

Queuing Systems

Maria Kihl



LUNDS UNIVERSITET

Institutionen för Telekommunikationssystem

© Maria Kihl

Printed in Sweden
Lund 2004

Table of Contents

	Table of Contents	i
1	Introduction	1
	1.1 Real-time systems	1
	1.2 Analysing a real-time system	2
	1.3 Performance measures	3
	1.4 System examples	4
2	General terminology for queuing systems	7
	2.1 Arrival processes	8
	2.2 Service times	8
	2.3 The queue	9
	2.4 System variables	9
	2.5 Queuing system classification	10

3	Basic queuing theoretic formulas	13
3.1	Little's theorem	13
3.2	Some basic performance metrics	15
3.3	Stability conditions	17
3.4	A general result for single server queues	18
3.5	Exercises	18
4	M/M/c-systems	23
4.1	The M/M/1-system with unlimited queue	23
4.2	Limited waiting room	29
4.3	Finite client population	32
4.4	Multiple servers	36
4.5	The M/M/c-system without queue (Erlang's model)	39
4.6	Exercises	42
5	M/G/1-systems	49
5.1	Pollaczek-Khintchin's formula	50
5.2	Stationary probability distribution	53
5.3	Waiting time distributions	55
5.4	Exercises	56
6	Queueing networks	61
6.1	Feed-forward networks	61
6.2	Networks with feedback	65
6.3	Exercises	69
7	Probability theory	77
7.1	Basic principles	77
7.2	Random variables	78
7.3	Z-transforms of probability distributions	82
7.4	Laplace transforms of distribution functions	83
7.5	Sum of two random variables	84
7.6	Exercises	86

8	Stochastic processes	87
8.1	The Poisson process	87
8.2	Markov chains	89
8.3	Discrete-time Markov chains	90
8.4	Continuous-time Markov chains	92
8.5	Exercises	96
	References	99

Chapter 1

Introduction

Assume that we have a system of some kind, for example a telephone network, Internet, a web server or some other system or part of a system. A system can formally be defined as a collection of entities, for example people or machines, that act and interact together toward the accomplishment of some task. In practice, what is meant by “the system” depends on the objectives of a particular study. The collection of entities that comprise a system for one study might be only a subset of the overall system for another.

1.1 Real-time systems

This book is focussed on so-called *real-time systems*. A real-time system has the basic structure as shown in Figure 1.1. It consists of a computer system with a processor that can serve jobs that arrive from the “outside world”. When a job has been served, the computer system delivers an answer of some kind. A job that arrives and cannot be served at once is placed in a job queue. Usually, a job has a so-called *deadline*, that is, an upper limit for the time it takes before the job has been served, the so-called response time for the system.

There are two classes of real-time systems: *Hard* real-time systems and *Soft* real-time systems. In a hard real-time system, all jobs must be served within their deadlines, otherwise the system may crash or fail. One example of a hard real-time system is the steering mechanism for an airplane. In a soft real-time system, on the other hand, some jobs may miss their deadlines or even be blocked with-

out the system taking any damage. One example of a soft real-time system is a web site that processes Hyper Text Transfer Protocol (HTTP) requests coming from clients. In this book, only soft real-time systems are considered.

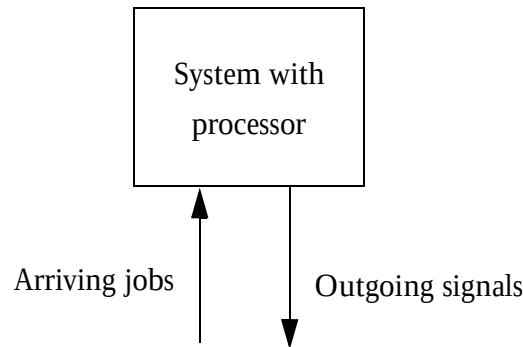


Figure 1.1. *A real-time system.*

1.2 Analysing a real-time system

If we are interested in the performance of a soft real-time system, there are two questions that we often want to find an answer to:

1. How long time does it take for the system to complete a service?
2. If the system is limited, for example with finite buffers, then how high is the probability that a request sent to the system will be blocked?

