OPERATIONS RESEARCH AND MANAGEMENT SCIENCE

HANDBOOK

Editor: A. Ravi Ravindran

September 29, 2006

Chapter 9

Queueing Theory

N. Gautam

Dept. of Industrial & Systems Engineering

Texas A&M University, College Station

gautam@tamu.edu

9.1 Introduction

What is common between a fast food restaurant, an amusement park, a bank, an airport security check point, and a post office? *Answer:* you are certainly bound to wait in a line before getting served at all these places. Such types of queues or waiting lines are found everywhere: computer-communication networks, production systems, transportation services, etc. In order to efficiently utilize manufacturing and service enterprises, it is critical to effectively manage queues. To do that, in this chapter we present a set of analytical techniques collectively called *queueing theory*. The main objective of queueing theory is to

develop formulae, expressions or algorithms for performance metrics such as: average number of entities in a queue, mean time spent in the system, resource availability, probability of rejection, etc. The results from queueing theory can directly be used to solve design and capacity planning problems such as: determining the number of servers, an optimum queueing discipline, schedule for service, number of queues, system architecture, etc. Besides making such strategic design decisions, queueing theory can also be used for tactical as well as operational decisions and controls.

The objective of this chapter is to introduce fundamental concepts in queues, clarify assumptions used to derive results, motivate models using examples, and point to software available for analysis. The presentation in this chapter is classified into four categories depending on types of customers (one or many) and number of stations (one or many). Examples of the four types are summarized in the table below.

	single-class	*multi-class*
single-station	post office	multi-lingual call center
multi-station	theme park	multi-ward hospital

Table 9.1: Examples to illustrate various types of queueing systems

The results presented in this chapter are a compilation of several excellent books and papers on various aspects of queueing theory. In particular, bulk of the single-station and single-class analysis (which forms over half the chapter) is from Gross and Harris [3] which arguably is one of the most popular texts in queueing theory. The book by Bolch et al [1] does a fantastic job presenting algorithms, approximations and bounds especially for multi-stage queues (i.e. queueing networks). For multi-class queues, the foundations are borrowed from the well-articulated chapters of Wolff [10] as well as Buzacott and Shanthikumar [2]. The set of papers by Whitt [9] explaining the queueing network analyzer is used for the multi-stage

and multi-class queues. Most of the notation used in this chapter and the fundamental results are from Kulkarni [7]. If one is interested in a single site with information about various aspects of queues (including Humor!), the place to visit is the page maintained by Hlynka [4]. In fact the page among other things illustrates various books on queueing, course notes, and, a list of software. A few software tools would be pointed out in this chapter, but it would be an excellent idea to visit Hlynka's site [5] for an up-to-date list of queueing software. In there, software that run on various other applications (such as MATLAB, Mathematica, Excel, etc.) are explained and the most suitable one for the reader can be adopted.

This chapter is organized as follows. First some basic results that are used throughout the chapter are explained in Section 9.2 on queueing theory basics. The bulk of this chapter is Section 9.3 which lays the foundation for the other types of systems by initially considering the single-station and single-class queue. Then in Section 9.4, the results for single-station and multiple classes are presented. Following that, the chapter moves from analyzing a single station to a network of queues in Section 9.5 where only one class is considered. This is extended to the most general form (of which all the previous models are special cases) of multi-station and multi-class queue in Section 9.6. Finally some concluding remarks are made in Section 9.7.

9.2 Queueing Theory Basics

Consider a single station queueing system as shown in Figure 9.1. This is also called a single stage queue. There is a single waiting line and one or more servers. A typical example can be found at a bank or post office. Arriving customers enter the queueing system and wait in the waiting area if a server is not free (otherwise they go straight to a server). When a server becomes free, one customer is selected and service begins. Upon service completion,

the customer departs the system. Few key assumptions are needed to analyze the basic queueing system.

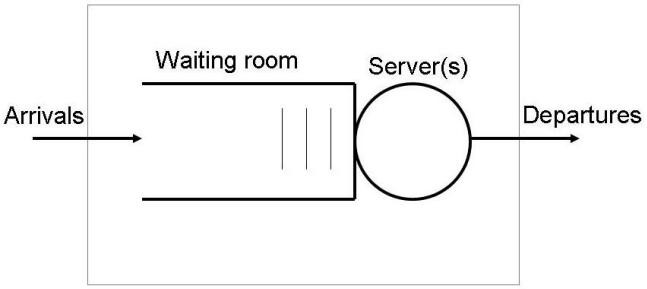

Figure 9.1: A single station queueing system

Assumption 1 *The customer inter-arrival times, i.e. the time between arrivals, are independent and identically distributed (usually written as "iid"). Thereby the arrival process is what is called a renewal process. All arriving customers enter the system if there is room to wait. Also all customers wait till their service is completed in order to depart.*

Assumption 2 *The service times are independent and identically distributed random variables. Also, the servers are stochastically identical, i.e. the service times are sampled from a single distribution. In addition, the servers adopt a work-conservation policy, i.e. the server is never idle when there are customers in the system.*

The above assumptions can certainly be relaxed. There are a few models that do not require the above assumptions. However, for the rest of this chapter, unless explicitly stated otherwise, we will assume that Assumptions 1 and 2 hold.

In order to standardize description for queues, Kendall developed a notation with five fields: *AP/ST/NS/Cap/SD*. In the Kendall notation, *AP* denotes arrival process characterized by the inter-arrival distribution, *ST* denotes the service time distribution, *NS* is the

number of servers in the system, *Cap* is the maximum number of customers in the whole system (with a default value of infinite), and *SD* denotes service discipline which describes the service order such as First Come First Served (FCFS) which is the default, Last Come First Served (LCFS), Random Order of Service (ROS), Shortest Processing Time First (SPTF), etc. The fields AP and ST can be specific distributions such as exponential (denoted by M which stands for memoryless or Markovian), Erlang (denoted by E_k), phase-type (PH), hyper-exponential (H), deterministic (D), etc. Sometimes instead of a specific distribution, AP and ST fields could be G or GI which denote general distribution (although GI explicitly says "general independent", G also assumes independence). Table 9.2 depicts values that can be found in the 5 fields of Kendall notation.

AP	M, G, E_k, H, PH, D, GI, etc.
ST	M, G, E_k, H, PH, D, GI, etc.
NS	denoted by s, typically 1, 2, ..., ∞
Cap	denoted by k, typically 1, 2, ..., ∞
	default : ∞
SD	FCFS, LCFS, ROS, SPTF, etc.
	default: FCFS

Table 9.2: Fields in the Kendall Notation

For example $GI/H/4/6/LCFS$ implies that the arrivals are according to a renewal process with general distribution, service times are according to a hyperexponential distribution, there are 4 servers, a maximum of 6 customers are permitted in the system at a time (including 4 at the server), and the service discipline is LCFS. Also, $M/G/4/9$ implies that the inter-arrival times are exponential (thereby the arrivals are according to a Poisson process), service times are according to some general distribution, there are 4 servers, the system capacity is 9 customers in total, and the customers are served according to FCFS.

Finally, in an $M/M/1$ queue, the arrivals are according to a Poisson process, service times exponentially distributed, there is 1 server, the waiting space is infinite and the customers are served according to FCFS.

9.2.1 Fundamental Queueing Relations

Consider a single station queueing system such as the one shown in Figure 9.1. Assume that this system can be described using Kendall notation. That means the inter-arrival time distribution, service time distribution, number of servers, system capacity and service discipline are given. For such a system we now describe some parameters and measures of performance. Assume that customers (or entities) that enter the queueing system are assigned numbers with the n^{th} arriving customer called customer-n. Most of the results presented in this section are available in Kulkarni [7] with possibly different notation.

In that light, let A_n denote the time when n^{th} customer arrives, and thereby $A_n - A_{n-1}$, an inter-arrival time. Let S_n be the service time for the n^{th} customer. Let D_n be the time when the n^{th} customer departs. We denote $X(t)$ as the number of customers in the system at time t, X_n as the number of customers in the system just after n^{th} customer departs, and X_n^* as the number of customers in the system just before n^{th} customer arrives. Although in this chapter we would not go into details, but it is worthwhile mentioning that $X(t)$, X_n and X_n^* are usually modeled as stochastic processes. We also define two other variables, which are usually not explicitly modeled but can be characterized in steady state. These are W_n, the waiting time of the n^{th} customer and $W(t)$, the total remaining workload at time t (this is the total time it would take to serve all the customers in the system at time t). The above variables are described in Table 9.3 for easy reference, where customer n denotes the n^{th} arriving customer.

Variable	Mathematical expression	Meaning
A_n		Arrival time of customer n
S_n		Service time of customer n
D_n		Departure time of customer n
$X(t)$		Number of customers in the system at time t
X_n	$X(D_n+)$	No. in system just after customer n's departure
X_n^*	$X(A_n-)$	No. in system just before customer n's arrival
W_n	$D_n - A_n$	Waiting time of customer n
$W(t)$		Total remainging workload at time t

Table 9.3: Variables and their mathematical as well as english meanings

It is usually very difficult to obtain distributions of the random variables $X(t)$, X_n, X_n^*, $W(t)$ and W_n. However the corresponding steady state values can be obtained, i.e. the limiting distributions as n and t go to infinite. In that light, let p_j be the probability that there are j customers in the system in steady state, and, let π_j and π_j^* be the respective probabilities that in steady state a departing and an arriving customer would see j other customers in the system. In addition, let $G(x)$ and $F(x)$ be the cummulative distribution functions of the workload and waiting time respectively in steady state. Finally, define L as the time-averaged number of customers in the system, and define W as the average waiting time (averaged across all customers). One of the primary objective of queueing models is to obtain closed form expressions for the performance metrics p_j, π_j, π_j^*, $G(x)$, $F(x)$, L and W. These performance metrics can be mathematically be represented as follows:

$$\begin{aligned} p_j &= \lim_{t\to\infty} P\{X(t) = j\} \\ \pi_j &= \lim_{n\to\infty} P\{X_n = j\} \end{aligned}$$

$$\begin{aligned}
\pi_j^* &= \lim_{n\to\infty} P\{X_n^* = j\} \\
G(x) &= \lim_{t\to\infty} P\{W(t) \leq x\} \\
F(x) &= \lim_{n\to\infty} P\{W_n \leq x\} \\
L &= \lim_{t\to\infty} E[X(t)] \\
W &= \lim_{n\to\infty} E[W_n]
\end{aligned}$$

Let $\overline{\lambda}$ be the average number of customers that enter the queueing system per unit time, refered to as the mean entering rate. Note that if the system capacity is finite, all arriving customers do not enter and therefore $\overline{\lambda}$ is specifically refered to as average rate of "entering" and not "arrival". The relation between L and W is given by Little's law (a result by Prof. John D.C. Little of MIT):

$$L = \overline{\lambda} W. \tag{9.1}$$

It is important to note two things. Firstly, for the finite capacity case W must be interpreted as the mean time in the system for customers that actually "enter" the system (and does not include customers that were turned away). Secondly, note that the average rate of departure from the system (if the system is stable) is also $\overline{\lambda}$. This is called conservation of customers whereby customers are neither created not destroyed, therefore average customer entering rate equals averate customer departure rate (if the system is stable).

We now forcus our attention to infinite capacity queues in the next section. However while we present results, if applicable, finite capacity queues' extensions will be explained. In addition, in future sections too we will mainly concentrate on infinite capcity queues (with some exceptions) due to issues of practicailty and ease of analysis. From a practical standpoint, if a queue actually has finite capacity but the capacity is seldom reached, approximating the queue as an inifite capacity queue is reasonable.

9.2.2 Preliminary Results for the $GI/G/s$ queue

Define the following for a single stage $GI/G/s$ queue (inter-arrival times independent and identically distributed, service time any general distribution, s servers, infinite waiting room, FCFS service discipline):

- λ : Average arrival rate into the system (inverse of the average time between 2 arrivals); notice that since the capacity is finite, all customers that *arrive*, also *enter* the system.
- μ : Average service rate of a server (inverse of the average time to serve a customer); it is important that the units for λ and μ be the same, i.e. both should be per second or both should be per minute, etc.
- L_q : Average number of customers in the queue, not including ones in service (L defined earlier, includes the customers at the servers).
- W_q : Average time spent in the queue, not including in service (W defined earlier, includes customer service times). Note that, in units of $1/\mu$,

$$W = W_q + \frac{1}{\mu}. \tag{9.2}$$

- $\rho = \dfrac{\lambda}{s\mu}$: the traffic intensity, which is a dimensionless quantity.

