1990

Queueing Models of Secondary Storage Devices

Edward G. Coffman

Report Number:

90-943

QUEUEING MODELS OF SECONDARY STORAGE DEVICES

Edward G. Coffman, Jr.
Micha Hofri

CSD-TR-943
January 1990

QUEUEING MODELS OF SECONDARY STORAGE DEVICES†

Edward G. Coffman Jr., AT&T Bell Laboratories, Murray Hill, NJ, 07974.
Micha Hofri, Department of Computer Science, The Technion, 32000 Haifa.‡

ABSTRACT

This chapter concerns the mathematical modeling and analysis of secondary (or auxiliary) storage devices, which often comprise the principal bottleneck in the overall performance of computer systems. The presentation begins with descriptions of the more important devices, such as disks and drums, and a general discussion of related queueing models. Server motion and dependent successive services are salient features of these models. Widely used, generic results and methods are presented and then applied to specific devices. The chapter concludes with a discussion of open problems and desirable extensions of the basic models.

† This is an update of the identically named chapter in H. Takagi (Ed.): *Stochastic Analysis of Computer and Communication Systems,* Elsevier Science Publ. B.V. 1990.

‡Currently at the Department of Computer Science, Purdue University, W. Lafayette, IN 47907.

1. Introduction

Secondary storage devices constitute an important, often critical, part of a computing system. Programs in the machine access these devices to record (write) or retrieve (read) data, and use them as stable repositories for long-term files, or as buffers and working storage for short-term files. The correct and efficient use of these devices raises numerous engineering problems dealing with different aspects of their operation.

1.1. The role of queueing theory

Secondary storage devices are regarded here as *service systems*. The programs using the computer generate input and output commands, which are the services requested of the storage subsystem. The interleaved operation of several programs gives rise to an essentially nondeterministic sequence of inter-request time intervals. The interleaving is largely uncontrollable by (and usually invisible to) the programs involved. Thus the modeling of this sequence by a stochastic process seems appropriate, even if each participating program is entirely deterministic.

As will appear from the descriptions below the devices do not enjoy the random access property of computer main storage. In a non-random device the time required to satisfy a request, i.e. to access data, depends not only on the amount and location of the data, but on the "state" of the device, which in turn is usually determined entirely by the previously accessed data. Since successive accesses performed by the device are often requested by distinct programs, the service times again appear amenable to representation by a stochastic process. Hence the application of queueing theory to analyze the behavior of secondary storage subsystems is natural. It has been in evidence for over twenty years, and began shortly after the initial research into the performance evaluation of computer systems.

1.2. Common devices

The types of secondary storage devices one finds in computer systems run the gamut from magnetic tapes through disks, drums, shift-registers (of different technologies) to various combinations of the above, such as tape libraries with "staging" disks or the rather uncommon data-cell and block-addressable solid-state devices.

Magnetic tapes are inherently non-sharable, accessed sequentially, and leave little scope for manipulation by the users or the creativity of the system engineer and the performance analyst. Specifically, their being used by a single process over a long time implies that a queueing model is inappropriate to describe their interaction with the rest of the system. Hence they will not be further considered here.

The most common device, by far, is the *moving arm disk*. Its salient geometrical properties are as follows (see Fig. 1):

(*i*) The data are read/written from/to a magnetic layer coating the rotating surfaces. The

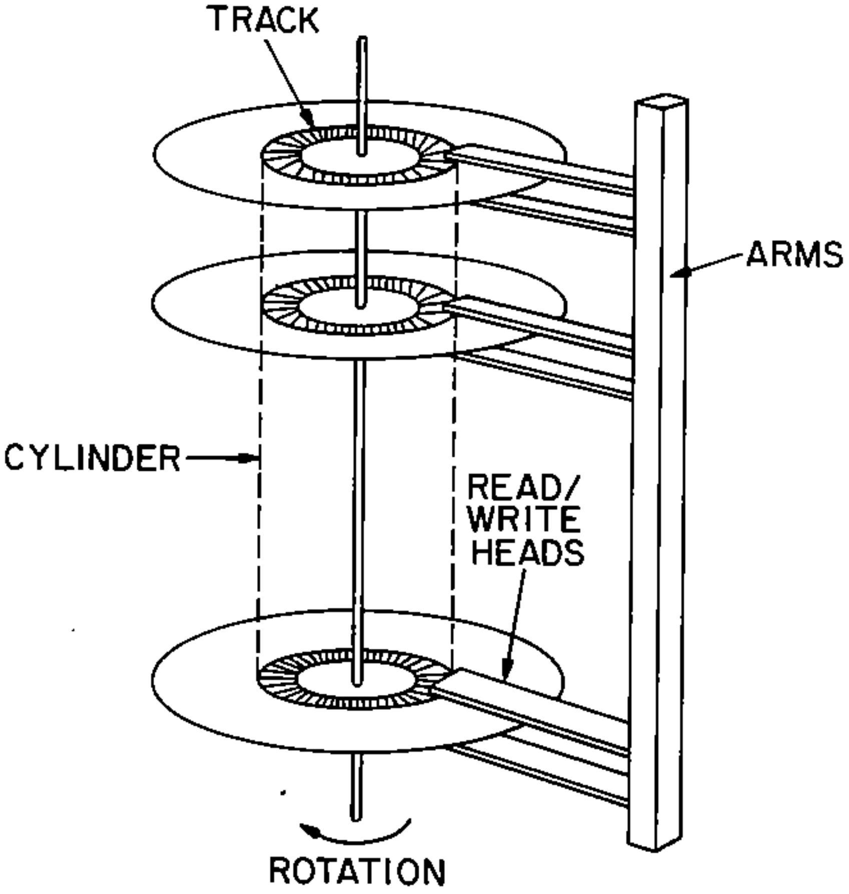

Figure 1: Moving-arm disk geometry

number of surfaces is typically between 2 and 40. A common value for the rotation period is 16⅔ milliseconds (with 60 revolutions per second).

(*ii*) Each surface has an associated *read/write head.* When the head is stationary it can access one *track:* an annular element of the surface. The number of tracks per surface is typically between 200 and 1000.

(*iii*) The heads are rigidly attached to an arm. The totality of tracks that are accessible when the arm is stationary is called a *cylinder.*

(*iv*) The arm can move, according to commands issued by the computer to the disk controller, so that different cylinders can be accessed. This motion is called a *seek.*

Some variations of the standard disk have more than one arm, or more than one read/write head per surface. In discussions of specific models below we shall need to consider a few more details concerning disk operation.

A cognate device, much more common in older systems, is the *drum.* Geometrically speaking, we may think of a drum as a single cylinder of a disk, with the heads permanently positioned to access any track (see Fig. 2). The number of tracks is usually much larger than the number in a disk cylinder, and may reach a few hundreds, but as there is only one set of tracks, the storage capacity is normally much lower. Since the heads are stationary, they can be positioned closer to the surface, permitting a higher recording density; this, coupled with the (usually) faster rotation produces higher transmission rates.

An intermediate device is the *fixed-head disk*. Here each track, rather than an entire surface, has an associated read/write head. For our purposes this may be viewed as either a set of parallel, co-axial drums, or one longer drum. Whichever point of view is adopted, the fixed-head disk can be lumped together with the ordinary drum for modeling purposes. Such disks are used for applications where continuously high data transmission rates are critical.

In Section 3 we shall present some variations on these devices as well as some more exotic breeds. A comprehensive reference on secondary storage devices is (Matick, 1977).

1.3. Elements of the queueing model

We describe the basic components of the model.

(a) The input process. As mentioned above, the concurrent execution of several processes in a computer system produces inter-request intervals that are highly irregular. Usually there are no patent, identifiable mechanisms which suggest to the model-builder a representative stochastic process. Responding to a universal dictum, with computational tractability the chief criterion, nearly all queueing models assume that requests are generated according to a time-homogeneous Poisson process. Sometimes several such processes are assumed to address independently distinct portions of the device address-space. While this assumption is rarely

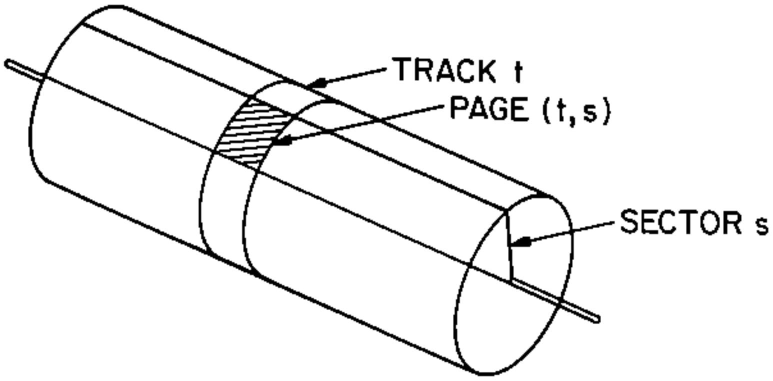

Figure 2: Magnetic drum geometry

well-approximated in reality, the analyst hopes that qualitative statements, especially comparisons between different operating regimes and system configurations, are robust under this idealization. The main departure from reality is apparently not in the assumption of exponentially distributed intervals (which has been observed occasionally to produce a reasonable fit), but in the assumption of time-homogeneity: usually a program that issues a request is blocked until request completion (especially if the request is for input). Since the population of active programs is finite, long queues would imply a lower request arrival rate, thus producing a stabilizing effect which is not reflected in the models. This implies that the quantitative results produced by the models described below may be only accurate at low or medium loads. Results for high loads can still be used to estimate the service capacity of the systems.

(b) The service process. It is here naturally that the geometry of a device comes to the fore. In drum systems, for example, the time to perform a service (e.g. read operation) depends on the rotational speed and angular position of the read head in relation to the beginning of the desired data. The head position just after reading or writing a record clearly influences the time required to access the next record. Hence successive drum services are not independent; they involve a *rotational latency,* which is the time that passes between the instants the heads complete the scan of one record and reach the beginning of the next.

Disks, which have an additional mechanical degree of freedom, have an additional source of latency. Unlike rotational latency, this arises from a controllable source: the positioning of the head-carrying arm. The difference will be very clearly reflected in the models below.

Economic considerations sometimes produce even more complicated structures for the service times. A moving-arm disk system is the standard example: The disk is not connected directly to the main memory, which is the source/destination of all data transfers, but through a distinct programmable device called a *channel.* One channel is commonly connectable to several disks, and connection is established both to issue control commands (such as initiating a seek) and to effect the data transfer. It may happen that when a seek or a rotational latency terminates, the channel is not available for further instructions or the data transfer, since it is then connected to another device. Thus additional delays are introduced, called *reconnect delays,* which increase the time required to service a request. Configurations with multiple disks often have additional switchable stages in the "data path", with vaguely descriptive titles such as *storage directors,* or *access facilities,* with the most common being *control unit.* Such a unit is normally connected to the device for the duration of the request processing, and hence produces no additional delay. From a logical point of view it may be considered a part of the device.

(c) Service regimes. Virtually no secondary storage system allows preemption of a service operation. Thus, the differences between regimes reduce to the order of selection for service. All of the models we encounter assume that the requests all have the same inherent priority. Service not in order of arrival is adopted only when taking advantage of the non-random structure of the device. Manipulations by the I/O scheduler (a part of the operating system that

actually dispatches the requests to the queues) in some instances use the different executing priorities of the originating processes to determine the position in queue of a newly generated request; once in the queue, the position of the request is thereafter preserved. To mask the non-random nature of the device the scheduler will often maintain distinct queues for distinct regions of the device address-space. These regions are sectors or tracks for drums, cylinders for disks etc. In these cases the regimes will differ according to the policy used to switch between the queues. The various policies will be discussed when we review the corresponding models.

We note that as a rule, in order to keep the computational overhead of the secondary storage subsystem as low as possible, only simple scheduling algorithms are implemented. Thus, adaptive algorithms are probably never used (at least we do not know of a single exception, although some have been proposed and discussed qualitatively).

(d) Performance measures. The criteria by which the various service policies are judged are the following:

(i) The sojourn times of requests; the main criterion is the first moment but the variance is usually important as well. If the variance is too large it results in some of the requests getting a much poorer service than others. Such inequitable service is usually considered inacceptable. Moreover, we may expect it to lead to inefficient use of the devices. To improve their utilization one could consider keeping in the system a larger population of active processes. However, a larger population is likelier to create higher congestion, leading to longer response times and a host of undesirable effects.

(ii) Queue lengths. These are necessary to estimate sizes of required data structures and the number of active processes that can be tolerated.

(iii) System operating capacity, defined as the maximum input rate the system can sustain without saturating.

We mention briefly that these devices also give rise to several types of problems of a stochastic nature, which are not amenable to a queueing-theoretic approach. For example there are placement problems (of files and records) based on access frequencies, compression and relocation problems, and others. Several are considered in (Wong, 1983). See also Section 4.

The following notation will be used in the rest of the paper:

S – service time duration.

μ^{-1} – first moment of S.

λ – rate of Poisson arrival processes.

$\rho \equiv \lambda/\mu$ – traffic intensity or load.

X – queue length (often at a specific set of epochs).

$A(z)$ – probability generating function (pgf) for the discrete random variable (rv) A.

$A(s)$ – Laplace-Stieltjes-Transform (LST) for the continuous rv A.

(When A denotes a process, the function $A(\cdot)$ refers to its steady-state distribution).

$\tilde{U}$ – the number of arrivals to a queue during a period denoted by the rv U. Thus $\tilde{U}(z) = U(\lambda-\lambda z)$.

Random variables with their parameters and associated transforms will often be subscripted by a class index, e.g. S_i, μ_i, λ_i, etc.

Vectors will be distinguished by bold (heavy) type. With no fear of ambiguity we shall need no explicit distinction between column and row vectors. Matrices are denoted by capital letters, their elements usually by the corresponding lower case characters.