If we want to design the system for certain applications and traffic scenarios, these two questions must be answered. But to get accurate answers it is necessary to analyse the system in some way.

Before analysing a system, we usually first need to find a good *model* of the system, and thereafter analyse the system with some kind of *method*.

Models

To most people, the word “model” evokes images as cockpits disconnected from their airplanes to be used in pilot training. This is an example of a *physical model* and are not typical of the kinds of models that are usually of interest in operations research systems analysis. The vast majority of models built for such purposes are mathematical, representing a system in terms of logical and quantitative relationships that are then manipulated and changed to see how the model reacts, and thus how the system would react - if the *mathematical model* is a valid one.

Although computers are completely deterministic machines, our analyses will use the theory of probability extensively. In fact, computer performance modelling has been one of the driving forces behind many recent advances in applied probability theory. There are two reasons why we use probability theory so extensively. First, many of the processes

involved are inherently random. For example, the successful transmission of a packet in a computer network is a random event since there is a chance that there will be some interference with the communication, and it will need to be retransmitted. Secondly, it is more convenient. Even if we had complete knowledge of all the times involved at each stage in the process, the size of the data needed as input to the model would be extremely large. Also, the results and conclusions that we could draw would be needlessly specific. By representing the essential characteristics of the system with a few parameters, we can evaluate the performance and hope to demonstrate the effects of changes in the parameter values.

Therefore, we will in this book develop so-called *stochastic models*, which have one or more random inputs. Many of our problems that we shall investigate are approached using *queuing theory*. The application of queuing theory started 1917 when the Danish engineer A.K. Erlang analysed the behaviour of simple queues to assist him in designing telephone exchanges. This telephonic flavour continued until the mid 1960s when the first analysis of a computer time-sharing system was undertaken by Scherr.

Methods

Once we have built a mathematical model, it must then be examined to see how it can be used to answer the questions of interest about the system it is supposed to represent. If the model is simple enough, it may be possible to work with its relationships and quantities to get an exact *analytical solution*.

However, many systems are highly complex, so that valid mathematical models of them are themselves complex, which means that analytical solutions becomes impossible to obtain.

In this case, the model can be studied by means of *simulation*. A simulation is the imitation of the operation of a real-world process or system over time. Whether done by hand or on a computer, simulation involves the generation of an artificial history of a system, and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system.

1.3 Performance measures

There are many reasons for why we may wish to calculate the performance of our systems, but for the most part long run costs and revenues are the interest of management. Freak occurrences that happen only in peculiar circumstances are of little interest. We will be examining the long run performance of systems. The notion of *statistical equilibrium* or *steady state* will often be used. This does not necessarily mean that the system behaves in a particularly “steady” manner, but that the probabilities of the system being in a particular state have settled down and are not changing with time.

When designing a new system, the *user-perceived performance* is a very important issue. The users have certain demands on the Quality of Service (QoS), and if the system cannot meet these demands, the system is not working as it should. When a user submits a job to a system, the question that usually needs to be answered is “How long before I

have my answers?”. *Customer waiting time* and its distribution is one of our main interests. The management of the system will be interested in the *number of jobs* waiting, since they must be stored somewhere, and the *throughput* of the system, since it indicates the rate at which revenue will be generated if a charge is made for each job using the system. The *utilisation* of the system, that is the proportion of the time that the system is busy, gives some indication of the scope for increasing throughput and hence revenue, without increasing the resources committed. Note that an increase in throughput may only be possible at the expense of increased delay to customers.

1.4 System examples

We will throughout this book use three examples of telecommunication systems, a web site, a cell in a mobile telephony network, and a packet switched network.