It is important to note that while extending the results to finite capacity queues, all the above definitions pertain only to customers that enter the system (and do not include those that were turned away when the system was full). However the first result below is applicable only for infinite capacity queues as finite capacity queues are always stable.

Result 1 *A necessary condition for stability of a queueing system is*

$$\rho \leq 1$$

For most cases the above condition is also sufficient (the sufficient condition actually is $\rho < 1$). However in the case of queues with multi-class traffic that traverse through multi-station queues, this condition may not be sufficient.

Little's law and other results using Little's law

As described in the previous section, we once again present Little's law, here $\overline{\lambda} = \lambda$.

Result 2 *For a $GI/G/s$ queue,*

$$L = \lambda W \tag{9.3}$$

and

$$L_q = \lambda W_q. \tag{9.4}$$

Notice that if we can compute one of L, L_q, W or W_q, the other three can be obtained using the above relations. Little's law holds under very general conditions. In fact even the service discipline does not have to be FCFS and the servers do not need to be work conserving. The result holds for any system with inputs and outputs. As an example, Equation (9.4) is nothing but using Little's law for the waiting space and not including the server. Note that if the system is stable, the output rate on an average is also λ. Using Little's law some more interesting results can be obtained.

Result 3 *The probability that a particular server is busy p_b is given by*

$$p_b = \rho$$

which can be derived from $W = W_q + 1/\mu$ and Little's law via the relation $L = L_q + \lambda/\mu$. Also, for the special single server case of $s = 1$, i.e. $GI/G/1$ queues, the probability that the system is empty, p_0 is

$$p_0 = 1 - \rho.$$

Based on the definition of L, it can be written as

$$L = \sum_{j=0}^{\infty} j p_j.$$

In a similar manner, let $L_{(k)}$ be the k^{th} factorial moment of the number of customers in the system in steady state, i.e.

$$L_{(k)} = \sum_{j=k}^{\infty} k! \binom{j}{k} p_j.$$

Also, let $W^{(k)}$ be the k^{th} moment of the waiting time in steady state, i.e.,

$$W^{(k)} = \lim_{n \to \infty} E[\{W_n\}^k].$$

Little's law can be extended for the $M/G/s$ queue in the following manner.

Result 4 *For an $M/G/s$ queue,*

$$L_{(k)} = \lambda^k W^{(k)}. \tag{9.5}$$

Of course the special case of $k = 1$ is Little's law itself. However, the interesting result is that all moments of the queue lengths are related to corresponding moments of waiting times. Notice that from factorial moments it is easy to obtain actual moments.

Limiting distributions of $X(t)$, X_n and X_n^*

In some situations it may not be possible to obtain p_j easily. But it may be possible to get π_j or π_j^*. In that light, two results based on the limiting distributions (π_j, π_j^* and p_j) will be presented. The first result relates the X_n and X_n^* processes in the limit (i.e. the relation between π_j and π_j^*). The second result illustrates the relation between the limiting distributions of $X(t)$ and X_n^* (i.e. p_j and π_j^*). They hold under very general cases beyond the cases presented here. However one must be very careful while using the results in the more general situations.

Result 5 *Let π_j and π_j^* be as defined earlier as the limiting distributions of X_n and X_n^* respectively. When either one of those limits exist, so does the other and*

$$\pi_j = \pi_j^* \quad \text{for all } j \geq 0.$$

It can easily be shown that the limits described in the above result exists for queue length processes of $M/M/1$, $M/M/s$, $M/G/1$ and $G/M/s$ queueing systems. However the limit for the more general $G/G/s$ case is harder to show but the result does hold. The result also holds for the $G/G/s/k$ case, whether we look at "entering" or "arriving" customers (in the "arriving" case, departing customers denote both the ones rejected as well as the ones that leave after service).

Result 6 *If the arrival process is Poisson (i.e. an $M/G/s$ queue), then the probability that an arriving customer in steady state will see the queueing system in state j is the probability that the system is in state j in the long run, i.e.,*

$$p_j = \lim_{t \to \infty} P\{X(t) = j\} = \pi_j^*.$$

The above result is called PASTA (Poisson Arrivals See Time Averages). PASTA is a powerful result that can be used in situations beyond queueing. For example if one is interested in computing an average over time, instead of observing the system continuously, it can be observed from time to time such that the inter-arrival times are exponentially distributed. In fact one common mistake made by a lot of people is to compute averages by sampling at equally spaced intervals. In fact sampling must be done in such a manner that the time between samples are exponentially distributed. Only then the averages obtained across such a sample will be equal to the average across time.

Therefore when one out of L, W, L_q or W_q is known, the other three can be computed. Also under certain conditions when one out of p_j, π_j or π_j^* is known, the others could be

computed. In the next few sections we will see how to compute "one" of those terms.

9.3 Single-station and single-class queues

In this section we consider a single queue at a single station handling a single class of customers. We start with the simplest case of $M/M/1$ queue and work our way through more complex cases. Note that all the results can be found in standard texts such as Gross and Harris [3] especially until Section 9.3.11.

9.3.1 The classic $M/M/1$ queue: main results

Consider a single stage queueing system where the arrivals are according to a Poisson process with average arrival rate λ per unit time (which is written as $PP(\lambda)$), i.e. the time between arrivals is according to an exponential distribution with mean $1/\lambda$. For this system the service times are exponentially distributed with mean $1/\mu$ and there is a single server.

The number in the system at time t in the $M/M/1$ queue, i.e. $X(t)$ can be modeled as a Continuous Time Markoc Chain (CTMC), specifically a birth and death process. The condition for stability for the CTMC and subsequently the $M/M/1$ queue is that the traffic intensity ρ should be less than 1, i.e. $\rho = \lambda/\mu < 1$. This means that the average arrival rate should be smaller than the average service rate. This is intuitive because the server would be able to handle all the arrivals only if the arrival rate is slower than the rate at which the server can process on an average.

The long run probability that the number of customers in the system is j (when $\rho < 1$)

is given by

$$p_j = \lim_{t\to\infty} P\{X(t) = j\} = (1-\rho)\rho^j \quad \text{for all } j \geq 0.$$

Therefore the long run probability that there are more than n customers in the system is ρ^n. In addition, the key performance measures can also be obtained. Using p_j we have

$$L = \sum_{j=0}^{\infty} jp_j = \frac{\lambda}{\mu - \lambda},$$

and

$$L_q = 0p_0 + \sum_{j=0}^{\infty} jp_{j+1} = \frac{\lambda^2}{\mu(\mu-\lambda)}.$$

Recall that W_n is the waiting time of the n^{th} arriving customer and $F(x) = \lim_{n\to\infty} P\{W_n \leq x\}$. Then

$$F(x) = 1 - e^{-(\mu-\lambda)x} \quad \text{for } x \geq 0$$

and therefore the waiting time for a customer arriving in steady-state is exponentially distributed with mean $1/(\mu-\lambda)$. Therefore

$$W = \frac{1}{\mu - \lambda},$$

which can also be derived using Little's law and the expression for L. In addition, using Little's law for L_q,

$$W_q = \frac{\lambda}{\mu(\mu-\lambda)}.$$

$M/M/1$ type queues with balking

When an arriving customer sees j other customers in the system, this customer joins the queue with probability α_j. In other words, this customer balks from the queueing system with probability $(1-\alpha_j)$. This can be modeled as a CTMC which is a birth and death process with birth parameters (i.e. rate for going from state n to $n+1$) $\lambda_{n+1} = \alpha_n\lambda$ for $n \geq 0$ and death parameters (i.e. rate for going from state n to $n-1$) $\mu_n = \mu$ for $n \geq 1$. It

is not possible to obtain p_j in closed form (and thereby L) except for some special cases. In general,

$$p_j = \frac{\prod_{k=0}^{j}(\lambda_k/\mu_k)}{1 + \sum_{n=1}^{\infty}\prod_{i=1}^{n}\frac{\lambda_i}{\mu_i}},$$

with $\lambda_0 = \mu_0 = 1$ and when the denominator exists. Also,

$$L = \sum_{j=0}^{\infty} j p_j$$

and using $\overline{\lambda} = \sum_{n=0}^{\infty} \lambda_{n+1} p_n$, W can be obtained as $L/\overline{\lambda}$.

$M/M/1$ type queues with reneging

Every customer that joins a queue waits for an $\exp(\theta)$ amount of time before which if the service does not begin, the customer leaves the queueing system (which is called reneging from the queueing system). This can be modeled as a birth and death CTMC with birth parameters (see above for $M/M/1$ with balking for definition) $\lambda_{n+1} = \lambda$ for $n \geq 0$ and death parameters $\mu_n = \mu + (n-1)\theta$ for $n \geq 1$. It is not possible to obtain p_j in closed form (and thereby L) except for some special cases. In general,

$$p_j = \frac{\prod_{k=0}^{j}(\lambda_k/\mu_k)}{1 + \sum_{n=1}^{\infty}\prod_{i=1}^{n}\frac{\lambda_i}{\mu_i}},$$

with $\lambda_0 = \mu_0 = 1$ and when the denominator exists. Also,

$$L = \sum_{j=0}^{\infty} j p_j$$

and using $\overline{\lambda} = \lambda$, it is possible to obtain W as L/λ. It is crucial to note that W is the time in the system for all customers, so it includes those customers that reneged as well as those that were served. A seperate analysis must be performed to obtain the departure rate of customers after service. Using this departure rate as $\overline{\lambda}$, if W is obtained then it would be the average waiting time for customers that were served.

In case there is a queueing system with balking and reneging, then the analysis can be combined. However it must be noted that if the reneging times are not exponential, then the analysis is a lot harder.

$M/M/1$ queue with state dependent service

Consider an $M/M/1$ type queue where the mean service rate depends on the state of the system. Many times when the number of customers waiting increases, the server starts working faster. This is typical when the servers are humans. Therefore, if there are n customers in the system, the mean service rate is μ_n. Note that in the middle of service if the number in service increases to $n+1$, the mean service rate also changes to μ_{n+1} (further if it increases to $n+2$ then service rate becomes μ_{n+2} and so on). This can also be modeled as a birth and death CTMC with birth parameters (defined in $M/M/1$ with balking) $\lambda_{n+1} = \lambda$ for $n \geq 0$ and death parameters are μ_n for $n \geq 1$. It is not possible to obtain p_j in closed form (and thereby L) except for some special cases. In general,

$$p_j = \frac{\prod_{k=0}^{j}(\lambda_k/\mu_k)}{1 + \sum_{n=1}^{\infty}\prod_{i=1}^{n}\frac{\lambda_i}{\mu_i}},$$

with $\lambda_0 = \mu_0 = 1$ and when the denominator exists. Also,

$$L = \sum_{j=0}^{\infty} j p_j$$

and using $\overline{\lambda} = \lambda$, W can be obtained as L/λ.

Note that if the mean service rate has to picked and retained throughout the service of a customer, that system cannot be modeled as a birth and death process.

$M/M/1$ queue with processor sharing

For p_j, L and W it does not matter what the service discipline is (FCFS, LCFS, ROS, etc.) The results are the same as long as the customers are served one at a time. Now what if the customers are served using a processor sharing discipline? Customers arrive according to a Poisson process with mean arrival rate λ customers per unit time. The amount of work each customer brings is according to $\exp(\mu)$, i.e. if each customer were served individually it would take $\exp(\mu)$ time for service. However, the processor is shared among all customers. So if the system has i customers, each customer gets only an i^{th} of the processing power. Therefore each of the i customers get a service rate of μ/i. However, the time for the first of the i to complete service is according to $\exp(i \times \mu/i)$. Therefore the CTMC for the number of customers in the system is identical to that of an FCFS $M/M/1$ queue. And so even the processor sharing discipline will have identical p_j, L and W as that of the FCFS $M/M/1$ queue.

