WILEY SERIES IN PROBABILITY AND STATISTICS

FUNDAMENTALS OF QUEUEING THEORY

Donald Gross • John F. Shortle
James M. Thompson • Carl M. Harris

FOURTH EDITION

WILEY

WWW. LINK AVAILABLE

Fundamentals of Queueing Theory

WILEY SERIES IN PROBABILITY AND STATISTICS

Established by WALTER A. SHEWHART and SAMUEL S. WILKS

A complete list of the titles in this series appears at the end of this volume.

Fundamentals of Queueing Theory

Fourth Edition

Donald Gross
George Mason University
Fairfax, Virginia

John F. Shortle
George Mason University
Fairfax, Virginia

James M. Thompson
Freddie Mac Corporation
McLean, Virginia

Carl M. Harris

WILEY

A JOHN WILEY & SONS, INC., PUBLICATION

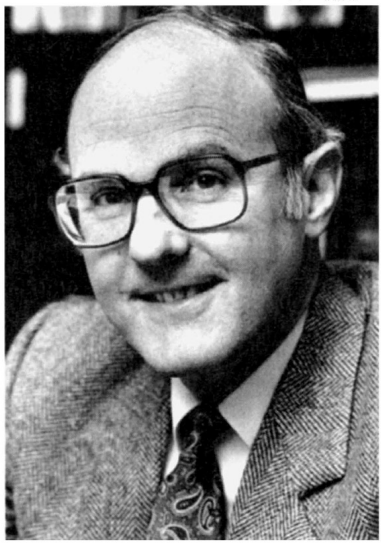
Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Gross, Donald.
Fundamentals of queueing theory / Donald Gross, John F. Shortle, Carl M. Harris. — 4th ed.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-471-79127-0 (cloth)
1. Queuing theory. I. Shortle, John F., 1969– II. Harris, Carl M., 1940– III. Title.
T57.9.G76 2008
519.8'2—dc22 2008003734

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1



Carl Harris, 1940–2000

This book is dedicated to the memory of Carl M. Harris. Carl and I first entertained the idea of a queueing text back in 1968 and collaborated on the first three editions. We were friends and colleagues from that time until his untimely death from a heart attack while exercising at a local gym, one month after his 60th birthday.

Carl was the BDM International Professor of Operations Research and the founding chair of the Systems Engineering and Operations Research Department for the Volgenau School of Information Technology and Engineering at George Mason University, Fairfax, Virginia. In 1999, he was awarded the Institute for Operations Research and the Management Sciences (INFORMS) Kimball Medal in recognition of distinguished service to the Operations Research profession and the Operations Research Society of America (INFORMS' predecessor). He was the society's 39th president. Carl's research interests were in the areas of applied probability and statistics, particularly queueing theory and stochastic processes. He authored or co-authored about 80 scholarly papers (on many of which I was fortunate enough to have worked with him.) In addition to the first 3 editions of this book, he co-authored with Saul I. Gass, *The Encyclopedia of Operations Research and Management Science.*

His warmth, collegiality, and friendship are sorely missed.

DONALD GROSS

CONTENTS

PREFACE

The changes in this fourth edition reflect the feedback from numerous students, teachers, and colleagues since the third edition came out ten years ago. Almost all the material from the third edition is kept in the fourth, however, with a fair amount of editing and reorganization.

Chapter 2 contains a new section on choosing the number of servers and a new subsection on computational issues of the Erlang B formula. The chapter now begins with a section on birth–death processes, which was the old Section 1.10 from the previous edition. Chapter 3 is substantially edited and contains a new section on retrial queues. Chapter 5 contains an expanded discussion of the level crossing method developed by Percy Brill. Chapter 7 is now split into two separate chapters: Chapter 7, Bounds and Approximations, and Chapter 8, Numerical Techniques and Simulation. Chapter 7 includes a new section on network approximations, and Chapter 8 includes a new section on numerical inversion of transforms.

Two appendices are added back to this edition, one on transforms and generating functions and the other on differential and difference equations (by popular request). The appendix on the QtsPlus software is completely rewritten to reflect the changes and expansion made to the software. Also, a subsection on how to use the software is added to Chapter 1. Finally, many more examples and problems are added. In this edition we do not denote which problems are solvable on the computer — we leave that up to the discretion of the student and/or instructor. We also do not include here

a table on suggested text material for various course lengths (quarter, semester, etc.). Again, we believe that the instructor is the best one to decide.

For errata, updates, and other information about the text and associated QtsPlus software, see the text website:

<http://mason.gmu.edu/~jshortle/fqt4th.html>.

