} = \underline{n}$. It follows from (5.23) , (5.24) and (5.25) , that

$$\frac{1}{m_{\underline{n}}} = \frac{1}{(1-q_0)m_0} \;;\; \underline{n} \in s_1 \;;\; \frac{1}{m_{\underline{n}}} = \left[\sum_{i=1}^{M} \delta(n_i)m_i\right]^{-1} + \frac{1}{(1-q_0)m_0} \;;$$
$$\underline{n} \in s_0 \qquad (5.26)$$

We can find now the average first passage times $t_{\underline{n},\underline{n}}$ from state $\underline{n}$ to state $\underline{n}$, for all $\underline{n} \in s$:

$$t_{\underline{n},\underline{n}} = \frac{1}{\tilde{p}(\underline{n})}\left\{\sum_{\underline{n}'\in s}\left[\tilde{p}(\underline{n}')/m_{\underline{n}'}\right]\right\} \quad ; \ \underline{n}\in s \qquad (5.27)$$

(expression (5.27) is a translation of (4.10) in terms of the present semi-Markov process).

Substitution of (5.26) into (5.27) yields

$$t_{\underline{n},\underline{n}} = \frac{1}{\tilde{p}(\underline{n})}\left\{\frac{1}{(1-q_0)m_0} + \sum_{\underline{n}'\in s_0}\left[\tilde{p}(\underline{n}')/\sum_{i=1}^{M}\delta(n_i')m_i\right]\right\} ;\underline{n}\in s \quad (5.28)$$

Let us say that an S_0-idle period is of 'type $\underline{n}$ ', for $\underline{n}\in s_0$, if at its beginning the Markov chain was in state $\underline{n}$. Thus there are as many types of S_0-idle periods as there are vectors in s_0 and each S_0-idle period belongs to one of these types. The average length of an S_0-idle period of type $\underline{n}$ is equal to

$$1/\sum_{i=1}^{M}\delta(n_i)m_i \ .$$

Since each first passage time from state $\underline{n}$ to state $\underline{n}$, for $\underline{n}\in s_0$, contains exactly one S_0-idle period of type $\underline{n}$ (it begins with it) , the steady-state proportion of time that S_0 is idle 'of type $\underline{n}$ ' is equal to

$$I_{\underline{n}} = \left[1/\sum_{i=1}^{M}\delta(n_i)m_i\right]/t_{\underline{n},\underline{n}} \quad ; \quad \underline{n}\in s_0 \qquad (5.29)$$

The steady-state proportion of time that S_0 is idle is equal to

$$I = \sum_{\underline{n}\in s_0} I_{\underline{n}} \qquad (5.30)$$

and the steady-state proportion of time that S_0 is busy -- the central processor utilisation factor -- is equal to

$$U = 1 - I$$

which, after substitution of (5.30) , (5.29) and (5.28) becomes

$$U = \left\{1 + (1-q_0)m_0 \sum_{\underline{n}\in S_0} \left[\tilde{p}(\underline{n}) / \sum_{i=1}^{M} \delta(n_i)m_i\right]\right\}^{-1} \qquad (5.31)$$

(There is a similarity in form between (5.31) and (4.14) ; also, they both give the same value for U when M = 1 and $F_0(x)$ and $F_1(x)$ are exponential distributions.)

To find the steady-state rate of departures from the system (L) and the steady-state average residence time of a job (W) , we substitute (5.31) into (5.18) and (5.19) respectively (these two expressions are independent of the service times distributions).

5.6 A remark on time-sharing systems.

When a large number of jobs are multiprogrammed with time-sharing, their total memory requirements usually exceed the main storage capacity of the computer. This means that either entire jobs, or parts of them, have to be moved frequently in and out of main storage by the system. One or more peripheral processors are reserved for such 'system' input/output operations and the traffic intensities at these processors depend on the number and size of jobs competing for main storage.

The model described in this chapter could possibly be used for studying time-sharing systems by assuming that N varies and that one or more of the traffic intensities r_1, r_2, ..., r_M depend on N . Turning our attention to formula (5.17) we can see, for instance, that if some r_i increases significantly with N , there will be a drop in the central processor utilisation factor for large values of N . This phenomenon is sometimes called 'thrashing' (excessive level of multiprogramming).

CHAPTER 6.

6.0 Summary.

In this chapter we shall study a priority multiprogramming system with one central and one peripheral processor. A cyclic queuing model of the system will be defined and analysed in the steady-state.

Two versions of the model will be considered:

In the first version, both the central processor queue and the input/output queue are served according to the 'preemptive resume' priority discipline. The service times of all but the lowest priority job are assumed to be distributed exponentially.

In the second version, the input/output queue is served according to the 'head-of-the-line' priority discipline; in the central processor queue the priorities remain preemptive. Exponentially distributed central processor service times and general input/output service times are assumed for all jobs.

Procedures for finding the steady-state average residence-in-the-system time of a job and the steady-state central processor utilisation factor will be derived. In both versions of the model, these procedures will be based on determining the steady-state average cycle time for a job of given priority. The results obtained in some special cases will be used to compare the performance of the two types of systems (interruptable and non-interruptable input/output operations) , and to draw attention to the problem of efficient allocation of priorities to job classes with markedly different central processor and input/output requirements.