2. Generic Models

Many results and techniques from queueing theory have been brought to bear on problems arising in the study of secondary storage systems. They have ranged widely in their appropriateness, sophistication and success. The characteristics of the devices and their modes of operation single out several techniques as particularly apt. These are reviewed below. Illustrations of their use appear in Section 3.

2.1. A single server with vacations

The first analysis of this model is in Skinner (1967). It has enjoyed high popularity since then; a comprehensive summary is given in Doshi (1986). Interesting applications and more recent results and generalizations may be found in Fuhrmann and Cooper (1985), Keilson and Servi (1987) and Loris-Teghem (1988). Takagi (1987) collects many of the results and applications in a text-book format. The model is a variation on the standard M/G/1 queueing model: Whenever the server becomes idle (i.e. when a departing customer leaves an empty queue), the server "takes a vacation". When the vacation terminates the server returns and inspects the queue. If it is still empty, he embarks on another vacation. Once the queue length at a vacation end is non-zero, the server resumes normal service activity until the queue re-empties, whereupon the cycle is repeated.

We use the well known fact that the queue length at departure epochs evolves as a Markov chain. Assuming stability, the following equations need to be satisfied

$$\begin{aligned} x_n &\equiv P(X = n \text{ immediately following a departure}) \\ &= x_0 \sum_{r=1}^{n+1} P(\tilde{U}_1 = r) P(\tilde{S} = n - r + 1) + \sum_{r=1}^{n+1} x_r P(\tilde{S} = n - r + 1) \end{aligned} \tag{1}$$

where U is the length of a single vacation, and $\tilde{U}_1$ is the number of arrivals during a vacation, given that there is at least one such arrival. The pgf for X is then

$$X(z) \equiv \sum_{n \geq 0} x_n z^n = x_0 \tilde{S}(z) \frac{\tilde{U}_1(z) - 1}{z - \tilde{S}(z)}, \quad x_0 = (1 - \rho)(1 - U(\lambda))/\lambda E(U). \tag{2a}$$

The reasoning used for the standard M/G/1 queue, with arrivals and departures occurring singly, shows that $X(z)$ is the pgf for the queue length observed by an arriving request, as well as by a random observer.

Since the input process and the service mechanism are independent when the queue is non-empty, one has for the LST of the sojourn time, when the selection for service is FCFS

$$H(s) = W(s)S(s) = X(1 - s/\lambda), \tag{3}$$

which also provides $W(s)$, the LST for the distribution of the steady-state waiting time, from request arrival till the beginning of its service.

The above analysis is implied in (Skinner, 1967). He considers the situation where following each service S there is an "inspection period" (T) after which either a service or a vacation takes place. The analysis is entirely similar; we use the notation $S_1 = S + T$, and observe that a vacation will only start following a departure at which the queue is empty, and no arrivals come during the subsequent inspection period. As $T(\lambda)$ equals the probability of no arrivals during T, we find for this system

$$X(z) = x_0 T(\lambda)\tilde{S}(z)\,\frac{\tilde{U}_1(z) - 1}{z - \tilde{S}_1(z)}, \quad x_0 = (1 - \rho_1)\frac{1 - U(\lambda)}{\lambda E(U)T(\lambda)}, \quad \rho_1 \equiv \rho + \lambda E(T). \tag{2b}$$

Skinner also computes an additional distribution which we use in the next section. He considers the distribution of the queue length not at departure epochs, but at inspection-end epochs, just before a service (S) or a vacation (U) commences. We denote this random variable by X_1. Referring to Fig. 3, the distribution we computed above is for the queue length when the server is at point A, whereas we compute now the probabilities at the epochs the server is at point B. These probabilities need to satisfy the equations

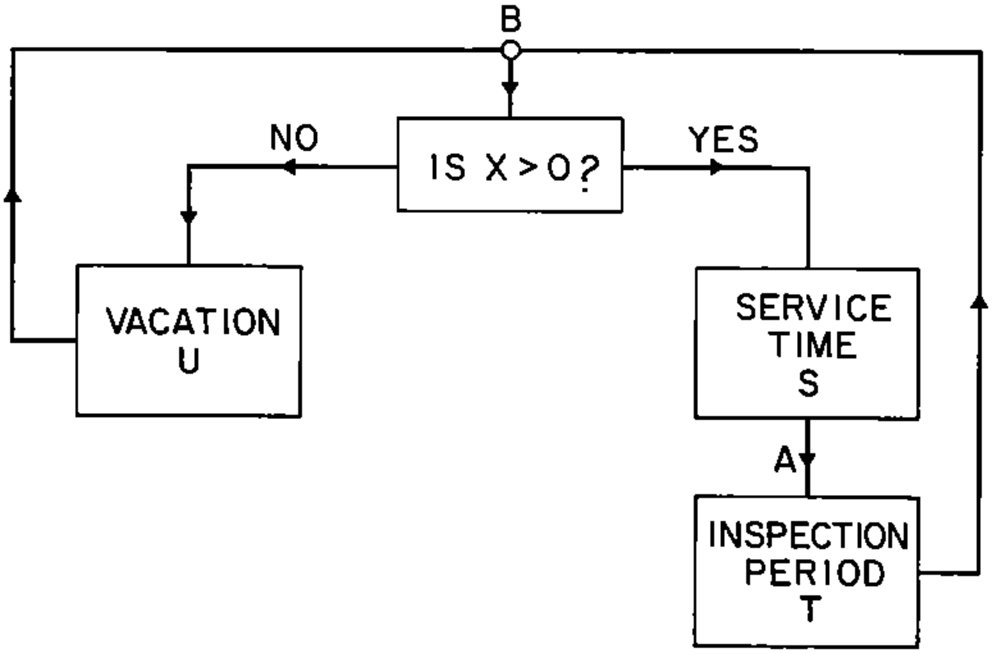

Figure 3: A generalized M/G/1 server itinerary.

$$x_{1,n} \equiv P(X = n \text{ immediately following an inspection or vacation})$$
$$= x_{1,0}P(\tilde{U} = n) + \sum_{r=1}^{n+1} x_{1,r}P(\tilde{S}_1 = n - r + 1), \tag{4}$$

and so, with obvious notation,

$$X_1(z) = x_{1,0}\frac{z\tilde{U}(z) - \tilde{S}_1(z)}{z - \tilde{S}_1(z)} \qquad x_{1,0} = \frac{1 - \rho_1}{1 - \rho_1 + \lambda E(U)}, \qquad \rho_1 = \lambda E(S_1). \tag{5}$$

Our version of the model of a single server with vacations has enjoyed a few generalizations. Doshi (1985) considers the case where successive vacations not separated by a busy-period have different (independent) distributions. Another straightforward generalization is analyzed by Hofri (1985), where the vacation sequence is terminated only when $X \geq m$, for some predetermined integer m. The Skinner model would correspond to $m = 1$.

Much of the recent work on vacation models concentrated on relating them to the underlying M/G/1 model (where the server is always available). Fuhrmann and Cooper (1985) provide an elegant representation of such relations, showing that the steady-state pgf for the number of customers in the system is a product of the corresponding pgf of the M/G/1 model—denoted by $\pi(z)$—and other, vacation-related pgf's that usually have intuitive interpretations.

In particular, they allow the server to take vacations at epochs other than when the queue is empty, with the following limitations:

(a) A vacation may only start at service- (or another vacation-) termination.

(b) Vacation beginning and end times must be independent of the future of the arrival process ("non-anticipatory vacations").

(c) The number of customers in the queue at vacation beginning has a stationary distribution, with a pgf denoted by $\zeta(z)$, and the duration of the vacation must be independent of this number.

Letting $\tilde{U}(z)$ denote as above the pgf for the number of arrivals during a vacation, and $\chi(z)$ the pgf of the numbers of customers in the system at a random instant during a vacation, they show the obvious relation

$$\chi(z) = \zeta(z)\frac{1 - \tilde{U}(z)}{E(U)(1 - z)}. \tag{6}$$

Then $X(z)$, the pgf of the number of customers left in the system behind a departing one, given above in equation (2), has the product form

$$X(z) = \chi(z)\pi(z), \tag{7}$$

with $\pi(z)$ given familiarly by the Khintchine-Pollaczek formula

$$\pi(z) = (1 - \rho)\frac{(z - 1)\tilde{S}(z)}{z - \tilde{S}(z)}. \tag{8}$$

One has only to note that the pgf $\tilde{U}_1(z)$ used in equation (2) equals $[\tilde{U}(z) - U(\lambda)]/[1 - U(\lambda)]$ to see that the two expressions agree.

It is remarkable that the relation (7) holds even when requirement (c) is not satisfied. Naturally equation (6) cannot hold then as well.

A related interesting decomposition can be effected with respect to the sojourn time of a request in a system with a FCFS regime. Using equations (3) and (7) we see

$$H(s) = \chi(1 - s/\lambda)\pi(1 - s/\lambda). \tag{9}$$

Now $\pi(1 - s/\lambda)$ is just the LST of $H_{M/G/1}$, the sojourn time in the underlying M/G/1 system. When $\zeta(z) = 1$ (exhaustive service between vacations) $\chi(1 - s/\lambda)$ is simply the LST of U', the vacation residual life-time (forward recurrence-time) at a random instant during a vacation. This means that H may be represented as $H_{M/G/1} + U'$, when the two components are considered independent.

2.2. Exhaustive service with one server and several queues (Eisenberg, 1972)

This model consists of M queues, fed by independent Poisson arrival processes with rates λ_m and service requirements S_m, $1 \le m \le M$. The server has a fixed cyclical itinerary of *I stages,* where stage i is associated with a queue m_i. The cycle may have any form; thus for $m = 3$, $I = 6$ the server could be required to visit the queues repeatedly in the sequence (1,2,1,3,1,2). The server remains at a queue until it empties, at which point the server switches to the next queue of the cycle. This switching starts the next stage. The switching time in stage i, from queue m_{i-1} to queue m_i requires a random duration denoted by C_i. If queue m_i is empty when the server arrives there, switching to queue m_{i+1} starts immediately. Note that the switching time is allowed to depend not only on the queues involved, but also on the stage index. In the above example we allow C_1 and C_5 to be differently distributed even though they both refer to switching from queue 1 to queue 2. Arithmetic in stage indices is always 1-modulo I (e.g., $i-1 = I$ when $i = 1$).

This model may be viewed as a generalization of the previous one, but it is considerably more ambitious, as the objective is to obtain the *joint* distribution of the M queue-lengths at two sets of epochs. One consists of the instants when a queue just emptied; such an event will be called "end of stage". The second set of epochs is embedded at all service completions. Note that the queue lengths at both sets of epochs form Markov chains.

The first set of epochs is "coarser" and easier to start with. Define $\beta_{i,\mathbf{x}}$ as the steady state probability that at the end of stage i the queue lengths are $\mathbf{x} = (x_1, \cdots, x_M)$, and let $\alpha_{i,\mathbf{x}}$ denote the corresponding probabilities at stage i service beginnings. Queue m_i is empty at a stage-i ending, so $\beta_{i,\mathbf{x}}=0$ for all $x_{m_i} \ne 0$, by definition. To compute these distributions we start by defining the pgf

$$\beta_i(z) = \sum_{\mathbf{x}} \beta_{i,\mathbf{x}} z^{\mathbf{x}} = \sum_{x_1 \ge 0} \cdots \sum_{x_M \ge 0} \beta_{i,x_1 \ldots x_M} z_1^{x_1} \cdots z_M^{x_M}, \tag{10}$$

and similarly the pgf $\alpha_i(z)$. It is easier to proceed directly in terms of the pgfs, exploiting the independence of services, inter-arrival and switching times. The definition of C_i leads to a relation "across" a switching time:

$$\alpha_i(z) = \beta_{i-1}(z)\tilde{C}_i(z), \tag{11}$$

where

$$\tilde{C}_i(z) = C_i\Big(\sum_{m=1}^{M} \lambda_m(1-z_m)\Big). \tag{12}$$

A similar relation across a stage service is derived as follows. Stage i starts service in state $\boldsymbol{x}$, with the pgf $\alpha_i(z)$, hence its duration has the LST $B_{m_i}^{x_{m_i}}(\cdot)$ where $B_m(\cdot)$ is the LST of a busy-period in queue m initiated by a single customer:

$$B_m(s) = S_m[s+\lambda_m - \lambda_m B_m(s)]. \tag{13}$$

The pgf of the number of arrivals to queue r during this stage is then $B_{m_i}^{x_{m_i}}(\lambda_r - \lambda_r z)$. These arrivals have to be added to the x_r customers present when the stage service started; hence a straightforward if seemingly forbidding summation (Coffman and Hofri, 1982, p. 63) provides the relation

$$\beta_i(z) = \alpha_i(f_i(z)) \equiv \alpha_i(z_1, \cdots, z_{m_i-1}, \eta_i(z), z_{m_i+1}, \cdots, z_M), \tag{14}$$

where

$$\eta_i(z) = B_m\Big(\sum_{\substack{r=1\\ r\neq m_i}}^{M} \lambda_r(1-z_r)\Big). \tag{15}$$

Then from equation (11) we obtain the recurrence relation

$$\beta_i(z) = \beta_{i-1}\big(f_i(z)\big)\tilde{C}_i\big(f_i(z)\big). \tag{16}$$

Pursuing equation (16) is difficult in its present form, so we further define

$$f_i^{(1)}(z) = f_i(z), \qquad f_i^{(l)}(z) = f_{i-l+1}\big(f_i^{(l-1)}(z)\big) \quad (l \geq 2),$$

where the subscripts regress from 1 to I. Applying equation (16) k times yields

$$\beta_i(z) = \beta_{i-k}[f_i^{(k)}(z)] \prod_{l=1}^{k} \tilde{C}_{i-l+1}[f_i^{(l)}(z)]. \tag{17}$$

Eisenberg (1972) showed that $f_i^{(l)}(z) \xrightarrow[l\to\infty]{} \boldsymbol{e}$, where $\boldsymbol{e} = (1, \cdots, 1) \in R^M$, for z in the multidisk $|z_m| \leq 1$, $1 \leq m \leq M$, when $\rho = \Sigma\,\rho_m < 1$; since $\beta_i(\boldsymbol{e}) = 1$, (17) provides the formal solution

$$\beta_i(z) = \prod_{l\geq 1} \tilde{C}_{i-l+1}[f_i^{(l)}(z)]. \tag{18}$$

We comment later on computing numbers from this solution.

