OPERATING SYSTEMS THEORY

EDWARD G. COFFMAN, JR.
Pennsylvania State University

PETER J. DENNING
Purdue University

PRENTICE-HALL, INC.
ENGLEWOOD CLIFFS, NEW JERSEY

Library of Congress Cataloging in Publication Data

COFFMAN, EDWARD GRADY.
Operating systems theory.

(Prentice-Hall series in automatic computation)
Includes bibliographical references.
1. Electronic digital computers—Programming.
2. Algorithms. I. Denning, Peter J., joint author.
II. Title.
QA76.6.C62 001.6'42 73-18
ISBN 0-13-637868-4

© 1973 by Prentice-Hall, Inc., Englewood Cliffs, N.J.

All rights reserved. No part of this book may be reproduced in any form or by any means without permission in writing from the publisher.

10 9 8 7 6 5 4 3 2

Printed in the United States of America

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA, PTY. LTD., *Sydney*
PRENTICE-HALL OF CANADA, LTD., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*

4 PROBABILITY MODELS OF COMPUTER SEQUENCING PROBLEMS

4.1. INTRODUCTION

4.1.1. Basic Definitions

The analysis of probability models of computer operation constitutes a useful technique for preliminary studies of operating systems. As a mathematical approach probability models are generally more realistic than deterministic models, because they can represent the irregular and unpredictable demands made by computer users. These demands are reflected in the complex of queues that are developed and controlled by the operating system for the use of auxiliary and main storage units, processors, input/output devices, and system routines. Thus when probability models are formulated to study the properties of dynamic scheduling techniques they take the form of queueing systems.

In this chapter we shall concentrate our attention on the scheduling of processor queues and take the macroscopic view that the remaining queueing activities are subsumed in a general execution requirement. (In Chapter 5 we shall examine input/output queueing problems.) Our specific object will be the study of time-sharing algorithms and certain batch-processing algorithms that improve system performance by making use of information assumed known about execution requirements. Such studies can help provide insight into the properties of new sequencing algorithms even though idealizations or simplifying assumptions have to be made in order to keep the models mathematically tractable. These mathematical models may also be formulated to help validate simulation studies, and in certain cases they can be used to obtain direct measures or predictions of actual system performance—this

again in spite of simplifications found necessary in the modeling process. The remainder of this section will be devoted to a characterization of mathematical queueing systems with the appropriate specializations pertinent to our modeling of processor scheduling. Our treatment of queueing theory will be introductory and can be found in the frequently cited references [1–8].

As pictured in Fig. 4.1-1, the principal components of a queueing system

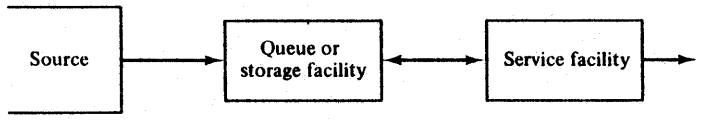

Fig. 4.1-1 A queueing system.

are a *server*, a *queue* consisting of a waiting or storage area to accommodate customers that must be delayed, and a *source*, which is a collection of system users or customers. In the computer application the server will normally consist of one or more processors; the waiting facilities may be input/output devices, auxiliary storage, or even main storage; and the customers will be jobs batched on magnetic tape, for example, or service requests of various types made by users connected on-line to the computer system. Throughout this chapter the terms server/processor, service/execution/operation, and jobs/arrivals will be used interchangeably in order to minimize unpleasant repetition of words. The terms user and (service) request will occasionally be employed when appropriate in our discussion of time-sharing systems.

Mathematical descriptions of probability models for queueing systems require specifications of the probability distribution functions describing the times between successive arrivals (to be called interarrival times) and the running or service times of jobs, along with a specification of the queueing discipline. By queueing or service discipline we mean simply a rule or procedure embodied in an operating system which determines the sequence in which jobs are executed and how much execution time a job is allocated each time it is selected for service.

We shall consistently assume that interarrival and service times are statistically independent. Moreover, with the exception of Section 4.3 both the successive interarrival times and the service times of successive arrivals will be assumed independent. We let T and S be the random variables corresponding to interarrival and service times, respectively. We adopt the notation $A(x) = \Pr[T \leq x]$ and $B(x) = \Pr[S \leq x]$ for the interarrival and service time distribution functions. The random variables T and S will be either discrete or continuous. In the latter case we shall use lowercase symbols to denote the corresponding density functions (when these exist); e.g. $f(x) = dF(x)/dx$ denotes the density function corresponding to the distribution function $F(x)$.

In general, although the existence of density functions will almost always be assumed in this chapter, the ith moment of a nonnegative random variable X with distribution function $F(x)$ will normally be expressed by the Stieltjes integral

$$\mathrm{E}(X^i) = \int_0^\infty x^i \, dF(x), \quad i = 1, 2, \ldots$$

rather than

$$\mathrm{E}(X^i) = \int_0^\infty x^i f(x) \, dx$$

In so doing, the advantages, apart from the applicability to more general distributions, are an economy of notation and the limiting of descriptions of random variables to their distribution functions.

4.1.2. The Arrival Process

For all the models we analyze we shall assume that $A(x)$ is a (negative) exponential distribution given by

$$A(x) = 1 - e^{-\lambda x}, \quad x \geq 0 \tag{4.1.1}$$

The corresponding density function is

$$a(x) = \lambda e^{-\lambda x}, \quad x \geq 0 \tag{4.1.2}$$

Working out the first two moments we have

$$\mathrm{E}(T) = \frac{1}{\lambda} \tag{4.1.3}$$

$$\mathrm{E}(T^2) = \frac{2}{\lambda^2} \tag{4.1.4}$$

from which the variance is found to be

$$\mathrm{Var}(T) = \frac{1}{\lambda^2} \tag{4.1.5}$$

Later on we shall consider arrival processes in which the parameter of the exponential distribution is a function of the state of the system. For the present, however, we shall assume that λ is independent of the system state. The assumption of independent and exponentially distributed interarrival times is perhaps the single most characteristic assumption of queueing theory. Consequently, we shall take time to study its properties in more detail.

First, let us investigate the so-called *memoryless* or *Markov* property of the exponential distribution, which accounts for the central importance of this assumption. Suppose that at the time we begin waiting for an event the distribution of our waiting time is known to be exponential with parameter λ, as in (4.1.1). After having waited t time units we inquire as to the distribution function $R_t(x)$ governing the remaining (residual) time we must wait for the event. With T denoting the random variable whose value is the total waiting time we have

$$R_t(x) = \Pr[T \leq x + t \mid T > t] = \frac{\Pr[t < T \leq t + x]}{\Pr[T > t]}$$

Using (4.1.1)

$$R_t(x) = \frac{\int_t^{t+x} \lambda e^{-\lambda y} \, dy}{\int_t^\infty \lambda e^{-\lambda y} \, dy} = \frac{e^{-\lambda t}[1 - e^{-\lambda x}]}{e^{-\lambda t}}$$

from which

$$R_t(x) = 1 - e^{-\lambda x} \tag{4.1.6}$$

Thus $R_t(x)$ is independent of t and identical to the original waiting time distribution.† Clearly, the system does not "age" with the passage of time.

The memoryless property is also exhibited in the following problem, stated in terms of arrival processes. Let arrivals occur according to a process in which interarrival times are independent and have the common distribution function (4.1.1). For an arbitrary but fixed $t > 0$, what is the distribution of the time until the next arrival? Again, this distribution is independent of t and given by (4.1.1). This result will be verified later when we calculate the corresponding result under the assumption that the times between events have a general distribution. In summary, the assumption of exponential interarrival times means that the time we must wait for a new arrival is statistically independent of how long we have already spent waiting for it.

The discrete analog of the exponential distribution is the geometric distribution. For example, consider a sequence of independent coin tossings (Bernoulli trials) where the probability of a head is p and the probability of a tail is $1 - p$. Starting with any given coin toss the distribution of the number, N, of consecutive heads required before a tail appears is geometric

$$\Pr[N = n] = (1 - p)p^n, \quad n = 0, 1, \ldots$$

independent of when the last tail occurred. This follows from the property

†It can also be shown without much difficulty that the exponential distribution is the only continuous distribution having this memoryless property [1].

that the probability of a tail occurring on any given toss is $1 - p$, independent of the results of previous coin tossings. Thus we have the memoryless property in discrete time events (arrivals) whose interarrival times are independently and geometrically distributed. Such arrivals constitute so-called Bernoulli arrival processes.

The arrival process we have defined in continuous time is the well-known Poisson process in which λ is simply called the arrival rate. We shall now give another definition of the Poisson process and show that it leads to the exponential distribution for interarrival times. Let $o(\Delta t)$ denote any quantity having an order of magnitude smaller than Δt. Suppose that there exists a constant λ such that for any small element of time Δt, the probability of no arrivals in $(t, t + \Delta t)$ is $1 - \lambda\,\Delta t + o(\Delta t)$ and the probability of one arrival is $\lambda\,\Delta t + o(\Delta t)$, where the events in the interval $(t, t + \Delta t)$ are statistically independent of t and of the events in any other nonoverlapping interval. A process satisfying the above properties is a Poisson process and the corresponding interarrival time distribution can be computed as follows. For an arbitrary time t_0 let $A_c(t)$ denote the probability that the time x of the next arrival exceeds $t_0 + t$; i.e., $A_c(t)$ is the probability that there were no arrivals in $(t_0, t_0 + t)$. Letting $z = x - t_0$ we have

$$A_c(t + \Delta t) = \Pr[z > t + \Delta t]$$

or

$$\begin{aligned} A_c(t + \Delta t) &= \Pr[z > t]\Pr[\text{no arrivals in } \Delta t] \\ &= A_c(t)[1 - \lambda\,\Delta t + o(\Delta t)] \end{aligned} \tag{4.1.7}$$

where we have used the assumption of independent events in disjoint intervals. Rearranging (4.1.7), dividing by Δt, and neglecting terms of order $o(\Delta t)$

$$\lim_{\Delta t \to 0} \frac{A_c(t + \Delta t) - A_c(t)}{\Delta t} = -\lambda A_c(t)$$

Thus,

$$A_c'(t) = -\lambda A_c(t) \tag{4.1.8}$$

Subject to the boundary condition $A_c(0) = 1$ we find the following solution to (4.1.8):

$$A_c(t) = e^{-\lambda t}, \quad t \geq 0$$

This is obviously independent of t_0 and we have

$$A(t) = 1 - A_c(t) = 1 - e^{-\lambda t}, \quad t \geq 0$$

which demonstrates that the exponential distribution for interarrival times is implied by the properties defining the Poisson process.

Finally, let us indicate briefly a method of determining the discrete probability mass function (pmf) $f_i(t)$ $(i = 0, 1, 2, \ldots)$ that specifies the probability of i Poisson arrivals in a time interval t. We immediately make use of the fact that this probability is independent of where the interval begins. We set $t = m\,\Delta t$, use the statistical independence of disjoint intervals, and compute the probability of one arrival in i of the m intervals of length Δt and no arrival in each of the remaining $m - i$ intervals. We find $f_i(t)$ by taking the limit of the binomial probability as shown below:

$$f_i(t) = \lim_{\Delta t \to 0} \binom{m}{i} [\lambda\,\Delta t + o(\Delta t)]^i [1 - \lambda\,\Delta t + o(\Delta t)]^{m-i}, \quad m = \frac{t}{\Delta t}$$

This can be reduced to

$$f_i(t) = \frac{(\lambda t)^i}{i!} \lim_{m \to \infty} \frac{m(m-1)\cdots(m-i+1)}{m^i} \lim_{m \to \infty} \left[1 - \frac{\lambda t}{m}\right]^m$$

from which we obtain

$$f_i(t) = \frac{(\lambda t)^i}{i!} e^{-\lambda t} \tag{4.1.9}$$

Equation (4.1.9) defines the *Poisson distribution*. Note that the interarrival time distribution is given by $A(t) = 1 - f_0(t)$. It is easily verified from (4.1.9) that

$$f_i(\Delta t) = \begin{cases} 1 - \lambda\,\Delta t + o(\Delta t), & i = 0 \\ \lambda\,\Delta t + o(\Delta t), & i = 1 \\ o(\Delta t), & i > 1 \end{cases} \tag{4.1.10}$$

thus showing the consistency of the definitions we have given for the Poisson process. The mean value of (4.1.9) is easily computed to be λt. In other words, the mean number of arrivals in a time interval t is the (mean) arrival rate λ times the length of the interval. The second moment of the Poisson distribution is given by $\lambda t(1 + \lambda t)$ from which we observe that the variance is λt, the same as the mean. For reasons based on the properties we have exhibited, Poisson arrivals are often called *random* arrivals.

Two important properties of parallel Poisson arrival processes will be of subsequent use. In Fig. 4.1-2(a) we have shown the confluence or sum of k independent Poisson "streams" with parameters λ_i, $i = 1, 2, \ldots, k$. The resulting process is also Poisson with rate $\lambda' = \lambda_1 + \lambda_2 + \cdots + \lambda_k$. This result is easily motivated by observing that the probability of an arrival in the sum process in a small time interval Δt is $\lambda'\,\Delta t + o(\Delta t)$ and the probability of no arrival is $1 - \lambda'\,\Delta t + o(\Delta t)$.