6.1 The model.

Consider a cyclic queuing system consisting of two servers, S_0 and S_1 (they represent the central and the peripheral processors) , in tandem. The system serves a constant number N ($N \geqslant 1$) of customers (jobs) , each of whom is assigned a distinct priority. Jobs can leave the system: whenever one does so, it is replaced instantaneously by a new job of the same priority from 'outside' . In other words, the supply of jobs of all priorities is inexhaustible (heavy demand conditions).

Because of the one-to-one correspondence between jobs in the system and priorities, both can be indexed by the integers 1,2,...,N (one can thus talk about 'job i' meaning 'the job which has priority i' ; i=1,2,...,N). We shall number the priorities in reverse order, i.e. priority 1 will be the highest, priority 2 the second highest, etc.

The structure of the model is pictured in figure 6.1 . Jobs require alternative S_0- and S_1-services (alternative central processor and input/output intervals). After receiving an S_0-service, job i joins the S_1-queue with probability q_i ($0 < q_i \leqslant 1$; $i=1,2,\dots,N$) , and leaves the system with probability $1-q_i$ (in the latter case a new job i replaces it in the S_0-queue). After receiving an S_1-service, jobs join again the S_0-queue ('queue' includes the job, if any, being served).

The period of time between two consecutive joinings of the S_0-queue by the same job will be called a 'cycle' . Thus, if K_i ($K_i=1,2,\dots$) is the number of S_0-services required by a job i , its residence in the system consists

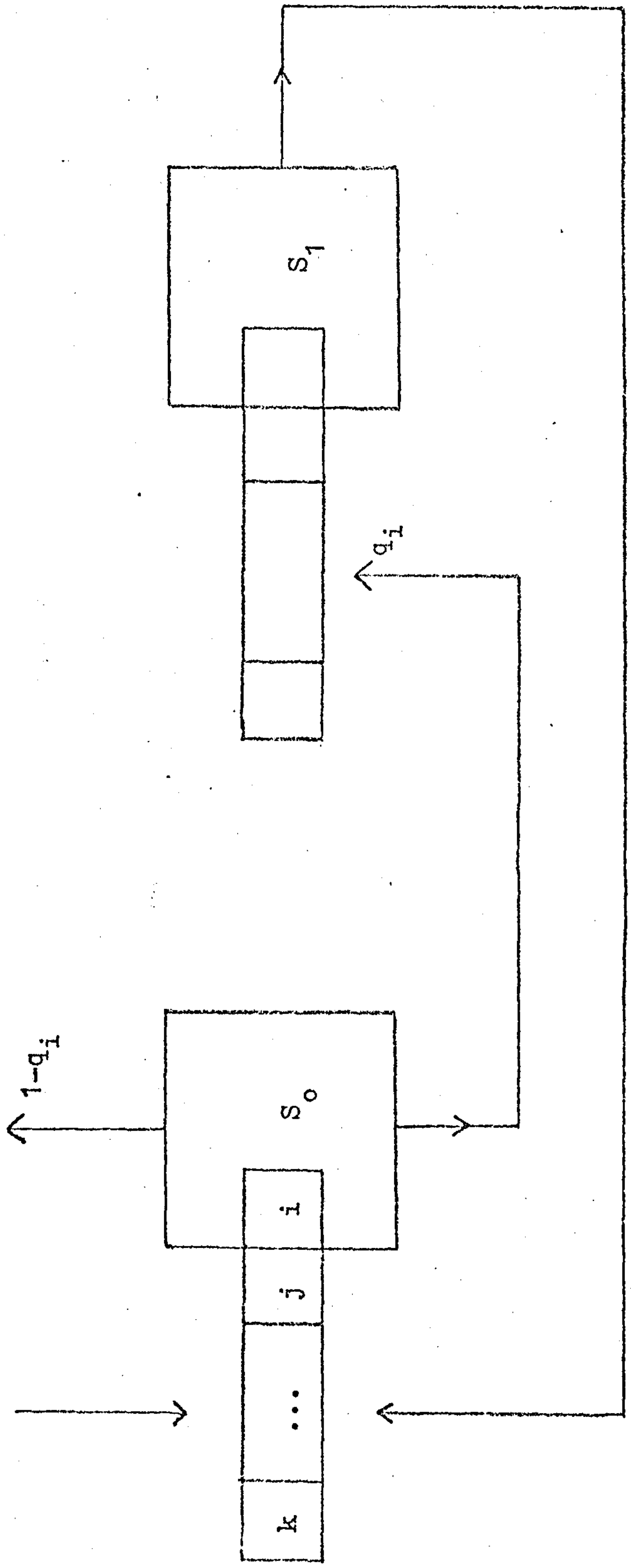


Figure 6.1

of K_i-1 cycles followed by a final stay in the S_0-queue . The random variable K_i can be seen to be distributed geometrically with parameter q_i :

$$P(K_i = k) = q_i^{k-1}(1-q_i) \ ; \ k=1,2,\ldots \ ; \ i=1,2,\ldots,N$$

The order in which the S_0-queue and the S_1-queue are served depends on the priorities of the jobs present there. In the S_0-queue, job i $(i=1,2,\ldots,N-1)$ has preemptive priority over jobs i+1, i+2, ...,N , i.e. if one of those jobs is being served when job i joins the queue, its service is interrupted and that of job i is started. An interrupted service is resumed from the point of interruption when there are no more higher priority jobs in the queue. This means, for instance, that when a job i leaves the system, the new job i which joins the S_0-queue goes straight into service (since the departing job i was able to complete its S_0-service, all other jobs, if any, in the S_0-queue must be of lower priority).