In order to compute waiting times in queue m, we need the (marginal) queue length distribution in queue m at service completions. This can be acquired directly from the marginal $\alpha_i(z)|_{z_m=1,\, m\neq m_i}$ (A similar computation was made by Hofri, 1986). However, Eisenberg uses a nice device that merits attention. Consider a time interval $[0, t)$ and random variables over it:

$\omega_i(t;\boldsymbol{x})$ = The number of service beginnings within stage i during $[0, t)$, when the state is $\boldsymbol{x}$ (necessarily $x_{m_i} > 0$).

$\pi_i(t;x)$ = The number of service completions within stage i during $[0, t)$, when the state is x.
$\alpha_i(t;x)$ = The number of stage i service beginnings during $[0, t)$, when the state is x.
$\beta_i(t;x)$ = The number of stage i completions during $[0, t)$, when the state is x.

The dependence of these random variables on the initial state will be disregarded, as we intend to consider only ratios of these variables when $t \to \infty$. Focusing on a particular state x and stage i, the key observation is that the number of service beginnings which do not start a stage service is equal to the number of service completions which do not conclude a stage, i.e.

$$\omega_i(t;x) - \alpha_i(t;x) = \pi_i(t;x) - \beta_i(t;x). \tag{19}$$

Now define

$$\omega(t) = \sum_i \sum_x \omega_i(t;x), \qquad \pi(t) = \sum_i \sum_x \pi_i(t;x),$$

$$\omega_{i,x} = \lim_{t\to\infty}[\omega_i(t;x)/\omega(t)], \qquad \pi_{i,x} = \lim_{t\to\infty}[\pi_i(t;x)/\pi(t)].$$

Note that $\omega_{i,x}$ and $\pi_{i,x}$ are the respective probabilities that service beginnings and completions occur in stage i and find state x. The stage service beginning and ending probabilities $\alpha_{i,x}$ and $\beta_{i,x}$ result from a similar procedure; i.e.

$$\beta_i(t) = \sum_x \beta_i(t;x), \qquad \alpha_i(t) = \sum_x \alpha_i(t;x),$$

and

$$\beta_{i,x} = \lim_{t\to\infty}[\beta_i(t;x)/\beta_i(t)], \qquad \alpha_{i,x} = \lim_{t\to\infty}[\alpha_i(t;x)/\alpha_i(t)].$$

For any t we have

$$|\omega(t) - \pi(t)| \le 1, \quad |\alpha_i(t) - \beta_i(t)| \le 1, \quad |\beta_i(t) - \beta_j(t)| \le 1. \tag{20}$$

Hence, the limiting ratio $\gamma \equiv \lim_{t\to\infty}[\beta_i(t)/\pi(t)]$ is independent of i. Dividing equation (19) by $\pi(t)$, letting $t \to \infty$ and using the last observation we obtain

$$\omega_{i,x} - \gamma\,\alpha_{i,x} = \pi_{i,x} - \gamma\,\beta_{i,x},$$

which in terms of pgfs becomes

$$\gamma\,\alpha_i(z) + \pi_i(z) = \omega_i(z) + \gamma\,\beta_i(z). \tag{21}$$

Since clearly,

$$\pi_i(z) = \omega_i(z)\widetilde{S}_{m_i}(z)/z_{m_i}, \tag{22}$$

equations (11) and (21) relate $\pi_i(z)$ to the pgf at stage completions,

$$\pi_i(z) = \gamma\widetilde{S}_{m_i}(z)\,\frac{\beta_{i-1}(z)\widetilde{C}_i(z) - \beta_i(z)}{z_{m_i} - S_{m_i}(z)}\ . \tag{23}$$

We still need the parameter γ that appears in equation (23). This may be obtained directly from equation (23) by setting $z = e$, but a more elegant way is to observe that the lengths of successive cycles of the server form a derivative Markov process with a limit distribution. If the mean cycle length is D, then the mean number of services per cycle is λD, where $\lambda = \Sigma_m \lambda_m$. Observing that stage i is visited precisely once per cycle, it follows that

$$\gamma = \frac{1}{\lambda D} \,. \tag{24}$$

The mean cycle length is easy to evaluate by mean-value arguments. During each cycle queue m is served for a fraction ρ_m, of the time. We conclude that

$$D = \rho D + \sum_i E(C_i) \;\text{ or }\; D = \sum_i E(C_i)/(1-\rho), \tag{25}$$

where $\rho = \sum_m \rho_m$, and hence that

$$\gamma = \frac{\lambda(1-\rho)}{\Sigma E(C_i)} \;. \tag{26}$$

This completes the specification of $\pi_i(z)$.

Note that in this system arrivals to queue m and departures from queue m see the same distribution of X_m. These two sets of epochs will not normally provide the same distribution for the *other* queues (X_n, $n \neq m$). Nor is it generally true that if $m_i = m = m_j$, $i \neq j$, then $\pi_i(z) = \pi_j(z)$. Thus none of these distributions is equal to the distribution of X_m as seen by arrivals to queue m. Nevertheless, $\pi_i(z)$ gives us the distribution of the waiting time of requests that are serviced during stage i, and such a request leaves behind in queue m precisely those that arrived during its sojourn. Since $\pi_i(z) \equiv \frac{1}{\pi_i(e)} \pi_i(z) \Big|_{z_m = 1,\, m \neq m_i}^{z_{m_i} = z}$ is the marginal pgf for X_{m_i} at departures during stage i, we have $\pi_i(z) = \widetilde{W}_i(z) \widetilde{S}_{m_i}(z)$ or

$$W_i(s) = \pi_i(1 - s/\lambda_{m_i})/S_{m_i}(s). \tag{27}$$

This result does not yet complete the determination of waiting times at queue m, since customers at this queue may be served at any stage i for which $m = m_i$. All that remains is to weight the contributions of these stages according to the fraction of the load they carry, namely, $\pi_i(1)/\sum_{j \mid m_j = m} \pi_j(1)$. Thus the LST of the waiting time in queue m is given by

$$W_{(m)}(s) = \sum_{i \mid m_i = m} \pi_i(1) W_i(s) / \sum_{j \mid m_j = m} \pi_j(1). \tag{28}$$

Eisenberg also discusses briefly a variant regime, usually called *gated service* (arrivals at the queue where the server is located are not served until the server reaches the same queue again). The required differences in the analysis are mild.

Evaluating the moments of queue lengths or waiting times via equations (18), (23), (27) and (28) calls for derivatives of $\beta_i(z)$ with respect to z_{m_i} and $z_{m_{i+1}}$ at $z = e$. Obtaining these directly from equation (18) is not as simple as one would like. Instead, they can be obtained from equation (16), by successive differentiation for every value of i and iterating the resultant equations. The procedure is given explicitly in Coffman and Hofri (1982). It is straightforward and numerically stable, but the iteration calls for a complex program. Recently, Baker and Rubin (1987) have shown, at the price of a slightly more complicated notation, a scheme to compute the expected waiting-times that is considerably simpler to do. It even has a slightly lower computational complexity at $I(I-1)$ equations, rather than the I^2 required in the above scheme.

An element of this model essential to its tractability is the assumption that the itinerary does not depend on the states of the queues. Giving this up is not likely to result in a compact analysis. Hofri (1986), has made a partial generalization: The server does not move when a queue is vacated unless the next queue on the itinerary has a length not less than a given threshold. Thus, a certain reordering of the services is allowed to depend on the states of the queues. Only the simplest case ($M=I=2$) has been treated, and even this results in rather heavy going for general threshold levels (the required numerical computations are particularly daunting).

2.3. Single-server queue with dependent services (Neuts, 1977)

We mentioned earlier that in modeling a secondary storage device, the standard M/G/1 queueing analysis will not normally apply because of the dependence of successive services. Indeed, overlooking this dependence has been the undoing of a few published analyses of such devices. An exact analysis requires that this dependence be taken into account explicitly. The most remarkable analyses of such models were begun in the mid '70s by Marcel F. Neuts and several of his students. Much of the underlying mathematics may be found in (Neuts, 1981). Here we shall present a more specific treatment that has been found useful for modeling secondary storage devices. We follow the development in Neuts (1977), except for two changes. First, we do not treat batch arrivals; giving up this feature entails no sacrifice with our modeling needs, but substantial simplifications accrue in the analysis. Secondly, we assume that, as often happens, the first service in a busy period has a distribution different from that of the remaining services. For the M/G/1 system this elaboration was analyzed by Welch (1964). We shall incorporate it in the following analysis (this was done in part by Coffman and Hofri, 1978).

(a) Model Specification. We consider a single queue with unlimited capacity, fed by Poisson arrivals at rate λ. Services are of m *types*, with transitions governed by the matrix $P = \{p_{ij}\}$, $1 \le i, j \le m$, where p_{ij} is the probability that a service of type j follows a service of type i. Thus successive service types form a Markov Chain. The stationary probability vector (spv) of P is denoted by π.

We let $\hat{A}(x) = \{\hat{a}_{ij}(x)\}$ denote the joint distribution of the type and service time, possibly dependent, which follow a type i service. The matrix $\hat{B}(x) = \{\hat{b}_{ij}(x)\}$ is similarly defined, when the type i and type j services are separated by an idle period. The LSTs of $\hat{A}(x)$ and $\hat{B}(x)$ are denoted by $A(s)$ and $B(s)$, respectively. Note that the dependence between successive services is via their types, not their durations. Thus, we have $P = A(0) = B(0)$. This structure may be used to model a dependence inherent in the service mechanism or, when selection for service depends on the order of arrivals, e.g. FCFS, it may be used to model sources that generate customers of m possible types according to a first-order Markov chain governed by P. For services of customers that follow a type i service we then have the vectors of distribution functions

$$H_i(x) = \sum_{k=1}^{m} \hat{a}_{ik}(x) = (\hat{A}(x)e)_i, \quad 1 \le i \le m,$$

and (29)

$$J_i(x) = \sum_{k=1}^{m} \hat{b}_{ik}(x) = (\hat{B}(x)e)_i, \quad 1 \le i \le m,$$

when the service is within a busy period, or initiates one, respectively.

Note that the distribution functions of a service of type i are given by $S_i(x) = \left(\Sigma_{k=1}^{m} \pi_k \hat{a}_{ki}(x)\right)/\pi_i$ or $S_i^{(init)}(x) = \left(\Sigma_{k=1}^{m} \pi_k \hat{b}_{ki}(x)\right)/\pi_i$, when the service is within a busy period, or initiates one, respectively. We shall normally ascribe type to a customer, but throughout the rest of this section one may replace "a customer of type r" with "a customer which received type r service".

The analysis presented below has two interesting features: the method used to compute boundary probabilities, and the calculation of queue-length and waiting time moments.

For details of the following we refer the reader to Neuts (1977).

(b) Queue length distribution. We start with the analysis of the queue-length distribution at departure epochs, assuming ergodicity (and hence, that P is aperiodic and irreducible). Define $\boldsymbol{x}_i = (x_{i,1}, \cdots, x_{i,m})$, where $x_{i,r}$ is the probability that a departing customer was of type r and left i in the queue. This state will be denoted by (i,r). In Neuts (1977), the same symbol is used but with r denoting the type of the *next* service. Our definition is more natural for our purposes, and the two are trivially related: Neuts' $\boldsymbol{x}_i$ is given by $(\boldsymbol{x}P)_i$ in our notation. Also, let $\hat{a}_{ij,k}(t)$ denote the probability that following a type i customer, a type j customer was served for a duration up to t, and during the service k arrivals occurred. We have

$$\hat{a}_{ij,k}(t) = \int_{x=0}^{t} e^{-\lambda x} \frac{(\lambda x)^k}{k!} \, d\hat{a}_{ij}(x), \tag{30}$$

and similarly,

$$\hat{b}_{ij,k}(t) = \int_{x=0}^{t} e^{-\lambda x} \frac{(\lambda x)^k}{k!} \, d\hat{b}_{ij}(x). \tag{31}$$

The LST's of $\hat{A}_k(x)$ and $\hat{B}_k(x)$ are denoted by $A_k(s)$ and $B_k(s)$; further, define $A_k = A_k(0)$, and $B_k = B_k(0)$, so the following equations obtain for the steady-state queue-length distribution

$$\boldsymbol{x}_i = \boldsymbol{x}_0 B_i + \sum_{k=1}^{i+1} \boldsymbol{x}_k A_{i-k+1}, \qquad i \ge 0. \tag{32}$$

For the vector pgf $\boldsymbol{X}(z) = \sum_{i\ge0} \boldsymbol{x}_i z^i$, equation (32) immediately provides

$$X(z)[zI - A(\lambda - \lambda z)] = x_0[zB(\lambda - \lambda z) - A(\lambda - \lambda z)]. \tag{33}$$

Note that from equation (33), $X(1) = \pi$. Now consider the vector $Y(z)$, that denotes the joint pgf of queue length and service type as seen at a random time (or by an arriving customer). It is interesting to note that $Y(z)$ is *not* equal to $X(z)$, but has the value

$$Y(z) = \left[x_0\, diag(H(\lambda - \lambda z) - zJ(\lambda - \lambda z)) + X(z) diag(e - H(\lambda - \lambda z))\right]/(1 - z). \tag{34}$$

The symbol $diag(\boldsymbol{a})$ stands for a diagonal matrix, with the vector $\boldsymbol{a}$ strung along the main diagonal. The vector x_0 is presented in equation (45) below. Naturally $Y(z)e = X(z)e$, i.e. the same *marginal* queue length distribution is observed at arrivals and departures.