Donald Gross
John F. Shortle
James M. Thompson

Fairfax, Virginia
March 2008

ACKNOWLEDGMENTS

We are grateful for the assistance given to us over the years by many professional colleagues and students, whose numerous comments and suggestions have been so helpful in improving our work. We particularly thank Prof. Andrew Ross, Eastern Michigan University, Prof. Percy Brill, University of Windsor, and Prof. John Mullen, New Mexico State University, for their very detailed comments, many of which are reflected in this edition. We also appreciate the help and encouragement of our research colleagues, Dr. Martin Fischer and Dr. Denise Masi of Noblis (formerly Mitretek Systems) and Prof. Brian Mark of George Mason University.

With heartfelt thanks, we extend special appreciation once more to our families for their unlimited and continuing encouragement and to all the people at John Wiley & Sons who have been wonderfully supportive of us through these four editions. We also appreciate the support of the Volgenau School of Information Technology and Engineering and the Department of Systems Engineering and Operations Research at George Mason University.

D. G.
J. F. S.
J. M. T.

CHAPTER 1

INTRODUCTION

All of us have experienced the annoyance of having to wait in line. Unfortunately, this phenomenon continues to be common in congested, urbanized, "high-tech" societies. We wait in line in our cars in traffic jams or at toll booths; we wait on hold for an operator to pick up our telephone calls; we wait in line at supermarkets to check out; we wait in line at fast-food restaurants; and we wait in line at banks and post offices. We, as customers, do not generally like these waits, and the managers of the establishments at which we wait also do not like us to wait, since it may cost them business. Why then is there waiting?

The answer is simple: There is more demand for service than there is facility for service available. Why is this so? There may be many reasons; for example, there may be a shortage of available servers, it may be infeasible economically for a business to provide the level of service necessary to prevent waiting, or there may be a space limit to the amount of service that can be provided. Generally these limitations can be removed with the expenditure of capital, and to know how much service should then be made available, one would need to know answers to such questions as, "How long must a customer wait?" and "How many people will form in the line?" Queueing theory attempts (and in many cases succeeds) to answer these questions through detailed mathematical analysis. The word "queue" is in more common usage

Fundamentals of Queueing Theory, Fourth Edition.
By D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris

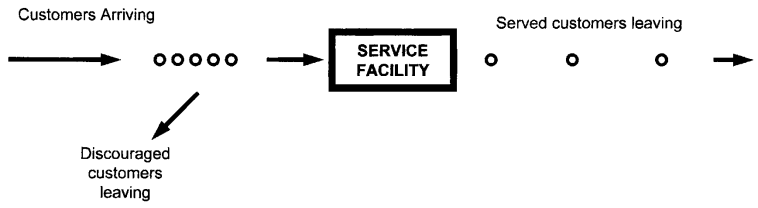


Figure 1.1 A typical queueing process.

in Great Britain and other countries than in the United States, but it is rapidly gaining acceptance in this country, although it must be admitted that it is just as displeasing to spend time in a queue as in a waiting line.

1.1 Description of the Queueing Problem

A queueing system can be described as customers arriving for service, waiting for service if it is not immediate, and if having waited for service, leaving the system after being served. The term "customer" is used in a general sense and does not imply necessarily a human customer. For example, a customer could be a ball bearing waiting to be polished, an airplane waiting in line to take off, or a computer program waiting to be run. Such a basic system can be schematically shown as in Figure 1.1. Although any queueing system may be diagrammed in this manner, it should be clear that a reasonably accurate representation of such a system would require a detailed characterization of the underlying processes.

Queueing theory was developed to provide models to predict the behavior of systems that attempt to provide service for randomly arising demands; not unnaturally, then, the earliest problems studied where those of telephone traffic congestion. The pioneer investigator was the Danish mathematician A. K. Erlang, who, in 1909, published "The Theory of Probabilities and Telephone Conversations." In later works he observed that a telephone system was generally characterized by either (1) Poisson input, exponential holding (service) times, and multiple channels (servers), or (2) Poisson input, constant holding times, and a single channel. Erlang was also responsible for the notion of stationary equilibrium, for the introduction of the so-called balance-of-state equations, and for the first consideration of the optimization of a queueing system.

Work on the application of the theory to telephony continued after Erlang. In 1927, E. C. Molina published his paper "Application of the Theory of Probability to Telephone Trunking Problems," which was followed one year later by Thornton Fry's book *Probability and Its Engineering Uses*, which expanded much of Erlang's earlier work. In the early 1930s, Felix Pollaczek did some further pioneering work on Poisson input, arbitrary output, and single- and multiple-channel problems. Additional work was done at that time in Russia by Kolmogorov and Khintchine, in France by Crommelin, and in Sweden by Palm. The work in queueing theory picked

up momentum rather slowly in its early days, but accelerated in the 1950s, and there has been a great deal of work in the area since then.

There are many valuable applications of the theory, most of which have been well documented in the literature of probability, operations research, management science, and industrial engineering. Some examples are traffic flow (vehicles, aircraft, people, communications), scheduling (patients in hospitals, jobs on machines, programs on a computer), and facility design (banks, post offices, amusement parks, fast-food restaurants).