Regarding the S_1-queue, we shall consider two cases:

1. The S_1-queue is served under the same preemptive priority discipline as the S_0-queue.

2. The priorities in the S_1-queue are of the head-of-the-line type, i.e. no interruption of an S_1-service is allowed; when S_1 is ready to begin a new service, it selects the job with the highest priority among those present in the S_1-queue.

Real-life multiprogramming computing systems (except time-sharing systems) usually operate under preemptive priority discipline for the central processor queue and head-of-the-line priority discipline for the input/output queues . The reason for our considering a model with preemptive priorities for input/output is that it is interesting to compare the performance of the two models. Under certain circumstanses it may be desirable to implement a system with interruptable input/output operations.

6.2 Preemptive priority disciplines for both queues.

Because of the ability of higher priority jobs to interrupt the S_0- and S_1-services of lower priority jobs, our model has now the following property:

The execution of job i (i=1,2,...,N-1) is not affected in any way by the existence of jobs i+1, i+2, ..., N .

This property allows us to assume that N = i if we are interested in a quantity which is connected only with jobs 1, 2, ..., i .

Before proceding with the analysis, specific assumptions have to be made regarding the S_0- and S_1-service times of the jobs in the system. We shall assume that consecutive S_0-service times and consecutive S_1-service times are independent random variables. Furthermore, for i=1,2,...,N-1 , the distribution function of the S_0-service times of job i is given by

$$F_{i,0}(x) = 1 - e^{-m_{i,0}x} \; ; \; i=1,2,\ldots,N-1$$

and the distribution function of the S_1-service times of job i is given by

$$F_{i,1}(x) = 1 - e^{-m_{i,1}x} \; ; \; i=1,2,\ldots,N-1$$

The S_0-service times of job N have a general distribution function $F_{N,0}(x)$ with finite mean $1/m_{N,0}$ and the S_1-service times of job N have a general distribution function $F_{N,1}(x)$ with finite mean $1/m_{N,1}$.

Our aim is to find the steady-state average residence-in-the-system time of job i $(i=1,2,\ldots,N)$ and the steady-state S_0-utilisation factor.

Consider the group of jobs $1,2,\ldots,n$ $(1 \leq n \leq N)$. Denote by $p_n(0)$ the steady-state probability that none of these jobs is in the S_0-queue ; denote by $p_n(i,j,\ldots,k)$, where $i<j<$ $<\ldots<k\leq n$, the steady-state probability that jobs $i,j,$ $\ldots,k$ are in the S_0-queue (job i being served and jobs $j,\ldots,k$ waiting) and the rest of the n jobs are in the S_1-queue. In this notation, the steady-state S_0-utilisation factor is given by

$$U = 1 - p_N(0) \tag{6.1}$$

and the steady-state S_1-utilisation factor — by

$$1 - p_N(1,2,\ldots,N)$$

The S_l-queue $(l=0,1)$ goes through alternative periods of containing and not containing jobs from the group $1,2,$ $\ldots,n$. These periods will be called 'S_l-busy periods of type n' and 'S_l-idle periods of type n' $(l=0,1)$ respectively.

We saw that when one job i leaves the system, the job i which replaces it immediately begins its S_0-service. As far as the occupancy of S_0 is concerned, the S_0-service of the new job i can be considered as an extention of the S_0-service of the old one. Bearing this in mind, we shall modify the model by assuming that jobs do not leave the system and that the distribution functions of their S_0-service times are equal to

$$\begin{aligned} G_i(x) &= 1 - e^{-q_i m_{i,0} x} \;; \; i=1,2,\dots,N-1 \;; \\ G_N(x) &= \sum_{k=1}^{\infty} q_N(1-q_N)^{k-1} F_{N,0}{}^{(k)}(x) \end{aligned} \qquad (6.2)$$

where $F_{N,0}{}^{(k)}(x)$ denotes the k-fold convolution of $F_{N,0}(x)$. The steady-state probabilities $p_n(0)$ and $p_n(i,j,\dots,k)$ have the same values in the original model and in the modified model but in the latter, jobs repeat their 'S_0-queue — S_1-queue' cycles unceasingly.

Suppose that the steady-state average cycle length for job n (in the modified model) has been found. Denote it by c_n $(n=1,2,\dots,N)$. The argument can then proceed as follows:

In the steady state, job n makes an average of $1/c_n$ cycles per unit time, therefore it visits the S_0-queue, on the average, $1/c_n$ times per unit time. Since job n receives exactly one S_0-service per residence in the S_0-queue and since, according to (6.2) , the average length of the S_0-services of job n is equal to $1/(q_n m_{n,0})$, the steady-state proportion of time that job n is being served by S_0 is equal to $1/(q_n m_{n,0} c_n)$. This is also the steady-state probability that job n is in the S_0-queue but no higher priority jobs are :

$$p_n(n) = \frac{1}{q_n m_{n,0} c_n} \quad ; \; n=1,2,\ldots,N \qquad (6.3)$$

Similarly, the steady-state probability that job n is in the S_1-queue but no higher priority jobs are, is equal to

$$\begin{aligned} p_n(1,2,\ldots,n-1) &= \frac{1}{m_{n,1} c_n} \quad ; \; n=2,3,\ldots,N \; ; \\ p_1(0) &= \frac{1}{m_{1,1} c_1} \end{aligned} \qquad (6.4)$$