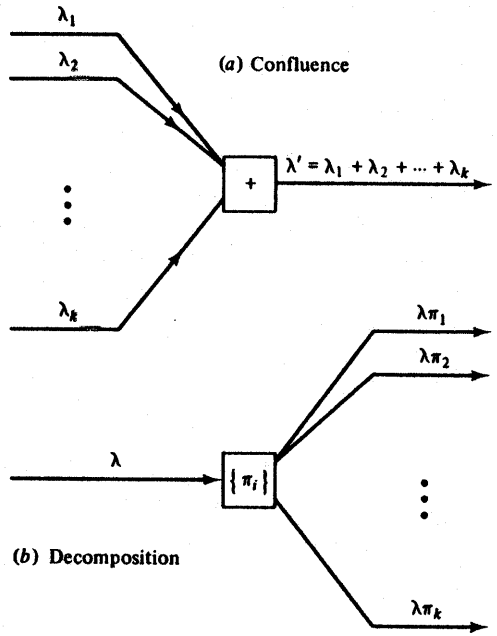

Fig. 4.1-2 Combination of Poisson processes.

The decomposition of a Poisson process according to a stationary probability distribution $\{\pi_i\}_{i=1}^k$ is shown in Fig. 4.1-2(b). Each time an arrival occurs from the source process it is assigned to one of the branch processes. The probability that the ith branch is assigned the arrival is π_i $(1 \leq i \leq k)$. The proof that the branch processes are independent and Poisson with the respective rates $\lambda\pi_i$ $(1 \leq i \leq k)$ is straightforward and left as an exercise.

4.1.3. The Service Mechanism

Although most of the models we shall analyze assume an arbitrary service time distribution, the exponential distribution will again play a special role in a couple of important models. In particular, with the latter models we shall have need of the memoryless property of the exponential distribution. This property will enable us to say that no matter how much service any given job has received at the time we choose to observe the system, the distribution function governing the service still required will be identical to the original exponential distribution.

The third component of a queueing system that we must define is the service discipline. First, we shall distinguish *preemptive* from *nonpreemptive* disciplines. According to nonpreemptive disciplines, jobs, once they commence execution, must be run to completion. With preemptive disciplines we shall assume that jobs once begun can be interrupted at any point and removed from the processor. Preempted jobs are returned to a queueing or storage facility and subsequently reallocated to the processor when their turn for service comes up again. No limits will be placed on the number of preemptions.

The amount of information used by a service discipline can vary widely and depends, of course, on what information can be assumed available. In the computer application the information used takes the form of prior knowledge of execution time, execution time already received, time of arrival, storage and input/output requirements, and priorities reflecting the importance or urgency of jobs. The scope of this book will be restricted to service disciplines using information concerning

1. *Time of arrival.* For example, the common first-in-first-out (FIFO) rule simply executes the jobs in the order in which they arrive.
2. *Job execution times.* As an example, we have the shortest-processing-time (SPT) rule according to which the next job executed is the one having the shortest execution time of those jobs currently waiting.
3. *Simple priorities.* The basic priority model that we analyze assumes that each job in the system is assigned to a priority class at (or by) the time it arrives. The set of priority classes is assumed countable. Such priorities can be used to discriminate between jobs on the basis of storage requirements, etc.; however, we shall not concern ourselves with the mechanism of priority assignment, primarily because it depends so strongly on the objectives and limitations of specific installations.

The above disciplines, along with several others, will be defined more fully in the remaining sections of this chapter. We shall conclude this section with a brief discussion of the performance measures generally sought in the analysis of queueing models of computer systems.

4.1.4. Performance Measures

Usually, the first and most basic measure of performance (or congestion) to be calculated is the distribution of the number in the system (server and queue) as a function of time. For most systems of interest the complexity of this calculation is extreme and the form of the results such as to make them difficult to interpret. Thus one usually settles for the long-run or steady-state behavior in terms of stationary probability distributions. The latter results are much simpler in general and are further justified by the fact that the transient behavior of systems is sufficiently short-lived in many cases to be of little interest.

Perhaps the most important performance measure from the point of view of the user is the distribution of the time he must spend in the system. This distribution will normally be conditioned on the amount of service he requires in those systems which discriminate between jobs on the basis of some known measure of execution times. (Hereafter, conditional waiting times will refer to waiting times conditioned specifically on the execution time required.) The distribution of waiting times in queue is also of interest in those systems where this distribution is essentially different from the previous one. In the models examined in the subsequent sections our main interest will be in finding the mean values of conditional waiting time distributions.

A third measure of congestion that is commonly studied is the distribution of the length of *busy periods*. In a single-server system a busy period begins when a job arrives to find an empty system and ends at the instant the system again becomes empty. For several of the models that we shall analyze the busy-period distributions are the same. We shall develop a functional equation for the transform of this equilibrium busy-period distribution. Calculation of the first two moments is given in the Problems.

In the next section we shall be presenting only the most basic results in applied queueing theory. Our primary purpose is to provide the reader lacking experience in this area with enough background to study basic applications to probability models of computer scheduling problems (Sections 3.4-3.8). These results will also provide a useful background to many of the techniques employed in Chapters 5 and 6 on other resource allocation problems.

Especially in the calculation of waiting times, queueing theory deals frequently with sums of independent random variables. This is one of the principal reasons why the use of Laplace transforms will be extremely convenient in certain of the following sections. We shall also make heavy use of generating functions (z-transforms) in dealing with discrete probability distributions, particularly in solving the difference equations describing stochastic processes. Because of their importance we have provided in Appendix A the definitions and basic properties of generating functions and Laplace transforms relevant to probability theory.

4.2. BASIC QUEUEING RESULTS

4.2.1. The M/M/1 Queueing System

We shall consider a single-processor queue with Poisson arrivals and service times exponentially distributed according to

$$B(x) = 1 - e^{-\mu x}, \quad x \geq 0 \tag{4.2.1}$$

with the first and second moments $\mathrm{E}(S) = 1/\mu$ and $\mathrm{E}(S^2) = 2/\mu^2$. Our main objective is the probability distribution describing the number in the system during statistical equilibrium. The results obtained will be valid for any service discipline that does not use any information concerning known execution times in determining the sequence of job executions. Examples of such disciplines include FIFO, LIFO (last-in-first-out), and random sequencing, as well as priority disciplines in which priority assignments are independent of job execution times.

In queueing parlance the systems defined above are members of the class of M/M/1 queueing systems (also called Poisson queues because of the exponential assumptions). In this notation the first element denotes the interarrival time distribution, the second element denotes the service time distribution, and the third element denotes the number of servers. M stands for the (Markov) exponential distribution, G for a general distribution, and D for the (deterministic) assumption of constant interarrival or service times.

Now suppose that the system commences operation at time $t = 0$. Let $X(t)$, $t \geq 0$, denote the number of jobs (in service and waiting) in the system at time t. The transitions $X(t) \to X(t')$, $t' > t$, are clearly determined by the chance effects introduced by our assumption of random arrivals and service times that are subject to the distribution $B(x)$. Accordingly, $X(t)$ is called a *stochastic* or *random process*. Since the system state represented by $X(t)$ is the number of jobs in the system we can associate a state space with $X(t)$ which consists simply of the nonnegative integers. In general, $X(t)$ will be defined when we specify an initial value $X(0)$ and the probability distributions governing the transitions $X(t) \to X(t')$ for all t and $t' > t \geq 0$. These transitions will be completely determined by our assumptions regarding arrivals, service times, and the service discipline.† As stated in the last section our interest will be in characterizing $X(t)$ by computing the probability distribution

$$p_n(t) = \Pr[X(t) = n], \quad t > 0, n = 0, 1, 2, \ldots \tag{4.2.2}$$

assuming that we are given $X(0)$ and hence

$$\begin{aligned} p_n(0) &= 1, \quad X(0) = n \\ &= 0, \quad \text{otherwise} \end{aligned} \tag{4.2.3}$$

We calculate the $p_n(t)$ as follows. Let us suppose that the system is in state $X(t) \geq 1$. We want to evaluate the possible transitions $X(t) \to X(t + \Delta t)$ for a small time interval Δt. According to our assumptions, the probability of an arrival in Δt is $\lambda \Delta t + o(\Delta t)$ and the probability of no arrivals is $1 - \lambda \Delta t + o(\Delta t)$, all other arrival events occurring with probability $o(\Delta t)$. Simi-

†For the present purposes the service discipline will have no effect, since we have assumed that sequencing is independent of execution times. (Also, it is understood that we shall be concerned only with disciplines that keep the processor occupied whenever there is one or more jobs in the system.)

larly, since $X(t) \geq 1$, the probabilities of one departure and no departures are $\mu\,\Delta t + o(\Delta t)$ and $1 - \mu\,\Delta t + o(\Delta t)$, respectively, with all other departure events having probability $o(\Delta t)$. Thus, using the independence of interarrival and service times and considering only events whose occurrence has probability on the order of magnitude of Δt, we can write (for $X(t) \geq 1$)

$$\begin{aligned}
\Pr[X(t+\Delta t) = X(t)+1] &= \Pr[\text{one arrival and no departures}].\\
&= [\lambda\,\Delta t + o(\Delta t)][1 - \mu\,\Delta t + o(\Delta t)]\\
&= \lambda\,\Delta t + o(\Delta t)\\
\Pr[X(t+\Delta t) = X(t)] &= \Pr[\text{no arrivals and no departures}].\\
&= [1 - \lambda\,\Delta t + o(\Delta t)][1 - \mu\,\Delta t + o(\Delta t)]\\
&= 1 - (\lambda+\mu)\,\Delta t + o(\Delta t)\\
\Pr[X(t+\Delta t) = X(t)-1] &= \Pr[\text{no arrivals and one departure}]\\
&= [\mu\,\Delta t + o(\Delta t)][1 - \lambda\,\Delta t + o(\Delta t)]\\
&= \mu\,\Delta t + o(\Delta t)
\end{aligned}$$

Relative to $X(t)$ all other possibilities for $X(t + \Delta t)$ have probabilities $o(\Delta t)$. Consequently, in terms of our notation we can accumulate the above equations into

$$\begin{aligned}
p_n(t+\Delta t) = p_n(t)[1 - (\lambda+\mu)\,\Delta t + o(\Delta t)] + p_{n+1}(t)[\mu\,\Delta t + o(\Delta t)]\\
+ p_{n-1}(t)[\lambda\,\Delta t + o(\Delta t)]
\end{aligned}$$

Neglecting terms whose order of magnitude is less than Δt we find on regrouping terms and dividing by Δt

$$\lim_{\Delta t \to 0} \frac{p_n(t+\Delta t) - p_n(t)}{\Delta t} = \mu p_{n+1}(t) - (\lambda+\mu)p_n(t) + \lambda p_{n-1}(t)$$

Thus,

$$p_n'(t) = \mu p_{n+1}(t) - (\lambda+\mu)p_n(t) + \lambda p_{n-1}(t) \tag{4.2.4}$$

Now suppose that $X(t) = 0$. In this case we can ignore the possibility of departures and write

$$p_0(t+\Delta t) = (1 - \lambda\,\Delta t)p_0(t) + \mu\,\Delta t p_1(t)$$

from which we obtain

$$p_0'(t) = \mu p_1(t) - \lambda p_0(t) \tag{4.2.5}$$

The solution to (4.2.4) and (4.2.5) with the given initial condition in (4.2.3) and the constraint $\sum_{n=0}^{\infty} p_n(t) = 1$ describes the time-dependent behavior of the system and can be found in standard texts on queueing theory.† The solution even for this simple queue is rather complex and superficially difficult to interpret. For this reason and for the more fundamental reason that such solutions have not been found for most of the other models we shall consider, we shall not dwell any further on the transient or time-dependent behavior described by these solutions. Instead, we shall restrict our interest to the limiting behavior of the probabilities $p_n(t)$ for large t.

If in fact the probabilities $p_n(t)$ approach a limit we say that these limiting probabilities describe the behavior of the system during *statistical equilibrium* (or the *steady state*). Also, the probabilities $p_n = \lim_{t\to\infty} p_n(t)$ constitute the *stationary probability distribution*. Intuitively, one might expect the existence of a stationary distribution under the simple condition that the arrival rate λ of jobs is less than the rate μ at which jobs can be executed. This can indeed be proved along with the fact that this distribution is independent of the initial condition $X(0)$. However, we shall proceed on the assumption that a stationary distribution exists and then show that these probabilities are well defined only if $\lambda < \mu$. We shall proceed in a similar way with the other models analyzed.