Queueing theory originated as a very practical subject, but much of the literature up to the middle 1980s was of little direct practical value. However, queueing theorists have once again become concerned about the application of the sophisticated theory that has largely arisen since the close of World War II. The emphasis in the literature on the exact solution of queueing problems with clever mathematical tricks is now becoming secondary to model building and the direct use of these techniques in management decisionmaking. Most real problems do not correspond exactly to a mathematical model, and increasing attention is being paid to complex computational analysis, approximate solutions, sensitivity analyses, and the like. The development of the practice of queueing theory must not be restricted by a lack of closed-form solutions, and problem solvers must be able to put the developed theory to good use. These points should be kept in mind by the reader, and we attempt to illustrate them whenever possible throughout this text.

1.2 Characteristics of Queueing Processes

In most cases, six basic characteristics of queueing processes provide an adequate description of a queueing system: (1) arrival pattern of customers, (2) service pattern of servers, (3) queue discipline, (4) system capacity, (5) number of service channels, and (6) number of service stages.

1.2.1 Arrival Pattern of Customers

In usual queueing situations, the process of arrivals is stochastic, and it is thus necessary to know the probability distribution describing the times between successive customer arrivals (interarrival times). It is also necessary to know whether customers can arrive simultaneously (batch or bulk arrivals), and if so, the probability distribution describing the size of the batch.

It is also necessary to know the reaction of a customer upon entering the system. A customer may decide to wait no matter how long the queue becomes, or, on the other hand, if the queue is too long, the customer may decide not to enter the system. If a customer decides not to enter the queue upon arrival, the customer is said to have *balked*. A customer may enter the queue, but after a time lose patience and decide to leave. In this case, the customer is said to have *reneged*. In the event that there are two or more parallel waiting lines, customers may switch from one to another, that is,

jockey for position. These three situations are all examples of queues with *impatient customers*.

One final factor to be considered regarding the arrival pattern is the manner in which the pattern changes with time. An arrival pattern that does not change with time (i.e., the probability distribution describing the input process is time-independent) is called a *stationary* arrival pattern. One that is not time-independent is called *nonstationary*.

1.2.2 Service Patterns

Much of the previous discussion concerning the arrival pattern is appropriate in discussing service. Most importantly, a probability distribution is needed to describe the sequence of customer service times. Service may also be single or batch. One generally thinks of one customer being served at a time by a given server, but there are many situations where customers may be served simultaneously by the same server, such as a computer with parallel processing, sightseers on a guided tour, or people boarding a train.

The service process may depend on the number of customers waiting for service. A server may work faster if the queue is building up or, on the contrary, may get flustered and become less efficient. The situation in which service depends on the number of customers waiting is referred to as *state-dependent* service. Although this term was not used in discussing arrival patterns, the problems of customer impatience can be looked upon as ones of state-dependent arrivals, since the arrival behavior depends on the amount of congestion in the system.

Service, like arrivals, can be stationary or nonstationary with respect to time. For example, learning may take place, so that service becomes more efficient as experience is gained. The dependence on time is not to be confused with dependence on state. The former does not depend on the number of customers in the system, but rather on how long it has been in operation. The latter does not depend on how long the system has been in operation, but only on the state of the system at a given time, that is, on how many customers are currently in the system. Of course, a queueing system can be both nonstationary and state-dependent.

Even if the service rate is high, it is very likely that some customers will be delayed by waiting in the line. In general, customers arrive and depart at irregular intervals; hence the queue length will assume no definitive pattern unless arrivals and service are deterministic. Thus it follows that a probability distribution for queue lengths will be the result of two separate processes—arrivals and services—which are generally, though not universally, assumed mutually independent.

1.2.3 Queue Discipline

Queue discipline refers to the manner in which customers are selected for service when a queue has formed. The most common discipline that can be observed in everyday life is first come, first served (FCFS). However, this is certainly not the only possible queue discipline. Some others in common usage are last come, first

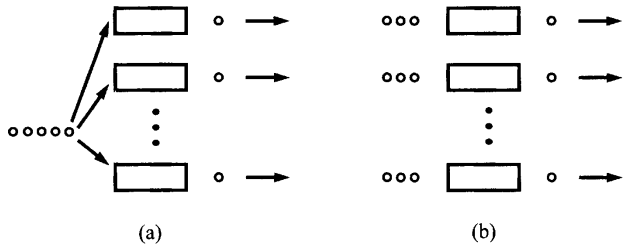


Figure 1.2 Multichannel queueing systems.

served (LCFS), which is applicable to many inventory systems when there is no obsolescence of stored units, as it is easier to reach the nearest items, which are the last in; selection for service in random order independent of the time of arrival to the queue (RSS); and a variety of *priority* schemes, where customers are given priorities upon entering the system, the ones with higher priorities to be selected for service ahead of those with lower priorities, regardless of their time of arrival to the system.

