Queueing Systems

One of life's more disagreeable activities, namely, waiting in line, is the delightful subject of this book. One might reasonably ask, "What does it profit a man to study such unpleasant phenomena?" The answer, of course, is that through understanding we gain compassion, and it is exactly this which we need since people will be waiting in longer and longer queues as civilization progresses, and we must find ways to tolerate these unpleasant situations. Think for a moment how much time is spent in one's daily activities waiting in some form of a queue: waiting for breakfast; stopped at a traffic light; slowed down on the highways and freeways; delayed at the entrance to one's parking facility; queued for access to an elevator; standing in line for the morning coffee; holding the telephone as it rings, and so on. The list is endless, and too often also are the queues.

The orderliness of queues varies from place to place around the world. For example, the English are terribly susceptible to formation of orderly queues, whereas some of the Mediterranean peoples consider the idea ludicrous (have you ever tried clearing the embarkation procedure at the Port of Brindisi?). A common slogan in the U.S. Army is, "Hurry up and wait." Such is the nature of the phenomena we wish to study.

1.1. SYSTEMS OF FLOW

Queueing systems represent an example of a much broader class of interesting dynamic systems, which, for convenience, we refer to as "systems of flow." A flow system is one in which some *commodity* flows, moves, or is transferred through one or more finite-capacity *channels* in order to go from one point to another. For example, consider the flow of automobile traffic through a road network, or the transfer of goods in a railway system, or the streaming of water through a dam, or the transmission of telephone or telegraph messages, or the passage of customers through a supermarket checkout counter, or the flow of computer programs through a time-sharing computer system. In these examples the commodities are the automobiles, the goods, the water, the telephone or telegraph messages, the customers, and the programs, respectively; the channel or channels are the road network,

the railway network, the dam, the telephone or telegraph network, the supermarket checkout counter, and the computer processing system, respectively. The "finite capacity" refers to the fact that the channel can satisfy the demands (placed upon it by the commodity) at a finite rate only. It is clear that the analyses of many of these systems require analytic tools drawn from a variety of disciplines and, as we shall see, queueing theory is just one such discipline.

When one analyzes systems of flow, they naturally break into two classes: *steady* and *unsteady* flow. The first class consists of those systems in which the flow proceeds in a predictable fashion. That is, the quantity of flow is exactly known and is constant over the interval of interest; the time when that flow appears at the channel, and how much of a demand that flow places upon the channel is known and constant. These systems are trivial to analyze in the case of a *single channel.* For example, consider a pineapple factory in which empty tin cans are being transported along a conveyor belt to a point at which they must be filled with pineapple slices and must then proceed further down the conveyor belt for additional operations. In this case, assume that the cans arrive at a constant rate of one can per second and that the pineapple-filling operation takes nine-tenths of one second per can. These numbers are constant for all cans and all filling operations. Clearly this system will function in a reliable and smooth fashion as long as the assumptions stated above continue to exist. We may say that the *arrival rate* R is one can per second and the maximum service rate (or *capacity*) C is $1/0.9 = 1.11111\cdots$ filling operations per second. The example above is for the case $R < C$. However, if we have the condition $R > C$, we all know what happens: cans and/or pineapple slices begin to inundate and overflow in the factory! Thus we see that *the mean capacity of the system must exceed the average flow requirements if chaotic congestion is to be avoided*; this is true for all systems of flow. This simple observation tells most of the story. Such systems are of little interest theoretically.

The more interesting case of steady flow is that of a *network* of channels. For stable flow, we obviously require that $R < C$ on each channel in the network. However we now run into some serious combinatorial problems. For example, let us consider a railway network in the fictitious land of Hatafla. See Figure 1.1. The scenario here is that figs grown in the city of Abra must be transported to the destination city of Cadabra, making use of the railway network shown. The numbers on each channel (section of railway) in Figure 1.1 refer to the maximum number of bushels of figs which that channel can handle per day. We are now confronted with the following fig flow problem: How many bushels of figs per day can be sent from Abra to Cadabra and in what fashion shall this flow of figs take place? The answer to such questions of maximal "traffic" flow in a variety of networks is nicely

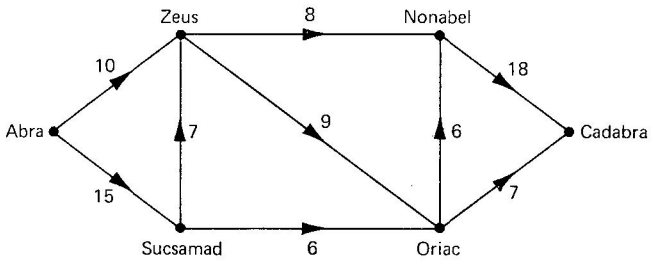

Figure 1.1 Maximal flow problem.

settled by a well-known result in network flow theory referred to as the *max-flow–min-cut* theorem. To state this theorem, we first define a *cut* as a set of channels which, once removed from the network, will separate all possible flow from the origin (Abra) to the destination (Cadabra). We define the *capacity* of such a cut to be the total fig flow that can travel across that cut in the direction from origin to destination. For example, one cut consists of the branches from Abra to Zeus, Sucsamad to Zeus, and Sucsamad to Oriac; the capacity of this cut is clearly 23 bushels of figs per day. The max-flow–min-cut theorem states that the maximum flow that can pass between an origin and a destination is the minimum capacity of all cuts. In our example it can be seen that the maximum flow is therefore 21 bushels of figs per day (work it out). In general, one must consider *all* cuts that separate a given origin and destination. This computation can be enormously time consuming. Fortunately, there exists an extremely powerful method for finding not only what is the maximum flow, but also which flow pattern achieves this maximum flow. This procedure is known as the *labeling algorithm* (due to Ford and Fulkerson [FORD 62]) and is efficient in that the computational requirement grows as a small power of the number of nodes; we present the algorithm in Volume II, Chapter 5.

In addition to maximal flow problems, one can pose numerous other interesting and worthwhile questions regarding flow in such networks. For example, one might inquire into the minimal cost network which will support a given flow if we assign costs to each of the channels. Also, one might ask the same questions in networks when more than one origin and destination exist. Complicating matters further, we might insist that a given network support flow of various kinds, for example, bushels of figs, cartons of cartridges and barrels of oil. This multicommodity flow problem is an extremely difficult one, and its solution typically requires considerable computational effort. These and numerous other significant problems in network flow theory are addressed in the comprehensive text by Frank and Frisch [FRAN 71] and we shall see them again in Volume II, Chapter 5. Network flow theory itself requires methods from graph theory, combinatorial

mathematics, optimization theory, mathematical programming, and heuristic programming.