To calculate the stationary probabilities p_n we use the fact that in the steady state $\lim_{t\to\infty} p_n'(t) = 0$. Simplifying (4.2.4) and (4.2.5) we have

$$\mu p_{n+1} - (\lambda+\mu)p_n + \lambda p_{n-1} = 0, \quad n \geq 1 \tag{4.2.6}$$

$$\mu p_1 - \lambda p_0 = 0 \tag{4.2.7}$$

Note that (4.2.6) and (4.2.7) have the characteristic feature of steady-state *balance* equations in which the rate of transitions or flow ($\mu p_{n+1} + \lambda p_{n-1}$) into a given state $n \geq 1$ is balanced by the flow ($\lambda p_n + \mu p_n$) out of that state. Defining $\rho = \lambda/\mu$ and solving (4.2.6) and (4.2.7) systematically, we find

$$p_n = \rho^n p_0, \quad n = 0, 1, 2, \ldots$$

Imposing the constraints $\rho < 1$ and $\sum_{n=0}^{\infty} p_n = 1$ gives $p_0 = 1 - \rho$ and hence

$$p_n = (1-\rho)\rho^n, \quad n = 0, 1, 2, \ldots \tag{4.2.8}$$

Note that $\rho = 1 - p_0$ is the equilibrium probability that the system is busy. As can be seen, (4.2.8) denotes a geometric distribution with parameter ρ, and $\rho < 1$ is the condition for the existence of this distribution. Clearly, $\rho < 1$

†Because of their application to population models, (4.2.4) and (4.2.5) are frequently called *birth-and-death equations* and the corresponding process $X(t)$ is called a birth-and-death process [1].

implies that $\lambda < \mu$ and hence an arrival rate that is less than the maximum departure rate. Because of its significance, the parameter ρ is frequently referred to as the *utilization factor* or *traffic intensity*. We shall use the former term.

Let us now derive (4.2.8) using the method of generating functions. Let $P(z) = \sum_{n=0}^{\infty} p_n z^n$ denote the (probability) generating function for the p_n. Multiplying (4.2.6) by z^n, summing over all $n \geq 1$, and then adding (4.2.7) we have

$$\sum_{n=0}^{\infty} p_{n+1} z^n + \rho \sum_{n=1}^{\infty} p_{n-1} z^n = (1 + \rho) \sum_{n=0}^{\infty} p_n z^n - p_0$$

Introducing $P(z)$ we find

$$\frac{1}{z}[P(z) - p_0] + \rho z P(z) = (1 + \rho)P(z) - p_0$$

from which we obtain

$$P(z) = \frac{p_0}{1 - \rho z}$$

Using $\lim_{z \to 1} P(z) = 1$ gives $p_0 = 1 - \rho$. Expressing $P(z)$ as a power series in z we find

$$P(z) = (1 - \rho) + (1 - \rho)\rho z + \cdots + (1 - \rho)\rho^n z^n + \cdots$$

which gives us (4.2.8) as the coefficient of the nth term.

We have gone through this exercise in computing $P(z)$ because in many similar problems we can obtain $P(z)$ but we cannot conveniently solve for the p_n directly. Having $P(z)$ enables us to compute the first two moments in a routine fashion (see Appendix A), and this will normally account for most of what we want to know about system congestion. Also, we can easily find $p_0 = \lim_{z \to 0} P(z)$ and hence the probability $1 - p_0$ of a busy system.

From (4.2.8) the mean and variance of the number in the system during statistical equilibrium are given by

$$\bar{n} = \frac{\rho}{1 - \rho} \tag{4.2.9}$$

$$\text{Var}(n) = \frac{\rho}{(1 - \rho)^2} \tag{4.2.10}$$

As a final remark we emphasize the applicability of the previous results to all service disciplines that do not base job sequencing on execution time information assumed known a priori about arriving jobs. We shall next examine

the more general M/G/1 system after which waiting time distributions will be derived for FIFO systems.

4.2.2. The M/G/1 Queueing System

Under the assumption of independent exponential interarrival and service times the process $X(t)$ has the following extremely important property: At any given time t_0 the future behavior of $X(t)$ depends only on the current state $X(t_0)$ and not on the past prior to t_0. Informally, this memoryless or Markov property defines *Markov processes*, and it introduces a substantial simplification into the analysis. Although this property will be retained in the arrival processes of subsequent queueing models, we shall be interested in carrying out the analysis of $X(t)$ assuming general service time distributions. With this assumption if at some point t_0 a job is in execution, then the behavior of $X(t)$, $t > t_0$, will depend not only on $X(t_0)$ but on the amount of service already received by the executing job. Thus $X(t)$ is no longer a Markov process. An extension to the previous approach which is suggested by the above property can be sketched as follows.

We can redefine $X(t)$ so that its state space includes the values of a supplementary variable x that specifies the amount of service already received by the job (if any) in execution at time t. Specifically, the value of the new process $X'(t)$ is a pair (n, x), where n is the number in the system at time t and x is defined as above. The new process now has the Markov property and the analysis can proceed on that basis. We shall avoid the difficult analysis of this approach and consider an alternative which is far simpler for our limited purposes.

The technique to be described is motivated by recognition of the fact that although $X(t)$ does not have the Markov property at every point in time, there exist many embedded sequences of time points at which $X(t)$ does have the Markov property. Suppose, for example, that we define the time instants (or epochs) $t_1, t_2, \ldots$ so that t_i is the instant immediately after the ith job execution in an M/G/1 system with service distribution $B(x)$.

Since the service and interarrival intervals are mutually independent, $X(t)$ at the epochs t_i has the Markov property. As a result the t_i are also called *regeneration points*. The discrete time process $X(t_i)$, $i = 1, 2, \ldots$, constitutes a Markov chain which is said to be an *embedded* Markov chain with respect to $X(t)$. For ease of notation we shall let $X_i \equiv X(t_i)$ and denote the Markov chain $\{X_i\}$.

The Markov chain $\{X_i\}$ is *homogeneous* in the sense that the *one-step transition probabilities*

$$\pi_{ij} = \Pr\{X_{k+1} = j \mid X_k = i\}, \quad k = 1, 2, \ldots \tag{4.2.11}$$

are not functions of the time parameter k. The stationary distribution for $\{X_i\}$

is defined as the solution to the system of *equilibrium equations*†

$$\pi_j = \sum_{i=0}^{\infty} \pi_{ij}\pi_i, \qquad j = 0, 1, 2, \ldots \tag{4.2.12}$$

subject to the normalization $\sum_{j=0}^{\infty} \pi_j = 1$. We shall now compute the generating function for the probabilities in (4.2.12) under the assumptions of Poisson arrivals and a general service time distribution. In Problem 4-10 we shall indicate a more direct technique for finding moments of the stationary distribution.

We begin by multiplying (4.2.12) by z^j and summing over all $j \geq 0$. On changing the order of summation we have the following generating function:

$$P(z) = \sum_{j=0}^{\infty} \pi_j z^j = \sum_{j=0}^{\infty} z^j \sum_{i=0}^{\infty} \pi_i \pi_{ij} = \sum_{i=0}^{\infty} \pi_i \sum_{j=0}^{\infty} \pi_{ij} z^j \tag{4.2.13}$$

Next, we shall compute the transition probabilities π_{ij}.

Suppose that $X_k = n > 0$; i.e., there is at least one job in the system and (t_k, t_{k+1}) corresponds to a service period. Since the job executing in (t_k, t_{k+1}) is the only job to depart, we must have $X_{k+1} \geq n - 1$. It follows that $\pi_{ij} = 0$ for $j < i - 1$ and that $X_{k+1} - X_k + 1$ represents the number of arrivals occurring in (t_k, t_{k+1}). Thus, if we let q_m, $m = 0, 1, 2, \ldots$, denote the probability of m arrivals in a service period, we have $\pi_{ij} = q_{j-i+1}$ for $i > 0$ and $j \geq i - 1$.

Now suppose that $X_k = 0$; i.e., t_k commences an idle period. Regardless of the length of this idle period the probability that $X_{k+1} = j$ is simply the probability that there are j arrivals during the execution of the job whose arrival terminated the idle period. Thus $\pi_{0j} = \pi_{1j} = q_j$ for $j \geq 0$. Let us now substitute for π_{ij} in (4.2.13). On separating out the first term of the summation, we have

$$P(z) = \pi_0 \sum_{j=0}^{\infty} q_j z^j + \sum_{i=1}^{\infty} \pi_i \sum_{j=i-1}^{\infty} q_{j-i+1} z^j \tag{4.2.14}$$

Letting $A(z) = \sum_{m=0}^{\infty} q_m z^m$ denote the generating function for the distribution of the number of arrivals in a service period, we can simplify (4.2.14) to

$$P(z) = \pi_0 A(z) + \frac{A(z)}{z}[P(z) - \pi_0] \tag{4.2.15}$$

Solving for $P(z)$ we get

$$P(z) = \pi_0 \frac{(1 - z)A(z)}{A(z) - z} \tag{4.2.16}$$

†The general conditions under which a stationary distribution exists for $\{X_i\}$ are given in Appendix A. These conditions will always hold for the applications discussed in this book.

To proceed further we need to examine $A(z)$. Now $f_m(t)$ as given by (4.1.9) specifies the probability of m Poisson arrivals in time t. Hence averaging this probability over the service time distribution we have

$$q_m = \int_0^{\infty} f_m(t)\, dB(t), \quad m = 0, 1, 2, \ldots$$

Substituting (4.1.9) and computing the generating function we have

$$A(z) = \sum_{m=0}^{\infty} z^m \int_0^{\infty} \frac{(\lambda t)^m}{m!} e^{-\lambda t}\, dB(t)$$

Reversing the order of summation and integration we obtain

$$A(z) = \int_0^{\infty} e^{-\lambda t(1-z)}\, dB(t) \tag{4.2.17}$$

Using the Laplace transform $B^*(s) = \mathrm{E}[e^{-sS}]$ of service times we get

$$A(z) = B^*(\lambda - \lambda z) \tag{4.2.18}$$

Using (4.2.17) we can return to (4.2.16) and compute π_0 and the moments of the distribution $\{\pi_i\}$. Making use of $\lim_{z \to 1} P(z) = 1$ and noting that both the numerator and denominator of (4.2.16) vanish at $z = 1$, we find on applying l'Hospital's rule

$$\frac{-\pi_0 A(1)}{A'(1) - 1} = 1$$

Now $A(1) = 1$ from (4.2.17). Differentiating (4.2.17) we find $A'(1) = \rho$ and hence

$$\pi_0 = 1 - \rho \tag{4.2.19}$$

where $\rho = \lambda \mathrm{E}(S)$ is the utilization factor. (Clearly, $0 \leq \rho < 1$ is necessary for statistical equilibrium.) Thus the stationary distribution is defined by the generating function

$$P(z) = \frac{(1 - \rho)(1 - z)B^*(\lambda - \lambda z)}{B^*(\lambda - \lambda z) - z} \tag{4.2.20}$$

We find for the first moment, again using l'Hospital's rule,

$$\bar{n} = \lim_{z \to 1} \frac{dP(z)}{dz} = \rho + \frac{A''(1)}{2(1 - \rho)}$$

From (4.2.17) we obtain $A''(1) = \lambda^2 E(S^2)$ and hence

$$\bar{n} = \rho + \frac{\lambda^2 E(S^2)}{2(1-\rho)} \tag{4.2.21}$$

Introducing the *coefficient of variation* $C(S)$ defined by

$$C^2(S) = \frac{\mathrm{Var}(S)}{E^2(S)} = \frac{E(S^2)}{E^2(S)} - 1 \tag{4.2.22}$$

we can put (4.2.21) into the more common form

$$\bar{n} = \rho + \frac{\rho^2[1 + C^2(S)]}{2(1-\rho)} \tag{4.2.23}$$

Higher moments of the distribution $\{\pi_j\}$ can be computed in a similar fashion.

According to the definition of $\{X_t\}$, the expression (4.2.23) represents the mean number of jobs that are left in the system by departing jobs. In general, since the state of the system is being oberved only at the instants just following departures, one might reasonably expect the distribution $\{\pi_j\}_{j=0}^{\infty}$ to differ from the stationary distribution for the continuous time-parameter process, $X(t)$; i.e., one's evaluation of a system is generally dependent on (biased by) the times at which he has chosen to observe it. But in fact these two distributions turn out to be precisely the same. In general, this property applies to any queueing system in which the only state-changes possible at any instant of time are $+1$ (arrival) and -1 (departure). Consistent with this property we find on substituting $E(S^2) = 2/\mu^2$ into (4.2.21) the result of (4.2.9) for the exponential service time distribution.

Note particularly that $\bar{n}$ is determined not only by the mean interarrival and service times but also by the second moment of the service time distribution. As shown by (4.2.21) $\bar{n}$ increases linearly with the variance of service times. Considering constant service times, for example, the variation in service times is zero and we have

$$\bar{n} = \rho + \frac{\rho^2}{2(1-\rho)} = \frac{\rho(1-\rho/2)}{(1-\rho)} \tag{4.2.24}$$

which is to be compared with the expression $\rho/(1-\rho)$ for exponential service times (whose coefficient of variation is 1). For small ρ the difference in the means is small, but as ρ approaches 1 (the *saturation* condition), the two differ by almost a factor of 2.