(c) Boundary probabilities – busy-period analysis. The next task is to compute $\{x_{0,r}\}$, the probabilities that a departing customer is of type r and leaves an empty queue. To this avail we consider an auxiliary, 'sparser' Markov chain, L, that effects transitions at queue-emptying epochs, and takes values in $\{1, \cdots, m\}$, which represent the types of departing customers. Clearly, the times when the chain L assumes the state r correspond precisely to occurrences of the state $(0, r)$ for the (queue-length, customer-type) process X. A transition of L corresponds to a busy cycle (an idle period combined with the subsequent busy period) of the X process. Because of the special distribution of the service initiating a busy-period we must first compute a related quantity, the *down-crossing-period* or dcp. This is the time that elapses from the occurrence of the state (k,r), for $k > 0$, until the instant when level $k-1$ is reached for the first time. Clearly, this variable is independent of k, and only regular services (governed by A, rather than B) need be considered. Given that a type i service has terminated, let $G_{ij}(z,s)$ be the conditional joint pgf-LST of the number of services during the following dcp (marked by z), its duration (marked by s) and the probability that the last service will be of type j. The usual "busy-period argument", due to Takács, conditions on events during the first service and shows that the matrix $G(z,s)$ must satisfy

$$G(z,s) = z\sum_{k\geq 0} A_k(s) G^k(z,s). \tag{35}$$

Let $L(z,s)$ be the matrix similar to $G(z,s)$, but based on busy-periods rather than dcp's. Then the same argument yields

$$L(z,s) = z\sum_{k\geq 0} B_k(s) G^k(z,s). \tag{36}$$

Note that the matrix $L(z,0)$ is the pgf of the number of services during a busy-period, and $L(1, s)$ the LST of its duration.

Let μ_i and $\bar{\mu}_i$ denote the mean number of services in a dcp, and its expected duration, given that it starts after a type i service, i.e.:

$$\mu = [\frac{\partial}{\partial z} G(z,s)e]_{z=1,s=0}, \qquad \bar{\mu} = -[\frac{\partial}{\partial s} G(z,s)e]_{z=1,s=0}. \tag{37}$$

We show below that these vectors have the following explicit forms:

$$\begin{aligned} \boldsymbol{\mu} &= (I - G + \overline{G})[I - P + \overline{G} - \lambda diag(\boldsymbol{\alpha})\overline{G}]^{-1} \boldsymbol{e}, \\ \overline{\boldsymbol{\mu}} &= (I - G + \overline{G})[I - P + \overline{G} - \lambda diag(\boldsymbol{\alpha})\overline{G}]^{-1} \boldsymbol{\alpha}, \end{aligned} \tag{38}$$

where

α_i is the mean of the service time distribution H_i in equation (29),
$G = G(1,0)$, is a stochastic matrix,
$\boldsymbol{g}$ is the spv of G

and

$\overline{G}$ is a matrix with all its rows equal to $\boldsymbol{g}$.

Efficient algorithms to compute G are discussed by Neuts (1976); Snyder and Stewart (1983) present acceleration techniques.

We now prove the first result in (38), starting from equation (35). We have

$$\frac{\partial G(z,s)}{\partial z} = \sum_{k \geq 0} A_k(s) G^k(z,s) + z \sum_{k \geq 0} A_k(s) \sum_{i=1}^{k} G^{i-1}(z,s) \frac{\partial G(z,s)}{\partial z} G^{k-i}(z,s).$$

Setting $z = 1$, $s = 0$, multiplying on the right by $\boldsymbol{e}$ and remembering that G is stochastic and $A_k(0) = A_k$, we obtain

$$\boldsymbol{\mu} = \sum_{k \geq 0} A_k \boldsymbol{e} + \sum_{k \geq 0} A_k \sum_{i=1}^{k} G^{i-1} \boldsymbol{\mu}. \tag{39}$$

Denote $\sum_{k \geq 0} A_k \sum_{i=1}^{k} G^{i-1}$ by T, and compute it from

$$T(I - G + \overline{G}) = \sum_{k \geq 0} A_k \sum_{i=1}^{k} G^{i-1}(I - G + \overline{G}),$$

and, since $G\overline{G} = \overline{G}$,

$$T(I - G + \overline{G}) = \sum_{k \geq 0} A_k (I - G^k + k\overline{G}) = P - G + \lambda diag(\boldsymbol{\alpha})\overline{G}.$$

Thus

$$T = (P - G + \lambda diag(\boldsymbol{\alpha})\overline{G})(I - G + \overline{G})^{-1}.$$

Using equation (39) and the stochasticity of P we conclude that

$$\boldsymbol{\mu} = \boldsymbol{e} + (P - G + \lambda diag(\boldsymbol{\alpha})\overline{G})(I - G + \overline{G})^{-1} \boldsymbol{\mu},$$

or

$$\boldsymbol{\mu}[I - (P - G + \lambda diag(\boldsymbol{\alpha})\overline{G})(I - G + \overline{G})^{-1}] = \boldsymbol{e}. \tag{40}$$

After some rearranging we obtain the desired result. The second part of (38) is proved in a similar way.

Corresponding to the matrix G, we have the stochastic matrix L, with the spv $\boldsymbol{l}$. The vectors corresponding to $\boldsymbol{\mu}$, $\overline{\boldsymbol{\mu}}$ for a busy-period are denoted by $\boldsymbol{\mu}^*$, $\overline{\boldsymbol{\mu}}^*$, and defined as in equation (37), with G replaced by L. The new vectors are easy to compute, starting with equation (36):

$$\begin{aligned} \boldsymbol{\mu}^* &= \boldsymbol{e} + [P - L + \lambda diag(\boldsymbol{\beta})\overline{G}][I - G + \overline{G}]^{-1}\boldsymbol{\mu}, \\ \overline{\boldsymbol{\mu}}^* &= \boldsymbol{\beta} + [P - L + \lambda diag(\boldsymbol{\beta})\overline{G}][I - G + \overline{G}]^{-1}\overline{\boldsymbol{\mu}}, \end{aligned} \tag{41}$$

where β_i is the mean of the service time distribution $J_i(x)$ defined in (29).

Useful relations satisfied by the various $\boldsymbol{\mu}$ vectors are:

$$\boldsymbol{\mu g} = 1/(1-\rho), \qquad \overline{\boldsymbol{\mu}}\boldsymbol{g} = \rho/[\lambda(1-\rho)], \qquad \rho = \lambda\boldsymbol{\pi\alpha} \tag{42}$$

$$\boldsymbol{\mu}^*\boldsymbol{l} = (1 - \lambda(\boldsymbol{l\alpha} - \lambda \boldsymbol{l\beta}))/(1-\rho), \qquad \overline{\boldsymbol{\mu}}^*\boldsymbol{l} = \rho(1 - \lambda(\boldsymbol{l\alpha} - \boldsymbol{l\beta}))/[\lambda(1-\rho)] \tag{43}$$

$$\boldsymbol{\mu} - \lambda\overline{\boldsymbol{\mu}} = \boldsymbol{e} = \boldsymbol{\mu}^* - \lambda\overline{\boldsymbol{\mu}}^* \tag{44}$$

We are finally in position to determine $\boldsymbol{x}_0$. Observe that $1/x_{0,r}$ is equal to the mean number of departures between returns of the process $\boldsymbol{X}$ to the state $(0, r)$, whereas $1/l_r$ is the mean number of busy periods between returns to the state r for the chain L. (l_r is component r of the vector $\boldsymbol{l}$). By Theorem 2.11 in Hunter (1969), the mean number of services between these returns is $(\boldsymbol{l\mu}^*)/l_r$. We conclude that

$$x_{0,r} = l_r/(\boldsymbol{l\mu}^*), \quad \text{or} \quad \boldsymbol{x}_0 = \boldsymbol{l}(1-\rho)/(1 - \lambda(\boldsymbol{l\alpha} - \boldsymbol{l\beta})). \tag{45}$$

(d) Moments of the queue lengths. For the derivative of equation (33), evaluated at $z=1$, we obtain

$$\boldsymbol{X}'(1)(I-P) + \boldsymbol{\pi}(I + \lambda A'(0)) = \boldsymbol{x}_0(P - \lambda B'(0) + \lambda A'(0)).$$

Since $I - P$ is singular, but $I - P + \overline{P}$ is not, we add $\boldsymbol{X}'(1)\overline{P} = \boldsymbol{\pi}(\boldsymbol{X}'(1)\boldsymbol{e})$ to both sides to obtain

$$\boldsymbol{X}'(1) = [\boldsymbol{x}_0(P - \lambda B'(0) + \lambda A'(0) - \boldsymbol{\pi}(I + \lambda A'(0))](I - P + \overline{P})^{-1} + \boldsymbol{\pi}(\boldsymbol{X}'(1)\boldsymbol{e}). \tag{46}$$

The quantity $E(X) = \boldsymbol{X}'(1)\boldsymbol{e}$ on the right-hand side seems to need a more circuitous approach. While we limit the following to the first moment, the higher moments can be found in precisely the same way. The expressions promptly become unwieldy, however, so symbolic manipulation by a computer is recommended.

The underlying idea is to examine the eigenvalue with highest absolute value of the matrix $A(s)$ (its Perron-Frobenius eigenvalue), which is denoted by $\delta(s)$ and has right and left eigenvectors $\boldsymbol{u}(s)$ and $\boldsymbol{v}(s)$, i.e.

$$A(s)\boldsymbol{u}(s) = \delta(s)\boldsymbol{u}(s), \qquad \boldsymbol{v}(s)A(s) = \delta(s)\boldsymbol{v}(s), \tag{47}$$

with

$$\boldsymbol{u}(s)\boldsymbol{v}(s) = 1, \qquad \boldsymbol{v}(s)\boldsymbol{e} = 1. \tag{48}$$

Since $A(0) = P$, we have

$$\delta(0) = 1, \quad \boldsymbol{u}(0) = \boldsymbol{e}, \quad \boldsymbol{v}(0) = \boldsymbol{\pi}. \tag{49}$$

The appropriateness of the normalization lies in the ease of computing derivatives of δ, $\boldsymbol{u}$ and $\boldsymbol{v}$ at $s=0$ to any desired order. Again, we proceed only as far as the first moment of X requires. The calculation of higher order moments is quite straightforward (see Neuts, 1977).

Differentiating the equation defining $\boldsymbol{u}(s)$ yields

$$[A'(s) - \delta'(s)I]\boldsymbol{u}(s) + [A(s) - \delta(s)I]\boldsymbol{u}'(s) = 0, \tag{50}$$

and at $s = 0$

$$[A'(0) - I\delta'(0)]\boldsymbol{e} + (P - I)\boldsymbol{u}'(0) = 0. \tag{51}$$

Multiplying equation (50) on the left by $\boldsymbol{v}(s)$ annihilates the second term, so that at $s = 0$,

$$\delta'(0) = \boldsymbol{\pi}A'(0)\boldsymbol{e} = -\boldsymbol{\pi\alpha} \equiv -E(S). \tag{52}$$

From equation (51) we then find for $\boldsymbol{u}'(0)$

$$(I - P)\boldsymbol{u}'(0) = -\boldsymbol{\alpha} + E(S)\boldsymbol{e},$$

or

$$\boldsymbol{u}'(0) = (I - P + \bar{P})^{-1}(-\boldsymbol{\alpha} + E(s)\boldsymbol{e} + \bar{P}\boldsymbol{u}'(0)).$$

However, $\bar{P}\boldsymbol{u}'(0) = (\boldsymbol{\pi u}'(0))\boldsymbol{e}$ and differentiating the relations between $\boldsymbol{u}(s)$ and $\boldsymbol{v}(s)$ at $s = 0$ reveals $\boldsymbol{\pi u}'(0) = 0$ (and $\boldsymbol{v}'(0)\boldsymbol{e} = 0$ as well). Hence

$$\boldsymbol{u}'(0) = (I - P + \bar{P})^{-1}(E(S)\boldsymbol{e} - \boldsymbol{\alpha}). \tag{53}$$

Similarly we derive

$$\boldsymbol{v}'(0) = \boldsymbol{\pi}[A'(0)(I - P + \bar{P})^{-1} + IE(S)]$$

and

$$\boldsymbol{u}''(0) = (I - P + \bar{P})^{-1}[2(A'(0) + IE(S))\boldsymbol{u}'(0) + (A''(0) - \delta''(0)\boldsymbol{e}] - 2(\boldsymbol{u}'(0)\boldsymbol{v}'(0))\boldsymbol{e}.$$

Thus we also need the second derivative of $\delta(s)$ at $s = 0$. We proceed as in the development of equations (50) and (52), with one more differentiation yielding

$$\delta''(0) = E(S^2) + 2\boldsymbol{\pi}A'(0)\boldsymbol{u}'(0), \tag{54}$$

where $E(S^2) = \boldsymbol{\pi}A''(0)\boldsymbol{e}$. Now, $E(X)$ is obtained by multiplying equation (33) on the right by $\boldsymbol{u}(\lambda - \lambda z)$, obtaining

$$\boldsymbol{X}(z)\boldsymbol{u}(\lambda - \lambda z) = \boldsymbol{x}_0 \frac{[zB(\lambda - \lambda z) - I\delta(\lambda - \lambda z)]\boldsymbol{u}(\lambda - \lambda z)}{z - \delta(\lambda - \lambda z)} . \tag{55}$$

Differentiating equation (55) and evaluating at $z = 1$ finally brings at some labor

$$\begin{aligned} E(X) &= \boldsymbol{X}'(1)\boldsymbol{e} \\ &= \boldsymbol{x}_0\big[\lambda^3\delta''(0)(I - P)\boldsymbol{u}'(0) - \lambda^2(1 - \rho)(I - P)\boldsymbol{u}''(0) \\ &+ 2\lambda(1 - \rho)(\rho I - P - \lambda B'(0))\boldsymbol{u}'(0) \\ &+ (1 - \rho)[2\lambda\boldsymbol{\beta} + \lambda^2\boldsymbol{\beta}^{(2)} - \lambda^2\delta''(0)\boldsymbol{e} + \lambda^2\delta''(0)(\boldsymbol{e} + \lambda\boldsymbol{\beta} - \rho\boldsymbol{e})]\big]/2(1 - \rho)^2, \end{aligned} \tag{56}$$

where $\beta_i^{(2)}$ is the second moment derived from the service time distribution $J_i(\cdot)$.