The *second* class into which systems of flow may be divided is the class of random or stochastic flow problems. By this we mean that the *times* at which demands for service (use of the channel) arrive are uncertain or unpredictable, and also that the size of the demands themselves that are placed upon the channel are unpredictable. The randomness, unpredictability, or unsteady nature of this flow lends considerable complexity to the solution and understanding of such problems. Furthermore, it is clear that most real-world systems fall into this category. Again, the simplest case is that of random flow through a *single* channel; whereas in the case of deterministic or steady flow discussed earlier in which the single-channel problems were trivial, we have now a case where these single-channel problems are extremely challenging and, in fact, techniques for solution to the single-channel or single-server problem comprise much of modern queueing theory.

For example, consider the case of a computer center in which computation requests are served making use of a batch service system. In such a system, requests for computation arrive at unpredictable times, and when they do arrive, they may well find the computer busy servicing other demands. If, in fact, the computer is idle, then typically a new demand will begin service and will be run until it is completed. On the other hand, if the system is busy, then this job will wait on a queue until it is selected for service from among those that are waiting. Until that job is carried to completion, it is usually the case that neither the computation center nor the individual who has submitted the program knows the extent of the demand in terms of computational effort that this program will place upon the system; in this sense the service requirement is indeed unpredictable.

A variety of natural questions present themselves to which we would like intelligent and complete answers. How long, for example, may a job expect to wait on queue before entering service? How many jobs will be serviced before the one just submitted? For what fraction of the day will the computation center be busy? How long will the intervals of continual busy work extend? Such questions require answers regarding the probability of certain periods and numbers or perhaps merely the average values for these quantities. Additional considerations, such as machine breakdown (a not uncommon condition), complicate the issue further; in this case it is clear that some preemptive event prevents the completion of the job currently in service. Other interesting effects can take place where jobs are not serviced according to their order of arrival. Time-shared computer systems, for example, employ rather complex scheduling and servicing algorithms, which, in fact, we explore in Volume II, Chapter 4.

The tools necessary for solving single-channel random-flow problems are

contained and described within queueing theory, to which much of this text devotes itself. This requires a background in probability theory as well as an understanding of complex variables and some of the usual transform-calculus methods; this material is reviewed in Appendices I and II.

As in the case of deterministic flow, we may enlarge our scope of problems to that of *networks* of channels in which random flow is encountered. An example of such a system would be that of a computer network. Such a system consists of computers connected together by a set of communication lines where the capacity of these lines for carrying information is finite. Let us return to the fictitious land of Hatafla and assume that the railway network considered earlier is now in fact a computer network. Assume that users located at Abra require computational effort on the facility at Cadabra. The particular times at which these requests are made are themselves unpredictable, and the commands or instructions that describe these requests are also of unpredictable length. It is these commands which must be transmitted to Cadabra over our communication net as messages. When a message is inserted into the network at Abra, and after an appropriate decision rule (referred to as a routing procedure) is accessed, then the message proceeds through the network along some path. If a portion of this path is busy, and it may well be, then the message must queue up in front of the busy channel and wait for it to become free. Constant decisions must be made regarding the flow of messages and routing procedures. Hopefully, the message will eventually emerge at Cadabra, the computation will be performed, and the results will then be inserted into the network for delivery back at Abra.

It is clear that the problems exemplified by our computer network involve a variety of extremely complex queueing problems, as well as network flow and decision problems. In an earlier work [KLEI 64] the author addressed himself to certain aspects of these questions. We develop the analysis of these systems later in Volume II, Chapter 5.

Having thus classified* systems of flow, we hope that the reader understands where in the general scheme of things the field of queueing theory may be placed. The methods from this theory are central to analyzing most stochastic flow problems, and it is clear from an examination of the current literature that the field and in particular its applications are growing in a viable and purposeful fashion.

* The classification described above places queueing systems within the class of systems of flow. This approach identifies and emphasizes the fields of application for queueing theory. An alternative approach would have been to place queueing theory as belonging to the field of applied stochastic processes; this classification would have emphasized the mathematical structure of queueing theory rather than its applications. The point of view taken in this two-volume book is the former one, namely, with application of the theory as its major goal rather than extension of the mathematical formalism and results.

1.2. THE SPECIFICATION AND MEASURE OF QUEUEING SYSTEMS

In order to completely specify a queueing system, one must identify the stochastic processes that describe the arriving stream as well as the structure and discipline of the service facility. Generally, the arrival process is described in terms of the probability distribution of the *interarrival times* of customers and is denoted $A(t)$, where*

$$A(t) = P[\text{time between arrivals} \leq t] \tag{1.1}$$

The assumption in most of queueing theory is that these interarrival times are independent, identically distributed random variables (and, therefore, the stream of arrivals forms a stationary renewal process; see Chapter 2). Thus, only the distribution $A(t)$, which describes the time between arrivals, is usually of significance. The second statistical quantity that must be described is the amount of demand these arrivals place upon the channel; this is usually referred to as the *service time* whose probability distribution is denoted by $B(x)$, that is,

$$B(x) = P[\text{service time} \leq x] \tag{1.2}$$

Here service time refers to the length of time that a customer spends in the service facility.

Now regarding the structure and discipline of the service facility, one must specify a variety of additional quantities. One of these is the extent of *storage capacity* available to hold waiting customers and typically this quantity is described in terms of the variable K; often K is taken to be infinite. An additional specification involves the *number of service stations* available, and if more than one is available, then perhaps the distribution of service time will differ for each, in which case the distribution $B(x)$ will include a subscript to indicate that fact. On the other hand, it is sometimes the case that the arriving stream consists of more than one identifiable *class* of customers; in such a case the interarrival distribution $A(t)$ as well as the service distribution $B(x)$ may each be characteristic of each class and will be identified again by use of a subscript on these distributions. Another important structural description of a queueing system is that of the queueing *discipline*; this describes the order in which customers are taken from the queue and allowed into service. For example, some standard queueing disciplines are first-come–first-serve (FCFS), last-come–first-serve (LCFS), and random order of service. When the arriving customers are distinguishable according to groups, then we encounter the case of *priority* queueing disciplines in which priority

* The notation $P[A]$ denotes, as usual, the "probability of the event A."

among groups may be established. A further statement regarding the *availability* of the service facility is also necessary in case the service facility is occasionally required to pay attention to other tasks (as, for example, its own breakdown). Beyond this, queueing systems may enjoy customer behavior in the form of *defections* from the queue, *jockeying* among the many queues, *balking* before entering a queue, *bribing* for queue position, *cheating* for queue position, and a variety of other interesting and not-unexpected humanlike characteristics. We will encounter these as we move through the text in an orderly fashion (first-come–first-serve according to page number).

Now that we have indicated how one must specify a queueing system, it is appropriate that we identify the measures of performance and effectiveness that we shall obtain by analysis. Basically, we are interested in the *waiting time* for a customer, the *number of customers* in the system, the *length of a busy period* (the continuous interval during which the server is busy), the *length of an idle period*, and the current *work backlog* expressed in units of time. All these quantities are random variables and thus we seek their complete probabilistic description (i.e., their probability distribution function). Usually, however, to give the distribution function is to give more than one can easily make use of. Consequently, we often settle for the first few moments (mean, variance, etc.).