9.3.2 The multi-server system – $M/M/s$

The description of an $M/M/s$ queue is similar to that of the classic $M/M/1$ queue with the exception that there are s servers. Note that by letting $s = 1$, all the results for the $M/M/1$ queue can be obtained. The number in the system at time t, $X(t)$, in the $M/M/s$ queue can be modeled as a CTMC, which again is a birth and death process. The condition for stability is $\rho = \lambda/(s\mu) < 1$ where ρ is called the traffic intensity. The long run probability that the number of customers in the system is j (when $\rho < 1$) is given by

$$p_j = \begin{cases} \frac{1}{j!}\left(\frac{\lambda}{\mu}\right)^j p_0 & \text{if } 0 \le j \le s-1 \\ \frac{1}{s!\, s^{j-s}}\left(\frac{\lambda}{\mu}\right)^j p_0 & \text{if } j \ge s \end{cases}$$

where $p_0 = \left[\sum_{n=0}^{s-1}\left\{\frac{1}{n!}(\lambda/\mu)^n\right\} + \frac{(\lambda/\mu)^s}{s!}\frac{1}{1-\lambda/(s\mu)}\right]^{-1}$. Thereby, using p_j, we can derive

$$L_q = \frac{p_0(\lambda/\mu)^s\lambda}{s!s\mu[1-\lambda/(s\mu)]^2}$$

Also, $W_q = L_q/\lambda$, $W = W_q + 1/\mu$, and $L = L_q + \lambda/\mu$. The steady state waiting time for a customer has a Cummulative Distribution Function (CDF) given by

$$F(x) = \frac{s(1-\rho)-w_0}{s(1-\rho)-1}(1-e^{-\mu x}) - \frac{1-w_0}{s(1-\rho)-1}(1-e^{-(s\mu-\lambda)x}),$$

where $w_0 = 1 - \frac{\lambda^s p_0}{s!\mu^s(1-\rho)}$.

9.3.3 Finite Capacity $M/M/s/K$ System

In fact this is one of the more general forms of the Poisson arrivals (with mean rate λ per unit time) and exponential service time (with mean $1/\mu$) queue. Using the results presented here, results for all the $M/M/\cdot/\cdot$ type queues can be obtained. For example letting $s = 1$ and $K = \infty$, the $M/M/1$ results can be obtained; $K = \infty$ would yield the $M/M/s$ results, $K = s$ would yield the $M/M/s/s$ results, $K = s = \infty$ would tield the $M/M/\infty$ results, etc. The special cases are popular because (a) the results are available in closed form, and (b) insights can be obtained especially while extending to the more general cases.

The number in the system at time t, $X(t)$, in the $M/M/s/K$ queue can be modeled as specifically a birth and death chain CTMC. Using the p_j values, one can derive

$$L_q = \frac{p_0(\lambda/\mu)^s\rho}{s!(1-\rho)^2}[1-\rho^{K-s}-(K-s)\rho^{K-s}(1-\rho)],$$

where $\rho = \lambda/(s\mu)$ and $p_0 = \left[\sum_{n=0}^{s}\left\{\frac{1}{n!}(\lambda/\mu)^n\right\} + \frac{(\lambda/\mu)^s}{s!}\sum_{n=s+1}^{K}\rho^{n-s}\right]^{-1}$. **Caution:** Since this is a finite capacity queue, ρ can be greater than 1. The probability that an arriving customer is rejected is p_K as is given by $p_K = \frac{(\lambda/\mu)^K}{s!s^{K-s}}p_0$. Therefore the average entering rate $\overline{\lambda}$ is given

by $\overline{\lambda} = (1 - p_K)\lambda$. Hence W_q can be derived as $L_q/\overline{\lambda}$. Also, W and L can be obtained using $W = W_q + 1/\mu$, and $L = L_q + \overline{\lambda}/\mu$.

9.3.4 The $M/M/1/K$ queue

The $M/M/1/K$ is another special case of the $M/M/s/K$ system with $s = 1$. However it is the most fundamental finite capacity queue example with Poisson arrivals (with mean rate λ) and exponential service times (with mean $1/\mu$). No more than K customers can be in the system at any time. The traffic intensity ρ (where $\rho = \lambda/\mu$) does not have to be less than one. Since the capacity is finite, the system cannot be unstable. However we assume for now that $\rho \neq 1$. For the case $\rho = 1$, limits results from calculus can be used (such as L'Hospital's rule) to obtain the corresponding value. The number of customers in the system, not including any at the server, is

$$L_q = \frac{\rho}{1-\rho} - \frac{\rho(K\rho^K + 1)}{1 - \rho^{K+1}}.$$

To obtain W_q, we use

$$W_q = L/\overline{\lambda},$$

where $\overline{\lambda}$ is the average entering rate into the system and can be expressed as $\lambda(1-p_K)$ where

$$p_K = \frac{(\lambda/\mu)^K[1 - \lambda/\mu]}{1 - (\lambda/\mu)^{K+1}}.$$

Also, W and L can be obtained using $W = W_q + 1/\mu$, and $L = L_q + \overline{\lambda}/\mu$.

9.3.5 The $M/M/s/s$ queue

Although the $M/M/s/s$ queue is a special case of the $M/M/s/K$ system with $K = s$, there are several interesting aspects and unique applications for it. Customers arrive according

to a Poisson process with mean rate λ per unit time. Essentially there are no queues. But there are s servers which can be thought of as s resources that customers hold on to for an exponential amount of time (with mean $1/\mu$). In fact queueing theory started with such a system by a Danish Mathematician A.K. Erlang who studied telephone switches with s lines. There is no waiting and if all s lines are being used, the customer gets a "busy" tone on their telephone. This system is also known as the Erlang Loss System. However there are many other applications for the $M/M/s/s$ queue such as: a rental agency with s items, gas stations where customers do not wait if a spot is not available among s possible spots, self-service area with maximum capacity s, etc. Many of these systems there are no explicit servers.

The probability that there are j (for $j = 0, \ldots, s$) customers in the system in the long run is

$$p_j = \frac{\frac{(\lambda/\mu)^j}{j!}}{\sum_{i=0}^{s} \frac{(\lambda/\mu)^i}{i!}}.$$

Therefore the "famous" Erlang loss formula is the probability that an arriving customer is rejected (loss probability) and is given by

$$p_s = \frac{\frac{(\lambda/\mu)^s}{s!}}{\sum_{i=0}^{s} \frac{(\lambda/\mu)^i}{i!}}.$$

A remarkable fact is that the above formula holds good even for the $M/G/s/s$ system with mean service time $1/\mu$. We will see that in the $M/G/s/s$ system explanation. We can derive

$$L = \frac{\lambda}{\mu}(1 - p_s).$$

Since the effective arrival rate is $\lambda(1 - p_s)$, $W = 1/\mu$, which is obvious since there is no waiting, the average time in the system W is indeed the average service time. Clearly $L_q = 0$ and $W_q = 0$ for that same reason.

9.3.6 The infinite server $M/M/\infty$ queue

This is identical to the $M/M/s/s$ system with $s = \infty$. Although in reality there are never infinite resources or servers. But when s is very large and a neglibible number of customers are rejected, the system can be assumed as $s = \infty$ as the results are expressed in closed-form. Systems such as the beach, grocery store (not counting the check out line), car rentals, etc. can be modeled as $M/M/\infty$ queues.

The probability that there are j customers in the system in the long run is $p_j = (\lambda/\mu)^j \frac{1}{j!} e^{-\lambda/\mu}$. Also, $L = \lambda/\mu$ and $W = 1/\mu$. Of course $L_q = 0$ and $W_q = 0$.

9.3.7 Finite population queues

Until this point we assumed that there are an infinite number of potential customers and the arrival rates did not depend on the number of customers in the system. Now we look at the case where the arrivals are state-dependent. Consider a finite population of N customers. Each customer after completion of service returns to the queue after spending $\exp(\lambda)$ time outside the queueing system. There is a single server who serves customers in $\exp(\mu)$ amount of time. Clearly the arrival rate would depend on the number of customers in the system. If $X(t)$ denotes the number of customers in the system, then its limiting distribution is

$$p_j = \lim_{t\to\infty} P\{X(t) = j\} = \frac{\binom{N}{j} j!(\lambda/\mu)^j}{\sum_{i=0}^{N} \binom{N}{i} i!(\lambda/\mu)^i}.$$

Clearly $L = \sum_{j=0}^{\infty} j p_j$, however L_q, W and W_q are tricky and need the effective arrival rate $\overline{\lambda}$. Using the fact that $\overline{\lambda} = \lambda(N - L)$ we can get

$$L_q = L - \lambda(N - L)/\mu, \quad W = \frac{L}{\lambda(N - L)} \quad \text{and } W_q = \frac{L_q}{\lambda(N - L)}.$$

Notice that the arrivals are not according to a Poisson process (which requires that

inter-arrival times be independent and identically distributed exponential random variables). Therefore PASTA cannot be applied. However it is possible to show that the probability that an arriving customer in steady state will see j in the system is

$$\pi_j^* = \frac{(N-j)p_j}{N-L}.$$

9.3.8 Bulk Arrivals and/or Service

So far we have only considered the case of single arrivals and single service. In fact in practice it is not uncommon to see bulk arrivals and/or bulk service. For example, arrivals into theme parks is usually in groups, arrivals and service in restaurants is in groups, shuttle busses perform service in batches, etc. We only present the cases where the inter-arrival times and service times are both exponentially distributed. However, unlike the cases seen thus far, here the CTMC models are not birth and death processes for the number in the system.

Bulk arrivals case: $M^{[X]}/M/1$ queue

Arrivals occur according to $PP(\lambda)$ and each arrival brings a random number X customers into the system. A single server processes the customers one by one spending $\exp(\mu)$ time with each customer. There is infinite waiting room and customers are processed according to FCFS. Such a system is denoted by $M^{[X]}/M/1$ queue. Let a_i be the probability that an arrival batch size is i, i.e. $P\{X = i\}$ for $i > 0$ (we do not allow batch size of zero). Let $E[X]$ and $E[X^2]$ be the first and second moments of X (where $E[X^2] = Var[X] + \{E[X]\}^2$). Define $\rho = \lambda E[X]/\mu$. The condition for stability is $\rho < 1$. We can derive the following results:

$$p_0 = 1 - \rho$$

$$L = \frac{\lambda\{E[X] + E[X^2]\}}{2\mu(1-\rho)}$$

Other p_j values can be computed in terms of λ, μ and a_i. Note that the average entering rate $\overline{\lambda} = \lambda E[X]$. Therefore using Little's law, $W = L/(\lambda E[X])$. Also, $W_q = W - 1/\mu$ and thereby $L_q = \lambda E[X] W_q$.

Bulk service case: $M/M^{[Y]}/1$ queue

Single arrivals occur according to $PP(\lambda)$. The server processes a maximum of K customers at a time and any arrivals that take place during a service can join service (provided the number is less than K). There is a single server, infinite waiting room and FCFS discipline. The service time for the entire batch is $\exp(\mu)$ whether the batch is of size K or not. In fact this is also sometimes known as the $M/M^{[K]}/1$ queue (besides the $M/M^{[Y]}/1$ queue). Notice that this system is identical to a shuttle bus type system where customers arrive according to $PP(\lambda)$ and busses arrive with $\exp(\mu)$ as the inter bus-arrival-distribution. As soon as a bus arrives, the first K customers (if there are less than K, then all customers) instantaneously enter the bus and the bus leaves. Then the queue denotes the number of customers waiting for a shuttle bus.

To obtain the distribution of the number of customers waiting, let $(r_1, \ldots, r_{K+1})$ be the $K+1$ roots of the characteristic equation (with D as the variable)

$$\mu D^{K+1} - (\lambda + \mu)D + \lambda = 0.$$

Let r_0 be the only root among the $K+1$ to be within 0 and 1. We can derive the following results:

$$p_n = (1 - r_0) r_0^n \quad \text{for } n \geq 0$$
$$L = \frac{r_0}{1 - r_0}, \quad L_q = L - \lambda/\mu$$

$$W = \frac{r_0}{\lambda(1-r_0)}, \; W_q = W - 1/\mu$$

9.3.9 The $M/G/1$ queue

Consider a queueing system with $PP(\lambda)$ arrivals and general service times. The service times are iid with CDF $G(\cdot)$, mean $1/\mu$ and variance σ^2. Notice that in terms of S_n, the service time for any arbitrary customer n,

$$\begin{aligned} G(t) &= P\{S_n \leq t\}, \\ 1/\mu &= E[S_n], \\ \sigma^2 &= Var[S_n]. \end{aligned}$$

There is a single server, infinite waiting room and customers are served according to FCFS service discipline. It is important to note for some of the results such as L and W, the CDF $G(\cdot)$ is not required. However for p_j and the waiting time distribution, the Laplace Steiltjes Transform (LST) of the CDF denoted by $\tilde{G}(s)$ and defined as

$$\tilde{G}(s) = E[e^{sS_n}] = \int_{t=0}^{\infty} e^{-st} dG(t)$$

is required.