Equation (56) can be simplified, if we note that $(I - P)(I - P + \bar{P})^{-1} = I - \bar{P}$, but there is not much to gain. By juggling in this way equations (33), (47), (48) one can obtain any moment in terms of lower order moments.

(e) Sojourn times. By the remark following equation (33) the marginal sojourn time is given by the LST

$$W(s) = \boldsymbol{X}(1 - s/\lambda)e. \tag{57}$$

Note that the marginal and virtual waiting times have the same distribution. However, when the types are associated with the *customers*, not the service mechanism, the LST of the sojourn time distribution of a type j-customer, $W_j(s) = X_j(1 - s/\lambda)$, is sometimes more informative, especially when the services required by the different types are highly heterogeneous. The computation of moments is simple once those for the queue length are available. The special services at busy-period initiation need no special consideration. Hofri (1984) computed the waiting time for an LCFS regime in a model of similar structure.

2.4. Queueing networks

Networks of queues have enjoyed enormous popularity among performance analysts since the mid '70s. Strictly speaking they are outside the scope of this survey, which focuses on models for isolated devices. We should mention, however, that they have been used extensively to model the performance of entire computing systems, where secondary storage devices, or subsystems, are represented by single nodes. The fine structure of the device operation is not compatible with known, efficient solution techniques for such models. Thus, approximations of the service times and delays in such a node have been used, with the parameters either empirically derived or computed from an isolated device model. Fuller and Baskett (1975) and Zahorjan, Hume and Sevcik (1978) consider interesting examples.

A short, enlightening text-book introduction to these models has been written by Gelenbe and Mitrani (1981). Kelly (1979) provides a much more elaborate discussion. Walrand (1988) provides also recent advances, including much on the control of such networks. Solution techniques are dealt with by Bruell and Balbo (1980) and Sauer and Chandy (1981).

3. Device Models

In this section we review several analyses of device models that are of special interest either from an engineering point of view or a methodological one. The order of presentation is arbitrary, except that the more exotic devices are treated later.

3.1. FIFO service discipline for disks and drums

We shall present specializations of the model described in Section 2.3 for these devices. A specialization is defined by the matrices $\hat{A}(x)$ and $\hat{B}(x)$, or equivalently, the transition probabilities $\{p_{ij}\}$ and the service distributions derived from these matrices.

(a) Parametrizing the FIFO disk. The *type* of a request is identified with the cylinder addressed. The cylinders are numbered 1 through M, and when successive requests are to cylinders i and j, $i \neq j$, a seek T_{ij} is needed. A reasonable estimate of this time is

$$T_{ij} = u + v\,|i-j\,|, \qquad\qquad 1 \le i, j \le M. \tag{58}$$

Typical values for u and v are 10 msec. and 100 μsec/cylinder respectively, and at the level of detail of our model T_{ij} may be taken as deterministic. The durations of rotational latency and data transmission can be lumped when reconnect delays are disregarded. A common practice, which is a reasonable approximation in many systems, is to let their sum equal the time of one full revolution of the disk, denoted by τ. We may then consider this part of the service deterministic as well. The total service is then $S_{ij} = \tau + T_{ij}$.

So far, we have considered services within a busy period. How do we treat a service that initiates a busy-period? This depends on the way the device is managed. A common practice is to leave the arm at rest when the device is not servicing, so that $b_{ij} = a_{ij}$. This approach minimizes system overhead, at the expense of occasionally leaving the arm near an edge of the disk. Such an event increases the expected seek length for the next request. Coffman and Hofri (1978) briefly consider an alternative, according to which the arm is dispatched to a popular, central location at the end of a busy-period.

The matrix driving the address sequence, P, is less obvious. It depends on the way programs address their data, on the number of disk drives in the system, and to some extent on the policy used to allocate areas for files. One usage (common among performance modelers) is to assume each cylinder has an associated probability of being addressed, independently of earlier requests. Thus $p_{ij} = p_j$. This is the assumption used by Coffman and Hofri (1978). Another approach is to recognize the fact that the vast majority of files are accessed sequentially and often allocated in contiguous blocks, and hence addresses tend to cluster in "local runs". There is some empirical evidence for this as well. A first order matrix that reflects this property is given by

$$p_{ij} = \begin{cases} p + (1-p)p_i & i = j, \\ (1-p)p_j & i \neq j, \end{cases} \tag{59}$$

where p is the probability that a request continues a local run. Hofri (1980) considered such a matrix (in the "zero order" approximation, where all p_i's were taken equal to $1/M$). Naturally, the richer the parametrization, the more difficult it is to use symmetry and obtain closed-form expressions for the pgfs and moments of the random variables of interest. Let us apply the results in Section 2.3 to this model, with the assumption $A_{ij} = B_{ij}$ (i.e., the arm rests when a busy-period terminates), so that $S_{ij} = \tau + T_{ij}$ for all requests.

The spv of P is the vector $\boldsymbol{p}$, used in equation (59). The matrices A_k do not display any useful structure, so G and g have to be determined numerically.

The process $\boldsymbol{X}$ has the pgf $\boldsymbol{X}(z)$ satisfying

$$\boldsymbol{X}(z)[zI - A(\lambda - \lambda z)] = \boldsymbol{g}(z-1)A(\lambda - \lambda z)(1 - \rho). \tag{60}$$

From equations (33) and the observation that $\boldsymbol{g} = \boldsymbol{l}$ when $A(s) = B(s)$, the expected service time following a type i customer is

$$\alpha_i = \tau + (1-p)\sum_{\substack{j=1 \\ j\neq i}}^{M} p_j(u + v\,|i-j\,|), \tag{61}$$

which is clearly smaller for centrally located cylinders, *regardless* of the distribution $\{p_i\}$.

The various quantities evaluated in Section 2.3.(d) do not appear to be reducible to any simpler form with this parametrization, but numerical evaluation is straightforward.

(b) Parametrizing the FIFO drum. The drum is assumed to comprise N logical sectors. The number of tracks is left unspecified. The time required to read or write sector i is a constant d_i, and hence the rotational latency is given by

$$t_{ij} = \begin{cases} \displaystyle\sum_{k=i+1}^{j-1} d_k, & 1 \le i < j \le N \\ D - \displaystyle\sum_{k=j}^{i} d_k, & 1 \le j \le i \le N \end{cases} \tag{62}$$

where $D = \Sigma_{k=1}^{N} d_k$. We assume that the physical motion of the drum is the only source of delays in the system; i.e., switching times between the heads of different tracks as well as software delays for interrupt processing and request scheduling are neglected. If the d_i's are all equal the device is called a paging drum, which is treated in the next section under a more natural and effective policy.

The input process is specialized by assuming that an arrival is a request for sector i with probability p_i, independently of previous requests and the state of the queue and server. A curious consequence of this specialization is that the expected value of rotational latency within a busy period depends on the $\{p_i\}$ and $\{d_i\}$, but not on the *relative arrangement* of the sectors. Indeed, within a busy-period

$$E(T) = \sum_{i=1}^{n} p_i \sum_{j=1}^{N} p_j t_{ij} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(t_{ij} + t_{ji})p_i p_j.$$

From equation (62) we find that

$$t_{ij} + t_{ji} = \begin{cases} D - d_i - d_j & i \neq j \\ 2D - 2d_i & i = j, \end{cases}$$

and hence

$$E(T) = \frac{D}{2}\left(1 + \sum_{i=1}^{N} p_i^2\right) - \sum_{i=1}^{N} p_i d_i, \tag{63}$$

in which the invariance is manifest. Coffman and Hofri (1978) have shown that this invariance does not hold when the rotational latency of busy-period initiating requests is taken into account as well.

Adapting the notation of Section 2.3 to this model, we first see that the matrix P is degenerate, with $p_{ij} = p_j$, and hence also $\pi = \boldsymbol{p}$. In addition,

$$A_{ij}(s) = p_j e^{-s(t_{ij} + d_j)}. \tag{64}$$

The evaluation of $B(\cdot)$ calls for a more involved computation; by conditioning on the length of the idle-period we find (Coffman and Hofri, 1978) that

$$B_{ij}(s) = \frac{\lambda p_j e^{-sd_j}}{(\lambda - s)(1 - e^{-\lambda D})} \left\{e^{-\lambda t_{ij}}[e^{-sD} - 1] - e^{-st_{ij}}\, [e^{-\lambda D} - 1]\right\}. \tag{65}$$

Although the form of equation (64) suggests that G might have an explicit closed form, none was found, so the rest of the computations need to be done numerically, according to the procedures outlined in Section 2.3.

3.2. The SLTF drum

Consider again the paging drum defined in the previous subsection, i.e. a drum with equal-sized sectors, $d_i = D/N,\ 1 \le i \le N$, and each accommodates a page of information. An obvious way to reduce latency is to process a queue of requests in the order that the starting addresses of the requested sectors appear at the read/write heads. This ordering changes dynamically as new page requests arrive and old ones depart. For obvious reasons this scheduling rule is called the shortest-latency-time-first (SLTF) policy.

The mathematical model can be represented as in Fig. 4, where for convenience the read/write heads are presented as rotating around the stationary drum surface. As shown in the figure the waiting requests are logically partitioned into queues based on the sectors they address. Each queue is served in FIFO order, with one request being served each drum revolution while the queue is non-empty. We assume that arrivals are governed by a Poisson process at rate $\lambda = \Sigma_{k=1}^{N} \lambda_k$, and that they are filtered so that $\lambda_k,\ 1 \le k \le N$ is the rate of arrivals to the k^{th} sector queue. Under this assumption, the individual sector queues can be analyzed in isolation to obtain queue-length and waiting-time distributions for each queue. While a direct Markov chain analysis is not difficult to work out (see e.g. Coffman, 1969), a simpler approach is to recognize each sector queue as a special case of the server-with-vacations models in Section 2.1 or the more general model of Section 2.3. In particular, consider the extended server-with-vacations model in Section 2.1, where each service period S is followed by an inspection period T. In associating this model with a sector queue of the paging drum, we choose a constant service time D/N (the time spent over sector k), a constant inspection period $(N-1)D/N$ (the time to rotate from the end of sector k back to the start of sector k), and a constant vacation time D (the time between successive visits to the start of sector k). For the generating functions of the number of arrivals during these periods we obtain

$$\widetilde{S}(z) = e^{-\lambda_k(1-z)D/N}, \quad \widetilde{T}(z) = e^{-\lambda_k(1-z)(N-1)D/N}, \quad \widetilde{U}(z) = e^{-\lambda_k(1-z)D}. \tag{66}$$

Replacing $\widetilde{S}(z)$ in equation (2) with $\widetilde{S}(z)\widetilde{T}(z)$ and then substituting the above generating functions gives us the generating functions for the desired queue-length probabilities, and hence the LST for the waiting times via equation (3).

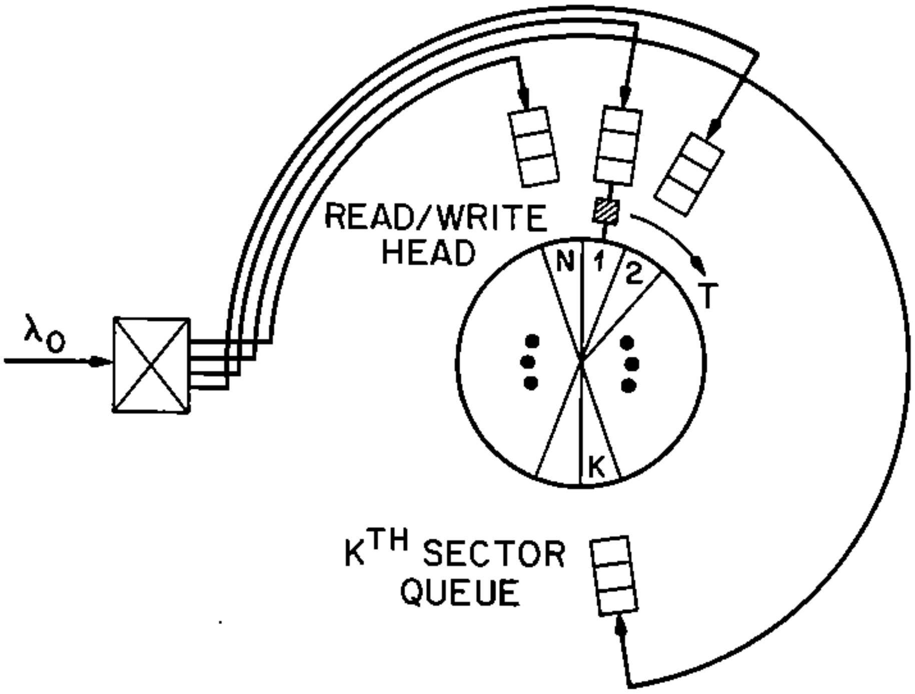

Figure 4: A mathematical model for the paging drum under SLTF

Note that this approach does not provide the joint distribution of the queue-lengths, which are dependent through the service ordering induced by the rotation of the drum.

The paging drum and certain variants of it have been studied by Fuller and Baskett (1975) and others (Gelenbe and Mitrani (1980) provide several references). In a useful generalization of the model, a request is assumed to be for K consecutive pages beginning at a specified sector, where K has a given stationary distribution. The analysis of this model appears to be much more difficult and remains as an open problem. (One may view it as a system of queues with batched arrivals and services.)