There are two general situations in priority disciplines. In the first, which is called *preemptive*, the customer with the highest priority is allowed to enter service immediately even if a customer with lower priority is already in service when the higher-priority customer enters the system; that is, the lower-priority customer in service is preempted, its service stopped, to be resumed again after the higher-priority customer is served. There are two possible additional variations: the preempted customer's service when resumed can either continue from the point of preemption or start anew. In the second general priority situation, called the *nonpreemptive* case, the highest-priority customer goes to the head of the queue but cannot get into service until the customer presently in service is completed, even though this customer has a lower priority.

1.2.4 System Capacity

In some queueing processes there is a physical limitation to the amount of waiting room, so that when the line reaches a certain length, no further customers are allowed to enter until space becomes available as the result of a service completion. These are referred to as finite queueing situations; that is, there is a finite limit to the maximum system size. A queue with limited waiting room can be viewed as one with forced balking where a customer is forced to balk if it arrives when the queue size is at its limit. This is a simple case, since it is known exactly under what circumstances arriving customers must balk.

1.2.5 Number of Service Channels

As we shortly explain in more detail, it is generally preferable to design multiserver queueing systems to be fed by a single line. Thus, when we specify the number of service channels, we are typically referring to the number of parallel service stations

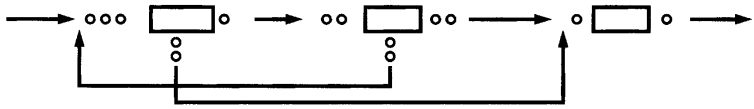


Figure 1.3 A multistage queueing system with feedback.

that can serve customers simultaneously. Figure 1.1 depicts an illustrative single-channel system, while Figure 1.2 shows two variations of multichannel systems. The two multichannel systems differ in that the first has a single queue, while the second allows a queue for each channel. A hair-styling salon with many chairs is an example of the first type of multichannel system (assuming no customer is waiting for any particular stylist), while a supermarket or fast-food restaurant might fit the second description. It is generally assumed that the service mechanisms of parallel channels operate independently of each other.

1.2.6 Stages of Service

A queueing system may have only a single stage of service, as in the hair-styling salon, or it may have several stages. An example of a multistage queueing system would be a physical examination procedure, where each patient must proceed through several stages, such as medical history; ear, nose, and throat examination; blood tests; electrocardiogram; eye examination; and so on. In some multistage queueing processes recycling or feedback may occur. Recycling is common in manufacturing processes, where quality control inspections are performed after certain stages, and parts that do not meet quality standards are sent back for reprocessing. Similarly, a telecommunications network may process messages through a randomly selected sequence of nodes, with the possibility that some messages will require rerouting on occasion through the same stage. A multistage queueing system with some feedback is depicted in Figure 1.3.

The six characteristics of queueing systems discussed in this section are generally sufficient to completely describe a process under study. Clearly, a wide variety of queueing systems can be encountered. Before performing any mathematical analyses, however, it is absolutely necessary to describe adequately the process being modeled. Knowledge of the basic six characteristics is essential in this task.

It is extremely important to use the correct model or at least the model that best describes the real situation being studied. A great deal of thought is often required in this *model selection procedure*. For example, let us reconsider the supermarket mentioned previously. Suppose there are c checkout counters. If customers choose a checkout counter on a purely random basis (without regard to the queue length in front of each counter) and never switch lines (no jockeying), then we truly have c independent single-channel models. If, on the other hand, there is a single waiting line and when a checker becomes idle, the customer at the head of the line (or with the lowest number if numbers are given out) enters service, we have a c-channel model. Neither, of course, is generally the case in most supermarkets. What usually happens

is that queues form in front of each counter, but new customers enter the queue that is the shortest (or has shopping carts that are lightly loaded). Also, there is a great deal of jockeying between lines. Now the question becomes which choice of models (c independent single channels or a single c-channel) is more appropriate. If there were complete jockeying, the single c-channel model would be quite appropriate, since even though in reality there are c lines, there is little difference, when jockeying is present, between these two cases. This is so because no servers will be idle as long as customers are waiting for service, which would not be the case with c truly independent single channels. As jockeying is rather easy to accomplish in supermarkets, the c-channel model will be more appropriate and realistic than the c-single-channels model, which one might have been tempted to choose initially prior to giving much thought to the process. Thus it is important not to jump to hasty conclusions but to select carefully the most appropriate model.