Happily, we shall begin with simple considerations and develop the tools in a straightforward fashion, paying attention to the essential details of analysis. In the following pages we will encounter a variety of simple queueing problems, simple at least in the sense of description and usually rather sophisticated in terms of solution. However, in order to do this properly, we first devote our efforts in the following chapter to describing some of the important random processes that make up the arrival and service processes in our queueing systems.

REFERENCES

FORD 62 Ford, L. K. and D. R. Fulkerson, *Flows in Networks*, Princeton University Press (Princeton, N.J.), 1962.

FRAN 71 Frank, H. and I. T. Frisch, *Communication, Transportation, and Transmission Networks*, Addison-Wesley (Reading, Mass.), 1971.

KLEI 64 Kleinrock, L., *Communication Nets; Stochastic Message Flow and Delay*, McGraw-Hill (New York), 1964, out of print. Reprinted by Dover (New York), 1972.

2

Some Important Random Processes*

We assume that the reader is familiar with the basic elementary notions, terminology, and concepts of probability theory. The particular aspects of that theory which we require are presented in summary fashion in Appendix II to serve as a review for those readers desiring a quick refresher and reminder; it is recommended that the material therein be reviewed, especially Section II.4 on transforms, generating functions, and characteristic functions.

Included in Appendix II are the following important definitions, concepts, and results:

- Sample space, events, and probability.
- Conditional probability, statistical independence, the law of total probability, and Bayes' theorem.
- A real random variable, its probability distribution function (PDF), its probability density function (pdf), and their simple properties.
- Events related to random variables and their probabilities.
- Joint distribution functions.
- Functions of a random variable and their density functions.
- Expectation.
- Laplace transforms, generating functions, and characteristic functions and their relationships and properties.†
- Inequalities and limit theorems.
- Definition of a stochastic process.

2.1. NOTATION AND STRUCTURE FOR BASIC QUEUEING SYSTEMS

Before we plunge headlong into a step-by-step development of queueing theory from its elementary notions to its intermediate and then finally to some advanced material, it is important first that we understand the basic

* Sections 2.2, 2.3, and 2.4 may be skipped on a first reading.
† Appendix I is a transform theory refresher. This material is also essential to the proper understanding of this text.

structure of queues. Also, we wish to provide the reader a glimpse as to where we are heading in this journey.

It is our purpose in this section to define some notation, both symbolic and graphic, and then to introduce one of the basic stochastic processes that we find in queueing systems. Further, we will derive a simple but significant result, which relates some first moments of importance in these systems. In so doing, we will be in a position to define the quantities and processes that we will spend many pages studying later in the text.

The system we consider is the very general queueing system G/G/m; recall (from the Preface) that this is a system whose interarrival time distribution $A(t)$ is completely arbitrary and whose service time distribution $B(x)$ is also completely arbitrary (all interarrival times and service times are assumed to be independent of each other). The system has m servers and order of service is also quite arbitrary (in particular, it need not be first-come–first-serve). We focus attention on the flow of customers as they arrive, pass through, and eventually leave this system; as such, we choose to number the customers with the subscript n and define C_n as follows:

$$C_n \text{ denotes the } n\text{th customer to enter the system} \tag{2.1}$$

Thus, we may portray our system as in Figure 2.1 in which the box represents the queueing system and the flow of customers both in and out of the system is shown. One can immediately define some random processes of interest. For example, we are interested in $N(t)$ where*

$$N(t) \triangleq \text{number of customers in the system at time } t \tag{2.2}$$

Another stochastic process of interest is the unfinished work $U(t)$ that exists in the system at time t, that is,

$$\begin{aligned} U(t) &\triangleq \text{the unfinished work in the system at time } t \\ &\triangleq \text{the remaining time required to empty the system of all} \\ &\qquad \text{customers present at time } t \end{aligned} \tag{2.3}$$

Whenever $U(t) > 0$, then the system is said to be busy, and only when $U(t) = 0$ is the system said to be idle. The duration and location of these busy and idle periods are also quantities of interest.

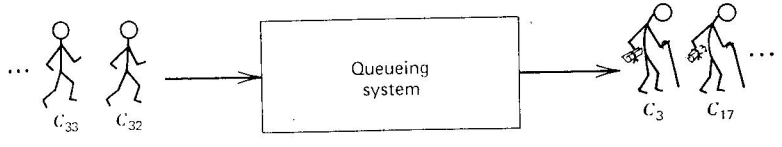

Figure 2.1 A general queueing system.

*The notation $\triangleq$ is to be read as "equals by definition."

The details of these stochastic processes may be observed first by defining the following variables and then by displaying these variables on an appropriate time diagram to be discussed below. We begin with the definitions. Recalling that the nth customer is denoted by C_n, we define his arrival time to the queueing system as

$$\tau_n \triangleq \text{arrival time for } C_n \tag{2.4}$$

and further define the interarrival time between C_{n-1} and C_n as

$$\begin{aligned} t_n &\triangleq \text{interarrival time between } C_{n-1} \text{ and } C_n \\ &= \tau_n - \tau_{n-1} \end{aligned} \tag{2.5}$$

Since we have assumed that all interarrival times are drawn from the distribution $A(t)$, we have that

$$P[t_n \leq t] = A(t) \tag{2.6}$$

which is independent of n. Similarly, we define the service time for C_n as

$$x_n \triangleq \text{service time for } C_n \tag{2.7}$$

and from our assumptions we have

$$P[x_n \leq x] = B(x) \tag{2.8}$$

The sequences $\{t_n\}$ and $\{x_n\}$ may be thought of as input variables for our queueing system; the way in which the system handles these customers gives rise to queues and waiting times that we must now define. Thus, we define the waiting time (time spent in the queue)* as

$$w_n \triangleq \text{waiting time (in queue) for } C_n \tag{2.9}$$

The total time spent in the system by C_n is the sum of his waiting time and service time, which we denote by

$$\begin{aligned} s_n &\triangleq \text{system time (queue plus service) for } C_n \\ &= w_n + x_n \end{aligned} \tag{2.10}$$

Thus we have defined for the nth customer his arrival time, "his" interarrival time, his service time, his waiting time, and his system time. We find it

* The terms "waiting time" and "queueing time" have conflicting definitions within the body of queueing-theory literature. The former sometimes refers to the total time spent in system, and the latter then refers to the total time spent on queue; however, these two definitions are occasionally reversed. We attempt to remove that confusion by defining waiting and queueing time to be the same quantity, namely, the time spent waiting on queue (but not being served); a more appropriate term perhaps would be "wasted time." The total time spent in the system will be referred to as "system time" (occasionally known as "flow time").