Reverting to the original model, we note that while S_0 is giving service to jobs n , they leave the system at rate $(1-q_n)m_{n,0}$. Therefore, the steady-state rate of departure (arrival) of jobs n from (into) the system is equal to

$$L_n = (1-q_n)m_{n,0}p_n(n) = \frac{1-q_n}{q_n c_n} \; ; \; n=1,2,\ldots,N \qquad (6.5)$$

The time that one job n spends in the system is equal to the interval between its and its successor's arrivals into the system. This means that the steady-state average residence-in-the-system time for job n is equal to

$$W_n = \frac{1}{L_n} = \frac{q_n c_n}{1-q_n} \quad ; \; n=1,2,\ldots,N \qquad (6.6)$$

If the averages c_i , and hence the probabilities $p_i(i)$ and $p_i(1,2,\ldots,i-1)$ were known for all $i=1,2,\ldots,n$, then the probability $p_n(0)$ could be found from

$$p_n(0) = 1 - \sum_{i=1}^{n} p_i(i) \; ; \; n=1,2,\ldots,N \qquad (6.7)$$

(the S_0-queue is free from jobs belonging to the group 1,2,

$\ldots, n$ iff neither job 1, nor job 2, ..., nor job n is being served by S_0). Similarly, the steady-state probability $p_n(1,2,\ldots,n)$ can be found from

$$p_n(1,2,\ldots,n) = 1 - \sum_{i=1}^{n} p_i(1,2,\ldots,i-1) \; ; \; i=1,2,\ldots,N \quad (6.8)$$

Derivation of expressions for c_n .

Our problem has been reduced to that of finding the steady-state average cycle length for job n $(n=1,2,\ldots,N)$, in the modified model. Let $s_{n,l}$ $(n=1,2,\ldots,N \; ; \; l=0,1)$ be the steady-state average time that job n spends in the S_l-queue. Obviously,

$$c_n = s_{n,0} + s_{n,1} \quad ; \; n=1,2,\ldots,N \qquad (6.9)$$

For $n = 1$ we have

$$s_{1,0} = \frac{1}{q_1 m_{1,0}} \; ; \; s_{1,1} = \frac{1}{m_{1,1}} \; ; \; c_1 = \frac{1}{q_1 m_{1,0}} + \frac{1}{m_{1,1}} \qquad (6.10)$$

because job 1 never waits.

Let $2 \leqslant n \leqslant N$. In the modified model, a job joining a queue must have just completed a service at the other server, i.e. must have been the highest priority job in the other queue prior to the joining. This implies that whenever job n joins the S_l-queue $(l=0,1)$ it finds jobs $1,2,\ldots,n-1$ in the S_l-queue.

The residence time of job n in a queue consists of an 'initial wait' — the time between joining the queue and beginning service — followed by the service, interspersed with preemptions. A little reflection convinces us that, because the service times of jobs $1,2,\ldots,n-1$ are distributed exponentially, the steady-state average initial wait of job n in the S_1-queue is equal to the steady-state average residence time of job n-1 in the S_1-queue, $s_{n-1,l}$ (l=0,1) .

Consider the interval between the beginning and the terminating of an S_0-service of job n . This interval consists of the actual service and of all wait periods initiated by preemptions.

At any time when job n is being served by S_0 , jobs 1, 2,...,n-1 are in the S_1-queue, job 1 being served by S_1 . Thus, if the S_0-service of job n is in progress at time t, a preemption will occur in the interval (t,t+dt) with probability $m_{1,1}dt$. It follows that the S_0-service of job n is preempted, on the average, $m_{1,1}/(q_n m_{n,0})$ times.

Each wait period caused by a preemption begins with the arrival in the S_0-queue of job 1 and ends with the departure from the S_0-queue of the last of jobs 1,2,...,n-1 . In other words, each such period is an S_0-busy period of type n-1 .

The steady-state average length of an S_0-idle period of type i is equal to $1/m_{1,1}$, for all i . Denoting the steady-state average length of an S_0-busy period of type i by $b_{i,0}$, we can write

$$p_i(0) = \frac{1/m_{1,1}}{b_{i,0} + 1/m_{1,1}} = \frac{1}{b_{i,0}m_{1,1} + 1}$$

which yields

$$b_{i,0} = \frac{1 - p_i(0)}{m_{1,1}p_i(0)}$$

Adding together the steady-state average lengths of the initial wait, the total wait due to preemptions and the S_0-service time, we obtain

$$s_{n,0} = s_{n-1,0} + \frac{m_{1,1}}{q_n m_{n,0}} \cdot \frac{1 - p_{n-1}(0)}{m_{1,1} p_{n-1}(0)} + \frac{1}{q_n m_{n,0}}$$

which is equivalent to

$$s_{n,0} = s_{n-1,0} + \frac{1}{q_n m_{n,0} p_{n-1}(0)} \tag{6.11}$$

A similar argument, applied to the residence of job n in the S_1-queue, leads to

$$s_{n,1} = s_{n-1,1} + \frac{1}{m_{n,1} p_{n-1}(1,2,\ldots,n-1)} \tag{6.12}$$

Substitution of (6.11) and (6.12) into (6.9) yields

$$c_n = c_{n-1} + \frac{1}{q_n m_{n,0} p_{n-1}(0)} + \frac{1}{m_{n,1} p_{n-1}(1,2,\ldots,n-1)} \tag{6.13}$$