A study of SLTF scheduling in the more general setting of a *file drum* can be found in Fuller and Baskett (1975). In this model the pages become files with both starting addresses and lengths considered as random variables. Once again, exact results are elusive even under the simplest of distributional assumptions. Fuller and Baskett (1975) have investigated a number of approximations in considerable detail. In the final section, we return to the file drum in our discussion of open problems.

3.3. The SCANing disk

The SCAN policy for disks is a popular approach to reducing the response time of the device by recognizing its non-random nature. A separate queue is held for requests addressing each cylinder, and each such queue is served exhaustively in (local) FIFO order. A "current direction" is maintained; when a queue empties, the arm seeks to the next cylinder in that direction, until the extreme cylinder is reached or there are no non-empty queues in that direction. The "current-direction" is reversed and the operation continues to cycle. For details about variants of this policy and its relation to others see Denning (1967).

The SCAN policy was analyzed by Coffman and Hofri (1982) under the following simplifying assumptions:

(*i*) The arrivals to each cylinder follow a time-homogeneous Poisson process with rate λ_i (to cylinder i); the total arrival rate is $\lambda = \Sigma\lambda_i$.

(*ii*) When a queue empties the arm seeks to the adjacent cylinder, in the "current-direction". The seek requires a fixed time, a. If that cylinder has no queued requests the next seek is instantly started. The "current direction" is reversed only when cylinder 1 (or M) is reached.

(*iii*) Satisfying a request requires a fixed time, T, equal for all cylinders. This assumption is immaterial for the tractability of the model, which could just as well have a differently distributed random duration T_i at each cylinder.

Under assumptions (*i*) and (*ii*) the model is entirely equivalent to the model described in Section 2.2 with the following specializations:

$I = 2M - 2,$

$m_i = m$ for $i = m$ or $i = 2M - m,\ 1 \le i \le 2M - m,\ 1 \le m \le M,$

$C_i = a.$

All the results in Section 2.2 then translate to the SCAN model. In particular equation (28) reduces to

$$W_{(m)}(s) = \frac{\pi_m(1)W_m(s) + \pi_{2M-m}(1)W_{2M-m}(s)}{\pi_m(1) + \pi_{2M-m}(1)} . \tag{67}$$

A fluid limit of the SCAN model is studied in Coffman and Gilbert (1987), where scanning is called "polling." In terms of the parameters introduced above, let $\tau = (M-1)a$ be the time that the arm spends in motion when crossing from cylinder 1 to cylinder M or vice versa. Let L be the distance then moved and $v = L/\tau$ the constant speed. They consider only the uniform model where $\lambda_i = \lambda/M$, for $1 \le i \le M$. The fluid limit is obtained by holding L, τ (or v), and the traffic intensity $\rho = \lambda T < 1$ constant, while allowing $\lambda, N \to \infty$ and $a, T \to 0$.

In this limit, *work* (required service) arrives deterministically at a rate ρ/L per unit time per unit distance on $[0, L]$. Waiting service requests in this limiting system are described in terms of a continuous function, a density $w(x)$ of work distributed over $0 \le x \le L$. We outline below the derivation of $w(x)$ and a related function $t(x)$ that describes the deterministic (but not constant) motion of the arm.

Define $w(x)$ specifically as the work per unit distance at point $0 \le x \le L$ when the arm is at the origin, and let $t(x)$ denote the time required by the arm to move from the origin to point x (then $t(L)$ just describes half of the cycle with the arm moving from left to right). Moving from x to $x+dx$ the arm encounters the work $w(x)dx$ originally present plus the new work $\rho t(x)dx/L$ that arrived during the motion from 0 to x. One obtains easily

$$\frac{d}{dx}\, t(x) = w(x) + \rho t(x)/L + 1/v\,. \tag{68}$$

For another relation between $t(x)$ and $w(x)$, we consider the reverse motion from L to 0, in which a time $t(x)$ is taken to move from L to $L-x$. We find from the definition of $w(x)$

$$w(x) = \rho[t(L) - t(L-x)]/L\,, \tag{69}$$

so equation (68) becomes

$$\frac{d}{dx}\, t(x) = \rho[t(L) + t(x) - t(L-x)]/L + 1/v\,. \tag{70}$$

Substituting $L-x$ for x changes equation (70) to

$$-\frac{d}{dx}\, t(L-x) = \rho[t(L) + t(L-x) - t(x)]/L + 1/v\,, \tag{71}$$

whereupon combination with equation (70) yields

$$\frac{d}{dx}\,[t(x) - t(L-x)] = 2\rho t(L)/L + 2/v\,,$$

and since $t(0)=0$, integration gives

$$t(x) - t(L-x) = [2\rho t(L)/L + 2/v]x - t(L)\,. \tag{72}$$

We see that $t(L)=\tau/(1-\rho)$, so substitution of equation (72) into equation (70) yields a first-order differential equation with the solution

$$t(x) = \frac{x}{v}\Big(1 + \frac{x\rho}{L(1-\rho)}\Big). \tag{73}$$

Then equations (69) and (73) give

$$w(x) = \frac{x\rho}{Lv(1-\rho)}\Big(1 + \rho - \frac{\rho x}{L}\Big). \tag{74}$$

According to equations (73) and (74), we see that the arm starting at position 0 sees work density $w(0)=0$ and moves at speed v. As it travels it encounters an increasing work density and its speed decreases. The total work present when the arm is at y and moving right is easily shown to be

$$\frac{\rho\tau}{1-\rho}\,\Big(\frac{1}{2} + \frac{\rho}{6} + (1-\rho)\frac{y}{L}(1 - \frac{y}{L})\Big)$$

which reaches a maximum of $\rho\tau(3/4 - \rho/12)/(1-\rho)$ at $y=L/2$ and a minimum of $\rho\tau(1/2 + \rho/6)/(1-\rho)$ at $y=0$ and $y=L$.

Expected waiting times are also easy to derive from (73). In particular, at a random point in time let $W(y)$ be the waiting time up to the arm's first return to point y. Then

$$W(y) = \frac{\tau}{2(1-\rho)}\,[1 + (1 - \frac{2y}{L})^2],$$

so the mean wait for service at a random point y, distributed uniformly over $[0, L]$, is

$$W = \int_0^L W(y)dy/L = \frac{2\tau}{3(1-\rho)}.$$

We remark that the fluid limit can be used as an effective approximation for any traffic intensity $\rho = \lambda T$, but for fixed parameters L, τ, and ρ it requires the rate of arrivals to be relatively high and the service times to be correspondingly small. A numerical study of the approximation can be found in Coffman and Gilbert (1987); see figure 10 there.

3.4. Disks with two read/write heads

In the few studies dealing with two-head systems, the difficult mathematical questions have focused on the calculation of seek time distributions under various head selection policies, i.e. rules which select for each request the head to perform the seek and read/write operation. As illustrated below, a variety of problems has emerged from the need to incorporate one or more physical constraints. Since performance in terms of queue lengths, waiting times and related measures has yet to be analyzed for two-head systems, this research takes us somewhat afield of classical queueing applications. Thus, we merely outline the basic models and results. It will be evident that the results mentioned below can be used, at least in principle, as a description of service times within the general queueing model of Section 2.3.

The assumption of a single data path between the disk system and primary memory is common to all models. The set of cylinder addresses is approximated by the unit interval [0,1]. Requests are modeled by a sequence of addresses $t_1, t_2, \cdots$, usually assumed to be independently and uniformly distributed over [0,1]. Each request must be served, in the order given, without the benefit of advance information on subsequent requests. The state of the system just after serving the i^{th} request is given by a pair (x_i, y_i), $0 \le x_i \le y_i \le 1$, denoting the positions of the two heads. Necessarily, either $x_i = t_i$ or $y_i = t_i$. With an initial state (x_o, y_o) given, the sequence $\{(x_i, y_i);\ i \ge 0\}$ constitutes a Markov chain in all cases of interest.

The models below are distinguished by whether there are two heads on the same arm or one on each of two arms. In the latter case two further models arise depending on whether the two arms are positioned by a single controller, or by two autonomous controllers. In all models the *nearer-server* (or *greedy*) rule is clearly of interest: if in state (x_i, y_i) both heads can be positioned at t_{i+1}, then the head at x_i is chosen for the seek if $|t_{i+1} - x_i| < |t_{i+1} - y_i|$; otherwise, the head at y_i is chosen.

(a) Two arms autonomously controlled. This is the most flexible system, since both heads can be moved concurrently to any pair of addresses in [0,1]. In serving a request one head performs the seek while the other *jockeys* so as to be in a favorable position for the next request. As shown by Hofri (1983), an optimal policy is relatively easy to find, provided the cost function to be minimized is limited to total or average seek time, i.e. no cost is incurred

by the jockeying motion. Indeed, it is not difficult to show that a sequence of locally optimum decisions is also globally optimum. In particular, the nearer head is always chosen for the seek, i.e. – the greedy rule is optimal. During a seek to point t the jockeying head is positioned at $t/3$ if $t \geq 1/2$ and at $1-(1-t)/3$ if $t < 1/2$; a calculation shows that this minimizes the expected seek to the next request. The same policy is optimal even when successive requests are dependent and not uniformly distributed, but the expression for the jockeying target is more involved.

(b) Two arms and a single controller. In this system the heads are moved independently but not concurrently; in serving a request only one head is moved. The minimization of expected seek times remains an open problem for this system, but it is conjectured that a head selection rule of threshold type is optimal, i.e. for each state (x_i,y_i) there is a threshold $\lambda(x_i,y_i)$ such that if $t_{i+1} < \lambda(x_i,y_i)$ the head at x_i serves the request at t_{i+1}, and if $t_{i+1} > \lambda(x_i,y_i)$ the head at y_i serves the request. By a combination of analysis and numerical evaluation of Bellman equations Calderbank, Coffman and Flatto (1985) compared an optimal policy with the nearer-server rule and found that expected seek times under the latter are very nearly minimum.

(c) A single arm with two fixed heads. In serving a request both heads move at a fixed distance, d, apart, and for each request one is selected to perform the read/write operation. Such a configuration was first investigated by Page and Wood (1981), who based their simulation study on actual systems. Calderbank, Coffman and Flatto (1984) analyzed the nearer-server rule under two assumptions: In the first model both heads had to be kept on the disk surface (i.e. in [0,1]). Thus, the left and right heads were restricted to $[0, d]$ and $[1-d, 1]$, respectively, and the nearer-server criterion was applied only to requests in $[d, 1-d]$. In the second model, whose analysis was very similar, there were no restrictions on head positioning other than the fixed separation distance. Explicit forms were obtained for the stationary distribution of the position of the left (and hence right) head. From these results expected seek times as functions of d were obtained. Interestingly, it was shown that an optimization with respect to d produced an expected seek time which was slightly less than that in the system with two independent arms and a single controller.

In each of the above systems the interval [0,1] can be divided in half with a head reserved exclusively for serving requests in each half. This clearly doubles the performance (halves the expected seek times) of single-head systems. An interesting result of all of the studies is that when the (implicit) interior boundary between the heads is removed, the expected seek time can be reduced to less than one half the expected seek time in a single-head system. (Two heads are *better* than twice as good as one head.)

As a final remark, we mention the extension of the above models to systems in which one or more directories are kept on the disk. In such a model t_i denotes a logical address (such as a file name). To convert this to a disk address an initial read must be done at one of the fixed directory locations on the disk. Thus, in general, the servicing of requests consists of alternating directory and file operations. Within the earlier mathematical models, Calderbank, Coffman and Flatto (1988) have analyzed the expected total seek time per request under an optimal head selection policy.

3.5. Controlling a CCD shift register

(Hofri and Rosberg, 1985). In this section we describe a mathematical model and its analysis for a different device. The treatment is also different, in that within the technological constraints we can determine an optimal operating policy for the device.

(a) Device Description. The device considered here is a register that may be viewed as one track on the surface of a disk or a drum. The technology however is entirely different (see Matick, 1977): the information is not magnetically recorded but is carried by electrical charges that rotate continuously at a rate u that is bounded above and below by physical requirements, $a \le u \le b$. We note that the ratio $r = b/a$ is fairly large and can exceed 10^4. The register is L bits long. It may be read or written through a single "port", an electric connection represented by the point P in Fig. 5. Reading or writing commence once a special portion of the contents (the register "signature") reaches the port. We shall denote this portion by "bit 0". In our model we do not distinguish between read and write requests, and we assume that they all require the entire contents of the register to be shifted across the port. (The complete system will typically have numerous such registers in parallel, but their operation is independent except for memory contention. Our model will consider a single register in isolation.)

Request arrivals are assumed to be triggered by a time-homogeneous Poisson process with rate λ. A request that arrives and finds the register busy servicing another is enqueued. Otherwise, the start of service awaits the arrival of bit 0 to P. If it arrives when bit 0 is at distance s from the port, its waiting time will be minimized if this portion of the cycle is completed at the

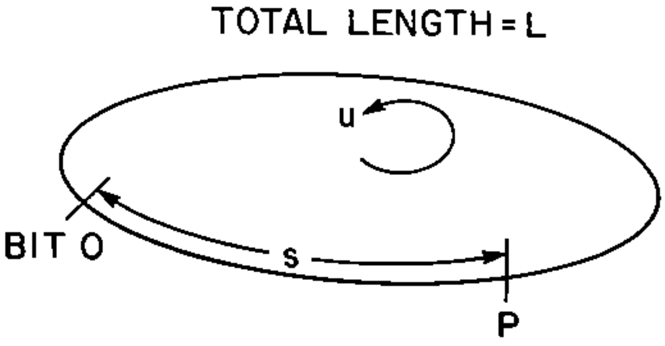

Figure 5: The geometry of a CCD shift register.

maximum rate, b. Thus far the model resembles the one discussed in Section 2.1. However, we have not yet specified the service or the "vacation" distribution. Since the rotation rate u can be fixed by the device controller, Hofri and Rosberg (1985) address the questions of determining the optimal rate during service and during a "vacation" a rotation that starts when there are no queued requests.