A web site

The web site is shown in Figure 1.2. Clients, connected with the web site through the Internet, request different web pages by sending HTTP Get messages. The web site consists of one or more web servers that process the requests and send a reply with the wanted web pages. A web page is either static or dynamic. A static web page may be stored as a single file. It is therefore easily processed by the site. A dynamic web page must be constructed by executing so called CGI scripts. This means that a request for a dynamic web page takes much longer to process than a request for a static page.

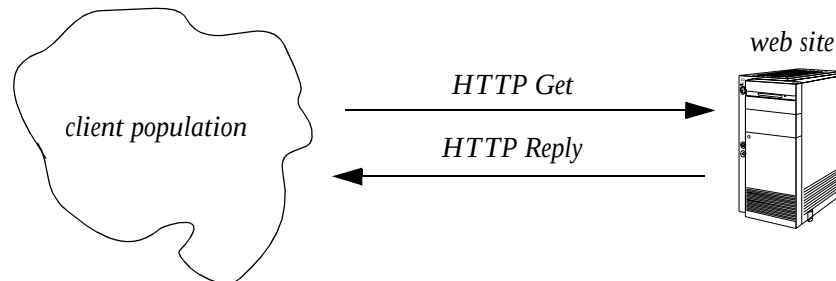


Figure 1.2. A web site.

A cell in a mobile telephony network

A mobile telephony network is divided into so called cells. In each cell there is a base station that communicates with the mobile phones that are located in the cell, see Figure 1.3. Each mobile phone sends data on a specific radio channel. There are a limited number of radio channels in each cell. When a user wants to make a telephone call, the base station decides which radio channel that must be used during the call. If all radio channels are occupied, the call is rejected.

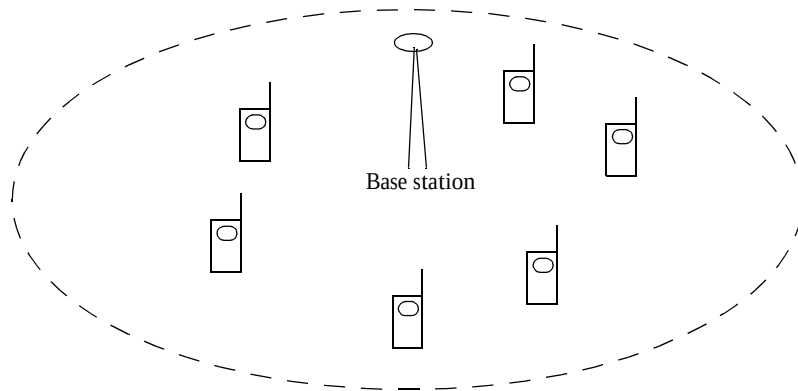


Figure 1.3. *A cell.*

A packet switched network

In a packet switched network, data is transferred in the form of packets. The general structure of a packet switched network is shown in Figure 1.4. The network consists of hosts, routers and links. The hosts are the terminals that contain the user applications. The objective of the routers is to transfer data between the hosts. Each host has an address so that the routers know where the packets should be sent. When data is sent across the network, it passes a number of links. The capacity of a link is usually measured in bits per second.