expedient at this point to elaborate somewhat further on notation. Let us consider the interarrival time t_n once again. We will have occasion to refer to the limiting random variable $\tilde{t}$ defined by

$$\tilde{t} \triangleq \lim_{n \to \infty} t_n \tag{2.11}$$

which we denote by $t_n \to \tilde{t}$. (We have already required that the interarrival times t_n have a distribution independent of n, but this will not necessarily be the case with many other random variables of interest.) The typical notation for the probability distribution function (PDF) will be

$$P[t_n \leq t] = A_n(t) \tag{2.12}$$

and for the limiting PDF

$$P[\tilde{t} \leq t] = A(t) \tag{2.13}$$

This we denote by $A_n(t) \to A(t)$; of course, for the interarrival time we have assumed that $A_n(t) = A(t)$, which gives rise to Eq. (2.6). Similarly, the probability density function (pdf) for t_n and $\tilde{t}$ will be $a_n(t)$ and $a(t)$, respectively, and will be denoted as $a_n(t) \to a(t)$. Finally, the Laplace transform (see Appendix II) of these pdf's will be denoted by $A_n^*(s)$ and $A^*(s)$, respectively, with the obvious notation $A_n^*(s) \to A^*(s)$. The use of the letter A (and a) is meant as a cue to remind the reader that they refer to the interarrival time. Of course, the moments of the interarrival time are of interest and they will be denoted as follows*:

$$E[t_n] \triangleq \bar{t}_n \tag{2.14}$$

According to our usual notation, the mean interarrival time for the limiting random variable will be given† by $\bar{t}$ in the sense that $\bar{t}_n \to \bar{t}$. As it turns out $\bar{t}$, which is the average interarrival time between customers, is used so frequently in our equations that a *special* notation has been adopted as follows:

$$\bar{t} \triangleq \frac{1}{\lambda} \tag{2.15}$$

Thus λ represents the *average arrival rate* of customers to our queueing system. Higher moments of the interarrival time are also of interest and so we define the kth moment by

$$E[\tilde{t}^k] \triangleq \overline{\tilde{t}^k} \triangleq a_k \qquad k = 0, 1, 2, \ldots \tag{2.16}$$

* The notation $E[\ ]$ denotes the expectation of the quantity within square brackets. As shown, we also adopt the overbar notation to denote expectation.

† Actually, we should use the notation t with a tilde *and* a bar, but this is excessive and will be simplified to $\bar{t}$. The same simplification will be applied to many of our other random variables.

In this last equation we have introduced the definition a_k to be the kth moment of the interarrival time $\tilde{t}$; this is fairly standard notation and we note immediately from the above that

$$\bar{t} = \frac{1}{\lambda} = a_1 \triangleq a \tag{2.17}$$

That is, three *special* notations exist for the mean interarrival time; in particular, the use of the symbol a is very common and various of these forms will be used throughout the text as appropriate. Summarizing the information with regard to the interarrival time we have the following shorthand glossary:

$$t_n = \text{interarrival time between } C_n \text{ and } C_{n-1}$$

$$t_n \to \tilde{t}, \quad A_n(t) \to A(t), \quad a_n(t) \to a(t), \quad A_n^*(s) \to A^*(s)$$

$$\tilde{t}_n \to \tilde{t} = \frac{1}{\lambda} = a_1 = a, \quad \overline{t_n^k} \to \overline{t^k} = a_k \tag{2.18}$$

In a similar manner we identify the notation associated with x_n, w_n, and s_n as follows:

$$x_n = \text{service time for } C_n$$

$$x_n \to \tilde{x}, \quad B_n(x) \to B(x), \quad b_n(x) \to b(x), \quad B_n^*(s) \to B^*(s)$$

$$\bar{x}_n \to \bar{x} = \frac{1}{\mu} = b_1 = b, \quad \overline{x_n^k} \to \overline{x^k} = b_k \tag{2.19}$$

$$w_n = \text{waiting time for } C_n$$

$$w_n \to \tilde{w}, \quad W_n(y) \to W(y), \quad w_n(y) \to w(y), \quad W_n^*(s) \to W^*(s)$$

$$\bar{w}_n \to \bar{w} = W, \quad \overline{w_n^k} \to \overline{w^k} \tag{2.20}$$

$$s_n = \text{system time for } C_n$$

$$s_n \to \tilde{s}, \quad S_n(y) \to S(y), \quad s_n(y) \to s(y), \quad S_n^*(s) \to S^*(s)$$

$$\bar{s}_n \to \bar{s} = T, \quad \overline{s_n^k} \to \overline{s^k} \tag{2.21}$$

All this notation is self-evident except perhaps for the occasional special symbols used for the first moment and occasionally the higher moments of the random variables involved (that is, the use of the symbols λ, a, μ, b, W, and T). The reader is, at this point, directed to the Glossary for a complete set of notation used in this book.

With the above notation we now suggest a *time-diagram notation* for queues, which permits a graphical view of the dynamics of our queueing system and also provides the details of the underlying stochastic processes. This diagram is shown in Figure 2.2. This particular figure is shown for a

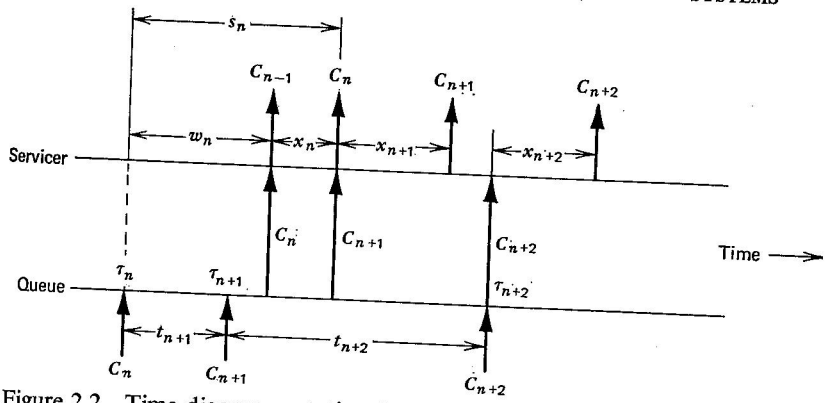

Figure 2.2 Time-diagram notation for queues.

first-come–first-serve order of service, but it is easy to see how the figure may also be made to represent any order of service. In this time diagram the lower horizontal time line represents the queue and the upper horizontal time line represents the service facility; moreover, the diagram shown is for the case of a single server, although this too is easily generalized. An arrow approaching the queue (or service) line from below indicates that an arrival has occurred to the queue (or service facility). Arrows emanating from the line indicate the departure of a customer from the queue (or service facility). In this figure we see that customer C_{n+1} arrives before customer C_n enters service; only when C_n departs from service may C_{n+1} enter service and, of course, these two events occur simultaneously. Notice that when C_{n+2} enters the system he finds it empty and so immediately proceeds through an empty queue directly into the service facility. In this diagram we have also shown the waiting time and the system time for C_n (note that $w_{n+2} = 0$). Thus, as time proceeds we can identify the number of customers in the system $N(t)$, the unfinished work $U(t)$, and also the idle and busy periods. We will find much use for this time-diagram notation in what follows.