Note that since the service time is not necessarily exponential, the number in the system for the $M/G/1$ queue cannot be modeled as a CTMC. However notice that if the system was observed at the time of departure, a Markovian structure is obtained. Let X_n be the number of customers in the system immediately after the n^{th} departure. Then it is possible to show that $\{X_n, n \geq 0\}$ is a discrete time Markov Chain (DTMC) whose transition probability matrix can be obtained. The stability condition is $\rho < 1$ where $\rho = \lambda/\mu$. Let π_j be the limiting probability (under stability) that in the long run a departing customer sees j

customers in the system, i.e.

$$\pi_j = \lim_{n\to\infty} P\{X_n = j\}.$$

Then, using PASTA, we have $p_j = \pi_j$ for all j. In order to obtain the π_j, consider the generating function $\phi(z)$ such that

$$\phi(z) = \sum_{j=0}^{\infty} \pi_j z^j.$$

If the system is stable,

$$\begin{aligned} \pi_0 &= 1-\rho, \\ \phi(z) &= \frac{(1-\rho)(1-z)\tilde{G}(\lambda-\lambda z)}{\tilde{G}(\lambda-\lambda z)-z}. \end{aligned}$$

Although π_j values cannot be obtained in closed form for the general case, they can be derived from $\phi(z)$ by repeatedly taking derivatives with respect to z and letting z go to zero.

However, L and W can be obtained in closed-form. The average number of customers in the system is

$$L = \rho + \frac{\lambda^2}{2}\frac{(\sigma^2 + 1/\mu^2)}{1-\rho}.$$

The average waiting time in the system is

$$W = 1/\mu + \frac{\lambda}{2}\frac{(\sigma^2 + 1/\mu^2)}{1-\rho}.$$

Also, $L_q = L - \rho$ and $W_q = W - 1/\mu$.

Recall that W_n is the waiting time of the n^{th} arriving customer and $F(x) = \lim_{n\to\infty} P\{W_n \leq x\}$. Although it is not easy to obtain $F(x)$ in closed form except for some special cases, it is possible to write it in terms of the Laplace Steiltjes Transform (LST) $\tilde{F}(s)$ defined as

$$\tilde{F}(s) = E[e^{sW_n}] = \int_{x=0}^{\infty} e^{-sx} dF(x)$$

in the following manner:

$$\tilde{F}(s) = \frac{(1-\rho)s\tilde{G}(s)}{s - \lambda(1-\tilde{G}(s))}.$$

It is important to realize that although inverting the LST in closed-form may not be easy, there are several software packages that can be used to invert it numerically. In addition, it is worthwhile to check that all the above results are true by trying exponential service times, as after all $M/M/1$ is a special case of the $M/G/1$ queue.

The $M/G/1$ queue with processor sharing

Similar to the $M/M/1$ system, for the $M/G/1$ system as well, the expressions for the number in system, L and waiting time, W, it does not matter what the service discipline is (FCFS, LCFS, ROS, etc). The results would be the same as long as the customers were served one at a time. Now what if the customers are served using a processor sharing discipline? Customers arrive according to a Poisson process with mean arrival rate λ customers per unit time. The amount of work each customer brings is according to some general distribution with CDF $G(\cdot)$ as described in the $M/G/1$ setting earlier. Also, if each customer were served individually it would take $1/\mu$ time for service on an average (and a variance of σ^2 for service time). However for this case, the processor is shared among all customers. So if the system has i customers, each customer gets only an i^{th} of the processing power. Therefore each of the i customers get a service rate of $1/i$ of the server speed. For this system, it can be shown that

$$W = \frac{1}{\mu - \lambda}.$$

The result indicates that the waiting time does not depend on the distribution of the service time but on the mean alone. Also $L = \lambda W$, $W_q = 0$ and $L_q = 0$.

The $M/G/\infty$ queue

Although this is an extension to the $M/M/\infty$ for the general service time case, the results are identical indicating they are independent of the distribution of service time. The probability that there are j customers in the system in the long run is

$$p_j = e^{-\lambda/\mu}\frac{(\lambda/\mu)^j}{j!} \quad for\ j \geq 0.$$

The departure process from the queue is $PP(\lambda)$. Also, $L = \lambda/\mu$ and $W = 1/\mu$. Of course $L_q = 0$ and $W_q = 0$.

The $M/G/s/s$ queue

This is a queueing system where the arrivals are according to a Poisson process with mean arrival rate λ. The service times (also called holding times) are generally distributed with mean $1/\mu$. There are s servers but no waiting space. The results are identical to that of the $M/M/s/s$ queue. In fact the Erlang loss formula was derived for this general case initially. For $0 \leq j \leq s$, the steady state probability that there are j customers in the system is

$$p_j = \frac{\frac{(\lambda/\mu)^j}{j!}}{\sum_{k=0}^{s}\frac{(\lambda/\mu)^k}{k!}}.$$

The departure process from the queue is $PP((1-p_s)\lambda)$. In addition, $L = \frac{\lambda}{\mu}(1-p_s)$, $W = 1/\mu$, $L_q = 0$ and $W_q = 0$.

9.3.10 The $G/M/1$ queue

Consider a queueing system where the inter-arrival times are according to some given general distribution and service times are according to an exponential distribution with mean $1/\mu$.

The inter-arrival times are iid with CDF $G(\cdot)$ and mean $1/\lambda$. This means that

$$\begin{aligned} G(t) &= P\{A_{n+1} - A_n \leq t\}, \\ 1/\lambda &= E[A_{n+1} - A_n] = \int_0^\infty t dG(t) \end{aligned}$$

Assume that $G(0) = 0$. Also, there is a single server, infinite waiting room and customers are served according to FCFS service discipline. It is important to note for most of the results the LST of the inter-arrival time CDF denoted by $\tilde{G}(s)$ and defined as

$$\tilde{G}(s) = E[e^{s(A_{n+1}-A_n)}] = \int_{t=0}^\infty e^{-st} dG(t)$$

is required.

Similar to the $M/G/1$ queue, since all the random events are not necessarily exponentially distributed, the number in the system for the $G/M/1$ queue cannot be modeled as a CTMC. However notice that if the system was observed at the time of arrivals, a Markovian structure is obtained. Let X_n^* be the number of customers in the system just before the n^{th} arrival. Then it is possible to model the stochastic process $\{X_n^*, n \geq 0\}$ as a DTMC. The DTMC is ergodic if

$$\rho = \lambda/\mu < 1$$

which is the stability condition. Let π_j^* be the limiting probability that in the long run an arriving customer sees j other customers in the system, i.e.

$$\pi_j^* = \lim_{n\to\infty} P\{X_n^* = j\}.$$

If $\rho < 1$, we can show that

$$\pi_j^* = (1 - \alpha)\alpha^j$$

where α is a unique solution in $(0, 1)$ to

$$\alpha = \tilde{G}(\mu - \mu\alpha).$$

Using the notation W_n as the waiting time of the n^{th} arriving customer under FCFS and $F(x) = \lim_{n\to\infty} P\{W_n \leq x\}$, we have

$$F(x) = 1 - e^{-\mu(1-\alpha)x}.$$

Therefore under FCFS, the waiting time in the system in the long run is exponenetially distributed with parameter $\mu(1-\alpha)$. Using that result, the average waiting time in the system is

$$W = \frac{1}{\mu(1-\alpha)}.$$

Using Little's law, the average number of customers in the system is

$$L = \frac{\lambda}{\mu(1-\alpha)}.$$

Note that we cannot use PASTA (as the arrivals are not Poisson, unlike the $M/G/1$ case). However, it is possible to obtain p_j using the following relation:

$$\begin{aligned} p_0 &= 1-\rho, \\ p_j &= \rho\pi^*_{j-1} \quad \text{when } j > 0. \end{aligned}$$

9.3.11 The $G/G/1$ queue

Consider a single server queue with infinite waiting room where the inter-arrival times and service times are according to general distributions. The service discipline is FCFS. Since the model is so general without a Markovian structure, it is difficult to model the number in the system as an analyzable stochastic process. Therefore it is not possible to get exact expressions for the various performance measures. However, bounds and approximations can be derived. There are several of them and none are considered absolutely better than others. In this subsection, almost all the results are from Bolch et al [1] unless otherwise noted.

Recall that W_n is the time in the system for the n^{th} customer, S_n is the service time for the n^{th} customer and A_n is the time of n^{th} arrival. In order to derive bounds and approximations for the $G/G/1$ queue, a few variables need to be defined. Define $I_{n+1} = \max(A_{n+1} - A_n - W_n, 0)$ and $T_{n+1} = A_{n+1} - A_n$. All the bounds and approximations are in terms of four parameters:

$$\begin{aligned} 1/\lambda &= E[T_n] \quad \text{average inter-arrival time} \\ C_a^2 &= Var[T_n]/\{E[T_n]\}^2 \quad \text{SCOV of inter-arrival times} \\ 1/\mu &= E[S_n] \quad \text{average service time} \\ C_s^2 &= Var[S_n]/\{E[S_n]\}^2 \quad \text{SCOV of service times} \end{aligned}$$

where SCOV is the "squared coefficient of variation" i.e. the ratio of the variance to the square of the mean (only for positive-valued random variables). Another parameter that is often used is $\rho = \lambda/\mu$ which is the traffic intensity.

Let random variables T, S and I be the limiting values as $n \to \infty$ of T_n, S_n and I_n respectively. Although the mean and variance of T and S are known, $E[I]$ can be computed as $E(I) = E(T) - E(S)$ which requires $E(T) > E(S)$, i.e. $\rho < 1$. It is possible to show that

$$W = \frac{E(S^2) - 2\{E(S)\}^2 - E(I^2) + E[T^2]}{2\{E(T) - E(S)\}}.$$

Notice that the only unknown quantity above is $E(I^2)$. Therefore approximations and bounds for W can be obtained through those of $E[I^2]$. Since $L = \lambda W$ (using Little's law), bounds and approximations for L can also be obtained.

Since on many occasions, the departure process from a queue is the arrival process to another queue in a queueing network setting, it is important to study the mean and SCOV of the departure process of a $G/G/1$ queue. Let D_n be the time of departure of the n^{th} customer. Define $\Delta_{n+1} = D_{n+1} - D_n$ as the inter-departure time with Δ being the inter-departure time in steady state. Then it is possible to show that under stability, $E(\Delta) =$

$E(I) + E(S) = E(T) = 1/\lambda$. Therefore the departure rate equals arrival rate (conservation of flow of customers). In addition, let $C_d^2 = Var(\Delta)/\{E[\Delta]\}^2$, then

$$C_d^2 = C_a^2 + 2\rho^2 C_s^2 + 2\rho(1-\rho) - 2\lambda W(1-\rho).$$

Note that C_d^2 is in terms of W and hence approximations and bounds for W would yield the same for C_d^2.