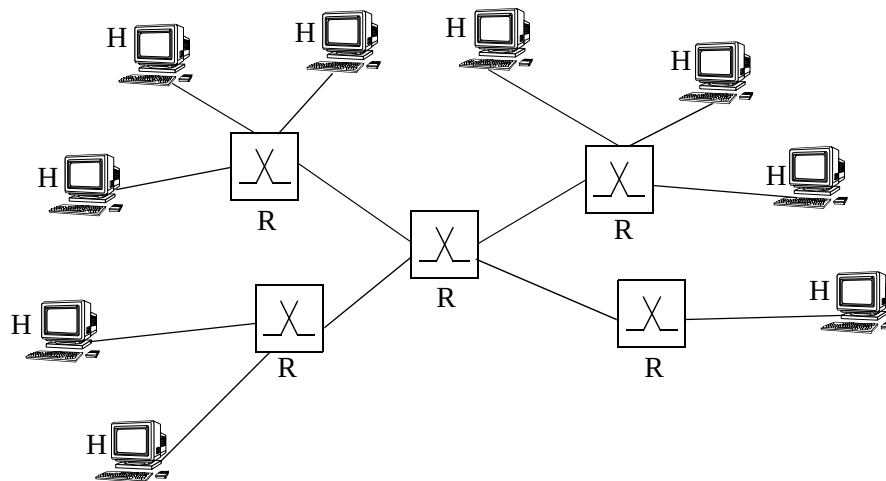


Figure 1.4. *A packet switched network: H=host, R=router.*

Chapter 2

General terminology for queuing systems

This chapter contains the general terminology used in queuing theory. Figure 2.1 shows the basic elements of a queuing system. A queuing system basically consists of an arrival process, a queue, and a number of servers. New jobs arrive according to a stochastic process. Each job should be treated by one of the servers. The length of the service times are distributed according to some distribution function. If all servers are busy when a job arrives, the job is placed in the queue. If the queue is full, the job is blocked. A job that has been serviced leaves the system. This is called a departure.

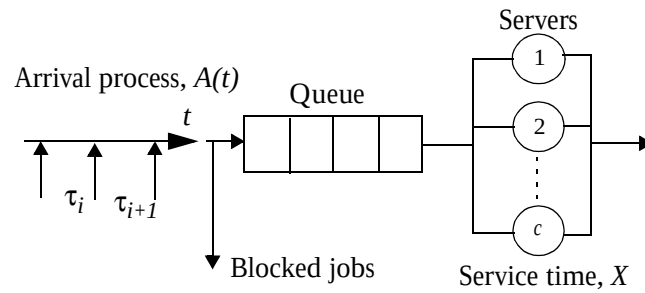


Figure 2.1. *A general queuing system.*

2.1 Arrival processes

Jobs arrive at the system with interarrival times $\tau_1, \tau_2, \dots, \tau_n$. It is assumed that the interarrival times are independent random variables with the same distribution. The average arrival rate to the system, λ , is given by

$$\lambda = \frac{1}{E[\tau]} \quad (2.1)$$

The arrival times are *exponential* if the interarrival times are exponential random variables with mean $E[\tau] = 1/\lambda$, which means that

$$P(\tau \leq t) = 1 - e^{-\lambda t} \quad t > 0 \quad (2.2)$$

The exponential distribution has the memoryless or Markov property. This means that the distribution of the time until the next arrival is independent of the time that has elapsed since the last arrival. It can be shown that the exponential distribution is the only continuous distribution with this property. Because knowledge of when the last arrival occurred does not change the distribution of the time until the next arrival, no information about arrivals need be kept to describe the state of the system.

It can be shown that when the interarrival times are exponential, then the number of arrivals in an interval of length t , $A(t)$, is given by a Poisson random variable with mean λt , which means that

$$P(A(t) = k) = \frac{(\lambda t)^k}{k!} \cdot e^{-\lambda t} \quad k = 0, 1, \dots \quad (2.3)$$

For this reason, an arrival process with exponential interarrival times is called an *Poisson arrival process*.

2.2 Service times

The resources in a queuing system are called *servers* because their function is to serve jobs sent by the customers. The time required to service a job is called the *service time* and is denoted X . X is a random variable, and the density function for X is usually denoted $b(t)$. The maximum rate at which a server can process jobs is attained when the server is continuously busy. When this is the case, the average time between job departures is equal to the average service time. The maximum processing capacity, μ , is therefore given by

$$\mu = \frac{1}{E[X]} \quad (2.4)$$

The processing capacity μ can be compared to the maximum flow that can be sustained over a pipe. The number of servers, c , in a queuing system can be greater than one. The total processing capacity of a queuing system is therefore given by $c\mu$.

Example 2.1: If the service times are *exponentially distributed*, the density function is given by:

$$b(t) = 1 - e^{-\mu t} \quad t > 0 \quad (2.5)$$

The distribution of the time until the next departure depends only on whether or not there is a customer in the server and not on the length of time that the current service, if any, has lasted (due to the Markov property).