Finally, (6.7), (6.8), (6.3) and (6.4) can be used to express $p_{n-1}(0)$ and $p_{n-1}(1,2,\ldots,n-1)$ in terms of c_1, $c_2,\ldots,c_{n-1}$. This gives

$$\begin{aligned} c_n = c_{n-1} &+ \left\{q_n m_{n,0}\left[1 - \sum_{i=1}^{n-1} \frac{1}{q_i m_{i,0} c_i}\right]\right\}^{-1} + \\ &+ \left\{m_{n,1}\left[1 - \sum_{i=1}^{n-1} \frac{1}{m_{i,1} c_i}\right]\right\}^{-1} \end{aligned} \tag{6.14}$$

All c_n $(n=2,3,\ldots,N)$ can be determined by repeated applications of (6.14), with c_1 given by (6.10).

Remark: As can be seen from the derivation, the quantities in which we are interested (e.g. the central processor utilisation factor) do not depend on the form of the distributions $F_{N,0}(x)$ and $F_{N,1}(x)$, only on their averages.

Special cases.

a) To reduce the number of parameters a little, suppose that $m_{1,1} = m_{2,1} = \dots = m_{N,1} = m_1$. Now the central processor utilisation factor is a function of the traffic intensities $r_n = m_1/(q_n m_{n,0})$; $n=1,2,\dots,N$. When $N = 3$, for instance, we have

$$U = \frac{r_1(1+r_2)(1+r_1+r_1r_3)}{1 + r_1 + r_1(1+r_2) + r_1^2(1+r_2)(1+r_3)}$$

The steady-state average residence-in-the-system times for jobs 1,2,3 are equal to

$$W_1 = \frac{q_1(1+r_1)}{m_1(1-q_1)} \quad ; \quad W_2 = \frac{q_2(1+r_1)(1+r_1+r_1r_2)}{m_1(1-q_2)r_1} \quad ;$$

$$W_3 = \frac{q_3(1+r_1+r_1r_2)\left[1 + r_1 + r_1(1+r_2) + r_1^2(1+r_2)(1+r_3)\right]}{m_1(1-q_3)r_1^2(1+r_2)}$$

Note that if r_1 is close to zero, i.e. if top priority is given to jobs whose central processor requirements are negligible compared to their input/output requirements, U is also close to zero, while W_2 and W_3 are close to infinity. The throughput (the total rate of departures from the system) is approximately equal to $m_1(1-q_1)/q_1$. If, on the other hand, r_2 (or r_3) is close to zero, it is easy to find a combination of values for the remaining parameters which will produce a greater central processor utilisation factor and a greater throughput. This example demonstrates that the commonly accepted practice of giving highest priority to jobs which are most 'input/output — bound' is not necessarily the most efficient.

We shall see that the situation is similar when the priorities for input/output are of the head-of-the-line type. The problem of optimal allocation of priorities is interesting and not trivial, especially when charges and revenue are to be taken into account.

b) When $q_1 = q_2 = \ldots = q_N = q$ and $m_{1,0} = m_{2,0} = \ldots = m_{N,0} = m_0$, as well as $m_{1,1} = m_{2,1} = \ldots = m_{N,1} = m_1$, all quantities of interest can be found explicitly. The steady-state S_0-utilisation factor is equal to

$$U = \frac{r(1 - r^N)}{1 - r^{N+1}}$$

where $r = m_1/(qm_0)$. The steady-state average residence-in-the-system time of job n is given by

$$W_n = \frac{(1 + r + \ldots + r^{n-1})(1 + r + \ldots + r^n)}{(1-q)m_0 r^n} \quad ; \; n=1,2,\ldots,N$$

Direct summation shows that the throughput is equal to

$$L = (1-q)m_0 \frac{r(1 + r + \ldots + r^{N-1})}{1 + r + \ldots + r^N} = (1-q)m_0 U$$

We see that the expressions for U and L are the same as in the case when the S_0-queue and the S_1-queue are served in order of arrival (chapter 3) . This is not surprising : when all jobs have the same characteristics, the central processor utilisation factor and the throughput are independent of the queuing disciplines.

6.3 Non-preemptive priorities for input/output.

In this version of the model, the S_1-queue is served according to the head-of-the-line priority discipline and the S_0-queue — according to the preemptive-resume priority discipline. We shall make the following assumptions regarding the S_0- and the S_1-service times :

Consecutive S_0-service times and consecutive S_1-service times are independent random variables. The S_0-service times of job i are distributed exponentially with mean $1/m_{i,0}$, for all $i = 1,2,\ldots,N$. The S_1-service times of job i have a general distribution $F_{i,1}(x)$, with finite mean $1/m_{i,1}$, for all $i = 1,2,\ldots,N$.