Bounds for L, W and C_d^2 for $G/G/1$ queue

First some bounds on W:

$$\begin{aligned}
W &\leq \frac{C_a^2 + \rho^2 C_s^2}{2\lambda(1-\rho)} + E(S) \\
W &\leq \frac{\rho(2-\rho)C_a^2 + \rho^2 C_s^2}{2\lambda(1-\rho)} + E(S) \\
W &\geq \frac{\rho(C_a^2 - 1 + \rho) + \rho^2 C_s^2}{2\lambda(1-\rho)} + E(S) \text{ if } T \text{ is DFR} \\
W &\leq \frac{\rho(C_a^2 - 1 + \rho) + \rho^2 C_s^2}{2\lambda(1-\rho)} + E(S) \text{ if } T \text{ is IFR}
\end{aligned}$$

where IFR and DFR are described subsequently. Note that $\rho(2-\rho) < 1$, therefore the first bound is always inferior to the second. Also IFR and DFR respectively denote increasing failure rate and decreasing failure rate random variables. Mathematically, the failure rate of a positive valued random variable X is defined as $h(x) = f_X(x)/[1-F_X(x)]$, where $f_X(x)$ and $F_X(x)$ are the probability density function and CDF of the random variable X. The reason they are called failure rate is because if X denotes the lifetime of a particular component, then $h(x)$ is the rate at which that component fails when it is x time units old. IFR and DFR imply that $h(x)$ is respectively increasing and decreasing functions of x. Note that all random variables need not be IFR or DFR, they could be neither. Also the exponential random variable has a constant failure rate.

We can also obtain bounds via the $M/G/1$ (disregarding C_a^2 and using Poisson arrival

process with mean rate λ arrival process) and $G/M/1$ (disregarding C_s^2 and using exponentially distributed service times with mean $1/\mu$) results. It is important to note that in order to use the $G/M/1$ results, the distribution of the inter-arrival times are needed, not just the mean and SCOV. See the table below with LB and UB referring to lower and upper bounds:

C_a^2	C_s^2	$M/G/1$	$G/M/1$
> 1	> 1	LB	LB
> 1	< 1	LB	UB
< 1	> 1	UB	LB
< 1	< 1	UB	UB

That means (see second result above) if $C_a^2 > 1$ and $C_s^2 < 1$ for the actual $G/G/1$ system, then W using $M/G/1$ analysis would be a lower bound and correspondingly $G/M/1$ would yield an upper bound.

Next let us see what we have for L when T is DFR (although all bounds above for W can be used by multiplying by λ)

$$\frac{\rho(C_a^2 - 1 + \rho) + \rho^2 C_s^2}{2(1-\rho)} + \rho \leq L \leq \frac{\rho(2-\rho)C_a^2 + \rho^2 C_s^2}{2(1-\rho)} + \rho.$$

Finally some bounds on C_d^2:

$$\begin{aligned} C_d^2 &\geq (1-\rho)^2 C_a^2 + \rho^2 C_s^2, \\ C_d^2 &\leq (1-\rho)C_a^2 + \rho^2 C_s^2 + \rho(1-\rho) \text{ if } T \text{ is DFR}, \\ C_d^2 &\geq (1-\rho)C_a^2 + \rho^2 C_s^2 + \rho(1-\rho) \text{ if } T \text{ is IFR}. \end{aligned}$$

Approximations for L, W and C_d^2 for $G/G/1$ queue

The following are some approximations for L and C_d^2 taken from Buzacott and Shanthikumar [2]. Approximations for W can be obtained by dividing L by λ. There are several other approximations available in the literature, many of which are empirical. Only a few are presented here as follows:

Approx.	L	C_d^2
1	$\left(\frac{\rho^2(1+C_s^2)}{1+\rho^2 C_s^2}\right)\left(\frac{C_a^2+\rho^2 C_s^2}{2(1-\rho)}\right)+\rho$	$(1-\rho^2)\left(\frac{C_a^2+\rho^2 C_s^2}{1+\rho^2 C_s^2}\right)+\rho^2 C_s^2$
2	$\left(\frac{\rho^2(1+C_s^2)}{2-\rho+\rho C_s^2}\right)\left(\frac{\rho(2-\rho)C_a^2+\rho^2 C_s^2}{2(1-\rho)}\right)+\rho$	$1-\rho^2+\rho^2 C_s^2+(C_a^2-1)\left(\frac{(1-\rho^2)(2-\rho)+\rho C_s^2(1-\rho)^2}{2-\rho+\rho C_s^2}\right)$
3	$\frac{\rho^2(C_a^2+C_s^2)}{2(1-\rho)}+\frac{(1-C_a^2)C_a^2\rho}{2}+\rho$	$(1-\rho)(1+\rho C_a^2)C_a^2+\rho^2 C_s^2$

9.3.12 The $G/G/m$ queue

Everything is similar to the $G/G/1$ queue explained before except that the number of servers is m. Getting closed-form expressions was impossible for $G/G/1$, so naturally for $G/G/m$ there is no question. However, several researchers have obtained bounds and approximations for the $G/G/m$ queue. In fact letting $m = 1$ for the $G/G/m$ results would produce great results for $G/G/1$. Notice that the traffic intensity $\rho = \lambda/(m\mu)$. The random variables S and T, as well as parameters C_a^2 and C_s^2 used in the following bounds and approximations have been defined in the $G/G/1$ system above.

- The Kingman upper bound:

$$W_q \leq \frac{Var(T) + Var(S)/m + (m-1)/(m^2\mu^2)}{2(1-\rho)}.$$

- The Brumelle and Marchal lower bound:

$$W_q \geq \frac{\rho^2 C_s^2 - \rho(2-\rho)}{2\lambda(1-\rho)} - \frac{m-1}{m}\frac{(C_s^2+1)}{2\mu}.$$

- Under heavy traffic conditions, for the $G/M/m$ systems,

$$W_q \approx \frac{Var(T) + Var(S)/m^2}{2(1-\rho)}\lambda$$

and waiting time in the queue is distributed approximately according to an exponential distribution with mean $1/W_q$. Note that "heavy traffic" implies that ρ is close to 1.

- And finally,

$$W_{G/G/m} \approx \frac{W_{M/M/m}}{W_{M/M/1}} W_{G/G/1} + E[S].$$

In the above approximation, the subscript for W denotes the type of queue. For exaple $W_{M/M/m}$ implies the mean waiting time for the $M/M/m$ queue using the same λ and μ as the $G/G/m$ case.

- There are several approximations available in the literature, many of which are empirical. The most popular one is the following. Choose α_m such that

$$\alpha_m = \begin{cases} \frac{\rho^m + \rho}{2} & \text{if } \rho > 0.7, \\ \rho^{\frac{m+1}{2}} & \text{if } \rho < 0.7. \end{cases}$$

The waiting time in the queue is given by the approximation

$$W_q \approx \frac{\alpha_m}{\mu}\left(\frac{1}{1-\rho}\right)\left(\frac{C_a^2 + C_s^2}{2m}\right).$$

9.4 Single-Station and Multi-Class Queues

In the models considered so far there was only a single class of customers in the system. However there are several applications where customers can be differentiated into classes and each class has its own characteristics. For example, consider a hospital emergency room. The patients can be classified into emergency, urgent and normal cases with varying arrival rates and service time requirements. Another example is a toll booth where the vehicles can

be classified based on type (cars, buses, trucks, etc) and each type has its own arrival rate and service time characteristics. There are several examples in production systems (routine maintenance versus break downs in repair shops) and communication systems (voice calls versus dial-up connection for Internet at a telephone switch) where entities must be classified due to the wide variability of arrival rates and service times.

Having made a case for splitting traffic in queues into multiple classes, it is also important to warn that unless absolutely necessary, due to the diffculty in analyzing such systems, one should not classify. There are two situations where it does make sense to classify. Firstly when the system has a natural classification where the various classes require their own performance measures (for example in a flexible manufacturing system, if a machine produces 3 types of parts and it is important to measure the in-process-inventory of each of them individually, then it makes sense to model them as 3 classes). Secondly when the service times are significantly different for the various classes that the distribution models would fit better, then it makes sense (for example if the service times have a bi-modal distribution, then classifying into two classes with uni-modal distribution for each class would possibly be better).

The next question to ask is how are the different classes of customers organized at the single station. There are two waiting line structures:

(a) All classes of customers wait in the same waiting room. Examples: buffer in flexible manufacturing system, packets on a router interface, vehicles at a 1-lane road traffic light, etc. Service scheduling policies are: FCFS, priority, ROS, LCFS, etc.

(b) Each class has a waiting room of its own and all classes of customers of a particular class wait in the same waiting room. Examples: robots handling several machine buffers, class-based queueing in routers, vehicles at a toll plaza (electronic payments, exact

change and full service), etc. Service scheduling policies (especially when there is only a single server) across the classes typically are: priority, polling, weighted round-robin, etc.

Within a class, the two waiting line structures use FCFS usually (LCFS and others are also possible). If the waiting room is of infinite capacity and there is no switch-over time from one queue to another, both (a) and (b) can be treated identically. However in the finite waiting room case, they are different and in fact one of the design decisions is to figure out the buffer sizes, admission/rejection rules, etc.

For the multi-class queues, several design decisions need to be made. These include:

- *Assigning classes*: how should the customers be classified? As alluded to before, it is critical especially when there is no clear cut classification, how customers should be classified and how many categories to consider.

- *Buffer sizing*: what should the size of the buffers be or how should a big buffer be partitioned for the various classes? These decisions can be made either one time (static) or changed as the system evolves (dynamic).

- *Scheduling rule*: how should the customers or entities by scheduled on the servers? For example FCFS, shortest expected processing time first (FCFS within a class), round-robin across the K classes, priority-based scheduling, etc. Sometimes these are "given" for the system and cannot be changed; other times these could be decisions that can be made.

- *Priority allocation*: if priority-based scheduling rule is used, then how should priorities be assigned? In systems like the hospital emergency room, the priorities are clear. However in many instances one has to trade off cost and resources to determine priorities.

- *Service capacity*: how to partition resources such as servers (wholly or partially) among classes? For example in a call center handling customers that speak different languages and some servers being multi-lingual, it is important to allocate servers to appropriate queues. Some times these capacity allocations are made in a static manner and other times dynamically based on system state.

There are several articles in the literature that discuss various versions of the above design problems. In this chapter we assume that the following are known or given: there are R classes already determined, infinite buffer size, scheduling rule already determined and a single server that serves customers one at a time. For such a system we first describe some general results for the $G/G/1$ case next and describe specific results for the $M/G/1$ case subsequently. Most of the results are adapted from Wolff [10] with possibly different notation.

9.4.1 Multi-class $G/G/1$ queue: general results

Consider a single station queue with a single server that caters to R classes of customers. Customers belonging to class i ($i \in \{1, 2, \ldots, R\}$) arrive into the system at a mean rate λ_i and the arrival process is independent of other classes, but is also independent and identically distributed within a class. Customers belonging to class i ($i \in \{1, 2, \ldots, R\}$) require an average service time of $1/\mu_i$. Upon completion of service, the customers depart the system. We assume that the distribution of inter-arrival times and service times are known for each class. However notice that the scheduling policy (i.e. service discipline) has not yet been specified. We describe some results that are invariant across scheduling policies (or at least a subset of policies).

For these systems, except for some special cases, it is difficult to obtain performance

measures such as "distribution" of waiting time and queue length like in the single class cases. Therefore we concentrate on obtaining average waiting time and queue length. Let L_i and W_i be the mean queue length and mean waiting time for class i customers. Irrespective of the scheduling policy, Little's law holds for each class, so for all $i \in [1, R]$,

$$L_i = \lambda_i W_i.$$

That means that one can think of each class as a mini system in itself. Also, similar results can be derived for L_{iq} and W_{iq} which respectively denote the average number waiting in queue (not including customers at servers) and average time spent waiting before service. In particular, for all $i \in [1, R]$,

$$\begin{aligned} L_{iq} &= \lambda_i W_{iq} \\ W_i &= W_{iq} + \frac{1}{\mu_i} \\ L_i &= L_{iq} + \rho_i \end{aligned}$$

where

$$\rho_i = \lambda_i / \mu_i.$$

In addition, L and W are the overall mean number of customers and mean waiting time averaged over all classes. Note that $L = L_1 + L_2 + \ldots + L_R$ and if $\lambda = \lambda_1 + \lambda_2 + \ldots + \lambda_R$, the net arrival rate, then $W = L/\lambda$. For the $G/G/1$ case with multiple classes, more results can be derived for a special class of scheduling policies called work-conserving disciplines which we describe next.

Work Conserving Disciplines under $G/G/1$

We now concentrate on a subset of service scheduling policies (i.e. service disciplines) called *work conserving disciplines* where more results for the $G/G/1$ queue can be obtained. In

fact many of these results have not been explained in the single class in the previous sections case but by letting $R = 1$, they can easily be accomplished.