(b) Optimal Service Policy. First, we note that it is not obvious that service should be invariably at the maximum rate. Consider a realization where at $t=0$ there is one queued request, bit 0 is at P and its service commences. The next request, in this realization, is due to arrive at $t=L/b+\varepsilon$. Service at rate b would have the next request wait almost an entire extra rotation, whereas a first service at rate $\approx Lb/(L+\varepsilon b)$ would slightly delay the first, halve the time in system for the second, and so provide a much shorter aggregate sojourn time. Nevertheless, it is shown by Hofri and Rosberg (1985) that the service *should* be at the constant rate b to minimize overall *expected* sojourn time.

The proof assumes that rate changes are instantaneous (which is reasonable at the level of detail of the model) and are effected at N equidistant points on the cycle of bit 0, for some integer N (which may be arbitrarily large).

Let $\Delta = L/N$. A policy that makes decisions at these points is called a Δ-policy. Under an optimal Δ-policy let $W_\Delta(i, s)$, $s \geq 0$, $i \geq 1$ denote the expected aggregate sojourn time experienced by requests in the system, beginning at time $t = 0$ and extending to the end of the current busy-period, given that at $t = 0$, bit 0 was at position s and i requests were present.

Let $W_\Delta(i, s, u)$ be similar to $W_\Delta(i, s)$, except that the action for the first Δ segment is known to be u (not necessarily optimal), and thereafter the optimal policy is to apply. An immediate calculation yields

$$\begin{aligned} W_\Delta(i, s, u) &= i\frac{\Delta}{u} + \int_0^{\Delta/u} \lambda e^{-\lambda t}\Big[\frac{\Delta}{u} - t + W_\Delta(i+1, s-\Delta)\Big]dt + e^{-\lambda\Delta/u} W_\Delta(i, s-\Delta) \\ &= (i+1)\frac{\Delta}{u} - \frac{1}{\lambda} + (1 - e^{-\lambda\Delta/u}) W_\Delta(i+1, s-\Delta) + e^{-\lambda\Delta/u}\Big[\frac{1}{\lambda} + W_\Delta(i, s-\Delta)\Big]. \end{aligned} \tag{75}$$

Noting that $W_\Delta(j, s-\Delta)$ does not depend on u, we differentiate equation (75) with respect to u:

$$\begin{aligned} &\frac{\partial}{\partial u} W_\Delta(i, s, u) = \\ &\quad -(i+1)\frac{\Delta}{u^2} + \frac{\Delta}{u^2} e^{-\lambda\Delta/u} - \frac{\lambda\Delta}{u^2} e^{-\lambda\Delta/u}[W_\Delta(i+1, s-\Delta) - W_\Delta(i, s-\Delta)]. \end{aligned} \tag{76}$$

Since the value in brackets is positive the right-hand-side of equation (76) is negative for $i \geq 0$ and $\lambda \geq 0$. This establishes the claim.

(c) Optimal Vacation Policy. At first glance it appears that the register should be left to rotate at rate b when there are no queued requests. This, however, is not the case. Since a request which arrives at a non-busy register and finds bit 0 at position s experiences a latency s/b, we want to minimize s at "interception time". It appears that a more reasonable policy would be

to shift fast (at b) when s is large, and then slow down, to tarry as long as possible at low s values, when bit 0 approaches the port. Indeed, Hofri and Rosberg prove the following:

(*i*) When control is effected every Δ time units, the optimal rate is b so long as $s > s_1$; it is a when $s < s_1$; and it may assume an intermediate value at the *single* decision point s_1. There is an algorithm to compute s_1 (and that intermediate value), which essentially amounts to solving a set of Bellman equations. It is rather simple because the form of the optimal policy is known. In actual computations the "intermediate" optimal value turns out to be very often either a or b, so the optimal policy is of a bang-bang type.

(*ii*) When control can be applied continuously at every s value, then the optimal policy is always pure bang-bang, assigning rate b for $s > s^*$, and rate a for $s \le s^*$, where s^* is determined through the equation

$$b\left(e^{-\lambda(L-s^*)/b} - e^{\lambda s^*/a}\right) + (b-a)\left(e^{\lambda s^*/a}-1\right) + \lambda L = 0.$$

For low arrival rates one may neglect terms of order $\left(\frac{\lambda}{a/L}\right)^3$. Note that $\lambda L < b$ is required for stability. One then finds

$$s^* \approx \frac{L}{1+\sqrt{r}}, \qquad r \equiv b/a. \tag{77}$$

It is interesting that the ratio of the optimal expected latency time to the value $L/2b$, obtained under the naive policy which maintains the rate b throughout, is given by $\frac{2}{1+\sqrt{r}}$. Since r can easily be in the thousands this is a substantial improvement.

(d) Performance analysis. Having determined the optimal service and idle period behavior the remaining analysis is routine. Note that the model is close, but not identical, to the model presented in Section 2.1, since the first arrival during a vacation will modify its duration. Equation (2) holds, but $\tilde{U}(z)$ should be assigned the value $zU_1(\lambda(1-z)/b)$, where $U_1(\cdot)$ here is the LST for the time between a first arrival during an "idle" rotation and the completion of that rotation. Let U denote the duration of an uninterrupted idle rotation. We have then $U = (L-s^*)/b + s^*/a$. The time of a first arrival during an idle-period, T, relative to the time when the rotation during which it occurred started, has a truncated exponential distribution (the truncation is at U); for a time-of-interception $T=t$ the position of bit 0 is given by

$$s(t) = \begin{cases} L - bt, & 0 \le t < t^* \equiv (L-s^*)/b \\ s^* - a(t-t^*), & t^* < t \le U. \end{cases}$$

The distribution of T induces one for $s(T)$. Since $U_1 = s(T)/b$ the computation of its LST is immediate, yielding

$$U_1(\xi/b) = \frac{1}{1-e^{-\lambda U}} \left\{ \frac{\lambda}{\lambda-\xi}\, e^{-L\xi/b} \left[1 - e^{-t^*(\lambda-\xi)} \right] \right. \\ \left. + \frac{\lambda}{\lambda-\xi a/b}\, e^{-\xi U a/b} \left[e^{-t^*(\lambda-\xi a/b)} - e^{-U(\lambda-\xi a/b)} \right] \right\}. \tag{78}$$

The expression for $U_1(\cdot)$ together with equations (2) and (3) allows us to compute moments of the response time.

3.6. Multidisk Systems

It is obvious that the analyses of individual device models do not cover the entire spectrum of congestion phenomena that occur when several storage modules operate concurrently. We have already mentioned one such phenomenon, the reconnect delay, but there are others as well.

The reconnect delay has been recently analyzed in detail, albeit only approximately, by Gavish and Sumita (1988). They consider a model consisting of two channels and two strings of disks. Their main contribution is in showing that a careful consideration of channel interference permits the model of Section 2.3 to provide a very good estimate of system performance.

The total sojourn time of a request for disk j in the subsystem is split into the following successive components:

W_j – waiting time in the queue;
C – time to send the seek command;
S_j – seek time;
W_j^c – waiting time for the channel to become free;
C – time to send the set sector/read command;
L_j – rotational latency time;
R_j^* – rotational delay (integral number of rotations);
H_j – time to find the head of the block within the sector;
T_j – time to transfer the block.

The command times, C, are constant. The seek times follow equation (58). Seek targets in their model obey the scheme of Hofri (1980) (each cylinder has a possibly different locality parameter; otherwise the address distribution is uniform). L_j and H_j are obtained from uniformity assumptions. The two deliberate approximations in the model involve the components W^c and R^*. W^c is estimated as the residual life-time of the channel-blocking duration, due to the other disks. This duration is computed in a state-independent way by a simple mixing of channel-use durations of the command and transfer times of the other disks, when they hold the channel, weighted by their traffic intensities. The variable R^* is assumed to be a geometrically distributed number of disk rotations; the parameter of the distributions is the channel utilization, which is obtained when W^c is computed.

In computing the waiting time W_j they recognize that a request arriving to an empty queue may be delayed from initiating the seek operation, due to the channel being busy with a different disk. Hence they adopt the approach of the model described in Section 2.1, and use the decomposition of the waiting time given by equation (9).

Their results were compared with a straightforward simulation of the model, to estimate the effect of the approximations inherent in it. The comparison revealed that the errors in mean values—device utilizations and expected response times—are usually bound by 4%, but may reach 8-9% at close to saturating loads.

It is of interest to note that the model was easy to adapt to reflect technology changes, such as using two arms per disk (that can only access *non-overlapping* regions on the disk surface) and multiple controllers with various access capabilities.

In dealing with even more ambitious models, techniques other than those we have used so far need to be invoked. For an interesting, well-considered approach see Bard (1981), which also contains a number of additional references.

4. Discussion and Open Problems

There are open problems of many types in the analysis of computer secondary storage systems. In this section we mention a few representative ones; in the literature the reader will find many more that are concerned with specific devices.

The idealizations introduced in the various models comprise the most obvious source of open problems. The tritest of these is the assumption of time-homogeneous exponential distributions for interarrival times, read/write times, etc. The value of these assumptions in producing tractable Markov processes is easy to see in virtually all cases. The sacrifices made are not so well understood; it is hoped but rarely proved that expected-value performance measures are sensitive only to the first one or two moments of the constituent probability distributions taken as exponential.

What is likely to introduce even rougher approximations is the frequent assumption of the independence of two or more random variables. Such an assumption usually provides a critical reduction in the dimensionality of a Markov model. An important example is the assumed independence of arrival rates and the number in queue. In many applications the number of active users of a storage device is simply too small for the Poisson assumption. One of the few concessions to this reality appears in the analysis of the SLTF file drum mentioned in Section 3. Fuller and Baskett (1975) analyze a closed two-server cyclic queue, where one server is a CPU and the other is a drum with a state-dependent service mechanism reflecting SLTF scheduling. Exponential service times with a rate parameter depending on queue length are assumed, so an analysis along conventional lines is possible. An added benefit of this simplification is that such storage models can be taken as elements (stations) in very general product-form queueing networks. Recent work by Mitra and McKenna (1984) provides efficient computational tools for analyzing networks with state dependent service times and multiple job classes.

The above model, however, represents another type of approximation involving the assumption of independence, viz. that a service time (more specifically, a latency delay in their case) depends only on the current state of the system. We return to this point later.

A third illustration of questionable independence assumptions was mentioned in Section 3. In the commonly adopted independent reference model, request addresses are taken as independent, when in fact a great deal of "locality" is often exhibited by these sequences in practice. Even a very simple model of locality, such as the one presented in Section 3.1, can make important improvements in the value of the mathematical analysis.

Thus far we have considered modeling issues and related open problems which arise in a very broad range of computer performance evaluation studies. In the present setting the more interesting open problems are those dealing with structures and algorithms peculiar to secondary storage devices. Two of these are presented next, one having to do with latency minimization and the other with seek-time minimization. In order to bring out clearly the essence of these problems, we shall adopt simplistic models, continuous in both time and space.

Consider first a model of the SLTF file drum. As shown in Fig. 4., assume that the server (read/write head) moves about a circle (the drum) at constant speed, rather than fixing the server and rotating the circle. Let us normalize the circumference of the circle to 1 and assume that arrivals constitute a Poisson process in the two dimensions of time and space, i.e. the probability of an arrival in $[t, t+dt] \times [x, x+dx]$ is $\lambda dtdx$. The arrivals represent starting addresses of files to be read or written.

Now according to the SLTF policy suppose the server has just completed the service of a request and is at position x. The next request served, say at y, is the first request encountered by the server in its constant circular motion starting at x. Beginning at the time the server is at point y, an exponentially distributed service period with parameter μ commences. From the point where the server is located at the end of this service period, it then moves to the next request as before and starts the next service.

Waiting times in this model were analyzed approximately by Fuller and Baskett (1975) under the assumption that on completion of a service the waiting requests (starting addresses) were distributed uniformly around the circle. It is not difficult to verify heuristically that this is not the case and that in fact, according to the actual distribution, the latency (motion to the next request) can be expected to be stochastically larger. Thus, even a conservative approximation to waiting times (i.e. one which provides a reasonable upper bound for the waiting times) under the SLTF policy remains an open problem.

A satisfactory analysis of SSTF sequencing, to obtain the request waiting time and a description of the server trajectory, is another intriguing open problem, even with a time-homogeneous arrival process. The problem resembles certain polling problems defined on the circle rather than the interval. In particular, if the server is constrained to move unidirectionally at a constant speed around a circle when not serving, we have a continuous polling model successfully analyzed by Coffman and Gilbert (1986). There, the assumption of constant service times was crucial to the formulation of a tractable Markov process.

The arm moves, however, on an interval. Using the simplified continuous model as before, we represent the set of cylinder addresses by the unit interval [0,1]. Arriving cylinder requests are again described by a Poisson process in two dimensions, just as for the SLTF model. In the simplest model we assume constant service times. After serving a request at point x, the head is moved to a waiting request nearest to x for its next service. The invariant measure describing the equilibrium server position, waiting times as a function of position and expected server motion per request are all important objectives of the analysis. A simulation study conducted by Hofri (1980) led to the following specific conjecture: The waiting time under a FCFS policy is *stochastically* larger than the waiting time under SSTF.

In special cases results of this sort have been proved. For example, in the absence of seek times (i.e. we have instantaneous transitions), the sum of queue lengths is invariant under the choice of a strategy among those that do not idle the disk when non-empty queues are present, assuming equal service times at each cylinder. When these times are not equal, Klimov (1974) gives a head-of-the-line priority rule by which the queues should be served (non-exhaustively) so as to minimize mean waiting time. Naturally, when switching incurs no cost, the optimal ordering depends on arrival and service characteristics only, and not on queue and server states.