2.3 The queue

An ideal queuing system is one where jobs arrive at equal intervals and in which they require a constant service time. As long as the total processing capacity is higher than the arrival rate, each job arrives at an available server and there is no waiting time.

In general, however, the interarrival times and service times are random. The combination of a long service time followed by a short interarrival time can lead to a situation in which all servers are busy when a job arrives.

For this reason, in many applications a queue is provided so that a job can wait for an available server. When a server becomes available the next job to receive service is selected according to the service discipline. Possible service disciplines are first-in-first-out (FIFO), last-in-first-out (LIFO), or service according to priority class. Note that FIFO is the same as first-come-first-served (FCFS) and that LIFO is the same as last-come-first-served (LCFS). If nothing else is described, the FIFO service discipline is assumed. The maximum number of jobs allowed in the system is denoted by L . Note that L includes both the jobs in queue and those in service.

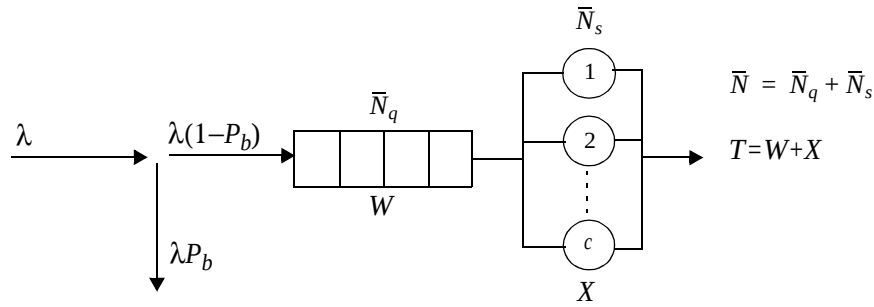


Figure 2.2. System variables

2.4 System variables

There are a number of system variables that are used when analysing a queuing system, see Figure 2.2. The average total time that a job spends in the system is denoted T , which consists of the average time spent waiting in the queue, W , plus the average time spent in service, X . The average total time is sometimes called the *response time*, that is the time it takes from a customer sends a job until the “response” is sent by the server. The average

total number of jobs in the system is denoted by $\bar{N}$, which consists of the average number in queue, $\bar{N}_q$, plus the average number in service, $\bar{N}_s$. P_b denotes the fraction of jobs that are blocked. If the queue is unlimited, $P_b=0$. The *effective arrival rate*, λ_{eff} , that is the actual rate of jobs that enters the system, is given by $\lambda_{eff} = \lambda(1 - P_b)$. The *blocked rate of jobs*, λ_b , is in the same given by $\lambda_b = \lambda P_b$.

2.5 Queuing system classification

Queuing systems are classified by a notation that specifies the following characteristics:

- Interarrival time distribution.
- Service time distribution.
- Number of servers.
- Maximum number in the system.

The general notation is shown in Figure 2.3. As can be seen in the figure, a queuing system is specified as $A/B/c/K$ where A corresponds to the arrival process, B corresponds to the service time distribution, c is the number of servers and K is the maximum occupancy. Some common examples are described below.

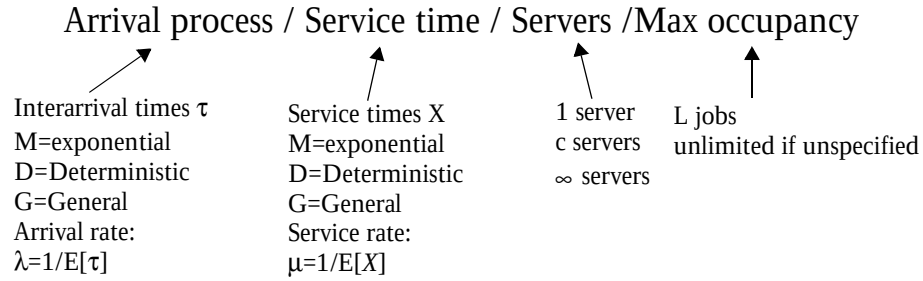


Figure 2.3. Queuing system classification.