We observe again that, if the model is modified by assuming that jobs do not leave the system and the distribution functions of the S_0-service times are given by

$$G_i(x) = 1 - e^{-q_i m_{i,0} x} \quad ; \quad i=1,2,\ldots,N$$

the steady-state probabilities of the various queue configurations will not change.

Denote by c_i $(i=1,2,\ldots,N)$ the steady-state average cycle length for job i , in the modified model. Since the arguments used in deriving (6.3) and (6.6) were independent of the S_1-queuing discipline, we can write, for the steady-state average residence-in-the-system time of job i

$$W_i = \frac{q_i c_i}{1 - q_i} \quad ; \; i=1,2,\ldots,N \tag{6.6a}$$

Similarly, the steady-state central processor utilisation factor is given by

$$U = \sum_{i=1}^{N} \frac{1}{q_i m_{i,0} c_i} \tag{6.7a}$$

(see (6.7) and its derivation). The throughput is equal to

$$L = \sum_{i=1}^{N} \frac{1 - q_i}{q_i c_i} \tag{6.5a}$$

(see (6.5) and its derivation).

The problem of finding W_i , U and L has thus been reduced, once more, to that of finding the averages c_i ($i= 1,2,\ldots,N$). However, because low-priority jobs can delay the S_1-services of high-priority jobs, the inductive approach described in the last section is no longer applicable. To find the c_i's, we shall use the method of the embedded Markov chain and semi-Markov process.

Let t_1, t_2, ..., t_n, ... be the consecutive moments of S_1-service completions. The state of the system at time t_n^+, i.e. just after the n-th S_1-service completion, is completely determined by the set of jobs, $Q(t_n^+)$, which are in the S_0-queue then. (For example, if $Q(t_n^+) = \{1,3,4\}$, then we know that job 1 is being served by S_0 , job 2 has just begun an S_1-service, jobs 3 and 4 are waiting in the S_0-queue and jobs $5,6,\ldots,N$ are waiting in the S_1-queue.)

There are $2^N - 1$ (since $Q(t_n^+)$ is never empty) possible states in which the system can be at t_n^+. These states can be conveniently indexed by the integers $1,2,\ldots,2^N-1$ in the following way :

Working from right to left, write a 1 or a 0 in position i , depending on whether job i is in $Q(t_n^+)$ or not, for $i=1,2,\ldots,N$; treat the resulting set of N binary digits as a binary representation of an integer ; denote that integer by $S(t_n^+)$ and use it to represent the state $Q(t_n^+)$. (For example, if $Q(t_n^+) = \{1,3,4\}$, then $S(t_n^+) =$ $= 00\ldots01101 = 13$.)

Because of the exponential distributions of the S_0-service times, the stochastic process

$$M = \{S(t_n^+) \ , \ n=1,2,\ldots\}$$

is a (finite-state) Markov chain. It is, furthermore, irreducible and aperiodic and therefore possesses a steady-state distribution $\underline{p} = (p_1, p_2, \ldots, p_{2^N-1})$, defined by

$$p_k = \lim_{n\to\infty} P\left[S(t_n^+) = k\right] \ ; \ k=1,2,\ldots,2^N-1 \qquad (6.15)$$

Let $V = (v_{j,k})_{j,k=1}^{2^N-1}$ be the transition probability matrix of M ; the elements of V are defined by

$$v_{j,k} = P\left[S(t_{n+1}^+)=k \middle| S(t_n^+)=j\right] \quad ; \ j,k=1,2,\ldots,2^N-1 \qquad (6.16)$$

The expressions for $v_{j,k}$ are given in the appendix of this chapter.

Let V_1 be the matrix obtained from V by subtracting 1 from the elements on its main diagonal and then replacing its last column with a column of 1's . The steady-state distribution $\underline{p}$ is given by

$$\underline{p} = I.V_1^{-1} \tag{6.17}$$

where I is the vector (of 2^N-1 elements)

$$I = (0,0,\ldots,0,1)$$

<u>Note.</u> Finding the inverse of V_1 is a difficult operation for large values of N . When doing it numerically, one should make use of the fact that V_1 is a very sparse matrix.

Consider now the times involved in the transitions of the Markov chain M , i.e. regard M as a semi-Markov process. More precisely, let m_k $(k=1,2,\ldots,2^N-1)$ be the average length of the interval between two successive Markov epochs, (t_n,t_{n+1}) , given that $S(t_n^+) = k$. When $k = 2^N-1$ we have

$$m_{2^N-1} = \frac{1}{q_1 m_{1,0}} + \frac{1}{m_{1,1}} \tag{6.18}$$

because, if all jobs are in the S_0-queue at t_n^+ , then (t_n,t_{n+1}) consists of the remaining S_0-service of job 1 plus its S_1-service. When $1 \leqslant k < 2^N-1$, the S_1-queue is not empty at t_n^+ and (t_n,t_{n+1}) consists of the S_1-service of the highest priority job in the S_1-queue. Hence

$$m_k = \frac{1}{m_{l,1}} \; ; \; k=1,2,\ldots,2^N-2 \tag{6.19}$$

where l is the position (counted from the right) of the rightmost zero in the N-digit binary representation of k .