1.3 Notation

As a shorthand for describing queueing processes, a notation has evolved, due for the most part to Kendall (1953), which is now rather standard throughout the queueing literature. A queueing process is described by a series of symbols and slashes such as $A/B/X/Y/Z$, where A indicates in some way the interarrival-time distribution, B the service pattern as described by the probability distribution for service time, X the number of parallel service channels, Y the restriction on system capacity, and Z the queue discipline (Appendix 1 contains a dictionary of symbols used throughout this text). Some standard symbols for these characteristics are presented in Table 1.1. For example, the notation $M/D/2/\infty/\text{FCFS}$ indicates a queueing process with exponential interarrival times, deterministic service times, two parallel servers, no restriction on the maximum number allowed in the system, and first-come, first-served queue discipline.

In many situations only the first three symbols are used. Current practice is to omit the service-capacity symbol if no restriction is imposed ($Y = \infty$) and to omit the queue discipline if it is first come, first served ($Z = \text{FCFS}$). Thus $M/D/2$ would be a queueing system with exponential input, deterministic service, two servers, no limit on system capacity, and first-come, first-served discipline.

The symbols in Table 1.1 are, for the most part, self-explanatory; however, a few require further comment. The symbol G represents a general probability distribution; that is, no assumption is made as to the precise form of the distribution. Results in these cases are applicable to any probability distribution. These general-time distributions, however, are required to represent independent and identically distributed random variables.

It may also appear strange that the symbol M is used for exponential. The use of the symbol E, as one might expect, would be too easily confused with E_k, which is used for the type-k Erlang distribution (a gamma with an integer shape parameter). So M is used instead; it stands for the Markovian or memoryless property of the exponential, which is developed in some detail in Section 1.9.

Table 1.1 Queueing Notation $A/B/X/Y/Z$

Characteristic	Symbol	Explanation
Interarrival-time distribution (A)	M	Exponential
	D	Deterministic
	E_k	Erlang type $k (k = 1, 2, \ldots)$
Service-time distribution (B)	H_k	Mixture of k exponentials
	PH	Phase type
	G	General
# of parallel servers (X)	$1, 2, \ldots, \infty$	
Max. system capacity (Y)	$1, 2, \ldots, \infty$	
Queue discipline (Z)	FCFS	First come, first served
	LCFS	Last come, first served
	RSS	Random selection for service
	PR	Priority
	GD	General discipline

The reader may have noticed that the list of symbols is not complete. For example, there is no indication of a symbol to represent bulk arrivals, to represent series queues, to denote any state dependence, and so on. If a suitable notation does exist for any previously unmentioned model, it is indicated when that particular model is brought up in the text. However, there still remain models for which no symbolism has either been developed or accepted as standard, and this is generally true for those models less frequently analyzed in the literature.

1.4 Measuring System Performance

Up to now the concentration has been on the physical description of queueing processes. What, then, might one like to know about the effectiveness of a queueing system? Generally there are three types of system responses of interest: (1) some measure of the waiting time that a typical customer might be forced to endure; (2) an indication of the manner in which customers may accumulate; and (3) a measure of the idle time of the servers. Since most queueing systems have stochastic elements, these measures are often random variables and their probability distributions, or at the very least their expected values, are desired.

There are two types of customer waiting times, the time a customer spends in the queue and the total time a customer spends in the system (queue plus service). Depending on the system being studied, one may be of more interest than the other. For example, if we are studying an amusement park, it is the time waiting in the queue that makes the customer unhappy. On the other hand, if we are dealing with

machines that require repair, then it is the total down time (queue wait plus repair time) that we wish to keep as small as possible. Correspondingly, there are two customer accumulation measures as well: the number of customers in the queue and the total number of customers in the system. The former would be of interest if we desire to determine a design for waiting space (say, the number of seats to have for customers waiting in a hair-styling salon), while the latter may be of interest for knowing how many of our machines may be unavailable for use. Idle-service measures can include the percentage of time any particular server may be idle, or the time the entire system is devoid of customers.

The task of the queueing analyst is generally one of two things. He or she is either to determine the values of appropriate measures of effectiveness for a given process, or to design an "optimal" (according to some criterion) system. To do the former, one must relate waiting delays, queue lengths, and such to the given properties of the input stream and the service procedures. On the other hand, for the design of a system the analyst might want to balance customer waiting time against the idle time of servers according to some inherent cost structure. If the costs of waiting and idle service can be obtained directly, they can be used to determine the optimum number of channels to maintain and the service rates at which to operate these channels. Also, to design the waiting facility it is necessary to have information regarding the possible size of the queue to plan for waiting room. There may also be a space cost that should be considered along with customer-waiting and idle-server costs to obtain the optimal system design. In any case, the analyst will strive to solve this problem by analytical means; however, if these fail, he or she must resort to simulation. Ultimately, the issue generally comes down to a trade-off of better customer service versus the expense of providing more service capability, that is, determining the increase in investment of service for a corresponding decrease in customer delay.