Interpreting the probability $1 - \pi_0 = \rho$ of a busy system as the average number of jobs in the processor one would expect that the second term in (4.2.23) is the mean number $\bar{n}_q$ waiting in queue. As a direct calculation would show, it is indeed true that

$$\bar{n}_q = \frac{\rho^2[1 + C^2(S)]}{2(1-\rho)} \tag{4.2.25}$$

Again, the above results for general service times are applicable to any nonpreemptive queue selection rule which does not use information about job execution times.

4.2.3. Waiting Times

4.2.3.1. Residual Waiting Times. Before computing waiting times for the FIFO system we shall deal with a simpler problem whose solution will be used later in deriving expected waiting times for nonpreemptive disciplines.

Consider a random arrival to an M/G/1 system in statistical equilibrium. Let $R(x)$ denote the (conditional) probability distribution governing the remaining execution time of the job in progress, given that the system is busy at the time of arrival. Our problem is to find the expected value R of the remaining execution time. In renewal theory R is referred to as the mean forward recurrence time and in the reliability context as the mean residual lifetime. We shall give below a brief, informal derivation of R; a full analysis leading to $R(x)$ is given in Appendix B. (See also Problem 4-12.)

One's first inclination might be to say that R is simply $E(S)/2$. That this is a mistake for general service time distributions is explained by the statement that our random arrival is more likely to occur during the execution of a long job than a short one. We shall exploit this observation in the following argument. Let $b(x)$ be the service time density and $b'(x)$ the density function for the length of the service period in which a random arrival occurs. Since we may interpret $b(x)\,dx$ as the relative frequency of service periods of length x, the density $b'(x)$ should be proportional to x and $b(x)$. Letting K be the constant of proportionality we find that in order for $Kxb(x)$ to be a probability density we must have $K = 1/E(S)$. Whence

$$b'(x) = \frac{xb(x)}{E(S)} \tag{4.2.26}$$

Now given a random arrival during a service period of length x, the conditional expectation of the remaining execution time is simply $x/2$. Hence

$$R = \int_0^{\infty} \left(\frac{x}{2}\right) \frac{xb(x)}{E(S)}\,dx = \frac{E(S^2)}{2E(S)} \tag{4.2.27}$$

Very shortly, we shall have occasion to verify this result by means of an independent argument. Note that R increases linearly with the variation in

service times and is equal to the minimum of $E(S)/2$ only when service times are constant. Also, for exponential service times we see that (4.2.27) reduces to $E(S)$, consistent with the memoryless property.

Let us now remove the conditioning on the existence of a busy system at arrival time. In other words, suppose that we ask for the mean time following the time of arrival which elapses before the processor becomes free for allocation to the next job selected from the queue. This quantity is denoted $E(S_r)$. Since an arrival finds the processor available immediately when the system† is empty and since ρ is the probability of a busy system, we have in statistical equilibrium

$$E(S_r) = \rho R$$

Substituting for R and $\rho = \lambda E(S)$ we have

$$E(S_r) = \frac{\lambda E(S^2)}{2} \tag{4.2.28}$$

This result is extended easily to systems in which the arriving jobs can be grouped into classes each having a distinct service time distribution. In particular, if we let λ_p and $B_p(x)$ $(p = 1, 2, \ldots)$ denote the arrival rate and service time distribution, respectively, for jobs of class p, we have by applying the same arguments

$$E(S_r) = \frac{1}{2} \sum_{p \geq 1} \lambda_p E(S_p^2) \tag{4.2.29}$$

under the assumption of statistical equilibrium. In Section 4.5 we shall make use of (4.2.29) in an analysis of priority queues.

4.2.3.2. FIFO Waiting Time Distributions. For the FIFO system the Laplace transform of the equilibrium waiting time distribution $W(x)$ is easily obtained from the generating function $P(z)$ of the number in system. Recall that $\{\pi_i\}$ gives the probability mass function for the number of jobs left behind by a departing job. However, the jobs left behind are precisely those which arrived during the departing job's stay in system. Now by (4.2.18), $A(z) = B^*(\lambda - \lambda z)$ describes the number of arrivals during service periods; by analogy, therefore, the number of arrivals during waiting times in the system is described by

$$P(z) = W^*(\lambda - \lambda z) \tag{4.2.30}$$

†We are making, and will continue to make, implicit use of an important property of queues *with Poisson arrivals*. Specifically, it can be shown that the equilibrium queue-length distribution is identical to the distribution encountered by arrivals during statistical equilibrium. Intuitively, the result follows from the "randomness" of Poisson arrivals. (For purposes of distinction the former distribution is also called the "external observer" distribution.)

Introducing the change of variable $s = \lambda - \lambda z$ into (4.2.30) we obtain from (4.2.20)

$$W^*(s) = \frac{(1 - \rho)sB^*(s)}{s - \lambda + \lambda B^*(s)} \tag{4.2.31}$$

The waiting time in the system is the sum of a queueing time (waiting time in queue) and a service period. Hence the Laplace transform of the queueing time distribution $V(x)$ can be found from $W^*(s) = B^*(s)V^*(s)$. Thus

$$V^*(s) = \frac{(1 - \rho)s}{s - \lambda + \lambda B^*(s)} \tag{4.2.32}$$

Denoting first moments by W and V we obtain by differentiation of (4.2.31) and (4.2.32)

$$W = E(S)\left[1 + \frac{\rho[1 + C^2(S)]}{2(1 - \rho)}\right] \tag{4.2.33}$$

and

$$V = E(S)\frac{\rho[1 + C^2(S)]}{2(1 - \rho)} \tag{4.2.34}$$

Equation (4.2.33) or (4.2.34) is frequently called the Pollaczek-Khintchine formula.

We may verify (4.2.31) for the M/M/1 system by the following independent argument. Let $W_n(x)$ denote the equilibrium distribution of waiting times, conditioned on the number n in the system at arrival time. Because of the memoryless property, the remaining execution time for the job in progress at arrival has the same exponential distribution as for the jobs in queue. Consequently, with the FIFO rule $W_n(x)$ is distributed as the sum of $n + 1$ identical and independent exponentials. This corresponds to the gamma-type (or special Erlangian) density described in Appendix A. Letting μ be the parameter of the service time distribution we thus have

$$W_n^*(s) = \left(\frac{\mu}{\mu + s}\right)^{n+1} \tag{4.2.35}$$

From the geometric stationary distribution in (4.2.8), for the unconditional distribution we find

$$W^*(s) = \sum_{n=0}^{\infty} p_n W_n^*(s) = (1 - \rho) \sum_{n=0}^{\infty} \rho^n \left(\frac{\mu}{\mu + s}\right)^{n+1} \tag{4.2.36}$$

or

$$W^*(s) = \frac{(1 - \rho)\mu}{(1 - \rho)\mu + s} \tag{4.2.37}$$

This is seen to be the transform of the following exponential distribution, also obtainable from (4.2.31):

$$W(x) = 1 - e^{-(1-\rho)\mu x}, \quad x \geq 0 \tag{4.2.38}$$

with mean value

$$W = \frac{1/\mu}{1-\rho} \tag{4.2.39}$$

Thus the waiting time in the system also has the Markov property. It is worth noting the fact illustrated in (4.2.38) that a geometrically distributed sum of independent random variables each with an identical exponential distribution is itself exponentially distributed. In the discrete case the resulting distribution would be geometric if the individual distributions were geometric.

The conditional queueing time distribution $V_n(x)$ given $n \geq 1$ in the system at arrival time is clearly $W_{n-1}(x)$. Since no waiting time in queue is experienced by an arrival to an empty system, we have

$$V(x) = p_0 + \sum_{n=1}^{\infty} p_n W_{n-1}(x)$$

Using transforms we readily obtain from (4.2.8)

$$V(x) = 1 - \rho e^{-(1-\rho)\mu x} \tag{4.2.40}$$

which is consistent with (4.2.32). Note that $V(x)$ is a mixed distribution with a probability mass $(1 - \rho)$ concentrated at $x = 0$.

The mean value expressions in (4.2.33) and (4.2.34) can also be arrived at directly. For example, V can be expressed as the sum of the expected time for the processor to become available for the next job plus the mean number $\bar{n}_q$ waiting in queue times the expected service time. Specifically,

$$V = \bar{n}_q \mathrm{E}(S) + \mathrm{E}(S_r) \tag{4.2.41}$$

Substituting (4.2.25) and (4.2.28) we obtain (4.2.34).†

4.2.3.3. Little's Result. On differentiating (4.2.30) and taking the limit $z \to 1$ we obtain *Little's result*

$$\bar{n} = \lambda W \tag{4.2.42}$$

†Note that by substituting (4.2.34) for V and (4.2.25) for $\bar{n}_q$ into (4.2.41) we verify independently the expression obtained for $E(S_r)$ in (4.2.28).

which states that the equilibrium mean number in the system is equal to the product of the arrival rate and the mean time in the system. Also, this result implies a conservation principle whereby the mean number $(\bar{n})$ in the system encountered by a new arrival is equal to the mean number (λW) it leaves behind on departure. The validity of (4.2.42) extends far beyond the contexts discussed so far and essentially depends only on the existence of a steady state in which the long-run rate of arrival is equal to the long-run rate of departure. Specifically, this result does not depend on the scheduling rule or on any particular properties of the arrival process. (See also Section 3.8, where Little's result is adapted to deterministic sequencing problems.) A detailed proof of these properties has been given by Little [9].

Little's result restricted to the number in queue leads to

$$\bar{n}_q = \lambda V \tag{4.2.43}$$

It is interesting to observe that on substituting (4.2.43) for $\bar{n}_q$ in (4.2.41) we could have solved for V without having known $\bar{n}_q$ [or $P(z)$] beforehand. Finally, in our study of priority queues we shall have occasion to apply Little's result to individual priority classes. Thus, for example,

$$\bar{n}_k = \lambda_k W_k, \quad k = 1, 2, \ldots \tag{4.2.44}$$

states that for each k the mean number of priority k jobs is equal to their arrival rate times their mean waiting time.

4.2.4. The Busy-Period Distribution

Let D be the random variable denoting the length of busy periods in an M/G/1 queue during statistical equilibrium. We seek the busy-period distribution $H(y) = \Pr[D \leq y]$. Suppose that a job J_0 initiating a busy period requires x time units and that during its execution there are n arriving jobs $J_1, J_2, \ldots, J_n$. Now the distribution $H(y)$ is insensitive to the sequence in which $J_1, J_2, \ldots, J_n$ and subsequent arrivals are executed, and so let us adopt the following discipline. After J_0 completes we execute J_1 and the subsequently arriving jobs until $J_2, \ldots, J_n$ are the only remaining jobs in the system. At this point we proceed in a similar manner executing J_2 and subsequent arrivals until $J_3, \ldots, J_n$ are the only jobs remaining. This process is repeated so that finally J_n is executed along with subsequent arrivals until the system again becomes empty and the busy period terminates.

Let D_i denote the time elapsing between the beginning of J_i's execution and the beginning of J_{i+1}'s execution, $i < n$, and let D_n be the time interval beginning with J_n's execution and ending with the termination of the busy period. Let X and N be the random variables denoting, respectively, J_0's

execution time and the number of arrivals during its execution time. A little reflection leads to the key observation that the D_i $(1 \leq i \leq n)$ are independent random variables, each having the same distribution $H(y)$ of a busy period. Thus the conditional distribution of a busy period given that J_0 requires x time units and n arrivals occur during these x time units has the following Laplace transform:

$$\begin{aligned} \mathrm{E}(e^{-sD} \mid X = x, N = n) &= e^{-sx}\mathrm{E}(e^{-s\sum_{i=1}^{n} D_i}) \\ &= e^{-sx}[H^*(s)]^n \end{aligned}$$

Removing the conditioning on n, the number of Poisson arrivals during x, and the service time x of J_0, we obtain

$$\mathrm{E}(e^{-sD}) = \int_0^\infty e^{-sx} \sum_{n=0}^{\infty} \frac{(\lambda x)^n}{n!}[H^*(s)]^n e^{-\lambda x}\, dB(x)$$

or

$$H^*(s) = \int_0^\infty e^{-sx} e^{-\lambda x[1-H^*(s)]}\, dB(x) \tag{4.2.45}$$

Finally, we get

$$H^*(s) = B^*(s + \lambda - \lambda H^*(s)) \tag{4.2.46}$$

Although this functional equation cannot usually be solved to give explicit solutions for $H^*(s)$, moments are readily derived. (See Problem 4-1.) The above analysis will also be exploited in Problem 4-13 to obtain waiting time distributions for the LIFO discipline.