In an **M/M/1-system** the jobs arrive at the system in a Poisson stream at rate λ , i.e., the time between successive job arrivals are exponential distributed. Further, each service of a customer takes a time which is exponential distributed in length, with mean $1/\mu$. There is one server and the queue is unlimited.

An **M/M/1/L-system** has the same properties as an M/M/1-system, that is one server with exponential interarrival times and service times. The only difference is that the queue is limited with $L-1$ places. This means that a job may be blocked when it arrives at the system.

In an **M/M/c-system**, there is an unlimited queue and there are c servers that can process the jobs. The servers are homogenous, which means that their processing capacities are the same. If there are more than one server available when a job arrives, the job randomly chooses one of the servers. This means that the servers equally share the work.

In an **M/M/c/c-system** there is no queue. This system is also called the *Erlang system*, named after the well-known Danish teletraffic scientist. It is a so-called *busy system*, which means that if all servers are busy when a job arrives, the job is blocked.

The **M/G/1-system** is more general than the M/M/1-system. In this system the service times can have any statistical distribution that must be further specified.

Example 2.2: A simple model of the web site is a so-called *single server queue*, see Figure 2.4. Requests sent by the clients have to be serviced by the Central Processor Unit (CPU). If the CPU is busy when a request arrives, the request is placed in a job queue. When the CPU has completed a request, it fetches another request from the queue according to some service discipline. Usually, the service discipline is First-Come-First-Served (FCFS) but other service disciplines are of course also possible. Two examples are Last-Come-First-Served (LCFS) and Earliest-Deadline-First (EDF).

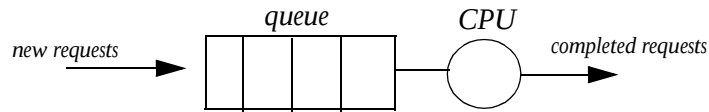


Figure 2.4. A model of the web site.

Example 2.3: The radio channels in the mobile network cell can be modelled as a so-called *loss system*, see Figure 2.5. Each radio channel can handle one telephone call at the time. If all radio channels are busy when a new call arrives, this call must be blocked.

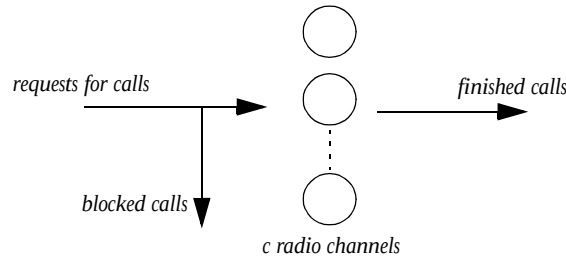


Figure 2.5. A model of the radio channels in a mobile network cell.

Example 2.4: Each router in the packet switched network consists of a number of incoming and outgoing links. Each link also have a buffer, in which packets are stored while waiting for service. A simple model of a router may therefore be a *queuing network* consisting of single server queues, see Figure 2.6. Packets arrive at the incoming links, and are placed in the input buffer. For each packet, the router must decide to which outgoing link the packet should be transferred. This takes some time, which is modelled by a service time in a server. The packet is then placed in the correct output buffer. The transmission of the packet also takes some time, which is modelled by a service time in a server.

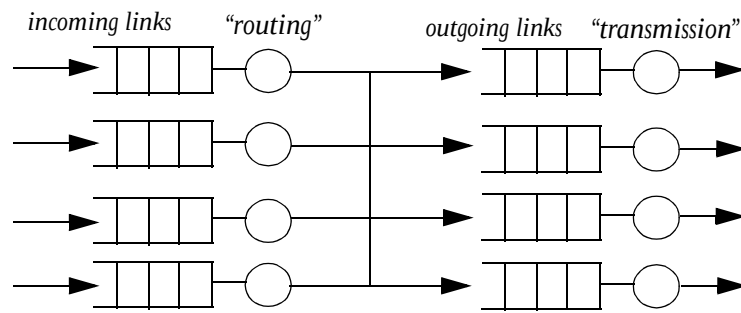


Figure 2.6. *A model of a router*

Chapter 3

Basic queuing theoretic formulas

This chapter contains some basic queuing theoretic formulas that are valid for all types of queuing systems.