The essence of *work conserving disciplines is that* the system workload at every instant of time remains unchanged over all work conserving service scheduling disciplines. Intuitively this means that the server never idles and does not do any wasteful work. The server continuously serves customers if there are any in the system. For example, FCFS, LCFS and ROS are work conserving. Certain priority policies that we will see later such as non-preemptive and pre-emptive resume policies are also work conserving. There are policies that are non-work-conserving such as the preemptive repeat (unless the service times are exponential). Usually when the server takes a vacation from service or if there is a switch over time (or set up time) during moving from classes, unless those can be explicitly accounted for in the service times, are non-work conserving.

To describe the results for the work conserving disciplines, consider the notation used in Section 9.4.1 above. Define ρ, the overall traffic intensity, as

$$\rho = \sum_{i=1}^{R} \rho_i.$$

An R-class $G/G/1$ queue with a work conserving scheduling discipline is stable if

$$\rho < 1.$$

In addition, when a $G/G/1$ system is work conserving, the probability that the system is emplty is $1 - \rho$.

Let S_i be the random variable denoting the service time of a class i customer. Then we have the second moment of the overall service time as

$$E[S^2] = \frac{1}{\lambda} \sum_{i=1}^{R} \lambda_i E[S_i^2].$$

We now present two results that are central to work conserving disciplines. These results

were not presented for the single-class case (easily doable by letting $R = 1$).

Result 7 *If the $G/G/1$ queue is stable, then when the system is in steady state, the expected remaining service time at an arbitrary time in steady state is $\lambda E[S^2]/2$.*

Since the total amount of work remains a constant across all work-conserving disciplines, and the above result represents the average work remaining for the customer at the server, the average work remaining due to all the customers waiting would also remain a constant across work conserving discipline. That result is described below.

Result 8 *Let W_{iq} be the average waiting time in the queue (not including service) for a class i customer, then the expression*

$$\sum_{i=1}^{R} \rho_i W_{iq}$$

is a **constant** *over all work conserving disciplines,*

However, quantities such as L, W, L_i and W_i (and the respective quantities with the q subscript) will depend on the service scheduling policies. It is possible to derive these expressions in closed form for $M/G/1$ queues which we describe next. The HOM software [8] can be used for numerical analysis of various scheduling policies for relatively general multiclass traffic.

9.4.2 $M/G/1$ Queue with Multiple Classes

Consider a special case of the $G/G/1$ queue with R classes where the arrival process is $PP(\lambda_i)$ for class i $(i = 1, 2, \ldots, R)$. The service times are iid with mean $E[S_i] = 1/\mu_i$, second moment $E[S_i^2]$ and CDF $G_i(\cdot)$ for class i $(i = 1, 2, \ldots, R)$ and $\rho_i = \lambda_i/\mu_i$. We present results for three work-conserving disciplnes: FCFS, non-preemptive priority and preemptive resume priority.

Multi-class $M/G/1$ with FCFS

In this service scheduling scheme, the customers are served according to FCFS. None of the classes receive any preferential treatment. The analysis assumes that all the R classes in some sense can be aggregated into one class since there is no differentiation. Hence the net arrival process is $PP(\lambda)$ with $\lambda = \lambda_1 + \lambda_2 + \ldots + \lambda_R$. Let S be a random variable denoting the "effective" service time for an arbitrary customer. Then

$$\begin{aligned} G(t) &= P(S \leq t) = \frac{1}{\lambda}\sum_{i=1}^{R} \lambda_i G_i(t), \\ E[S] &= \frac{1}{\mu} = \frac{1}{\lambda}\sum_{i=1}^{R} \lambda_i E[S_i], \\ E[S^2] &= \sigma^2 + 1/\mu^2 = \frac{1}{\lambda}\sum_{i=1}^{R} \lambda_i E[S_i^2], \\ \rho &= \lambda E[S]. \end{aligned}$$

Assume that the system is stable. Then using standard $M/G/1$ results with $X(t)$ being the total number of customers in the system at time t, we get when $\rho < 1$,

$$\begin{aligned} L &= \rho + \frac{1}{2}\frac{\lambda^2 E[S^2]}{1-\rho}, \\ W &= L/\lambda, \\ W_q &= W - 1/\mu, \\ L_q &= \frac{1}{2}\frac{\lambda^2 E[S^2]}{1-\rho}. \end{aligned}$$

The expected number of class i customers in the system (L_i) as well as in the queue (L_{iq}) and the expected waiting time in the system for class i (W_i) as well as in the queue (L_{iq}) are given by:

$$\begin{aligned} W_{iq} &= W_q = \frac{1}{2}\frac{\lambda E[S^2]}{1-\rho}, \\ L_{iq} &= \lambda_i W_{iq}, \end{aligned}$$

$$\begin{aligned} L_i &= \rho_i + L_{iq}, \\ W_i &= W_{iq} + 1/\mu_i. \end{aligned}$$

$M/G/1$ with Non-preemptive priority

Here we consider priorities among the various classes. For the following analysis assume that class 1 has highest priority and class R has the lowest. Service discipline within a class is FCFS. The server always starts serving a customer of the highest class among those waiting for service, and the first customer that arrived within that class. However the server completes serving a customer before considering who to serve next. The meaning of non-preemptive priority means that a customer in service does not get preempted while in service by another customer of high priority (however pre-emption does occur while waiting). Assume that the system is stable.

Let $\alpha_i = \rho_1 + \rho_2 + \ldots + \rho_i$ with $\alpha_0 = 0$. Then we get the following results:

$$\begin{aligned} E[W_i^q] &= \frac{\frac{1}{2}\sum_{j=1}^{R} \lambda_j E[S_j^2]}{(1-\alpha_i)(1-\alpha_{i-1})} \quad \text{for } 1 \le i \le R, \\ E[L_i^q] &= \lambda_i E[W_i^q], \\ W_i &= E[W_i^q] + E[S_i], \\ L_i &= E[L_i^q] + \rho_i. \end{aligned}$$

Sometimes performance measures for individual classes are required and other times aggregate performance measures across all classes. The results for the individual classes can also be used to obtain the overall or aggregate performance measures as follows:

$$\begin{aligned} L &= L_1 + L_2 + \ldots + L_R, \\ W &= L/\lambda, \end{aligned}$$

$$W_q = W - 1/\mu,$$
$$L_q = \lambda W_q.$$

Note: In the above analysis we assume that we are given which class should get the highest priority, second highest, etc. However, if we need to determine an optimal way of assigning priorities, one method is now provided. If you have R classes of customers and it costs the server C_j per unit time a customer of class j spends in the system (holding cost for class j customer), then in order to minimize the total expected cost per unit time in the long run, the optimal priority assignment is to give class i higher priority than class j if $C_i\mu_i > C_j\mu_j$. In other words, sort the classes in the decreasing order of the product $C_i\mu_i$ and assign first priority to the largest $C_i\mu_i$ and the last priority to the smallest $C_i\mu_i$ over all i. This is known as the $C\mu$ rule. Also note that if all the C_i values were equal, then this policy reduces to "serve the customer with the smallest expected processing time first".

$M/G/1$ with Preemptive resume priority

A slight modification to the $M/G/1$ non-preemptive priority considered above, is to allow preemption during service. During the service of a customer if another customer of higher priority arrives, then the customer in service is preempted and service begins for this new high priority customer. When the pre-empted customer returns to service, service resumes from where it was preempted. This is a work-conserving discipline (however if the service has to start from the beginning which is called preemptive repeat, then it is not work conserving because the server wasted some time serving). Here we consider the case where upon arrival, customer of class i can preempt a customer of class j in service if $j > i$. Also the total service time is unaffected by the interruptions, if any. Assume that the system is stable.

The waiting time of customers of class i is unaffected by customers of class j if $j > i$.

Thus class 1 customers face a standard single-class $M/G/1$ system with arrival rate λ_1 and service time distribution $G_1(\cdot)$. In addition, if only the first i classes of customers are considered, then the processing of these customers as a group is unaffected by the lower priority customers. The crux of the analysis is in realizing that the work content of this system (with only the top i classes) at all times is the same as an $M/G/1$ queue with FCFS and top i classes due to the work conserving nature. Therefore using the results for work-conserving systems, the performance analysis of this system is done.

Now consider an $M/G/1$ queue with only the first i classes and FCFS service. The net arrival rate is

$$\lambda(i) = \lambda_1 + \lambda_2 + \ldots + \lambda_i,$$

the average service times is

$$\frac{1}{\mu(i)} = \sum_{j=1}^{i} \frac{\lambda_j E[S_j]}{\lambda(i)},$$

and the second moment of service times is

$$S^2(i) = \sum_{j=1}^{i} \frac{\lambda_j E[S_j^2]}{\lambda(i)}.$$

Also let $\rho(i) = \lambda(i)/\mu(i)$. Let W_{jq}^{prp} be the waiting time in the queue for class j customers under pre-emptive resume policy. Using the principle of work conservation (see Result 8),

$$\sum_{j=1}^{i} \rho_j W_{jq}^{prp} = \rho(i) \frac{\lambda(i) S^2(i)}{2(1 - \lambda(i)/\mu(i))}.$$

Notice that the left hand side of the above expression is the first i classes under pre-emptive resume and the right hand side being FCFS with only the first i classes of customers. Now, we can recursively compute W_{1q}, then W_{2q}, and so on till W_{Rq} via the above equations for $i = 1, 2, \ldots R$.

Other average measures for the pre-emptive resume policy can be obtained as follows:

$$W_i^{prp} \;\; = \;\; W_{iq}^{prp} + E[S_i],$$

$$\begin{aligned} L_i^{prp} &= \lambda_i W_i^{prp}, \\ L_{iq}^{prp} &= L_i^{prp} - \rho_i. \end{aligned}$$

Sometimes performance measures for individual classes are required and other times aggregate performance measures across all classes. The results for the individual classes can also be used to obtain the overall performance measures as follows:

$$\begin{aligned} L^{prp} &= L_1^{prp} + L_2^{prp} + \ldots + L_R^{prp}, \\ W^{prp} &= L^{prp}/\lambda, \\ W_q^{prp} &= W^{prp} - 1/\mu, \\ L_q^{prp} &= \lambda W_q^{prp}. \end{aligned}$$

9.5 Multi-Station and Single-Class Queues

So far we have only considered single stage queues. However in practice there are several systems where customers go from one station (or stage) to other stations. For example, in a theme park the various rides are the different stations and customers wait in lines at each station and randomly move to other stations. Several engineering systems such as production, computer-communication and transportation systems can also be modeled as queueing networks.

In this section we only consider single class queueing networks. The network is analyzed by considering each of the individual stations one by one. Therefore the main technique would be to decompose the queueing network into individual queues or stations and develop characteristics of arrival processes for each individual station. Similar to the single station case, here too we start with networks with Poisson arrivals and exponential service times,

and then eventually move to more general cases. There are two types of networks: open queueing networks (customers enter and leave the networks) and closed queueing networks (the number of customers in the networks stay a constant).

9.5.1 Open Queueing Networks: Jackson Network

A Jackson Network is a special type of open queueing network where arrivals are Poisson and service times are exponential. In addition, a queueing network is called a Jackson network if it satisfies the following assumptions:

1. It consists of N service stations (nodes).
2. There are s_i servers at node i ($1 \leq s_i \leq \infty$), $1 \leq i \leq N$.
3. Service times of customers at node i are iid $\exp(\mu_i)$ random variables. They are independent of service times at other nodes.
4. There is infinite waiting room at each node.
5. Externally, customers arrive at node i in a Poisson fashion with rate λ_i. All arrival processes are independent of each other and the service times. At least one λ_i must be non-zero.
6. When a customer completes service at node i, he or she or it departs the system with probability r_i or joins the queue at node j with probability p_{ij}. Here $p_{ii} > 0$ is allowed. It is required that $r_i + \sum_{j=1}^{N} p_{ij} = 1$ as all customers after completing service at node i either depart the system or join another node. The routing of a customer does not depend on the state of the network.
7. Let $P = [P_{ij}]$ be the routing matrix. Assume that $I - P$ is invertible, where I is an

$N \times N$ identity matrix. The $I - P$ matrix is invertible if there is at least one node from where customers can leave the system.