In a general queueing system one expects that when the number of customers is large then so is the waiting time. One manifestation of this is a very simple relationship between the mean number in the queueing system, the mean arrival rate of customers to that system, and the mean system time for customers. It is our purpose next to derive that relationship and thereby familiarize ourselves a bit further with the underlying behavior of these systems. Referring back to Figure 2.1, let us position ourselves at the input of the queueing system and count how many customers enter as a function of time. We denote this by $\alpha(t)$ where

$$\alpha(t) \triangleq \text{number of arrivals in } (0, t) \tag{2.22}$$

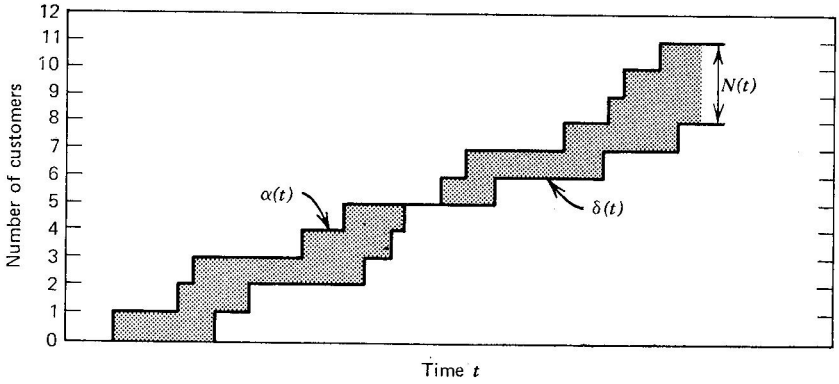

Figure 2.3 Arrivals and departures.

Alternatively, we may position ourselves at the output of the queueing system and count the number of departures that leave; this we denote by

$$\delta(t) \triangleq \text{number of departures in } (0, t) \tag{2.23}$$

Sample functions for these two stochastic processes are shown in Figure 2.3.

Clearly $N(t)$, the number in the system at time t, must be given by

$$N(t) = \alpha(t) - \delta(t)$$

On the other hand, the total area between these two curves up to some point, say t, represents the total time all customers have spent in the system (measured in units of customer-seconds) during the interval $(0, t)$; let us denote this cumulative area by $\gamma(t)$. Moreover, let λ_t be defined as the average arrival rate (customers per second) during the interval $(0, t)$; that is,

$$\lambda_t \triangleq \frac{\alpha(t)}{t} \tag{2.24}$$

We may define T_t as the system time per customer averaged over all customers in the interval $(0, t)$; since $\gamma(t)$ represents the accumulated customer-seconds up to time t, we may divide by the number of arrivals up to that point to obtain

$$T_t = \frac{\gamma(t)}{\alpha(t)}$$

Lastly, let us define $\bar{N}_t$ as the average number of customers in the queueing system during the interval $(0, t)$; this may be obtained by dividing the accumulated number of customer-seconds by the total interval length t

thusly

$$\bar{N}_t = \frac{\gamma(t)}{t}$$

From these last three equations we see

$$\bar{N}_t = \lambda_t T_t$$

Let us now assume that our queueing system is such that the following limits exist as $t \to \infty$:

$$\lambda = \lim_{t \to \infty} \lambda_t$$

$$T = \lim_{t \to \infty} T_t$$

Note that we are using our former definitions for λ and T representing the average customer arrival rate and the average system time, respectively. If these last two limits exist, then so will the limit for $\bar{N}_t$, which we denote by $\bar{N}$ now representing the average number of customers in the system; that is,

$$\bar{N} = \lambda T \qquad \blacksquare (2.25)$$

This last is the result we were seeking and is known as *Little's result*. It states that *the average number of customers in a queueing system is equal to the average arrival rate of customers to that system, times the average time spent in that system.** The above proof does not depend upon any specific assumptions regarding the arrival distribution $A(t)$ or the service time distribution $B(x)$; nor does it depend upon the number of servers in the system or upon the particular queueing discipline within the system. This result existed as a "folk theorem" for many years; the first to establish its validity in a formal way was J. D. C. Little [LITT 61] with some later simplifications by W. S. Jewell [JEWE 67] and S. Eilon [EILO 69]. It is important to note that we have not precisely defined the boundary around our queueing system. For example, the box in Figure 2.1 could apply to the entire system composed of queue and server, in which case $\bar{N}$ and T as defined refer to quantities for the entire system; on the other hand, we could have considered the boundary of the queueing system to contain only the queue itself, in which case the relationship would have been

$$\bar{N}_q = \lambda W \qquad \blacksquare (2.26)$$

where $\bar{N}_q$ represents the average number of customers in the queue and, as defined earlier, W refers to the average time spent waiting in the queue. As a third possible alternative the queueing system defined could have surrounded

* An intuitive proof of Little's result depends on the observation that an arriving customer should *find* the same average number, $\bar{N}$, in the system as he *leaves behind* upon his departure. This latter quantity is simply the arrival rate λ times his average time in system, T.

only the server (or servers) itself; in this case our equation would have reduced to

$$\bar{N}_s = \lambda \bar{x} \tag{2.27}$$

where $\bar{N}_s$ refers to the average number of customers in the service facility (or facilities) and $\bar{x}$, of course, refers to the average time spent in the service box. Note that it is always true that

$$T = \bar{x} + W \tag{2.28}$$

The queueing system could refer to a specific class of customers, perhaps based on priority or some other attribute of this class, in which case the same relationship would apply. In other words, the average arrival rate of customers to a "queueing system" times the average time spent by customers in that "system" is equal to the average number of customers in the "system," regardless of how we define that "system."

We now discuss a basic parameter ρ, which is commonly referred to as the *utilization factor*. The utilization factor is in a fundamental sense really the ratio R/C, which we introduced in Chapter 1. It is the ratio of the rate at which "work" enters the system to the maximum rate (capacity) at which the system can perform this work; the work an arriving customer brings into the system equals the number of seconds of service he requires. So, in the case of a single-server system, the definition for ρ becomes

$$\rho \triangleq \text{(average arrival rate of customers)} \times \text{(average service time)}$$
$$= \lambda \bar{x} \tag{2.29}$$

This last is true since a single-server system has a maximum capacity for doing work, which equals 1 sec/sec and each arriving customer brings an amount of work equal to $\bar{x}$ sec; since, on the average, λ customers arrive per second, then $\lambda \bar{x}$ sec of work are brought in by customers each second that passes, on the average. In the case of multiple servers (say, m servers) the definition remains the same when one considers the ratio R/C, where now the work capacity of the system is m sec/sec; expressed in terms of system parameters we then have

$$\rho \triangleq \frac{\lambda \bar{x}}{m} \tag{2.30}$$

Equations (2.29) and (2.30) apply in the case when the maximum service rate is independent of the system state; if this is not the case, then a more careful definition must be provided. The rate at which work enters the system is sometimes referred to as the *traffic intensity* of the system and is usually expressed in *Erlangs*; in single-server systems, the utilization factor is equal to the traffic intensity whereas for (m) multiple servers, the traffic intensity equals $m\rho$. So long as $0 \leq \rho < 1$, then ρ may be interpreted as

$$\rho = E[\text{fraction of busy servers}] \tag{2.31}$$

[In the case of an infinite number of servers, the utilization factor ρ plays no important part, and instead we are interested in the *number* of busy servers (and its expectation).]