1.5 Some General Results

We present some general results and relationships for $G/G/1$ and $G/G/c$ queues in this section, prior to specific model development. These results will prove useful in many of the following sections and chapters, as well as providing some insight at this early stage.

Denoting the average rate of customers entering the queueing system as λ and the average rate of serving customers as μ, a measure of traffic congestion for c-server systems is $\rho \equiv \lambda/c\mu$ (often called *traffic intensity*). When $\rho > 1$ ($\lambda > c\mu$), the average number of arrivals into the system exceeds the maximum average service rate of the system, and we would expect, as time goes on, the queue to get bigger and bigger, unless, at some point, customers were not allowed to join. If we are interested in steady-state conditions (the state of the system after it has been in operation a long time), when $\rho > 1$, the queue size never settles down (assuming customers are not prevented from entering the system) and there is no steady state. It turns out that for steady-state results to exist, ρ must be strictly less than 1 (again, assuming no denial of customer entry). When $\rho = 1$, unless arrivals and service are deterministic

and perfectly scheduled, no steady state exists, since randomness will prevent the queue from ever emptying out and allowing the servers to catch up, thus causing the queue to grow without bound. Therefore, if one knows the average arrival rate and average service rate, the minimum number of parallel servers required to guarantee a steady-state solution can be calculated immediately by finding the smallest c such that $\lambda/c\mu < 1$.

What we most often desire in solving queueing models is to find the probability distribution for the total number of customers in the system at time t, $N(t)$, which is made up of those waiting in queue, $N_q(t)$, plus those in service, $N_s(t)$. Let $p_n(t) = \Pr\{N(t) = n\}$, and $p_n = \Pr\{N = n\}$ in the steady state. Considering c-server queues in steady state, two expected-value measures of major interest are the mean number in the system,

$$L = \mathrm{E}[N] = \sum_{n=0}^{\infty} np_n,$$

and the expected number in queue,

$$L_q = \mathrm{E}[N_q] = \sum_{n=c+1}^{\infty} (n-c)p_n.$$

1.5.1 Little's Formulas

One of the most powerful relationships in queueing theory was developed by John D. C. Little in the early 1960s (see Little, 1961, for the original proof—a host of papers refining the proof followed in the ensuing decades). Little related the steady-state mean system sizes to the steady-state average customer waiting times as follows. Letting T_q represent the time a customer (transaction) spends waiting in the queue prior to entering service and T represent the total time a customer spends in the system ($T = T_q + S$, where S is the service time, and T, T_q, and S are random variables), two often used measures of system performance with respect to customers are $W_q = \mathrm{E}[T_q]$ and $W = \mathrm{E}[T]$, the mean waiting time in queue and the mean waiting time in the system, respectively. Little's formulas are

$$L = \lambda W \tag{1.1a}$$

and

$$L_q = \lambda W_q. \tag{1.1b}$$

Thus it is necessary to find only one of the four expected-value measures, in view of Little's formulas and the fact that $\mathrm{E}[T] = \mathrm{E}[T_q] + \mathrm{E}[S]$, or, equivalently, $W = W_q + 1/\mu$, where μ, as before, is the mean service rate.

Although the following does not constitute a proof, we illustrate the concept of Little's formulas by considering a *sample path* of one *busy period* (time from when

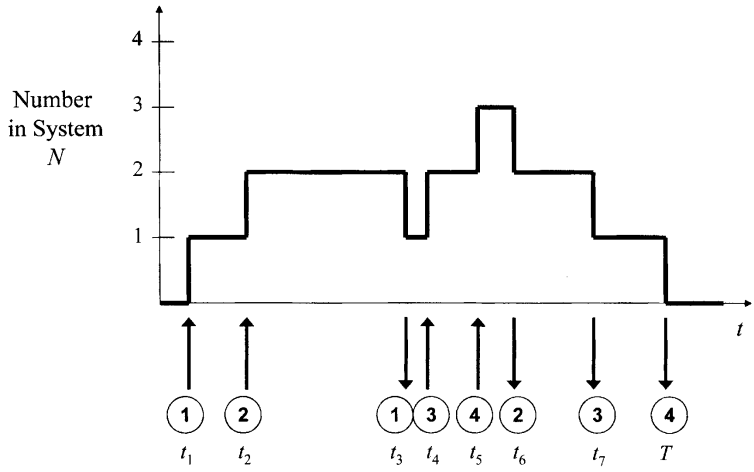


Figure 1.4 Busy-period sample path.

a customer enters an empty system until it next empties out again). Consider the illustration in Figure 1.4, where the number of customers (say, N_c) that arrive over the time period $(0, T)$ is 4.