When the seek times are non-zero, exhaustive service is optimal in minimizing the total queue length (Ross, 1985), and under certain combinations of parameters it appears to be optimal not to depart from an empty queue until the next to be served has a queue larger than some positive threshold. The latter was shown to be the case for two queues by Hofri (1986). The last few years have seen an increase in the number of works on optimal control of queues; we have yet to see an application of any of those results to operating systems.

Recently it was observed by Daniel and Geist (1983) (and elaborated in Geist and Daniel, 1987), partly from simulation data, and partly from measurements (in a lightly loaded system) that a policy intermediate between SCAN and SSTF, provides lower mean waiting times than both do. This policy, termed V-SCAN, computes seek distances and then 'seeks' to the closest non-empty cylinder, as SSTF does. However, the distance in front of the head (defined as the direction of its last motion) is compared to the corresponding distance behind the head plus a prespecified constant R. This yields the SSTF and SCAN rules, when R equals 0 and N—the number of cylinders—respectively. Simulations with $R \approx 0.2N$ showed that in the region of moderate arrival rates, mean waiting times under V-SCAN were a few percent lower than the minimum of those under SSTF and SCAN. The data also indicated that at high arrival rates SCAN was superior to SSTF. This has recently been confirmed in independent extensive simulation by Coffman and Gilbert, (1987); their data suggest that for all sufficiently large traffic intensities, SSTF is inferior to SCAN, but that they converge to the same heavy-traffic limit. A number of analytical results (described above in Section 3.3) for the simpler SCAN policy on both the circle and the interval support the heavy-traffic behavior seen in the simulation data.

Gopal and Rosberg (1986) take a different approach to the problem. They propose an algorithm to compute the scheduling of a given number of requests when there are no subsequent arrivals, prove its optimality and compare its performance (via simulation) with the standard scheduling policies. On the basis of this algorithm they propose QOPT, a quasi-optimal schedule when arrivals are allowed: after completing each request, on the basis of the requests then present, it finds the next request the optimal algorithm (with no more arrivals) would have served, and 'seeks' there. The required computation is quadratic in the number of requests (which is small in all reasonable situations). In simulation experiments this algorithm outperformed SSTF and SCAN for all arrival rates, in terms of waiting times. The differences exceeded a fraction of a percent only at the very highest rates.

An area of queueing theory of high engineering interest, but difficult to pursue for the problems we considered here, concerns transient, or time-dependent analysis. Going beyond the conventional steady-state formulation is especially important when we want to assess the "surge behavior" of a system and the rate at which it dissipates backlogs of requests which accumulate during periods when the arrival rate temporarily far exceeds the processing rate. When the service mechanism is independent of the state of the queue (as is the case in standard queueing models such as the M/G/1 queue), or when the dependence is easy to take into account, as is the case in the model displayed in Section 2.3, a busy-period or dcp analysis provides the answer. The treatment of the relaxation time of the M/G/1 model with vacations by Keilson and Servi (1987), provides this information for a more useful model. When the dependence is more contrived, as is often the case with the devices and policies we face here, results are much harder to come by. Consider for example a disk under the SSTF policy. Obviously the larger the backlog, the rarer *and shorter* are the seek motions the arm will have to perform. Put another way: the service improves as the state of the subsystem deteriorates. From an engineering point of view this is a definite advantage, but it does not make for an easy analysis. Some interesting results along these lines have recently been obtained for Jackson-type queueing networks by Massey (1984), but extending them to the models we have discussed may well be quite difficult.

Recently disk systems have been provided with a new feature: a cache—a solid-state device—that stores the recent blocks read or written to the disks. Accessing the cache is much faster than the disks proper, since there is no mechanical motion involved; when the hit-ratio (fraction of the requests satisfied from the cache) is high, as has often been observed, a very substantial speed-up results. No satisfactory analysis of such a system, e.g., an analysis that would provide recommendations on the desired cache management policy, or changes in the disk scheduling algorithms, is available yet.

There are many stochastic optimization questions closely related to the queueing models we have discussed. In very general terms a typical statement will take the form: Among a given class of policies for deciding the sequence in which requests are to be served, find one which minimizes the expected waiting time (or the expected number in system, expected head/drum motion per request, etc.). The case shown in Section 3.5 is a nice exception, but such problems are usually very difficult, even when attention is restricted to the class of stationary Markovian policies (Ross, l983) which base decisions only on the current state of the system.

While in most cases the analysis is quite different from that found in the theory of queues, a queueing analysis may be an integral part of a problem in optimal allocation. For example, consider a general FCFS drum model (as in Section 3.1) whose parameters include the sector access probabilities $\{p_i\}$. The p_i's are determined by the way records are assigned to sectors and, to some extent at least, are under the control of the designer. A reasonable objective would be to distribute a set of records with known access frequencies among the sectors so as to produce a distribution $\{p_i\}$ which minimizes expected waiting time.

A direct solution to this problem begins with a queueing analysis to obtain a formula expressing the expected waiting time as a function of $\{p_i\}$. After finding a set of values for the p_i's which minimizes this function, we have the essentially combinatorial problem of partitioning the records among the sectors so as to obtain the desired distribution $\{p_i\}$. Of course, similar problems can be formulated for disk systems, where cylinders play the role of sectors.

References

Aven O.I., Coffman E.G. Jr., Kogan Y.A. (1987): *Stochastic Analysis of Computer Storage.* D. Reidel Publ. Co., Dordrecht, The Netherlands.

Baker, J.E., Rubin, I. (1987): "Polling with a General Service-Order Table", *IEEE Trans. Comm.* **COM-35,** #3, 283-288.

Bard, Y. (1981): "A Simple Approach to System Modelling", *Performance Evaluation,* **1,** 225-248.

Bhandarkar, D.P. (1975):"On the Performance of Magnetic Bubble Memories in Computer Systems", *IEEE Trans. Comp.,* **C-24,** #11, 1125-1129.

Bruell, S.C., Balbo, G. (1980): *Computational Algorithms for Closed Queueing Networks,* Elsevier North-Holland, New York.

Calderbank, A.R., Coffman E.G. Jr., Flatto L. (1984): "Optimum Head Separation in a Disk with Two Read/Write Heads", *J. Assoc. Comput. Mach.* **31,** 826-838.

Calderbank, A.R., Coffman E.G. Jr., Flatto L. (1985): "Two-Server Sequencing Problems", *Math. Oper. Res.* **10,** 585-598.

Calderbank, A.R., Coffman E.G. Jr., Flatto L. (1988): "Optimal Placement of Directories on a Computer Disk", *J. Assoc. Comput. Mach.* **35,** 433-446.

Coffman, E.G. Jr. (1969): "Analysis of a Drum Input/Output Queue under Scheduled Operation in a Paged Computer System" *J. Assoc. Comput. Mach.* **16,** 73-90. *Corrigendum:* **16,** 646.

Coffman, E.G. Jr., Gilbert, E.N. (1986): "A Continuous Polling System with Constant Service", *IEEE Trans. Infor. Theory* **IT-32,** 584-591.

Coffman, E.G. Jr., Gilbert, E.N. (1987): "Polling and Greedy Servers on a Line", *Queueing*

Syst. **2,** #2, 115-145.

Coffman, E.G. Jr., M. Hofri (1978): "A Class of FIFO Queues Arising in Computer Systems", *Oper. Res.,* **25,** #5, 864-880.

Coffman, E.G. Jr., M. Hofri (1982): "On the Expected Performance of Scanning Disks", *SIAM J. Comput.* **11,** #1, 60-70.

Daniel, S., Geist, R. (1983): "V-SCAN: An Adaptive Disk Scheduling Algorithm". *Proc. of the IEEE Int. Symp. on Comp. Sys. Org.*, New Orleans 3/1983.

Denning, P.J. (1967): "Effects of Scheduling on File Memory Operations", *Proc. AFIPS, SJCC,* **31,** 9-21.

Doshi, B.T. (1986): "An M/G/1 Queue with Variable Vacations". In N. Abu El Ata (Ed.) *Modelling Techniques and Tools for Perf. Analysis.* pp 67-81, Elsevier, Amsterdam, The Netherlands.

Doshi, B.T. (1986): "Queueing Systems with Vacations A Survey", *Queueing Systems,* **1,** #1, 29-66.

Eisenberg, M. (1972): "Queues with Periodic Service and Changeover Time", *Oper. Res.,* **20,** 440-451.

Fuhrmann, S.W., Cooper, R.B. (1985): "Stochastic Decompositions in the M/G/1 Queue with Generalized Vacations", *Oper. Res.,* **33,** 1117-1129.

Fuller, S.H., Baskett, F. (1975): "An Analysis of Drum Storage Units", *J. Assoc. Comput. Mach.,* **22,** #1, 83-105.

Gavish, B., Sumita U. (1988): "Analysis of Channel and Disk Subsystems in Computer Systems". *Queueing Syst.* **3,** #1, 1-24.

Geist, R., Daniel, S. (1987): "A Continuum of Disk Scheduling Algorithms", *ACM Trans. Comput. Syst.* **5,** #1, 77-92.

Gelenbe, E., Mitrani, I. (1980): *Analysis and Synthesis of Computer Systems*, Academic Press, London.

Gopal, I.S., Rosberg Z. (1986): "Quasi-Optimal Disk-Arm Scheduling". RC 12462 (#55116) IBM TJW Research Center.

Hofri, M. (1980): "Disk Scheduling: FCFS vs. SSTF Revisited", *Comm. ACM,* **23,** #11, 645-653. *Corrigendum:* **24,** #11, p. 772.

Hofri, M. (1983): "Should the Two-Headed Disk be Greedy? - Yes, It Should", *Inform. Process. Lett.,* **16,** 83-85.

Hofri, M. (1984): "Analysis of Interleaved Storage via a Constant-Service Queueing System with Markov-Chain Driven Input", *J. Assoc. Comput. Mach.,* **31,** #3, 628-648.

Hofri, M. (1985): "An M/G/1 Queue with Vacations and a Threshold", Technical Report #375, Computer Science Dept., Technion, Haifa, Israel.

Hofri, M. (1986): "Queueing Systems with a Procrastinating Server", *Proc. Performance '86 – Performance Evaluation Review,* **14,** #1, May 1986, 245-253.

Hofri, M., Rosberg, Z. (1985): "Optimally Controlled CCD Shift Registers (Optimal Interception on a Recurrent Trajectory)". *Stochastic Models* **1,** #3, 341-360.

Hunter, J.J. (1969): "On the Moments of Markov Renewal Processes", *Adv. in Appl. Probab.,* **1,** 188-210.

Keilson, J., Servi L.D. (1987): "Dynamics of the M/G/1 Vacation Model", *Oper. Res.* **35,** 575-582.

Kelly, F.P. (1979): *Reversibility and Stochastic Networks,* John Wiley, Chichester.

Klimov, G.F. (1974): "Time Sharing Service Systems I", *Theory Probab. Appl,* **19,** 532-551.

Loris-Teghem J. (1988): "Vacation Policies in an M/G/1 Type Queueing System with Finite Capacity". *Queueing Systems,* **3,** 41-52.

Massey, W.A. (1984): "Open Networks of Queues: Their Algebraic Structure and Estimating Their Transient Behavior". *Adv. in App. Probab.* **16,** #1, 176-201.

Matick, R.E. (1977): *Computer Storage Systems and Technology.* Wiley-Interscience, New York.

McKenna, J., Mitra, D. (1984): "Asymptotic Expansions and Integral Representations of Moments of Queue Lengths in Closed Markovian Networks", *J. Assoc. Comput. Mach.,* **31,** #2, 346-360.

Meilijson, I., Weiss, G. (1977): "Multiple Feedback at a Single Server Station", *Stochastic Process. Appl.,* **5,** 195-205.

Neuts, M.F. (1976): "Moment Formulas for the Markov Renewal Branching Process", *Adv. in Appl. Probab.,* **8,** 690-711.

Neuts, M.F. (1977): "Some Explicit Formulas for the Steady-State Behavior of the Queue with Semi-Markovian Service Times", *Adv. in Appl. Probab.,* **9,** 141-157.

Neuts, M.F. (1981): *Matrix-Geometric Solutions in Stochastic Models, an Algorithmic Approach,* The John Hopkins Univ. Press, Baltimore.

Page, I.P., Wood, R.T. (1981): "Empirical Analysis of a Moving Head Disc Model with Two Heads separated by a Fixed Number of Tracks", *Comput. J.* **24,** 339-341.

Ross, K. (1985): Personal Communication. (Dept. of System Eng., University of Pennsylvania).)

Ross, S.M. (1983): *Introduction to Stochastic Dynamic Programming,* Academic Press, New York.

Sauer, C.H., Chandy, K.M. (1981): *Computer Systems Performance Modelling,* Prentice Hall, Englewood Cliffs, N.J.

Skinner, C.E. (1967): "A Priority Queueing System with Server Walking Time", *Oper. Res.,* **15,** 278-285.

Snyder, P.M., Stewart, W.J. (1983): "A Comparison of Two Numerical Methods for Solving Queueing Phenomena", Technical Report, Dept. of Computer Science, N. Carolina State University, Raleigh.

Takagi, H. (1987): "Queueing Analysis of Vacation Models", TRL Research Report TR87-0032, IBM Tokyo Research Laboratory.

Walrand J. (1988): *An Introduction to Queueing Networks.* Prentice-Hall, Englewood Cliffs, NJ.

Welch, P.D. (1964): "On a Generalized M/G/1 Queueing Process in which the First Customer in Each Busy-Period Receives Exceptional Service", *Oper. Res.,* **12,** 736-752.

Wong, C.K. (1983): *Algorithmic studies in Mass storage Systems.* Computer Science Press Inc.

Zahorjan, J., Hume, J.N.P., Sevcik, K.C. (1978): "A Queueing Model of a Rotational Position Sensing Disk System", *INFOR - Canadian J. Oper. Res. Inform. Process.*, **16,** 199-216.