Indeed, for the system G/G/1 to be stable, it must be that $R < C$, that is, $0 \leq \rho < 1$. Occasionally, we permit the case $\rho = 1$ within the range of stability (in particular for the system D/D/1). Stability here once again refers to the fact that limiting distributions for all random variables of interest exist, and that all customers are eventually served. In such a case we may carry out the following simple calculation. We let τ be an arbitrarily long time interval; during this interval we expect (by the law of large numbers) with probability 1 that the number of arrivals will be very nearly equal to $\lambda\tau$. Moreover, let us define p_0 as the probability that the server is idle at some randomly selected time. We may, therefore, say that during the interval τ, the server is busy for $\tau - \tau p_0$ sec, and so with probability 1, the number of customers served during the interval τ is very nearly $(\tau - \tau p_0)/\bar{x}$. We may now equate the number of arrivals to the number served during this interval, which gives, for large τ,

$$\lambda\tau \cong \frac{(\tau - \tau p_0)}{\bar{x}}$$

Thus, as $\tau \to \infty$ we have $\lambda\bar{x} = 1 - p_0$; using Definition (2.29) we finally have the important conclusion for G/G/1

$$\rho = 1 - p_0 \tag{2.32}$$

The interpretation here is that ρ is merely the fraction of time the server is busy; this supports the conclusion in Eq. (2.27) in which $\lambda\bar{x} = \rho$ was shown equal to the average number of customers in the service facility.

This, then, is a rapid look at an overall queueing system in which we have exposed some of the basic stochastic processes, as well as some of the important definitions and notation we will encounter. Moreover, we have established Little's result, which permits us to calculate the average number in the system once we have calculated the average time in the system (or vice versa). Now let us move on to a more careful study of the important stochastic processes in our queueing systems.

2.2*. DEFINITION AND CLASSIFICATION OF STOCHASTIC PROCESSES

At the end of Appendix II a definition is given for a stochastic process, which in essence states that it is a family of random variables $X(t)$ where the

* The reader may choose to skip Sections 2.2, 2.3, and 2.4 at this point and move directly to Section 2.5. He may then refer to this material only as he feels he needs to in the balance of the text.

random variables are "indexed" by the time parameter t. For example, the number of people sitting in a movie theater as a function of time is a stochastic process, as is also the atmospheric pressure in that movie theater as a function of time (at least those functions may be *modeled* as stochastic processes). Often we refer to a stochastic process as a random process. A random process may be thought of as describing the motion of a particle in some space. The classification of a random process depends upon three quantities: the *state space*; the *index (time) parameter*; and the *statistical dependencies* among the random variables $X(t)$ for different values of the index parameter t. Let us discuss each of these in order to provide the general framework for random processes.

First we consider the *state space*. The set of possible values (or states) that $X(t)$ may take on is called its state space. Referring to our analogy with regard to the motion of a particle, if the positions that particle may occupy are finite or countable, then we say we have a *discrete-state* process, often referred to as a *chain*. The state space for a chain is usually the set of integers $\{0, 1, 2, \ldots\}$. On the other hand, if the permitted positions of the particle are over a finite or infinite continuous interval (or set of such intervals), then we say that we have a *continuous-state* process.

Now for the *index (time) parameter*. If the permitted times at which changes in position may take place are finite or countable, then we say we have a *discrete-(time) parameter* process; if these changes in position may occur anywhere within (a set of) finite or infinite intervals on the time axis, then we say we have a *continuous-parameter* process. In the former case we often write X_n rather than $X(t)$. X_n is often referred to as a random or stochastic *sequence* whereas $X(t)$ is often referred to as a random or stochastic *process*.

The truly distinguishing feature of a stochastic process is the relationship of the random variables $X(t)$ or X_n to other members of the same family. As defined in Appendix II, one must specify the complete joint distribution function among the random variables (which we may think of as vectors denoted by the use of boldface) $\mathbf{X} = [X(t_1), X(t_2), \ldots]$, namely,

$$F_{\mathbf{X}}(\mathbf{x}; \mathbf{t}) \triangleq P[X(t_1) \leq x_1, \ldots, X(t_n) \leq x_n]. \tag{2.33}$$

for all $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $\mathbf{t} = (t_1, t_2, \ldots, t_n)$, and n. As mentioned there, this is a formidable task; fortunately, many interesting stochastic processes permit a simpler description. In any case, it is the function $F_{\mathbf{X}}(\mathbf{x}; \mathbf{t})$ that really describes the dependencies among the random variables of the stochastic process. Below we describe some of the usual types of stochastic processes that are characterized by different kinds of dependency relations among their random variables. We provide this classification in order to give the reader a global view of this field so that he may better understand in which particular

regions he is operating as we proceed with our study of queueing theory and its related stochastic processes.

(a) Stationary Processes. As we discuss at the very end of Appendix II, a stochastic process $X(t)$ is said to be stationary if $F_{\mathbf{X}}(\mathbf{x}; \mathbf{t})$ is invariant to shifts in time for all values of its arguments; that is, given any constant τ the following must hold:

$$F_{\mathbf{X}}(\mathbf{x}; \mathbf{t} + \tau) = F_{\mathbf{X}}(\mathbf{x}; \mathbf{t}) \tag{2.34}$$

where the notation $\mathbf{t} + \tau$ is defined as the vector $(t_1 + \tau, t_2 + \tau, \ldots, t_n + \tau)$.

An associated notion, that of *wide-sense stationarity*, is identified with the random process $X(t)$ if merely both the first and second moments are independent of the location on the time axis, that is, if $E[X(t)]$ is independent of t and if $E[X(t)X(t + \tau)]$ depends only upon τ and not upon t. Observe that all stationary processes are wide-sense stationary, but not conversely. The theory of stationary random processes is, as one might expect, simpler than that for nonstationary processes.

(b) Independent Processes. The simplest and most trivial stochastic process to consider is the random sequence in which $\{X_n\}$ forms a set of independent random variables, that is, the joint pdf defined for our stochastic process in Appendix II must factor into the product, thusly

$$\begin{aligned} f_{\mathbf{X}}(\mathbf{x}; \mathbf{t}) &\triangleq f_{X_1 \ldots X_n}(x_1, \ldots, x_n; t_1, \ldots, t_n) \\ &= f_{X_1}(x_1; t_1) \cdots f_{X_n}(x_n; t_n) \end{aligned} \tag{2.35}$$

In this case we are stretching things somewhat by calling such a sequence a random process since there is no structure or dependence among the random variables. In the case of a continuous random process, such an independent process may be defined, and it is commonly referred to as "white noise" (an example is the time derivative of Brownian motion).