3.1 Little's theorem

Some properties of a queuing system are extremely general and depend on very weak assumptions about the behaviour of the various processes. The most important of these results is known as Little's theorem after the author of the first formal proof. It was well-known to early workers in queuing and inventory theory but not proved in any formal sense until 1961.

Assume that we have a system as shown in Figure 3.1. The system can be seen as a black box. The effective arrival rate to the system, that is the rate of jobs that becomes served, is λ_{eff} . Since no jobs are destroyed or created in the system, i.e. the system is *conservative*, the average departure rate must also be λ_{eff} . There are in average $\bar{N}$ jobs in the system and the average total time each job spends in the system is T .

Little's formula relates the average time spent in the system to the effective arrival rate and the average number of jobs in the system by the following formula:

$$\bar{N} = \lambda_{eff} \cdot T \quad (3.1)$$

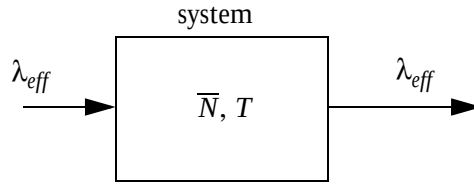


Figure 3.1. A system.

Proof of Little's theorem

Little's theorem can be proved in several ways. The proof given below is based heavily on our intuition about how numbers in a system and times in a system evolve.

Consider the evolution of a system in time. Jobs arrive at intervals and later depart. Notice that no mention need be made of any distribution of time intervals. Little's theorem applies to any particular sample of a system, as well as to the long term averages. Let $A(t)$ denote the number of arrivals to the system before time t . Similarly, let $D(t)$ represent the number of departures from the system before t .

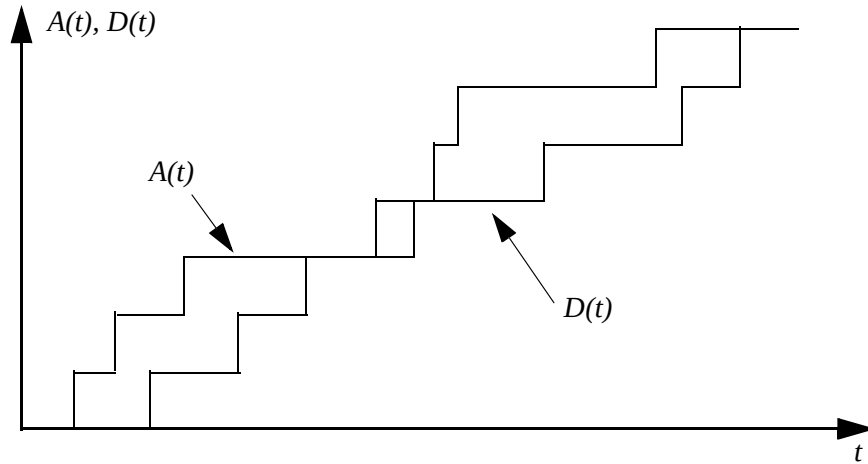


Figure 3.2. Sample of $A(t)$ and $D(t)$.