To analyze the Jackson network, as mentioned earlier, we decompose the queueing network into the N individual nodes (or stations). The results are adapted from Kulkarni [7]. In steady state, the total arrival rate into node j (external and internal) is denoted by a_j and is given by

$$a_j = \lambda_j + \sum_{i=1}^{N} a_i p_{ij} \quad j = 1, 2, \ldots, N.$$

Let $a = (a_1, a_2, \ldots, a_N)$. Then a can be solved as

$$a = \lambda(I - P)^{-1}.$$

The following results are used to decompose the system: (a) the departure process from an $M/M/s$ queue is a Poisson process; (b) the superposition of Poisson process forms a Poisson process; and (c) Bernoulli (i.e. probabilistic) splitting of Poisson processes forms Poisson processes. Therefore the resultant arrival into any node or station is Poisson. Then we can model node j as an $M/M/s_j$ queue with $PP(a_j)$ arrivals, $\exp(\mu_j)$ service and s_j servers (if the stability condition at each node j is satisfied, i.e. $a_j < s_j\mu_j$). Hence it is possible to obtain the steady state probability of having n customers in node j as

$$\phi_j(n) = \begin{cases} \frac{1}{n!}\left(\frac{a_j}{\mu_j}\right)^n \phi_j(0) & \text{if } 0 \le n \le s_j - 1 \\ \frac{1}{s_j!\, s_j^{n-s_j}}\left(\frac{a_j}{\mu_j}\right)^n \phi_j(0) & \text{if } n \ge s_j \end{cases}$$

where $\phi_j(0) = \left[\sum_{n=0}^{s_j-1}\left\{\frac{1}{n!}(a_j/\mu_j)^n\right\} + \frac{(a_j/\mu_j)_j^s}{s_j!}\frac{1}{1 - a_j/(s_j\mu_j)}\right]^{-1}$.

Now looking back into the network as a whole, let X_i be the steady state number of customers in node i. Then it is possible to show that

$$P\{X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N\} = \phi_1(x_1)\phi_2(x_2)\ldots\phi_N(x_N).$$

The above form of the joint distribution is known as product form. In steady state, the queue lengths at various nodes are independent random variables. Therefore what this implies is that each node (or station) in the network behaves as if it is an independent $M/M/s$ queue. Hence each node j can be analyzed as an independent system and performance measures can be obtained.

Specifically, it is possible to obtain performance measures at station j such as the average number of customers (L_j), average waiting time (W_j), time in queue not including service (W_{jq}), number in queue not including service (L_{jq}), distribution of waiting time ($F_j(x)$), and all other measures using the single station $M/M/s$ queue analysis in Section 9.3.2.

Besides the Jackson network, there are other Product-form Open Queueing Networks. The state dependent service rate and the state dependent arrival rate problems are two cases when product-form solution exists.

State dependent service

Assume that the service rate at node i when there are n customers at that node is given by $\mu_i(n)$ with $\mu_i(0) = 0$. Also assume that the service rate does not depend on the states of the remaining nodes. Then define the following: $\phi_i(0) = 1$ and

$$\phi_i(n) = \prod_{j=1}^{n} \left(\frac{a_i}{\mu_i(j)} \right) \quad n \geq 1$$

where a_j is as before, the effective arrival rate into node j.

The steady state probabilities are given by

$$P\{X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N\} = c \prod_{i=1}^{N} \phi_i(x_i),$$

where the normalizing constant c is

$$c = \left\{ \prod_{i=1}^{N} \left\{ \sum_{n=0}^{\infty} \phi_i(n) \right\} \right\}^{-1}.$$

Using the above joint distribution it is possible to obtain certain performance measures. However one of the difficulties is to obtain the normalizing constant. Once that is done, the marginal distribution at each node (or station) can be obtained. That can be used to get the distribution of the number of customers in the system as well as the mean (and even higher moments). Then using Little's law, the mean waiting time can also be obtained.

State dependent arrivals and service

In a fashion similar to the case of state dependent service, the analysis of state dependent arrivals and service can be extended. Let $\lambda(n)$ be the total arrival rate to the network as a whole when there are n customers in the entire network. Assume that u_i is the probability that an incoming customer joins node i, independently of other customers. Therefore external arrivals to node i are at rate $u_i\lambda(n)$. The service rate at node i when there are n_i customers at that node is given by $\mu_i(n_i)$ with $\mu_i(0) = 0$.

Let b_i be the unique solution to

$$b_j = u_j + \sum_{i=1}^{N} b_i p_{ij}.$$

Define the following: $\phi_i(0) = 1$ and

$$\phi_i(n) = \prod_{j=1}^{n} \left(\frac{b_i}{\mu_i(j)} \right) \quad \text{for } n \geq 1.$$

Define $\hat{x} = \sum_{i=1}^{N} x_i$. The steady state probabilities are given by

$$P\{X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N\} = c \prod_{i=1}^{N} \phi_i(x_i) \prod_{j=1}^{\hat{x}} \lambda(j),$$

where the normalizing constant c is

$$c = \left\{ \sum_{x} \prod_{i=1}^{N} \phi_i(x_i) \prod_{j=1}^{\hat{x}} \lambda(j) \right\}^{-1}.$$

9.5.2 Closed Queueing Networks (exponential service times)

Closed queueing networks are networks where there are no external arrivals to the system and no departures from the system. They are popular in population studies, multiprogrammed computer systems, window flow control, Kanban, etc. It is important to note that the number of customers being a constant is essentially what is required. This can happen if a new customer enters the network as soon as an existing customer leaves (a popular scheme in just-in-time manufacturing). Most of the results are adapted from Kulkarni [7]. We need a few assumptions to analyse these networks:

1. The network has N service stations and a total of C customers.
2. The service rate at node i, when there are n customers in that node is $\mu_i(n)$ with $\mu_i(0) = 0$ and $\mu_i(n) > 0$ for $1 \leq n \leq C$.
3. When a customer completes service at node i, he/she/it joins node j with probability p_{ij}.

Let the routing matrix $P = [p_{ij}]$ be such that it is irreducible. That means, it is possible to reach every node from every other node in one or more steps or hops. Define $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_N)$ such that

$$\pi = \pi P \quad \text{and} \quad \sum_{i=1}^{N} \pi_i = 1.$$

Indeed the P matrix is a stochastic matrix, which is a lot similar to the transition probability matrix of DTMCs. However it is important to note that nothing is modeled as a DTMC.

Define the following: $\phi_i(0) = 1$ and

$$\phi_i(n) = \prod_{j=1}^{n} \left(\frac{\pi_i}{\mu_i(j)} \right) \quad n \geq 1$$

The steady state probabilities are given by

$$P\{X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N\} = G(C) \prod_{i=1}^{N} \phi_i(x_i),$$

where the normalizing constant $G(C)$ is chosen such that

$$\sum_{x_1, x_2, \ldots, x_N} P(X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N) = 1.$$

Note that for this problem, similar to the two previous product-form cases, the difficulty arises is computing the normalizing constant. In general it is not computationally trivial.

Some additional results can be obtained such as the *Arrival Theorem* explained below.

Result 9 *In a closed product form queueing network, for any x, the probability that x jobs are seen at the time of arrival to node i when there are C jobs in the network is equal to the probability that there are x jobs at this node with one less job in the network (i.e. $C - 1$).*

This gives us the relationship between the arrival time probabilities and steady-state probabilities. If $\pi_{ij}(C)$ denotes the probability that in a closed-queueing network of C customers, an arriving customer into node i sees j customers ahead of him/her/it in steady state. Also, if $p_{ij}(C-1)$ denotes the probability that in a "hypothetical" closed-queueing network of $C - 1$ customers, there are j customers in node i in steady state. Result 9 states that

$$\pi_{ij}(C) = p_{ij}(C - 1).$$

Single Server Closed Queueing Networks

Assume that for all i, there is a single server at node i with service rate μ_i. Then the mean performance measures can be computed without going through the computation of

the normalizing constant. Define the following:

- $W_i(k)$: Average waiting time in node i when there are k customers in the network;
- $L_i(k)$: Average number in node i when there are k customers in the network;
- $\lambda(k)$: Overall throughput of the network when there are k customers in the network.

Initialize $L_i(0) = 0$ for $1 \leq i \leq N$. Then for $k = 1$ to C, iteratively compute for each i

$$\begin{aligned} W_i(k) &= \frac{1}{\mu_i}[1 + L_i(k-1)], \\ \lambda(k) &= \frac{k}{\sum_{i=1}^{N} a_i W_i(k)}, \\ L_i(k) &= \lambda(k) W_i(k) a_i. \end{aligned}$$

The first of the above 3 equations comes from the Arrival Theorem (Result 9). The second and third come from Little's law applied to the network and a single node respectively.

9.5.3 Algorithms for Non-product-form Networks

The product form for the joint distribution enables one to analyze each node independently. When the inter-arrival times or service times are not exponential, then a non-product form emerges. We will now see how to develop approximations for these non-product-form networks. We present only one algorithm here, namely the diffusion approximation. However the literature is rich with several others such as the maximum entropy method, QNA for single class, etc. We now illustrate the diffusion approximation algorithm as described in Bolch et al [1].

Diffusion Approximation: Open Queueing Networks

1. **Key Idea**: Substitute the discrete process $\{X_i(t), t \geq 0\}$ that counts the number in the node i, by a continuous diffusion process. Thus a product-form approximation can be obtained that works well under heavy traffic (i.e. traffic intensity in each node is above 0.95 at least).

2. **Assumptions**: Single server at each node. Service time at server i has mean $1/\mu_i$ and SCOV $C_{S_i}^2$. There is a single stream of arrivals into the network with inter-arrival times having a mean of $1/\lambda_i$ and SCOV C_A^2. There is a slight change of notations for the routing probabilities (consider the outside world as node 0):

   (a) if $i > 0$ then p_{ij} is the probability of going from node i to node j upon service completion in node i;

   (b) if $i = 0$ then p_{0j} is the probability that an external arrival joins node j;

   (c) if $j = 0$ then p_{i0} is the probability of exiting the queueing network upon service completion in node i.

3. **The Algorithm**:

   (a) Obtain visit ratios a_j for all $1 \leq j \leq N$ by solving

   $$a_j = \sum_{i=1}^{N} p_{ij} a_i + p_{0j},$$

   with $a_0 = 1$. Then for $1 \leq i, j \leq N$, if $P = [p_{ij}]$ an $N \times N$ matrix, then $a = [a_1 a_2 \ldots a_N] = [p_{01} p_{02} \ldots p_{0N}][I - P]^{-1}$.

   (b) For all $1 \leq i \leq N$, compute the following (assume $C_{S_0}^2 = C_A^2$):

   $$\begin{aligned} C_{A_i}^2 &= 1 + \sum_{j=0}^{N} (C_{S_j}^2 - 1) p_{ji}^2 a_j / a_i, \\ \rho_i &= \frac{\lambda a_i}{\mu_i}, \end{aligned}$$

$$\theta_i = \exp\left[\frac{-2(1-\rho_i)}{C_{A_i}^2\rho_i + C_{S_i}^2}\right],$$

$$\phi_i(x_i) = \begin{cases} 1-\rho_i & \text{if } x_i = 0, \\ \rho_i(1-\theta_i)\theta_i^{x_i-1} & \text{if } x_i > 0. \end{cases}$$

(c) The steady state joint probability is

$$p(x) = \prod_{i=1}^{N} \phi_i(x_i).$$

(d) The mean number of customers in node i is

$$L_i = \frac{\rho_i}{1-\theta_i}.$$

Diffusion Approximation: Closed Queueing Networks

All the parameters are identical to the open queueing network case. There are C customers in the closed queueing network. There are two algorithms, one for large C and other for small C.

1. **Algorithm Bottleneck (for large C)**

   (a) Obtain visit ratios a_j for all $1 \leq j \leq N$ by solving

   $$a_j = \sum_{i=1}^{N} p_{ij} a_i.$$

   As there will not be a unique solution, one can normalize by $a_1 + a_2 + \ldots + a_N = 1$.

   (b) Identify the bottleneck node b as the node with the largest a_i/μ_i value among all $i \in [1, N]$.