4.3. STATE-DEPENDENT ARRIVAL AND SERVICE TIMES IN POISSON QUEUES

The equilibrium equations of (4.2.6) and (4.2.7) can easily be generalized to apply to a variety of interesting and useful variations of the basic Poisson queue in which the service discipline is not influenced by service times. In particular, let us assume that the arrival rate and (maximum) service rates are functions of the system state (i.e., number in the system). Let λ_n and μ_n denote these arrival and service rates when the system contains n jobs, waiting and in execution. Reworking (4.2.4) and (4.2.5) we obtain

$$p_n'(t) = \mu_{n+1}p_{n+1}(t) - (\lambda_n + \mu_n)p_n(t) + \lambda_{n-1}p_{n-1}(t), \quad n \geq 1 \tag{4.3.1}$$

$$p_0'(t) = \mu_1 p_1(t) - \lambda_0 p_0(t) \tag{4.3.2}$$

The corresponding equilibrium equations become

$$\mu_{n+1}p_{n+1} - (\lambda_n + \mu_n)p_n + \lambda_{n-1}p_{n-1} = 0 \tag{4.3.3}$$

$$\mu_1 p_1 - \lambda_0 p_0 = 0 \tag{4.3.4}$$

Solving (4.3.3) and (4.3.4) systematically we find

$$p_1 = \frac{\lambda_0}{\mu_1}p_0, \qquad p_2 = \frac{\lambda_1}{\mu_2}\frac{\lambda_0}{\mu_1}p_0, \qquad p_3 = \frac{\lambda_2}{\mu_3}\frac{\lambda_1}{\mu_2}\frac{\lambda_0}{\mu_1}p_0, \qquad \ldots$$

or

$$p_n = p_0 \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i}, \quad n = 1, 2, \ldots \tag{4.3.5}$$

We obtain p_0 by equating the sum of the probabilities to 1. Thus

$$p_0 = \left\{1 + \sum_{n=1}^{\infty} \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i}\right\}^{-1} \tag{4.3.6}$$

We shall now work out some examples having meaning in the computer application.

In many queueing processes arising in computer operation the assumption of a waiting or storage facility of unlimited size is quite untenable. To model such systems under exponential assumptions suppose that a maximum of L jobs (including the one on the processor) can be accommodated in a single-server system. Assume that jobs arriving to find the system full $(n = L)$ leave without returning. We may use the results in (4.3.5) and (4.3.6) by assigning

$$\begin{aligned} \lambda_n &= \lambda, \quad n < L \\ &= 0, \quad n \geq L \\ \mu_n &= \mu, \quad n = 1, 2, \ldots, L \end{aligned} \tag{4.3.7}$$

Substituting into (4.3.5) and (4.3.6) and using $\rho = \lambda/\mu$ we find

$$p_n = \frac{\rho^n(1-\rho)}{1-\rho^{L+1}}, \qquad n = 0, 1, \ldots, L \tag{4.3.8}$$

from which the moments and waiting times can be found in the usual ways. Note that (4.3.8) is valid for all $\rho \geq 0$. [When $\rho = 1$ (4.3.8) becomes the uniform distribution $p_n = 1/(L+1)$.] No matter what the relative values of λ and μ are, the number in the system is bounded by L, and p_n is nonzero for all n, $0 \leq n \leq L$, and $0 < \rho < \infty$.

Another important constraint to many computer models is that repre-

sented by the *finite-source* assumption. In time-sharing applications especially, the number of users (e.g., consoles) may be so few as to make the infinite-source assumption inherent in the basic Poisson arrival mechanism a poor representation of the actual arrival process.

Suppose that we have N terminals (in a remote job entry system, say) and suppose that the time elapsing from the completion of one job entered at a terminal to the time the next job is entered is exponentially distributed with parameter λ. Assuming exponentially distributed service times as before we may use the results of (4.3.5) and (4.3.6) as follows. When there are n jobs in the system there are $N - n$ terminals available for entering new jobs. Thus

$$\begin{aligned} \lambda_n &= (N-n)\lambda, & n \le N \\ &= 0, & n > N \\ \mu_n &= \mu, & n \ge 1 \end{aligned} \tag{4.3.9}$$

Substituting into (4.3.5) and (4.3.6) and using $\rho = \lambda/\mu$ yields

$$p_0 = \left\{ \sum_{i=0}^{N} \frac{N!}{(N-i)!} \rho^i \right\}^{-1} \tag{4.3.10}$$

$$p_n = \frac{\rho^n [N!/(N-n)!]}{\sum_{i=0}^{N} [N!/(N-i)!\rho^i]}, \qquad n = 0, 1, \ldots, N \tag{4.3.11}$$

In the next section we shall return to this model and discuss its use in a study of time-sharing systems. The original motivation for models of the type analyzed here arose from the study of machine servicing problems in which the machines were the customers and the repairman was the server.

As a final example we shall consider the multiserver (M/M/m) Poisson queue. Suppose that there are m processors and a service discipline that assigns jobs to these processors as they become available. Clearly, if there are $n \le m$ jobs in the system, the queue will be empty and n of the processors busy. To apply our results we make

$$\begin{aligned} \lambda_n &= \lambda, & n = 0, 1, 2, \ldots \\ \mu_n &= n\mu, & 1 \le n \le m \\ &= m\mu, & n \ge m \end{aligned} \tag{4.3.12}$$

Working out the results of (4.3.5) and (4.3.6) and defining a new utilization factor for m processors, $\rho = \lambda/m\mu$, we obtain

$$p_n = \begin{cases} \dfrac{(m\rho)^n}{n!} p_0, & n < m \\ \dfrac{m^m}{m!} \rho^n p_0, & n \ge m \end{cases} \tag{4.3.13}$$

where

$$p_0 = \left\{ \frac{(m\rho)^m}{m!(1-\rho)} + \sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} \right\}^{-1} \tag{4.3.14}$$

The moments are found directly but no simple forms result.

4.4. THE ROUND-ROBIN SERVICE DISCIPLINE

The round-robin (RR) service discipline was one of the first to be used in time-sharing systems, primarily because of its simplicity and because it has the property of providing shorter waiting times for shorter jobs. After defining the RR discipline we shall set about the task of quantifying its waiting time property.

An RR system is pictured in Fig. 4.4-1. Poisson arrivals are assumed at an

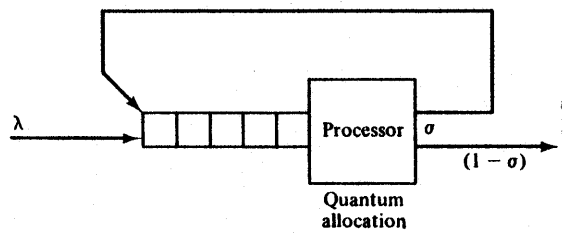

Fig. 4.4-1 The round-robin system.

average rate λ. Assuming an arbitrary service time distribution, job sequencing proceeds as follows. Each time a job is selected for operation it is selected from the head of the ordered queue and allocated a fixed amount of execution time called a *quantum* or *time slice*. We let the given quantum size be Q time units. If the job completes prior to the expiration of the quantum, then it simply departs from the system. If after Q time units the job requires additional execution time, it is returned (fed back) to the end of the queue to await its next turn at the processor. In this way a job makes as many passes through the queue as it requires quanta of service. New arrivals simply join the end of the queue at the time of arrival.

To simplify the analysis without altering the basic structure of the conditional waiting time distribution we shall assume that service times are integral multiples of the quantum size. We shall continue to make this assumption in our discussion of other disciplines according to which service is allocated a quantum at a time. With respect to the time interval Q the specific (discrete) service time distribution to be assumed is the geometric distribution

$$g_i = \sigma^{i-1}(1-\sigma), \quad i = 1, 2, \ldots \tag{4.4.1}$$

where $0 < \sigma < 1$. That is, the probability that a given job requires i quanta (i.e., iQ time units) is given by g_i. The first and second moments of the service time distribution are given by

$$\mathrm{E}(S) = \sum_{i=1}^{\infty} (iQ)g_i = \frac{Q}{1-\sigma} \tag{4.4.2}$$

$$\mathrm{E}(S^2) = \sum_{i=1}^{\infty} (iQ)^2 g_i = \frac{1+\sigma}{(1-\sigma)^2}Q^2 \tag{4.4.3}$$

Because of the memoryless property, the assumption of the geometric distribution (or the exponential distribution for continuous service times) is essential to the analysis of the RR system, for this property enables us to say that, regardless of the number of quanta already received by a job, the probability that it requires one more is always the same and given by the parameter σ.

Consider a random arrival to the system in statistical equilibrium and suppose that it finds j in the system and requires k quanta of service. Let $W_k(j)$ denote the conditional expectation of the time spent by the arrival in the system given j in the system at arrival time. Thus the conditional waiting time in the system of such a job is defined as

$$W_k = \sum_{j=0}^{\infty} p_j W_k(j) \tag{4.4.4}$$

where $\{p_j\}_{j=0}^{\infty}$ is the stationary probability distribution for the number in the RR system.† Our objective now is to develop an expression for W_k in terms of the parameters λ, σ, and Q. First, however, we shall calculate $W_k(j)$. In effect, our approach is to "tag" the arrival whose waiting time we seek and to follow it through the system.

A job requiring k quanta of service must make k passes through the queue; i.e., such jobs are fed back to the end of the queue $k-1$ times. The length of a pass is measured from the instant the job joins the queue to the instant it next rejoins the queue (or departs if the job's next quantum is its last). According to this definition, each pass except the first requires an integral number of quanta.

Let $U_i(j)$, $i = 1, 2, \ldots, k$, be the random variable denoting the time required on the ith pass, assuming j in the system at arrival. Clearly,

$$W_k(j) = \sum_{i=1}^{k} \mathrm{E}(U_i(j)) \tag{4.4.5}$$

Now if $U_i(j) = x$, $i \geq 2$, then $(x/Q) - 1$ denotes the number of jobs ahead of the tagged job on the ith pass. On the average $\sigma[(x/Q) - 1]$ of these will

†See footnote on p. 162.

return for a quantum on the $(i+1)$st pass (this is where we invoke the memoryless property of the geometric distribution). Therefore, since λx is the mean number of arrivals during the ith pass, for the $(i+1)$st pass we have

$$\mathrm{E}[U_{i+1}(j) \mid U_i(j) = x] = \sigma Q\left(\frac{x}{Q} - 1\right) + \lambda x Q + Q$$

from which

$$\mathrm{E}[U_{i+1}(j)] = (\lambda Q + \sigma)\mathrm{E}[U_i(j)] + Q(1-\sigma), \quad i = 2, 3, \ldots \tag{4.4.6}$$

Since j jobs are given to be ahead of the tagged job on the first pass, we have

$$\mathrm{E}[U_2(j)] = \lambda Q \mathrm{E}[U_1(j)] + Q(\sigma j + 1) \tag{4.4.7}$$

By a simple induction argument we can establish the following explicit expression for $\mathrm{E}[U_i(j)]$ $(2 \leq i \leq k)$:

$$\mathrm{E}[U_i(j)] = (\lambda Q + \sigma)^{i-2}\mathrm{E}[U_2(j)] + Q(1-\sigma)\frac{1-(\lambda Q+\sigma)^{i-2}}{1-\lambda Q-\sigma} \tag{4.4.8}$$

Substituting into (4.4.5) and carrying out the summation we find

$$W_k(j) = \mathrm{E}[U_1(j)] + \frac{(k-1)Q}{1-\rho} + Q\left[\lambda \mathrm{E}[U_1(j)] + \sigma j - \frac{\rho}{1-\rho}\right]\frac{1-\alpha^{k-1}}{1-\alpha} \tag{4.4.9}$$

where

$$\alpha = \lambda Q + \sigma \tag{4.4.10}$$

and where ρ is the utilization factor for the system:

$$\rho = \frac{\lambda Q}{1-\sigma} \tag{4.4.11}$$

Substituting into (4.4.4) and noting that

$$W_1 = \sum_{j=0}^{\infty} p_j \mathrm{E}[U_1(j)]$$

we have

$$W_k = W_1 + \frac{(k-1)Q}{1-\rho} + Q\left[\lambda W_1 + \sigma\bar{n} - \frac{\rho}{1-\rho}\right]\frac{1-\alpha^{k-1}}{1-\alpha}, \quad k \geq 1 \tag{4.4.12}$$

where $\bar{n}$ is the mean of the stationary distribution $\{p_j\}$. However, this distribu-

tion is precisely the same as the one applying to an M/G/1 system, as can be seen, for example, from the fact that the interdeparture intervals during busy periods in both systems have a geometric distribution with parameter σ. Hence, using (4.4.3) for $E(S^2)$ in (4.2.21), we have

$$\bar{n} = \rho + \frac{(1+\sigma)\rho^2}{2(1-\rho)} \tag{4.4.13}$$

Finally, for the expected time to make the first pass,

$$W_1 = \mathrm{E}(Q_r) + \bar{n}_q Q + Q \tag{4.4.14}$$

where $\mathrm{E}(Q_r)$ is the mean time to finish the quantum (if any) in progress at arrival time. Since quanta are of fixed length, from (4.2.28) we have $\mathrm{E}(Q_r) = \rho Q/2$. Using $\bar{n}_q = \bar{n} - \rho$ we have

$$W_1 = \frac{1-\rho}{2} Q + \bar{n} Q \tag{4.4.15}$$

With (4.4.13) and (4.4.15) W_k is completely determined in (4.4.12).

It is readily seen from (4.4.12) that for fixed parameter values the dependence of W_k on k is dominated by the linear term $(k-1)Q/(1-\rho)$ when k is large. We shall now show that this linear dependence is precise when, for fixed mean service time $Q/(1-\sigma)$, we consider the limit of (4.4.12) as Q approaches zero.