The calculations for L and W are

$$\begin{aligned} L &= [1(t_2 - t_1) + 2(t_3 - t_2) + 1(t_4 - t_3) + 2(t_5 - t_4) \\ &\quad + 3(t_6 - t_5) + 2(t_7 - t_6) + 1(T - t_7)]/T \\ &= (\text{area under curve})/T \\ &= (T + t_7 + t_6 - t_5 - t_4 + t_3 - t_2 - t_1)/T \end{aligned} \tag{1.2a}$$

and

$$\begin{aligned} W &= [(t_3 - t_1) + (t_6 - t_2) + (t_7 - t_4) + (T - t_5)]/4 \\ &= (T + t_7 + t_6 - t_5 - t_4 + t_3 - t_2 - t_1)/4 \\ &= (\text{area under curve})/N_c. \end{aligned} \tag{1.2b}$$

Thus we see from (1.2a) and (1.2b) that the area under curve is $LT = WN_c$, which yields $L = WN_c/T$. The fraction N_c/T is the number of customers arriving over the time T and is, for this period, the arrival rate λ, so that $L = \lambda W$. A similar argument would hold for a picture of the number in the queue N_q over the period $(0, T)$, yielding $L_q = \lambda W_q$. While this is not a proof (since it needs to be shown that these relationships hold in the limit over many busy periods as time goes to infinity), one can see the idea behind the relationships.

An interesting result that can be derived from Little's formulas [(1.1a) and (1.1b)] and the relation between W and W_q is

$$L - L_q = \lambda(W - W_q) = \lambda(1/\mu) = \lambda/\mu. \tag{1.3}$$

Table 1.2 Summary of General Results for $G/G/c$ Queues

$\rho = \lambda/c\mu$	Traffic intensity; offered work load rate to a server
$L = \lambda W$	Little's formula
$L_q = \lambda W_q$	Little's formula
$W = W_q + 1/\mu$	Expected-value argument
$p_b = \lambda/c\mu = \rho$	Busy probability for an arbitrary server
$r = \lambda/\mu$	Expected number of customers in service; offered work load rate
$L = L_q + r$	Combined result—(1.3)
$p_0 = 1 - \rho$	$G/G/1$ empty-system probability
$L = L_q + (1 - p_0)$	Combined result for $G/G/1$

But $L - L_q = \mathrm{E}[N] - \mathrm{E}[N_q] = \mathrm{E}[N - N_q] = \mathrm{E}[N_s]$, so that the expected number of customers in service in the steady state is λ/μ, which we will denote by r. Note for a single-server system that $r = \rho$ and it also follows from simple algebra that

$$L - L_q = \sum_{n=1}^{\infty} np_n - \sum_{n=1}^{\infty}(n-1)p_n = \sum_{n=1}^{\infty} p_n = 1 - p_0.$$

From this, we can easily derive the probability that any given server is busy in a multiserver system in the steady state. We denote this probability by p_b. Since we have just shown that the expected number present in service at any instant in the steady state is r, it follows from the symmetry of the c servers that the expected number present at one server is r/c. Then, by a simple expected-value argument, we can show that $p_b = \rho$, since

$$r/c = \rho = 0 \cdot (1 - p_b) + 1 \cdot p_b.$$

For a single-server queue $(G/G/1)$, the probability of the system being idle $(N = 0)$ is the same as the probability of a server being idle. Thus $p_0 = 1 - p_b$ in this case, and $p_0 = 1 - \rho = 1 - r = 1 - \lambda/\mu$. The quantity $r = \lambda/\mu$, the expected number of customers in service, has another interesting connotation. It is sometimes also referred to as the *offered load*, since, on average, each customer requires $1/\mu$ time units of service and the average number of customers arriving per unit time is λ, so that the product $\lambda(1/\mu)$ is the amount of work arriving to the system per unit time. Dividing this by the number of servers c (which yields ρ) gives the average amount of work coming to each server per unit time.

Table 1.2 summarizes the results of this section.

1.6 Simple Data Bookkeeping for Queues

At this point, it might be useful to use a table format to show how the random events of arrivals and service completions interact for a sample single-server system

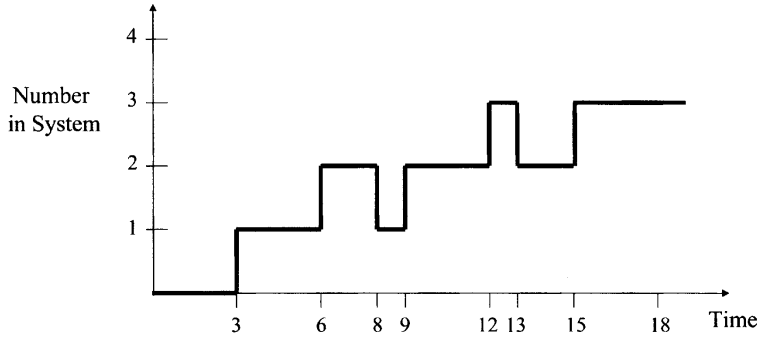


Figure 1.5 Sample path for queueing process.

to form a queue. In the following, we begin at time 0 with a first arrival and then update the system state when events (arrivals or departures) occur—thus the name *event-oriented bookkeeping* is used for this sort of table.