   (c) Set $\rho_b = 1$. Using the relation $\rho_b = \lambda a_b/\mu_b$, obtain $\lambda = \mu_b/a_b$. Then for all $i \neq b$, obtain $\rho_i = \lambda a_i/\mu_i$.

   (d) Follow the open queueing network algorithm now to obtain for all $i \neq b$, $C_{A_i}^2$, θ_i and $\phi_i(x_i)$.

(e) Then the average number of customers in node i ($i \neq b$) is

$$L_i = \frac{\rho_i}{1 - \theta_i}$$

and $L_b = C - \sum_{i \neq b} L_i$.

2. **Algorithm MVA (for small C)**:

   Consider MVA for product-form closed queueing networks (see Section 9.5.2). Use that analysis and iteratively compute for all $1 \leq k \leq C$, the quantities $W_i(k)$, $\lambda_i(k)$ and $L_i(k)$. Assume overall throughput $\lambda = \lambda(C)$. Then follow the open queueing network algorithm (in Section 9.5.3).

9.6 Multi-Station and Multi-Class Queues

Consider an open queueing network with multiple classes where the customers are served according to FCFS. To obtain performance measures we use a decomposition technique. For that, we first describe the problem setting, develop some notation and illustrate an algorithm.

9.6.1 Scenario

We first describe the setting, some of which are underlying assumptions needed to carry out the analysis.

1. There are N service stations (nodes) in the open queueing network. The outside world is denoted by node 0 and the others $1, 2, \ldots, N$.

2. There are m_i servers at node i ($1 \leq m_i \leq \infty$), for $1 \leq i \leq N$.

3. The network has R classes of traffic and class switching is not allowed.

4. Service times of class r customers at node i are iid with mean $1/\mu_{i,r}$ and SCOV $C^2_{S_{i,r}}$.
5. The service discipline is FCFS.
6. There is infinite waiting room at each node.
7. Externally, customers of class r arrive at node i according to a general interarrival time with mean $1/\lambda_{0i,r}$ and SCOV $C^2_{A_{i,r}}$.
8. When a customer of class r completes service at node i, he or she or it joins the queue at node j ($j \in [0, N]$) with probability $p_{ij,r}$.

After verifying that the above scenario (and assumptions) are applicable, the next task is to obtain all the input parameters for the model described above, i.e. for each $i \in [1, N]$ and $r \in [1, R]$, m_i, $1/\mu_{i,r}$, $C^2_{S_{i,r}}$, $1/\lambda_{0i,r}$, $C^2_{A_{i,r}}$, $p_{ij,r}$ (for $j \in [0, N]$).

9.6.2 Notation

Before describing the algorithm, some of the notations are in Table 9.4 for easy reference. A few of the notations are inputs to the algorithm (as described above) and others are derived in the algorithm. The algorithm is adapted from Bolch et al [1] albeit with different set of notations. The reader is also encouraged to refer to Bolch et al [1] for further insights into the algorithm.

9.6.3 Algorithm

The decomposition algorithm essentially breaks down the network into individual nodes and analyzes each node as an independent $GI/G/s$ queue with multiple classes (note that this is only FCFS and hence handling multiple classes is straightforward). For the $GI/G/s$ analysis,

N	Total number of nodes
Node 0	Outside world
R	Total number of classes
$\lambda_{ij,r}$	Mean arrival rate from node i to node j of class r
$\lambda_{i,r}$	Mean arrival rate to node i of class r (or mean departure rate from node i of class r)
$p_{ij,r}$	Fraction of traffic of class r that exit node i and join node j
λ_i	Mean arrival rate to node i
$\rho_{i,r}$	Utilization of node i due to customers of class r
ρ_i	Utilization of node i
μ_i	Mean service rate of node i
$C^2_{S_i}$	SCOV of service time of node i
$C^2_{ij,r}$	SCOV of time between two customers going from node i to node j
$C^2_{A_{i,r}}$	SCOV of class r inter arrival times into node i
$C^2_{A_i}$	SCOV of inter arrival times into node i
$C^2_{D_i}$	SCOV of inter departure times from node i

Table 9.4: Notation used in algorithm

we require for each node and each class the mean arrival and service rates as well as the SCOV of the inter arrival times and service times. The bulk of the algorithm in fact is to obtain them. There are three situations where this becomes hard: when multiple streams are merged (superposition), when traffic flows through a node (flow), and, when a single stream is forked into multiple streams (splitting). For convenience, we assume that just before entering a queue, the superposition takes place which results in one stream. Likewise we assume that upon service completion, there is only one stream which gets split into multiple streams. There are 3 basic steps in the algorithm (a software developed by Kamath [6] uses the algorithm and refinements; it can be downloaded for free and used for analysis).

Step 1: Calculate the mean arrival rates, utilizations and aggregate service rate parameters using the following:

$$\begin{aligned}
\lambda_{ij,r} &= \lambda_{i,r} p_{ij,r}, \\
\lambda_{i,r} &= \lambda_{0i,r} + \sum_{j=1}^{N} \lambda_{j,r} p_{ji,r}, \\
\lambda_i &= \sum_{r=1}^{R} \lambda_{i,r}, \\
\rho_{i,r} &= \frac{\lambda_{i,r}}{m_i \mu_{i,r}}, \\
\rho_i &= \sum_{r=1}^{R} \rho_{i,r} \quad \text{(condition for stability } \rho_1 < 1 \ \forall i) \ , \\
\mu_i &= \frac{1}{\sum_{r=1}^{R} \frac{\lambda_{i,r}}{\lambda_i} \frac{1}{m_i \mu_{i,r}}} = \frac{\lambda_i}{\rho_i}, \\
C_{S_i}^2 &= -1 + \sum_{r=1}^{R} \frac{\lambda_{i,r}}{\lambda_i} \left(\frac{\mu_i}{m_i \mu_{i,r}} \right)^2 (C_{S_{i,r}}^2 + 1).
\end{aligned}$$

Step 2: Iteratively calculate the coefficient of variation of inter-arrival times at each node. Initialize all $C_{ij,r} = 1$ for the iteration. Then until convergence perform (i), (ii) and (iii) cyclically.

(i) Superposition (aggregating customers from all nodes j and all classes r the SCOV of inter-arrival time into node i):

$$C_{A_{i,r}}^2 = \frac{1}{\lambda_{i,r}} \sum_{j=0}^{N} C_{ji,r}^2 \lambda_{j,r} p_{ji,r},$$

$$C_{A_i}^2 = \frac{1}{\lambda_i} \sum_{r=1}^{R} C_{A_{i,r}}^2 \lambda_{i,r}.$$

(ii) Flow (departing customers from node i have inter-departure time SCOV as a function of the arrival times, service times and traffic intensity into node i):

$$C_{D_i}^2 = 1 + \frac{\rho_i^2 (C_{S_i}^2 - 1)}{\sqrt{m_i}} + (1 - \rho_i^2)(C_{A_i}^2 - 1).$$

(iii) Splitting (computing the class-based SCOV for class r customers departing from node i and arriving at node j):

$$C_{ij,r}^2 = 1 + p_{ij,r}(C_{D_i}^2 - 1).$$

Note that the splitting formula is exact if the departure process is a renewal process. However, the superposition and flow formulae are approximations. Several researchers has provided expressions for the flow and superposition. The above is from Ward Whitt's QNA [9].

Step 3: Obtain performance measures such as mean queue length and mean waiting times using standard $GI/G/m$ queues. Treat each queue as independent. Choose α_{m_i} such that

$$\alpha_{m_i} = \begin{cases} \frac{\rho_i^{m_i} + \rho_i}{2} & \text{if } \rho_i > 0.7, \\ \rho_i^{\frac{m_i+1}{2}} & \text{if } \rho_i < 0.7. \end{cases}$$

Then the mean waiting time for class r customers in the queue (not including service) of node i is approximately

$$W_{iq} \approx \frac{\alpha_{m_i}}{\mu_i} \left(\frac{1}{1-\rho_i} \right) \left(\frac{C_{A_i}^2 + C_{S_i}^2}{2m_i} \right).$$

Notice that for all classes r at node i, W_{iq} is the waiting time in the queue. Other performance measures at node i and across the network can be obtained using standard relationships.

9.7 Concluding Remarks

In this chapter we presented some of the fundamental scenarios and results for single as well as multi-class queueing systems and networks. However, this by no means does justice to the vast amount of literature available in the field as the chapter has barely scratched the surface of queueing theory. But with this background it should be possible to read through relevant

articles and books that model several other queueing systems. In a nutshell, queueing theory can be described as an analytical approach for system performance analysis. There are other approaches for system performance analysis such as simulations. It is critical to understand and appreciate situations when it is more appropriate to use queueing theory as well as situations where one is better off using simulations.

Queueing theory is more appropriate when: (a) several what-if situations need to be analyzed expeditiously, viz. what happens if the arrival rate doubles, triples, etc.; (b) insights into relationship between variables are required, viz. how is the service time related to waiting time; (c) to determine best course of action for any set of parameters, viz. is it always better to have one queue with multiple servers than one queue for each server; (d) formulae are needed to plug into optimization routines, viz. to insert into a non-linear program the queue length must be written as a function to optimize service speed. Simulations on the other hand are more appropriate when: (a) system performance measures are required for a single set of numerical values; (b) performance of a set of given policies need to be evaluated numerically; (c) assumptions needed for queueing models are unrealistic (which is the most popular reason for using simulations). Having said that, in practice it is not uncommon to use a simulation model to verify analytical results from queueing models or use analytical models for special cases to verify simulations.

Another important aspect, especially for practitioners, is the trade off between using physics versus phychology. Queueing theory in general and this chapter in particular deals with the physics of waiting lines or queues. One should realize that the best solution is not necessarily one that uses physics of queues but maybe some phychological considerations. A classic example is a consultant who was approached by a hotel where customers were complaining about how long they waited to get to their rooms using the elevators. Instead of designing a new system with more elevators (and a huge cost thereby), the consultant

simply advised placing mirrors near the elevator and inside the elevator. By doing that, although the actual time in the system does not improve, but the perceived time surely does as the customers sometimes do not realize they are waiting while they busily stare at the mirrors!

From a research standpoint, there are several unsolved problems today and a few of them are described below. For the single-class and single-station systems issues such as: long-range dependent arrivals and service, time-dependent arrival and service rates, non-identical servers, and time varying capacity and number of servers have received limited attention. For the multi-class and single-station queues policies for scheduling customers especially when some classes have heavy-tailed service times (and there are more than one servers) are being actively pursued from a research standpoint. For the single-class and multi-station queues, the situation where arrival and service rates at a node depend on states of some of the other nodes has not been explored. For the multi-class and multi-station case, especially with re-entrant lines, performance analysis is being pursued for policies other than FCFS (such as pre-emptive and non-preemptive priority).

Acknowledgements

The author would like to thank the anonymous reviewers for their comments and suggestions that significantly improved the content and presentation of this book chapter. The author is also grateful to Prof. Ravindran for considering him to write this chapter.

References

[1] G. Bolch, S. Greiner, H. de Meer and K.S. Trivedi. *Queueing Networks and Markov Chains.* 1st Edition, John Wiley and Sons Inc., NY, 1998.

[2] J. A. Buzacott and J.G. Shanthikumar. *Stochastic Models of Manufacturing Systems.* Prentice-Hall, New York 1992.

[3] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory.* 3rd Ed., John Wiley and Sons Inc., NY, 1998

[4] M. Hlynka. *Queueing Theory Page.* http://www2.uwindsor.ca/˜hlynka/queue.html.

[5] M. Hlynka. *List of Queueing Theory Software.* http://www2.uwindsor.ca/˜hlynka/qsoft.html

[6] M. Kamath. *Rapid Analysis of Queueing Systems Software.* http://www.okstate.edu/cocim/raqs/

[7] V. G. Kulkarni. *Modeling and Analysis of Stochastic Systems.* Texts in Statistical Science Series. Chapman and Hall, Ltd., London, 1995.

[8] M. Moses, S. Seshadri and M. Yakirevich. *HOM Software* http://www.stern.nyu.edu/HOM

[9] W. Whitt. *The Queueing Network Analyzer.* The Bell System Technical Journal, 62, 9, pp. 2779-2815, 1983.

[10] R. W. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice Hall, Englewood Cliffs, N.J., 1989