Knowing the averages m_k and the steady-state probabilities p_k $(k=1,2,\ldots,2^N-1)$, we can find $t_{k,k}$, the steady-state average first passage times of M from state k to state k . The well-known formula in the theory of semi-Markov processes (see Barlow and Proschan [7]) yields

$$t_{k,k} = \frac{1}{p_k}\sum_{j=1}^{2^N-1} m_j p_j \quad ; k=1,2,\ldots,2^N-1 \qquad (6.20)$$

The reciprocal of $t_{k,k}$,

$$n_k = \frac{1}{t_{k,k}} \quad ; k=1,2,\ldots,2^N-1 \qquad (6.21)$$

represents the steady-state average number of times that M is in state k per unit time (the average number,per unit time, of Markov epochs t_n such that $S(t_n^+) = k$).

Denote by s_i $(i=2,3,\ldots,N)$ the set of integers $\{k \mid 1 \leqslant k < 2^N-2$ and the rightmost zero in the N-digit binary representation of k is in position i , counted from the right$\}$. s_i has the property that if $S(t_n^+) = k$ for some $k \in s_i$, then job i is the highest priority job in the S_1-queue at t_n^+ , i.e. job i has just begun an S_1-service. Denote by s_1 the set $\{2,4,6,\ldots,2^N-2,2^N-1\}$. s_1 has the property that if $S(t_n^+) = k$, for some $k \in s_1$, then job 1 has either just begun an S_1-service or will do so when its S_0- service is completed.

We can find now

$$n(i) = \sum_{k \in s_i} n_k \quad ; i=1,2,\ldots,N \qquad (6.22)$$

$n(i)$ is the steady-state average number of times, per unit time, that job i begins an S_1-service. The reciprocal of this number is the steady-state average length of the interval between two consecutive moments of admission into S_1-service for job i $(i=1,2,\ldots,N)$. This last quantity can be seen to be precisely the steady-state average cycle length for job i . Thus

$$c_i = \frac{1}{n(i)} \quad ; \; i=1,2,\ldots,N$$

which, after substitution of (6.22) ,(6.21) and (6.20) becomes

$$c_i = \frac{\sum_{j=1}^{2^N-1} m_j p_j}{\sum_{k\in s_i} p_k} \quad ; \; i=1,2,\ldots,N \qquad (6.23)$$

<u>Special cases</u>

a) When $m_{1,0}=m_{2,0}=\ldots=m_{N,0}$, $m_{1,1}=m_{2,1}=\ldots=m_{N,1}$ and $q_1=q_2=\ldots=q_N$, the numerator in the right-hand side of (6.23) is equal to

$$\frac{1}{m_{1,1}}(1 + r.p_{2^N-1})$$

where $r = m_{1,1}/(q_1 m_{1,0})$ is the traffic intensity. Substitution into (6.7a) now yields

$$U = (\frac{1}{r} + p_{2^N-1})^{-1}$$

This formula agrees with the one obtained in the case of FIFO queuing disciplines at both queues (chapter 4) , provided that the distributions, as well as the means, of the S_1-service times are identical.

b) When $m_{1,1}=m_{2,1}=\ldots=m_{N,1}$ and when the S_1-service times are distributed exponentially, the probabilities p_k and the S_0-utilisation factor can be expressed in terms of the traffic intensities $r_i = m_{1,1}/(q_i m_{i,0})$ $(i=1,2,\ldots,N)$. In the case of $N = 2$, for instance,

$$U = \frac{(1+r_1)(r_1+r_2+r_1r_2)}{1+(1+r_1)^2(1+r_2)} \qquad (6.24)$$

We note that if one job class has a very low traffic intensity, a greater S_0-utilisation factor may be achieved by assigning a lower priority to that class.

Finally, compare (6.24) with the expression for U in the 'preemptive priorities at S_1' model :

$$U = \frac{r_1(1+r_2)}{1+r_1+r_1r_2}$$

Stating the results of the comparison in general terms, we can say that the central processor utilisation factor is higher in systems with interruptable input/output operations when the low-priority jobs are very input/output - oriented ($r_2 \sim 0$) ; it is higher in systems with non-interruptable input/output operations when the top-priority jobs are very input/output -oriented ($r_1 \sim 0$) . There is little difference between the two types of systems when $r_1 \sim r_2$, or $r_1 \sim \infty$, or $r_2 \sim \infty$.

APPENDIX

Expressions for the transition probabilities (6.16) .

We note first that if $j = 2^N-1$ then

$$v_{j,k} = v_{j-1,k} \quad ; \; k=1,2,\dots,2^N-1 \quad .$$

because, if all jobs are in the S_0-queue at t_n^+ , job 1 must complete its S_0-service before anything else can happen; from that moment the system behaves as if it started in state 2^N-2 .

Suppose now that $1 \leqslant j \leqslant 2^N-2$. Let $j_{1,0} < j_{2,0} < \dots < j_{s,0}$ be the positions (counted from the right) of the ones, and j_1 be the position (counted from the right) of the rightmost zero, in the N-digit binary representation of j . In other words, let jobs $j_{1,0}, j_{2,0}, \dots, j_{s,0}$ be the jobs in the S_0-queue and job j_1 be the highest priority job in the S_1-queue at t_n^+ . At t_{n+1}^+ job j_1 will be in the S_0-queue and some, or none, or all, of jobs $j_{1,0}, j_{2,0}, \dots, j_{s,0}$ will be in the S_1-queue. The possible transitions, therefore, are :

to state $k_0 = j + 2^{j_1-1}$, if the S_1-service of job j_1 is completed before the S_0-service of job $j_{1,0}$;

to state $k_1 = j + 2^{j_1-1} - 2^{j_{1,0}-1}$, if the S_0-service of job $j_{1,0}$, but not that of job $j_{2,0}$, is completed before the S_1-service of job j_1 ;

- -

to state $k_s = j + 2^{j_1-1} - 2^{j_{1,0}-1} - \dots - 2^{j_{s,0}-1}$, if the S_0-services of jobs $j_{1,0}, j_{2,0}, \dots, j_{s,0}$ are completed before the S_1-service of job j_1 (k_s is, of course, equal to 2^{j_1-1}).