Note first that as Q becomes very small with $Q/(1-\sigma)$ fixed, jobs are cycled a large number of times, receiving on each cycle a very small amount of service. Clearly, the relative position of the jobs in queue has less effect as Q is made smaller. Thus in the limit $Q \to 0$ the geometrically distributed service times become exponentially distributed with the same mean and the resulting system corresponds to one in which the processor sharing of Chapter 3 is implemented in a dynamic scheduling discipline. More specifically, if at some point there are n jobs in the processor-sharing (PS) system, each of them is receiving service at $(1/n)$th the rate that is received by a job in sole possession of the processor.

To find a corresponding expression for conditional waiting times from (4.4.12) we denote the service requirement $t = kQ$ and the associated mean waiting time $W(t)$. Informally, we substitute t/Q for k in (4.4.12) and take the limit as Q approaches zero with $\rho = \lambda Q/(1-\sigma)$ held fixed. It is not difficult to verify that the second term in (4.4.12) is the only term not to vanish in the limit. Hence we find

$$W(t) = \frac{t}{1-\rho} \tag{4.4.16}$$

Thus the mean time spent in the PS system under exponential assumptions is directly proportional to the service required. Note especially that $W(t)$ is not affected by the variance of service times. Intuitively, this arises from the sharing property that prevents longer jobs from blocking the shorter jobs as in the FIFO system. Since the mean $(\bar{n})$ in the PS and FIFO system is the same under the assumption of exponential interarrival and service time distributions, then by Little's result we see that the mean (unconditional) waiting times for both systems are the same and given by

$$W = \frac{\mathrm{E}(S)}{1-\rho} \tag{4.4.17}$$

where $\mathrm{E}(S)$ is the mean of the service time distribution and $\rho = \lambda \mathrm{E}(S)$ is the utilization factor. It has been shown [10] that the mean time spent in the PS system is also given by (4.4.17) for *any* service time distribution with mean $\mathrm{E}(S)$.

As stated in the previous section it is more realistic to assume a finite source in those time-sharing systems having a relatively small number of user terminals. Under a processor-sharing discipline a finite-source system can be studied using the model of Section 4.3. The results in (4.3.10) and (4.3.11) carry over for the present purpose since the processor-sharing discipline does not alter the (exponential) distribution of interdeparture intervals during busy periods. In view of the systems being represented by finite-source time-sharing models the mean time in system is frequently called the *mean response time*, while the time taken to produce a new request after the previous request has been serviced is called *think time*.

The mean response time W is easily calculated using the following argument. Since the mean think time is $1/\lambda$ for each terminal, the (mean) fraction of the time spent in the system by a user is given by $W/(W + 1/\lambda)$. Thus the mean number $\bar{n}$ of users being processed by the system can be written as

$$\bar{n} = N \frac{W}{W + 1/\lambda} \tag{4.4.18}$$

Equating (4.4.18) to the mean calculated from the distribution in (4.3.10) we have

$$N \frac{W}{W + 1/\lambda} = \sum_{n=0}^{N} n p_n = p_0 \sum_{n=1}^{N} \frac{N!}{(N-n)!} n \rho^n$$

where

$$p_0 = \left\{ \sum_{n=0}^{N} \frac{N!}{(N-n)!} \rho^n \right\}^{-1} \tag{4.4.19}$$

Solving for W we obtain, after some routine algebra,

$$W = \frac{N(1/\mu)}{1 - p_0} - \frac{1}{\lambda} \tag{4.4.20}$$

The principal shortcoming of the time-sharing analysis so far has been the failure to take into account the costs (time delays) generally occasioned by the switching mechanism that removes one job from the processor and loads or prepares the next job to be executed. Such costs would be negligible only in a system where there is sufficient main memory to store all executing programs or at least to assure a high degree of overlapping of input/output and execution. One common but approximate technique for including these delays is to assume that a quantum always consists of a fixed, initial time interval devoted to these so-called program *swapping* time delays. One source of the approximation in this technique is the fact that while there is only one job in the system we cannot assume that this job is transferred in and out of main memory during its execution. Clearly, the extent of this approximation becomes less as the demand on the system (as measured by ρ) increases. With this method of accounting for swapping costs, experiments with a general purpose time-sharing system have shown remarkably good agreement between observed measures and those we have calculated here for a mathematical model [11].

Another source of approximation rests with the assumption that swapping times are fixed. However, a constant swap time should be a usable approximation for those systems having drum or disk auxiliary memories in which the time to swap a process is dominated by a single near-constant seek or latency delay. Further details on these systems are provided in Chapter 5.

A precise modeling of swap time, removing both of the approximations above, necessarily leads to a queueing process not having the Markov property. Similar to the analysis in Section 4.2 the approaches to the general model have consisted of applications of the theory of Markov chains. A Markov chain is defined at the epochs just following the service received during each quantum, including the swap times incurred. Note that one has a finite Markov chain in the case of the finite-source model. Thus stationary probability distributions can always be found by inverting (possibly large) matrices formulated from the one-step transition probabilities. In general, such Markov chain approaches to finite-source models lead to cumbersome waiting time results that cannot be interpreted directly. On the other hand, efficient computational methods have been developed, and, of course, the numerical studies have led to more realistic descriptions of system behavior.

Conditional waiting times for the infinite-source RR system were first obtained by Kleinrock [12] using a discrete time model (i.e., geometric interarrival and service times). The analysis in this section parallels more closely the treatments provided for the RR model in continuous time [13, 14]. Processor-sharing results were obtained as limits of the RR results [13, 14, 15]. It is worth noting that for the RR system [16] and the processor-sharing system [17] the complete distribution (in fact its Laplace transform) has been determined for conditional waiting times. Conditional waiting times for the finite-source RR models (including a swap time parameter) were first obtained by Coffman [18] and Krishnamoorthi and Wood [19]. Since the appearance of the above work a substantial literature has evolved on the RR system and certain variations of it. For further references the interested reader can consult the survey by McKinney [20].

4.5. NONPREEMPTIVE PRIORITY QUEUES

In this section we shall consider systems in which the jobs are given preferential treatment based on priorities associated with the jobs. We assume that the priority of a job is an integer fixed at arrival time; thus we shall speak of the kth priority class as being the class of jobs with priority k ($k = 1, 2, \ldots$). Since it is conventional in the literature, we shall say that one job has higher priority than another if it belongs to a priority class with lower index.

The priority queueing system to be studied is pictured in Fig. 4.5-1, where

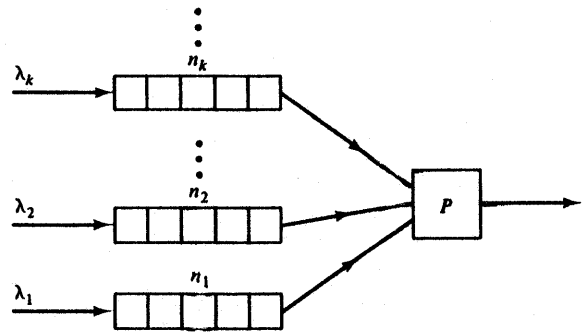

Fig. 4.5-1 Nonpreemptive priority queue.

the different queue levels correspond to the different priority classes. For the service discipline we assume that whenever a job is completed the processor is next assigned to that job at the head of the highest priority (lowest level) nonempty queue. Once a job is begun on a processor it is allowed to run to completion; i.e., the service discipline is nonpreemptive. Independent Poisson arrivals are assumed for the different priority classes with the arrival rate for the kth class denoted by λ_k ($k = 1, 2, \ldots$). Arbitrary, possibly different service time distributions exist for the priority classes. We denote the ith

moment of the distribution for the kth class by $E(S_k^i)$. [As usual, we let $E(S_k) = E(S_k^1)$.]

Let us now consider the mean waiting time W_k of random arrivals to the kth queue during statistical equilibrium. It follows from the definition of the service discipline that

$$W_k = V_k + E(S_k) \tag{4.5.1}$$

where V_k is the mean waiting time in queue for the priority k jobs. Thus we proceed to find an expression for V_k.

Let n_i ($i = 1, 2, \ldots$) be the number of jobs encountered in the ith queue at the time of arrival of a priority k job (to be called the tagged job as before). Clearly, the tagged job cannot commence processing until all jobs of equal or higher priority at queue levels 1 through k have been completed. Moreover, the tagged job must wait for the completion of all jobs of priority 1 through $k-1$ which arrive during its waiting time. For the purpose of computing V_k let us assume that all the $n_1 + n_2 + \cdots + n_k$ jobs of priorities 1 through k in the system at the time of arrival are executed first. Let us call the expected value of this time interval V_k'. Let V_k'' denote the expected value of the following interval during which the processor executes all jobs of higher priority arriving while the tagged job is waiting in queue. For V_k we can write

$$V_k = E(S_r) + V_k' + V_k'' \tag{4.5.2}$$

where $E(S_r)$ is the mean execution time that remains for the job on the processor at the time of arrival and is given by (4.2.29) as

$$E(S_r) = \frac{1}{2}\sum_{k=1}^{\infty} \lambda_k E(S_k^2) \tag{4.5.3}$$

Although (4.5.2) is based on a resequencing of the processing times for which the tagged job must wait, its validity is justified by the fact that the arrival process is Poisson and independent of the service mechanism and the state of the system. According to our definition,

$$V_k' = \sum_{i=1}^{k} E(n_i)E(S_i) \tag{4.5.4}$$

Letting $\rho_i = \lambda_i E(S_i)$ and applying Little's result to each of the first k queues we get

$$V_k' = \sum_{i=1}^{k} \lambda_i V_i E(S_i) = \sum_{i=1}^{k} \rho_i V_i \tag{4.5.5}$$

Now the mean number of ith priority arrivals during the time the tagged job spends in queue is $\lambda_i V_k$. Hence

$$V_k'' = \sum_{i=1}^{k-1} \lambda_i V_k E(S_i) = \sum_{i=1}^{k-1} \rho_i V_k \tag{4.5.6}$$

Substituting (4.5.5) and (4.5.6) into (4.5.2) gives

$$V_k = E(S_r) + \sum_{i=1}^{k} \rho_i V_i + \sum_{i=1}^{k-1} \rho_i V_k$$

or

$$V_k = \frac{E(S_r) + \sum_{i=1}^{k} \rho_i V_i}{1 - \sum_{i=1}^{k-1} \rho_i} \tag{4.5.7}$$

Using (4.5.3) for $E(S_r)$ it is readily shown by induction that the solution to (4.5.7) is given by

$$V_k = \frac{\sum_{i=1}^{\infty} \lambda_i E(S_i^2)}{2(1-\beta_k)(1-\beta_{k-1})} \tag{4.5.8}$$

where

$$\beta_j = \sum_{i=1}^{j} \rho_i \tag{4.5.9}$$

This result was first derived by Cobham [21]. Laplace transforms of the waiting-time distributions for the nonpreemptive priority queue were first calculated by Kesten and Runnenberg [22].

Using these results for the nonpreemptive priority queue we can obtain rather simply the analogous results for a so-called *shortest-processing-time* (SPT) discipline. For this discipline we assume that the execution time of a job is known at the time of arrival. Whenever the processor completes the execution of a job the next job executed by the SPT discipline is that one of those waiting with the least execution time.

To obtain mean waiting times we can simplify matters with the assumption of discrete service times. Let g_i, $i = 1, 2, \ldots$, denote the probability that a job requires iQ units of execution time, and let $E(S^i)$, $i = 1, 2, \ldots$, denote the moments of the service time distribution as before. If λ denotes the total arrival rate of jobs to the system, then λg_k denotes the arrival rate of jobs requiring k quanta. Since the decomposition of a Poisson process according to a stationary probability distribution gives rise to independent Poisson processes (see Section 4.2), the SPT system is representable as in Fig. 4.5-1 with $\lambda_k = \lambda g_k$. Substituting λg_k for λ_k and $iQ\lambda g_i$ for ρ_i (4.5.8) yields the following expression for the mean waiting time in queue for a job requiring

k quanta:

$$V_k = \frac{\lambda E(S^2)}{2(1 - \lambda H_k(S))(1 - \lambda H_{k-1}(S))} \tag{4.5.10}$$

where

$$H_j(S) = \sum_{i=1}^{j} (iQ)g_i \tag{4.5.11}$$

Informally, the result for a continuous service time distribution $B(x)$ is found by letting Q be a differential element of time and g_i be the probability mass function corresponding to $B(x)$ (in the sense that the geometric distribution is a discretization of the exponential distribution). Taking the limit $Q \to 0$ of (4.5.10) as we did in the previous section to obtain the processor-sharing result, we find Phipps' result [23] for the conditional waiting time in queue of a job whose service requirement is t:

$$V(t) = \frac{\lambda E(S^2)}{2\left[1 - \lambda \int_0^t x\, dB(x)\right]^2} \tag{4.5.12}$$

4.6. THE SHORTEST-ELAPSED-TIME DISCIPLINE

We now return to our investigation of time-sharing algorithms under the assumption that job running times are not known in advance. Besides the RR discipline of Section 4.4, the discrete versions of the so-called *shortest-elapsed-time* (SET) discipline have been investigated as a further means of providing favored service for jobs that have short execution times.