(c) Markov Processes. In 1907 A. A. Markov published a paper [MARK 07] in which he defined and investigated the properties of what are now known as Markov processes. In fact, what he created was a simple and highly useful form of dependency among the random variables forming a stochastic process, which we now describe.

A Markov process with a discrete state space is referred to as a Markov chain. The discrete-time Markov chain is the easiest to conceptualize and understand. A set of random variables $\{X_n\}$ forms a Markov chain if the probability that the next value (state) is X_{n+1} depends only upon the current value (state) X_n and not upon any previous values. Thus we have a random sequence in which the dependency extends backwards one unit in time. That

is, the way in which the entire past history affects the future of the process is completely summarized in the current value of the process.

In the case of a discrete-time Markov chain the instants when state changes may occur are preordained to be at the integers $0, 1, 2, \ldots, n, \ldots$. In the case of the continuous-time Markov chain, however, the transitions between states may take place at any instant in time. Thus we are led to consider the random variable that describes how long the process remains in its current (discrete) state before making a transition to some other state. Because the Markov property insists that the past history be completely summarized in the specification of the current state, then we are not free to require that a specification also be given as to how long the process has been in its current state! This imposes a heavy constraint on the distribution of time that the process may remain in a given state. In fact, as we shall see in Eq. (2.85), this state time must be *exponentially* distributed. In a real sense, then, the exponential distribution is a continuous distribution which is "memoryless" (we will discuss this notion at considerable length later in this chapter). Similarly, in the discrete-time Markov chain, the process may remain in the given state for a time that must be *geometrically* distributed; this is the only *discrete* probability mass function that is memoryless. This memoryless property is required of all Markov chains and restricts the generality of the processes one would like to consider.

Expressed analytically the *Markov property* may be written as

$$\begin{aligned} &P[X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \ldots, X(t_1) = x_1] \\ &\quad = P[X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n] \end{aligned} \tag{2.36}$$

where $t_1 < t_2 < \cdots < t_n < t_{n+1}$ and x_i is included in some discrete state space.

The consideration of Markov processes is central to the study of queueing theory and much of this text is devoted to that study. Therefore, a good portion of this chapter deals with discrete-and continuous-time Markov chains.

(d) Birth–death Processes. A very important special class of Markov chains has come to be known as the birth–death process. These may be either discrete-or continuous-time processes in which the defining condition is that state transitions take place between *neighboring* states only. That is, one may choose the set of integers as the discrete state space (with no loss of generality) and then the birth–death process requires that if $X_n = i$, then $X_{n+1} = i - 1$, i, or $i + 1$ and no other. As we shall see, birth–death processes have played a significant role in the development of queueing theory. For the moment, however, let us proceed with our general view of stochastic processes to see how each fits into the general scheme of things.

(e) Semi-Markov Processes. We begin by discussing discrete-time semi-Markov processes. The discrete-time Markov chain had the property that at every unit interval on the time axis the process was required to make a transition from the current state to some other state (possibly back to the same state). The transition probabilities were completely arbitrary; however, the requirement that a transition be made at every unit time (which really came about because of the Markov property) leads to the fact that the time spent in a state is geometrically distributed [as we shall see in Eq. (2.66)]. As mentioned earlier, this imposes a strong restriction on the kinds of processes we may consider. If we wish to relax that restriction, namely, to permit an arbitrary distribution of time the process may remain in a state, then we are led directly into the notion of a discrete-time *semi-Markov process*; specifically, we now permit the times between state transitions to obey an *arbitrary* probability distribution. Note, however, that at the instants of state transitions, the process behaves just like an ordinary Markov chain and, in fact, at those instants we say we have an *imbedded* Markov chain.

Now the definition of a continuous-time semi-Markov process follows directly. Here we permit state transitions at any instant in time. However, as opposed to the Markov process which required an exponentially distributed time in state, we now permit an arbitrary distribution. This then affords us much greater generality, which we are happy to employ in our study of queueing systems. Here, again, the imbedded Markov process is defined at those instants of state transition. Certainly, the class of Markov processes is contained within the class of semi-Markov processes.

(f) Random Walks. In the study of random processes one often encounters a process referred to as a *random walk*. A random walk may be thought of as a particle moving among states in some (say, discrete) state space. What is of interest is to identify the *location* of the particle in that state space. The salient feature of a random walk is that the next position the process occupies is equal to the previous position plus a random variable whose value is drawn independently from an arbitrary distribution; this distribution, however, does not change with the state of the process.* That is, a sequence of random variables $\{S_n\}$ is referred to as a random walk (starting at the origin) if

$$S_n = X_1 + X_2 + \cdots + X_n \qquad n = 1, 2, \ldots \tag{2.37}$$

where $S_0 = 0$ and $X_1, X_2, \ldots$ is a sequence of independent random variables with a common distribution. The index n merely counts the number of state transitions the process goes through; of course, if the instants of these transitions are taken from a discrete set, then we have a discrete-time random

* Except perhaps at some boundary states.

walk, whereas if they are taken from a continuum, then we have a continuous-time random walk. In any case, we assume that the interval between these transitions is distributed in an arbitrary way and so a random walk is a special case of a semi-Markov process.* In the case when the common distribution for X_n is a discrete distribution, then we have a discrete-state random walk; in this case the transition probability p_{ij} of going from state i to state j will depend only upon the difference in indices $j - i$ (which we denote by q_{j-i}).

An example of a continuous-time random walk is that of Brownian motion; in the discrete-time case an example is the total number of heads observed in a sequence of independent coin tosses.

A random walk is occasionally referred to as a process with "independent increments."

(g) Renewal Processes. A renewal process is related† to a random walk. However, the interest is not in following a particle among many states but rather in *counting transitions* that take place as a function of time. That is, we consider the real time axis on which is laid out a sequence of points; the distribution of time between adjacent points is an arbitrary *common* distribution and each point corresponds to an instant of a state transition. We assume that the process begins in state 0 [i.e., $X(0) = 0$] and increases by unity at each transition epoch; that is, $X(t)$ equals the *number* of state transitions that have taken place by t. In this sense it is a special case of a random walk in which $q_1 = 1$ and $q_i = 0$ for $i \neq 1$. We may think of Eq. (2.37) as describing a renewal process in which S_n is the random variable denoting the *time* at which the nth transition takes place. As earlier, the sequence $\{X_n\}$ is a set of independent identically distributed random variables where X_n now represents the time between the $(n-1)$th and nth transition. One should be careful to distinguish the interpretation of Eq. (2.37) when it applies to renewal processes as here and when it applies to a random walk as earlier. The difference is that here in the renewal process the equation describes the *time* of the nth renewal or transition, whereas in the random walk it describes the *state* of the process and the time between state transitions is some other random variable.

An important example of a renewal process is the set of arrival instants to the G/G/m queue. In this case, X_n is identified with the interarrival time.