Consider the elementary case of a constant rate of arrivals to a single channel that possesses a constant service rate. (Figure 1.5 is an illustration of this with interarrival times of 3 and serve times of 5.) These regularly spaced arrivals are to be served first come, first served (FCFS). Let it also be assumed that at time $t = 0$ there are no customers waiting and that the channel is empty. Let λ be defined as the number of arrivals per unit time, and $1/\lambda$ then will be the constant time between successive arrivals. The particular unit of time (minutes, hours, etc.) is up to the choice of the analyst. However, consistency must be adhered to once the unit is chosen so that the same basic unit is used throughout the analysis. Similarly, if μ is to be the rate of service in terms of completions per unit time when the server is busy, then $1/\mu$ is the constant service time. We would like to calculate the number in the system at an arbitrary time t, say, $n(t)$, and the time the nth arriving customer must wait in the queue to obtain service, say, $W_q{}^{(n)}$. From these, it then becomes easy to compute the major measures of effectiveness. Under the assumption that as soon as a service is completed another is begun, the number in the system (including the customer in service) at time t is determined by the equation

$$\begin{aligned} n(t) = &\{\text{number of arrivals in } (0,t]\} \\ &- \{\text{number of services completed in } (0,t]\}. \end{aligned} \tag{1.4}$$

It should be pointed out that there are usually three waiting times of interest—the time spent by the nth customer waiting for service (or line delay), which we write here as $W_q^{(n)}$; the time the nth customer spent in the system, which we shall call $W^{(n)}$; and what is called the virtual line wait $V(t)$, namely, the wait a fictitious arrival would have to endure if it arrived at time t. The reader is cautioned that various authors are not consistent and each of these quantities is sometimes referred to simply as the waiting time.

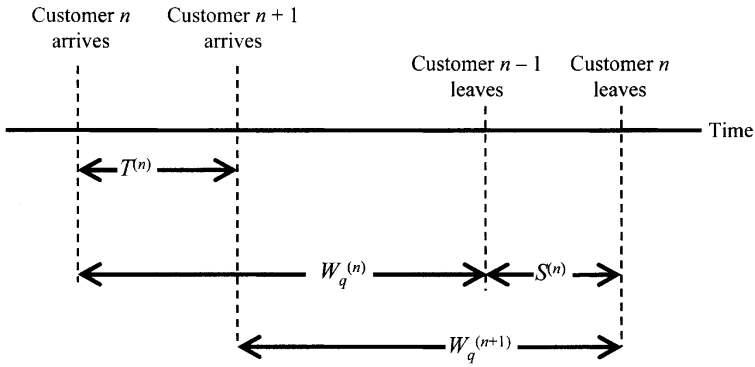


Figure 1.6 Successive $G/G/1$ waiting times.

To find the waiting times in queue until service begins, we observe that the line waits $W_q^{(n)}$ and $W_q^{(n+1)}$ of two successive customers in *any* single-server queue (deterministic or otherwise) are related by the simple recurrence relation

$$W_q^{(n+1)} = \begin{cases} W_q^{(n)} + S^{(n)} - T^{(n)} & (W_q^{(n)} + S^{(n)} - T^{(n)} > 0), \\ 0 & (W_q^{(n)} + S^{(n)} - T^{(n)} \leq 0), \end{cases} \tag{1.5}$$

where $S^{(n)}$ is the service time of the nth customer and $T^{(n)}$ is the interarrival time between the nth and $(n+1)$st customers. This can be seen by a simple diagram as shown in Figure 1.6. (This is an important general relation that is also utilized in later portions of the text.)

Bookkeeping has to do with updating the system status when events occur, recording items of interest, and calculating measures of effectiveness. Event-oriented bookkeeping updates the system state only when events (arrivals or departures) occur. Since there is not necessarily an event every basic time unit, in next-event bookkeeping the master clock is increased by a variable amount each time, rather than a fixed amount as it would be in time-oriented bookkeeping. The event-oriented approach will be illustrated here by an example, using the arrival and service data given in Table 1.3.

We see from simple averaging calculations for columns (5) and (6) in Table 1.4 that the mean line delay of the 12 customers was $40/12 = \frac{10}{3}$, while their mean system waiting time turned out to be $70/12 = \frac{35}{6}$. Furthermore, we observe that we can estimate the mean arrival rate as $\frac{12}{31}$ customers per unit time, since there were 12 arrivals over the 31-time-unit observation horizon. Thus the application of Little's law to these numbers tells us that the average system size L over the full time horizon was

$$L = \lambda W = \frac{70/12}{31/12} = \frac{70}{31}.$$

The mean queue size can be computed similarly.