A system using a discrete SET discipline is shown in Fig. 4.6-1. Because of the structure of Fig. 4.6-1 and because of the definition below, the discrete SET disciplines have also been called multiple-level, feedback queueing disciplines. For the particular system we have chosen to analyze, the jobs arrive in a Poisson stream at rate λ and are assumed to have service times taken from a discrete but otherwise general probability distribution. As before g_i ($i = 1, 2, \ldots$) will represent the probability that a job requires iQ units of execution time, and $G(k) = \sum_{i=1}^{k} g_i$ will denote the cumulative distribution function.

A general statement of the service discipline is as follows. After the processor has completed the quantum allocated to a given job the next job to be allocated a quantum of execution time is the one having received the fewest quanta of all those jobs currently waiting. If there is a tie among several jobs having received the least service, then the job selected is the one with the earliest arrival time. This discipline can be put in terms of priority queues, as shown in Fig. 4.6-1, by simply associating the kth ($k = 1, 2, \ldots$)

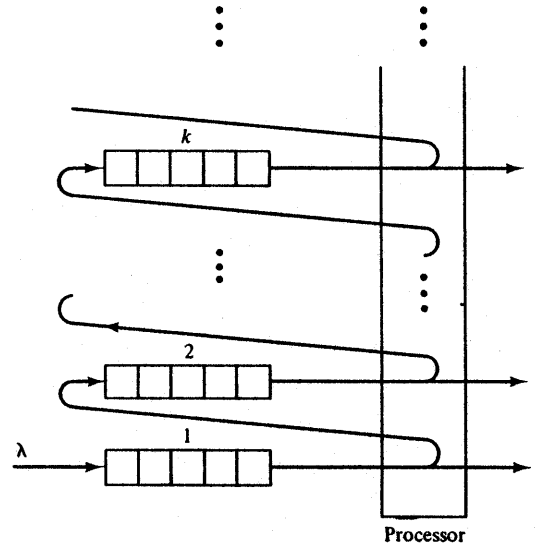

Fig. 4.6-1 The discrete SET system.

queue level (priority) with the set of jobs having already received $k - 1$ quanta and waiting for their kth quantum. At the time a job at the kth level begins a quantum the first $k - 1$ queues must be empty. After such a job completes its kth quantum the job either leaves the system because it is complete or it joins the end of the $(k + 1)$st queue level to await its $(k + 1)$st quantum.

A superficial comparison of the RR and SET disciplines leads one to expect that the SET discipline favors short jobs more than the RR discipline but at the expense of longer waiting times for long jobs. We shall now develop waiting time results for the SET discipline with which a more specific comparison can be made. For the mean waiting time of a random arrival requiring k quanta we have

$$W_k = V_k + kQ \tag{4.6.1}$$

To compute V_k we first partition it into

$$V_k = V_k' + V_k'' \tag{4.6.2}$$

where V_k' and V_k'' are defined as follows. V_k' consists of the mean time to complete the quantum in progress and to complete the allocation of up to k quanta to the jobs in the system at the time of arrival. In other words V_k' is

the mean time to finish the job in progress plus the mean time to service the jobs in the first k queue levels at the time of arrival.

Now, according to our priority discipline, every job that arrives while the new arrival (the tagged job) is waiting in queue and receiving its first $k-1$ quanta must be allocated up to a maximum of $k-1$ quanta of execution time. That is, each such job must be allowed to ascend to the kth queue level if it requires in excess of $(k-1)Q$ units of execution time. The total execution time required by these new arrivals has a mean value which we define as V_k''.

Let $E_k(S)$ denote the mean amount of execution time used by a job to which kQ time units have been allocated. We have

$$E_k(S) = \sum_{i=1}^{k} (iQ)g_i + kQ[1 - G(k)] \tag{4.6.3}$$

For later use we also define the second moment

$$E_k(S^2) = \sum_{i=1}^{k} (iQ)^2 g_i + (kQ)^2[1 - G(k)]$$

It follows immediately from our definition of V_k'' that

$$V_k'' = \lambda[V_k + (k-1)Q]E_{k-1}(S) \tag{4.6.4}$$

To compute V_k' we simplify matters as follows. Suppose that we consider a modified SET system in which the jobs served at the first queue level are allocated kQ units of execution time and those served at each higher level are allocated one quantum of service. The new system is pictured in Fig. 4.6-2. Observe that, from the point of view of a job requiring k quanta, the mean time spent in the first queue of the new system is precisely V_k', the mean time spent waiting for jobs in the original system at the time of arrival to receive their maximum of k quanta. Thus by examining the new system we find

$$V_k' = E'(S_r) + N_1' E_k(S) \tag{4.6.5}$$

where $E'(S_r)$ is the mean time to complete the job in progress and N_1' is the mean number encountered in the first queue level at the time of arrival in the *new* system. Using Little's result we have

$$V_k' = E'(S_r) + \lambda V_k' E_k(S) \tag{4.6.6}$$

or

$$V_k' = \frac{E'(S_r)}{1 - \lambda E_k(S)} \tag{4.6.7}$$

To compute $E'(S_r)$ we must take into account the fact that in the new system

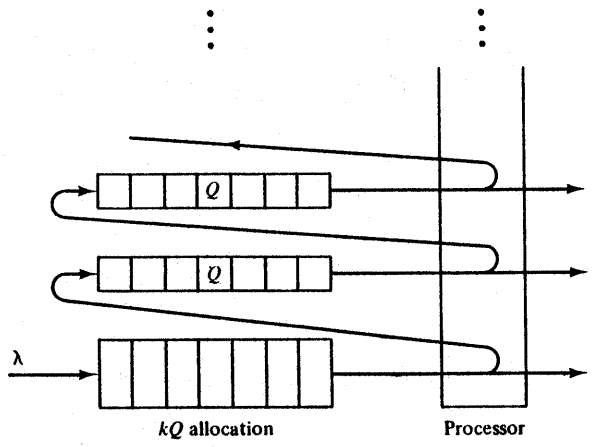

Fig. 4.6-2 Modified SET system.

we have two classes of executing jobs: those receiving an allocation of kQ units of execution time, having just waited through the first queue, and those receiving but one quantum, having just waited through a higher-level queue. The arrival rate to the first queue level in the new system is simply λ. The arrival rate λ_i' to the ith $(i = 2, 3, \ldots)$ queue level is determined by that fraction of the arrivals which require greater than or equal to $(k + i - 1)Q$ units of execution time. Thus

$$\lambda_i' = \lambda \sum_{j=k+i-1}^{\infty} g_j = \lambda[1 - G(k + i - 2)], \quad i = 2, 3, \ldots \tag{4.6.8}$$

Now the arrival process for the first queue is Poisson, but the arrivals to the higher-level queues do not constitute Poisson processes. (Note that arrivals to the higher-level queues occur only within processor busy periods.) However, the tagged job is assumed to be a random (i.e., Poisson) arrival and as a result the arguments leading to (4.2.29) still apply.† Consequently,

$$E'(S_r) = \frac{\lambda}{2}\left[E_k(S^2) + Q^2 \sum_{i=0}^{\infty} [1 - G(k + i)]\right] \tag{4.6.9}$$

Substituting (4.6.7) and (4.6.4) into (4.6.2) we have

$$V_k = \lambda E_{k-1}(S) V_k + \frac{E'(S_r)}{1 - \lambda E_k(S)} + \lambda E_{k-1}(S)(k-1)Q \tag{4.6.10}$$

†We shall be making a similar observation in the next section in the discussion of the shortest-remaining-processing-time discipline. A rigorous justification of this observation can be found in [8].

Solving for V_k and using (4.6.9) we obtain the final result:

$$V_k = \frac{\lambda\left[\mathrm{E}_k(S^2) + Q^2 \sum_{i=k}^{\infty}[1 - G(i)]\right]}{2[1 - \lambda \mathrm{E}_{k-1}(S)][1 - \lambda \mathrm{E}_k(S)]} + \frac{\lambda \mathrm{E}_{k-1}(S)}{1 - \lambda \mathrm{E}_{k-1}(S)}(k-1)Q \tag{4.6.11}$$

Using (4.6.1) we have

$$W_k = \frac{\lambda\left[\mathrm{E}_k(S^2) + Q^2 \sum_{i=k}^{\infty}[1 - G(i)]\right]}{2[1 - \lambda \mathrm{E}_{k-1}(S)][1 - \lambda \mathrm{E}_k(S)]} + \frac{(k-1)Q}{1 - \lambda \mathrm{E}_{k-1}(S)} + Q \tag{4.6.12}$$

The corresponding result for the basic SET discipline and a continuous service time distribution $B(x)$ is easily obtained from (4.6.12) by taking the limit $Q \to 0$ with $t = kQ$ held fixed. Specifically, the mean time in system for a job requiring t time units of service is given by

$$W(t) = \frac{(\lambda/2)\int_0^{\infty} x^2\, dB(x)}{\left[1 - \lambda \int_0^t x\, dB(x) - \lambda t[1 - B(t)]\right]^2} + \frac{t}{1 - \lambda \int_0^t x\, dB(x) - \lambda t[1 - B(t)]} \tag{4.6.13}$$

The results in (4.6.11) and (4.6.12) were obtained by Schrage [24] along with the Laplace transform of the waiting time distributions assuming arbitrary quantum sizes at each level.

As stated earlier the SET discipline always allocates the processor to the job having received the least execution time. In the event of ties the SET rule requires that the set of jobs having received the least execution time must share the processor in the processor-sharing mode discussed in Section 4.4. Also, a new arrival always preempts the job currently sharing the processor. This new arrival retains the processor until it departs, until the next arrival appears, or until it obtains an amount of service equal to that received by the jobs preempted on arrival, whichever occurs first. In the last case it must then share the processor with the jobs preempted at the time of arrival. As a result of the processor sharing produced by the SET discipline the waiting time in queue clearly does not have the same significance that it does in systems without processor sharing.

4.7. THE SHORTEST-REMAINING-PROCESSING-TIME DISCIPLINE

When processing times are known in advance and preemptive disciplines can be employed there are two basic preemptive versions of the SPT rule to consider. The most direct one simply sequences jobs, preempting when necessary, so that the job being executed at each point in time is the one of those currently in the system whose *original* service requirement was least. In the interest of reducing the steady-state mean number in the system we can use a second preemptive version of the SPT rule, viz., the shortest-remaining-processing-time (SRPT) rule. The SRPT discipline sequences jobs so that at each point in time the job on the processor is always the one with the least remaining processing time of all those jobs currently in the system. It is known [25] that the mean number in the system resulting from SRPT sequencing is in fact minimal over all possible service disciplines. Because of this important property, the remainder of this section will be devoted to an analysis of the SRPT discipline.

As before we shall first treat a discrete version of the SRPT discipline in which we assume that jobs require service times that are integral numbers of quanta. However, in the present case it will simplify matters somewhat if we assume that quanta are preemptible; i.e., an arriving job immediately preempts the job on the processor if the former has a service requirement which is less than that remaining for the latter. As before g_i will represent the general probability mass function for the number of quanta required by jobs, and $G(k) = \sum_{i=1}^{k} g_i$ will denote the cumulative distribution function.

A schematic of the discrete SRPT discipline is shown in Fig. 4.7-1, where the Poisson arrival mechanism is decomposed into separate Poisson arrivals to each of the queue levels $1, 2, \ldots, k, \ldots$ with λ_k denoting the arrival rate to the kth queue level. The queue level into which an arrival is placed is determined by the service it requires, jobs requiring k quanta being placed into the kth queue level. Hence $\lambda_k = \lambda g_k$, $k = 1, 2, \ldots$. A job at the service point of the kth queue level does not commence service until the first $k - 1$

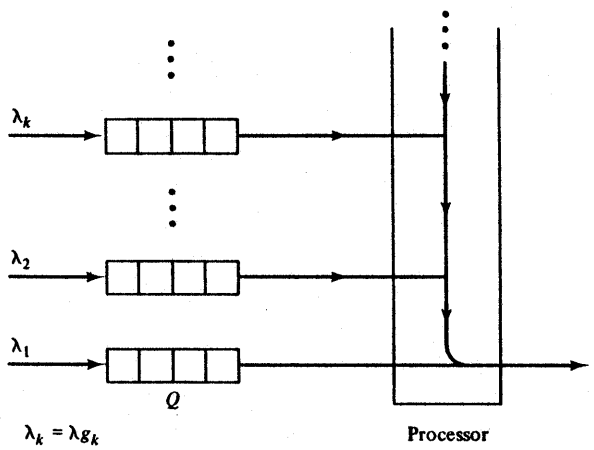

Fig. 4.7-1 The SRPT system.

queue levels are empty. On completing its quantum at the kth level a job immediately begins its next quantum at the $(k-1)$st level. As stated earlier a job is immediately preempted (and put back at the head of the queue determined by its *remaining* processing time) whenever a job arrives at a lower queue level.

We shall now compute the mean time in the system of a random arrival during statistical equilibrium which requires kQ units of execution time. This mean waiting time is decomposed into

$$W_k = W_k' + W_k'' \tag{4.7.1}$$

where W_k' is the mean time elapsing from the moment of arrival until the tagged job enters service for the first time and W_k'' is the mean of the remaining time spent in the system. W_k'' is also called the mean *residence* time.