* Usually, the distribution of time between intervals is of little concern in a random walk; emphasis is placed on the value (position) S_n after n transitions. Often, it is assumed that this distribution of interval time is memoryless, thereby making the random walk a special case of Markov processes; we are more generous in our definition here and permit an arbitrary distribution.

† It may be considered to be a special case of the random walk as defined in (f) above. A renewal process is occasionally referred to as a *recurrent* process.

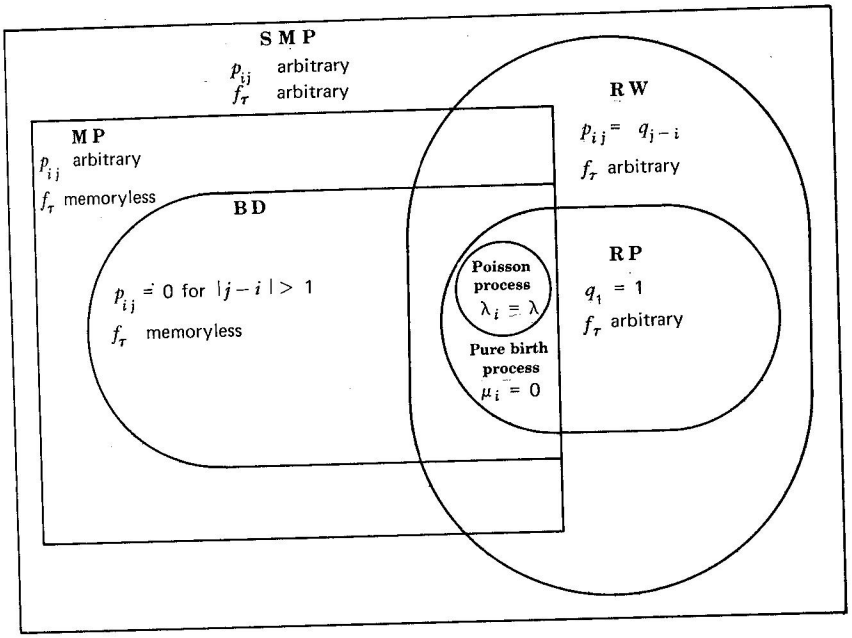

Figure 2.4 Relationships among the interesting random processes. **SMP**: Semi-Markov process; **MP**: Markov process; **RW**: Random walk; **RP**: Renewal process; **BD**: Birth-Death Process.

So there we have it—a self-consistent classification of some interesting stochastic processes. In order to aid the reader in understanding the relationship among Markov processes, semi-Markov processes, and their special cases, we have prepared the diagram of Figure 2.4, which shows this relationship for discrete-state systems. The figure is in the form of a Venn diagram. Moreover, the symbol p_{ij} denotes the probability of making a transition next to state j given that the process is currently in state i. Also, f_τ denotes the distribution of time between transitions; to say that "f_τ is memoryless" implies that if it is a discrete-time process, then f_τ is a geometric distribution, whereas if it is a continuous-time process, then f_τ is an exponential distribution. Furthermore, it is implied that f_τ may be a function both of the current and the next state for the process.

The figure shows that birth–death processes form a subset of Markov processes, which themselves form a subset of the class of semi-Markov processes. Similarly, renewal processes form a subset of random walk processes which also are a subset of semi-Markov processes. Moreover, there are some renewal processes that may also be classified as birth–death

processes. Similarly, those Markov processes for which $p_{ij} = q_{j-i}$ (that is, where the transition probabilities depend only upon the difference of the indices) overlap those random walks where f_τ is memoryless. A random walk for which f_τ is memoryless and for which $q_{j-i} = 0$ when $|j - i| > 1$ overlaps the class of birth–death processes. If in addition to this last requirement our random walk has $q_1 = 1$, then we have a process that lies at the intersection of all five of the processes shown in the figure. This is referred to as a "pure birth" process; although f_τ must be memoryless, it may be a distribution which depends upon the state itself. If f_τ is independent of the state (thus giving a constant "birth rate") then we have a process that is figuratively and literally at the "center" of the study of stochastic processes and enjoys the nice properties of each! This very special case is referred to as the *Poisson process* and plays a major role in queueing theory. We shall develop its properties later in this chapter.

So much for the classification of stochastic processes at this point. Let us now elaborate upon the definition and properties of discrete-state Markov processes. This will lead us naturally into some of the elementary queueing systems. Some of the required theory behind the more sophisticated continuous-state Markov processes will be developed later in this work as the need arises. We begin with the simpler discrete-state, discrete-time Markov chains in the next section and follow that with a section on discrete-state, continuous-time Markov chains.

2.3. DISCRETE-TIME MARKOV CHAINS*

As we have said, Markov processes may be used to describe the motion of a particle in some space. We now consider discrete-time Markov chains, which permit the particle to occupy discrete positions and permit transitions between these positions to take place only at discrete times. We present the elements of the theory by carrying along the following contemporary example.

Consider the hippie who hitchhikes from city to city across the country. Let X_n denote the city in which we find our hippie at noon on day n. When he is in some particular city i, he will accept the first ride leaving in the evening from that city. We assume that the travel time between any two cities is negligible. Of course, it is possible that no ride comes along, in which case he will remain in city i until the next evening. Since vehicles heading for various neighboring cities come along in some unpredictable fashion, the hippie's position at some time in the future is clearly a random variable. It turns out that this random variable may properly be described through the use of a Markov chain.

* See footnote on p. 19.

APPENDIX II

Probability Theory Refresher

In this appendix we review selected topics from probability theory, which are relevant to our discussion of queueing systems. Mostly, we merely list the important definitions and results with an occasional example. The reader is expected to be familiar with this material, which corresponds to a good first course in probability theory. Such a course would typically use one of the following texts that contain additional details and derivations: Feller, Volume I [FELL 68]; Papoulis [PAPO 65]; Parzen [PARZ 60]; or Davenport [DAVE 70].

Probability theory concerns itself with describing random events. A typical dictionary definition of a random event is an event lacking aim, purpose, or regularity. Nothing could be further from the truth! In fact, it is the *extreme regularity* that manifests itself in collections of random events, that makes probability theory interesting and useful. The notion of statistical regularity is central to our studies. For example, if one were to toss a fair coin four times, one expects on the average two heads and two tails. Of course, there is one chance in sixteen that no heads will occur. As a consequence, if an unusual sequence came up (that is, no heads), we would not be terribly surprised nor would we suspect the coin was unfair. On the other hand, if we tossed the coin a million times, then once again we expect approximately half heads and half tails, but in this case, if no heads occurred, we would be more than surprised, we would be indignant and with overwhelming assurance could state that this coin was clearly unfair. In fact, the odds are better than 10^{88} to 1 that at least 490,000 heads will occur! This is what we mean by statistical regularity, namely, that we can make some very precise statements about large collections of random events.

II.1. RULES OF THE GAME

We now describe the rules of the game for creating a mathematical model for probabilistic situations, which is to correspond to real-world experiments. Typically one examines three features of such experiments:

1. A set of possible experimental *outcomes*.