Denote by $F(i_1,i_2,\ldots,i_n;x)$ the distribution function of the sum of n independent random variables distributed exponentially with parameters $q_{i_1}m_{i_1,0}$, $q_{i_2}m_{i_2,0}$, $\ldots$, $q_{i_n}m_{i_n,0}$ respectively $(1\leqslant i_1<i_2<\ldots<i_n\leqslant N)$. The probability that the S_0-services of jobs $j_{1,0}$, $j_{2,0}$, $\ldots$, $j_{n,0}$ $(1\leqslant n\leqslant s)$ are completed before the S_1-service of job j_1 is equal to

$$b(j_{1,0},j_{2,0},\ldots,j_{n,0};j_1) = \int_0^\infty F(j_{1,0},j_{2,0},\ldots,j_{n,0};x)\,dF_{j_1,1}(x)$$

We can write now

$$v_{j,k_0} = 1 - b(j_{1,0};j_1) \quad ;$$

$$v_{j,k_1} = b(j_{1,0};j_1) - b(j_{1,0},j_{2,0};j_1) \quad ;$$

- -

$$v_{j,k_{s-1}} = b(j_{1,0},j_{2,0},\ldots,j_{s-1,0};j_1) - b(j_{1,0},j_{2,0},\ldots,j_{s,0};j_1)$$

$$v_{j,k_s} = b(j_{1,0},j_{2,0},\ldots,j_{s,0};j_1)$$

For all other values of k,

$$v_{j,k} = 0$$

References.

[1a] Feller, W., An Introduction to Probability Theory and Its Applications, vol.1, Wiley, New York, 1950.

[1b] Feller, W., An introduction to Probability Theory and Its Applications, vol.2, Wiley, New York, 1957.

[2] Parzen, E., Stochastic Processes, Holden-Day, 1962.

[3] Lewis, P.A.W. and Shedler, G.S., A Cyclic-Queue Model of System Overhead in Multiprogrammed Computer Systems, J. ACM, 18, 2, pp. 199-220, 1971.

[4] Little, J.D.C., A proof of the queuing formula L= W, Oper. Res., 9, pp. 383-387, 1961.

[5] Saaty, T.L., Elements of Queuing Theory, McGraw-Hill, London, 1961.

[6] Barraclough, E.D., The use of an IBM 2829 Basic Counter Unit to monitor the hardware performance of the 360/67 computer, Computing Laboratory, University of Newcastle upon Tyne, May, 1970.

[7] Barlow, R.E., and Proschan, F., Mathematical Theory of Reliability, Wiley, New York, 1965.

[8] Gordon, W.J. and Newell, G.F., Closed Queuing Networks with Exponential Servers, Oper. Res., 15, pp. 254-265, 1967.

[9] Doob, J.L., Stochastic Processes, Wiley, New York, 1953.

[10] Gaver, D.P., Probability Models for Multiprogramming Computer Systems, J. ACM, 14, 3, pp. 423-438, 1967.

[11] Wallace, V.L. and Mason, D.L., Degree of Multiprogramming in Page-on-Demand Systems, Comm. ACM, 13, 6, pp. 305-308, 1969.

[12] Chen, Y.C. and Shedler, G.S., A Cyclic Queue Network Model for Demand Paging Computer Systems, IBM Research Report RC-2398, 1969.

[13] Shedler, G.S., A Cyclic Queue Model of a Paging Machine, IBM Research Report RC-2814, 1970.

[14] Adiri, I., Hofri, M. and Yadin, M., A Multiprogramming Queue, IBM Research Report RC-3566, 1971.

[15] Omahen, K. and Schrage, L., A Queuing Analysis of a Multiprocessor System with Shared Memory, Proceedings of the Symposium on Computer-Comunications Networks and Teletraffic, Polytechnic Institute of Brooklyn, April 1972.

[16] Koenigsberg, E., Cyclic Queues, Oper, Res. Quart., 9, pp. 22-35, 1958.

[17] Mitrani, I., Nonpriority Multiprogramming Systems Under Heavy Demand Conditions - Customers' Viewpoint, J. ACM, 19, 3, pp. 445-452, 1972.

[18] Mitrani, I., Multiprogramming Systems with One Processor and Many Input-Output Devices, University of Newcastle upon Tyne Technical Report Series, No. 22, 1971.

[19] O'Donovan, T., Mathematical Models of Time-Sharing Systems, Ph.D. Thesis, Trinity College, Dublin, 1971.

[20] McKinney, J.M., A Survey of Analytical Time-Sharing Models, Comp. Surveys, 1, pp. 105-116, 1969.