To calculate W_k' we adopt a stratagem which is similar to that used in the analysis of the SET discipline. Specifically, we consider the modified system shown in Fig. 4.7-2 where the queue levels of Fig. 4.7-1 have been organized into a conventional priority system. The arrivals to a queue level $i \leq k$ require i quanta, but the arrivals to queues above the kth are all assumed to require only k quanta of service. Thus the priority sequencing in the first k queues amounts to SPT sequencing.

On examining this new system we are led to the observation that the mean waiting time *in queue* for an arrival requiring k quanta is precisely W_k', the mean waiting time of the tagged job up to the point when it begins its

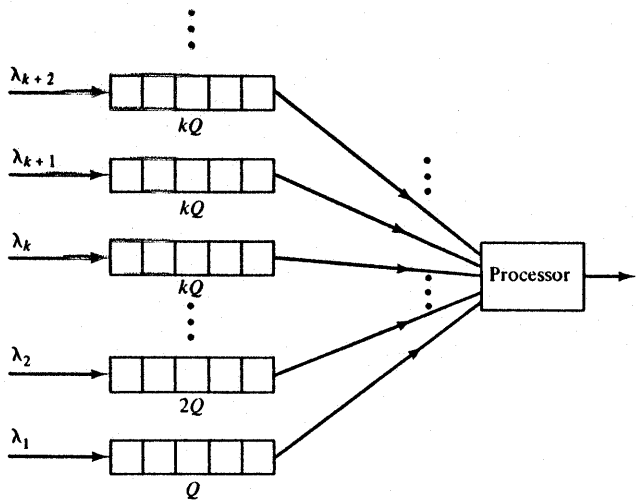

Fig. 4.7-2 A priority system for computing W_k'.

first quantum of service in the original system. In effect, we are observing from the preemption rule that lower-priority arrivals affect the tagged job during its initial wait in queue as though they arrived with a service requirement of only k quanta. The observation is also based on the fact that the jobs originally requiring k or fewer quanta affect (interfere with) the tagged job during its initial wait in queue in precisely the same way that they would affect the tagged job in the analogous, discrete SPT system during the time spent in queue. Thus an analysis of the new system for W_k' follows precisely the lines in Section 4.5 and results in

$$W_k' = \frac{\mathrm{E}'(S_r)}{[1-\lambda H_k(S)][1-\lambda H_{k-1}(S)]} \tag{4.7.2}$$

where

$$H_k(S) = \sum_{i=1}^{k} (iQ) g_i \tag{4.7.3}$$

and $\mathrm{E}'(S_r)$ is the mean time to finish the job (if any) in progress at the time of arrival. In computing $\mathrm{E}'(S_r)$ we must consider two classes of executing jobs: those originally requiring $i \leq k$ quanta and those from the queues beyond the kth which originally required in excess of k quanta in the SRPT system. On the arrival of the tagged job the job in progress would be allocated i quanta and k quanta, respectively. The arrival process for the jobs whose service requirement has been reduced to k quanta in the original system is not Poisson, but as before only the arrival rates need be known. Thus, making use of (4.2.29) applied to the new system, we obtain

$$\mathrm{E}'(S_r) = \frac{\lambda}{2}\left[\sum_{i=1}^{k} (iQ)^2 g_i + (kQ)^2[1-G(k)]\right] = \frac{\lambda \mathrm{E}_k(S^2)}{2} \tag{4.7.4}$$

Having determined W_k' we now calculate the mean residence time W_k''. The tagged job must receive a quantum at each level from 1 to k, and so we can write

$$W_k'' = \sum_{j=1}^{k} (Q + Y_j) \tag{4.7.5}$$

where Y_j is the mean time the tagged job has to wait in the jth queue from the moment it begins to the moment it completes its $(k-j+1)$st quantum. Clearly, the waiting time Y_j is occasioned by the arrival of higher-priority jobs (at levels lower than the jth) which preempt the tagged job. To compute Y_j we take into account separately the arrivals preempting the tagged job during its Q time units of execution and the remaining arrivals occurring during Y_j. Thus

$$Y_j = \sum_{i=1}^{j-1} (\lambda_i Q) iQ + \sum_{i=1}^{j-1} (\lambda_i Y_j)(iQ) \tag{4.7.6}$$

Substituting λg_i for λ_i we have

$$Y_j = \lambda H_{j-1}(S)Q + \lambda H_{j-1}(S)Y_j \tag{4.7.7}$$

where $H_{j-1}(S)$ is given by (4.7.3). Solving (4.7.7) for Y_j yields

$$Y_j = \frac{\lambda H_{j-1}(S)}{1 - \lambda H_{j-1}(S)}Q \tag{4.7.8}$$

whereupon substitution into (4.7.5) gives

$$W_k'' = \sum_{j=1}^{k} \frac{Q}{1 - \lambda H_{j-1}(S)} \tag{4.7.9}$$

Adding (4.7.2) and (4.7.9) we obtain as our final result

$$W_k = \frac{\lambda E_k(S^2)}{2[1 - \lambda H_k(S)][1 - \lambda H_{k-1}(S)]} + \sum_{j=1}^{k} \frac{Q}{1 - \lambda H_{j-1}(S)} \tag{4.7.10}$$

For a continuous service time distribution $B(x)$ we can take the limit $Q \to 0$ of (4.7.10) and obtain the following general result for the SRPT discipline:

$$W(t) = \frac{\lambda \int_0^t x^2\, dB(x) + \lambda t^2[1 - B(t)]}{2[1 - \lambda \int_0^t x\, dB(x)]^2} + \int_0^t \frac{dx}{1 - \lambda \int_0^x y\, dB(y)} \tag{4.7.11}$$

with $V(t) = W(t) - t$. The results of this section can be found in Miller and Schrage [25].

4.8. A COMPARISON OF PROCESSING TIME PRIORITY DISCIPLINES [8, 26]

In the preceding sections we dwelt on the analytical techniques that have been applied to the study of probability models of computer operation. We focused on service disciplines according to which priority decisions are based on the processing times of jobs. In this section our purpose is to describe the relative merits of these disciplines and to identify those system performance criteria for which they are best suited.

We shall restrict our discussion primarily to the disciplines arising as limiting cases when the basic time interval or quantum is made to approach zero. The general properties of the latter disciplines will resemble those of the nonzero quantum systems for suitably small quantum sizes, i.e., for quantum sizes small compared to processing times. As will be seen it is much easier to characterize these disciplines.

First, let us consider the shortest-remaining-processing-time (SRPT) discipline just analyzed in the previous section. Since the SRPT rule assumes complete information on processing times, it is intuitively apparent that SRPT sequencing represents the best one can do in minimizing interdeparture intervals. It would follow, again intuitively, that the SRPT discipline incorporates that rule giving minimum mean waiting times in system. This has in fact been shown for general arrival and service time distributions, even when all arrival times are known in advance. Using Little's result it is observed as a consequence that the mean number in the system is also minimized. In general, this will tend to minimize the storage requirements for jobs; however, an exact statement regarding storage requirements must take into account the correlation, if any, between the size of a job and its execution time.

A limitation of the SRPT rule results from the potentially high cost of preemptions, particularly when this involves transferring jobs between a CPU and auxiliary storage devices. Variations of the SRPT rule which decrease the rate of preemptions, at the expense of a certain increase in mean waiting time, have also been studied [14]. If we must limit ourselves completely to nonpreemptive disciplines, then the SPT rule of Section 4.4 occupies the role for this class of disciplines that the SRPT rule occupies for the class of all disciplines. (Recall that SPT sequencing was also shown to provide for minimum mean time in the system for the deterministic models discussed in Section 3.8.)

The SRPT and SPT disciplines have also been analyzed [8] when processing time information is assumed to be imperfect; i.e., only estimates of processing times are assumed known. The fact that the mean waiting time is significantly reduced with even crude estimates of processing times greatly extends the class of systems to which these rules are applicable. For an interesting quantitative study of the effects of processing-time-dependent disciplines on mean flow time, see Conway et al. [8].

The objectives accomplished by the PS and SET rule are somewhat more subtle to describe. Unlike the SRPT and SPT disciplines the PS and SET disciplines assume that processing times are not known in advance, an assumption that is commonly necessary in modeling computer systems.

To characterize the SET rule we observe first that, without prior knowledge of processing times, the expected job processing time is a nondecreasing function of the elapsed processing time. Thus we see that the SET discipline favors shorter jobs in the sense that it schedules next that job or jobs whose expected total processing time (based on elapsed time) is least among those jobs currently in the system. In the PS system no job characteristics (fixed or time-varying) are explicitly used in making scheduling decisions. Thus no favoritism whatever is shown by the PS rule; all jobs are given an equal share of the processor capacity regardless of their relative characteristics. However, it is true that compared to the usual standard of reference, the FIFO rule, the PS discipline does get the shorter jobs through faster on the average. On

the other hand, the FIFO rule schedules those jobs with the longest waiting times in queue; as a result the shorter jobs experience more interference from the longer jobs than in the PS system where no discrimination at all is shown. Finally, to clarify the distinction between the PS and SET disciplines we can state that the jobs in the PS system at any instant will complete in the order of least *remaining* processing time, whereas the jobs in the SET system at any instant will depart in the order of least *total* processing time.

It is important to realize that the service time distribution has an effect on the performance of the above disciplines. When the service time distribution is exponential any discipline that does not use processing time information yields the same mean time in system. For this case, then, the SET and PS rules do not change the system mean waiting time (although the higher-order moments of the waiting time are affected). For more general service times the mean time in system is affected by the variation in service times. It is intuitively clear that the PS and SET disciplines perform poorly with very small variations in service times. As an extreme case, with no variation (all service times are equal) we note that with the SET rule no job completes until the end of a busy period, while with FIFO all but one job completes earlier. On the other hand, if we have only very short jobs (say 1 second) and very long jobs (say 1 hour), which corresponds to a very large variation, it is clear that the PS and SET disciplines will complete the short jobs quickly, while with FIFO they may be delayed considerably by the longer jobs.

We can make the above remarks more precise for the PS rule by comparing the mean waiting time given in (4.4.17) for the PS system

$$W_{\text{PS}} = \frac{\text{E}(S)}{1-\rho}$$

with the corresponding FIFO result in (4.2.33)

$$W_{\text{FIFO}} = \text{E}(S) + \rho\frac{[1 + C^2(S)]}{2(1-\rho)}\text{E}(S)$$

Thus we have

$$W_{\text{FIFO}} - W_{\text{PS}} = \frac{\rho[C^2(S) - 1]}{2(1-\rho)}\text{E}(S)$$

from which we see the dependence of relative performance on the coefficient of variation C. For the exponential distribution $C(S) = 1$ and $W_{\text{FIFO}} = W_{\text{PS}}$, as stated earlier. For larger variations in service times ($C(S) > 1$) the PS rule gives smaller mean times in the system, whereas the opposite is true for smaller variations in service times. A similar result for the SET discipline is not available but by considering the "extreme-case" service distributions

again it seems likely that

$$W_{\text{SET}} < W_{\text{PS}}, \quad C > 1$$
$$W_{\text{SET}} > W_{\text{PS}}, \quad C < 1$$

In summary, we see that while the PS and SET disciplines have advantages in expediting the service of the shorter jobs they can degrade the response time for a large class of service time distributions. Numerical studies of these disciplines can be found in Coffman and Kleinrock [13] and Schrage [14].

PROBLEMS

✓ **4-1.** Let a cycle of the process $X(t)$ be defined as the time interval composed of an idle period and the subsequent busy period. In statistical equilibrium we observe that the average number of arrivals in a cycle of the M/G/1 system is equal to the average number served in a busy period. Using this observation verify the following expression for the mean length $\text{E}(D)$ of a busy period in equilibrium:

$$\text{E}(D) = \frac{\text{E}(S)}{1 - \lambda\text{E}(S)}$$

Verify this result by computing the first moment from (4.2.46). Also, compute

$$\text{E}(D^2) = \frac{\text{E}(S^2)}{[1 - \lambda\text{E}(S)]^3}$$

4-2. Suppose that arrivals occur in groups but that interarrival times are still exponential with parameter λ. (This is the so-called *bulk arrival* mechanism.) The probability that an arriving group has size k is stationary and given by g_k. Let $G(z) = \sum_{k=1}^{\infty} g_k z^k$ be the corresponding generating function. Using the notation of Section 4.2 show that

$$A(z) = B^*(\lambda - \lambda G(z))$$

Substitution into the expression for $P(z)$ in (4.2.16) gives the solution for $\{\pi_i\}$ in the bulk arrival queue with general service times.

4-3. Using the *preemptive resume* rule the priority discipline of Section 4.5 is modified as follows. Whenever a higher-priority job arrives when a lower-priority job is on the processor, the latter is immediately removed from the processor and replaced at the head of the queue from which it originally came. The higher-priority job begins service immediately, and when the preempted job eventually returns to the processor it commences execution from the point at which it was interrupted by the preemption; i.e., no service