Figure 3.2 shows an example of the $A(t)$ and $D(t)$ functions for some system. Clearly, the number of jobs in the system at time t , denoted $N(t)$, is given by

$$N(t) = A(t) - D(t) \quad (3.2)$$

The total amount of time that *all* jobs have spent in the system is:

$$T(t) = \int_0^t N(s) ds \quad (3.3)$$

Now the number of arrivals is $A(t)$, so the average time that *one* job spends in the system before t is given by

$$T_t = \frac{T(t)}{A(t)} \quad (3.4)$$

The average number of jobs in the system, before t is

$$\bar{N}_t = \frac{1}{t} \cdot \int_0^t N(s) ds = \frac{T(t)}{t} \quad (3.5)$$

The average effective arrival rate before t is

$$\lambda_t = \frac{A(t)}{t} \quad (3.6)$$

By combining equations 3.3-3.5, it follows that:

$$\bar{N}_t = \lambda_t \cdot T_t \quad (3.7)$$

Now T_t will tend to the average response time for a job, as long as the system empties infinitely often. This happens when the system is stable, that is when the effective arrival rate is lower than the maximum processing capacity of the system. Taking the limits as $t \rightarrow \infty$, Little's result follows:

$$\bar{N} = \lambda_{eff} \cdot T \quad (3.8)$$

3.2 Some basic performance metrics

The basic performance metrics described below are valid for very general queueing systems, as shown in Figure 3.3. The system has a queue (unlimited or limited), and c servers. The average service time for a job is $E[X]$. The average arrival rate is λ and the effective arrival rate is λ_{eff} . If the queue is unlimited, $\lambda = \lambda_{eff}$, since there are no rejected jobs. If the queue is limited, the fraction of jobs that are rejected is P_b .

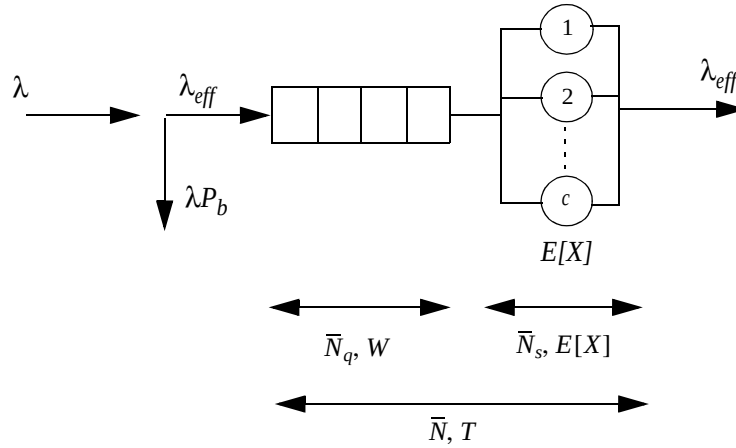


Figure 3.3. A general queueing system.

Average number of jobs

By applying Little's theorem to different parts of the system, we can obtain expressions for the average number of jobs in the system, queue, and servers.

System. The average number of jobs in the system, $\bar{N}$, is as before given by

$$\bar{N} = \lambda_{eff} \cdot T \quad (3.9)$$

where T is the average time a job spend in the system (queue plus servers) before it is completed.

Queue. By applying Little's formula to the queue, it follows that the average number of jobs in the queue, $\bar{N}_q$, is given by

$$\bar{N}_q = \lambda_{eff} \cdot W \quad (3.10)$$

since the effective arrival rate to the queue must be the same as the effective rate to the system, and the average time spent in the queue is W .

Servers. By applying Little's formula to the servers, it follows that the average number of jobs in the servers, $\bar{N}_s$, is given by

$$\bar{N}_s = \lambda_{eff} \cdot E[X] \quad (3.11)$$

since the effective arrival rate to the servers must be the same as the effective rate to the system, and the average time spent in a server is $E[X]$.

Utilisation

The utilisation, U , is defined as the average fraction of servers that are in use. The utilisation is given by

$$U = \frac{\bar{N}_s}{c} \quad (3.12)$$

The utilisation can never be larger than one. Note that when the system has a single server, then the utilisation is also equal to the proportion of time that the server is in use.

Throughput

The throughput of the system is defined as the average departure rate. Since no jobs are destroyed or created in the system, the average departure rate must be equal to the effective arrival rate, λ_{eff} .

Carried load

The carried load, ρ_c , can be seen as the amount of “work” that the system performs. It is given by

$$\rho_c = \lambda_{eff} \cdot E[X] \quad (3.13)$$

that is, by the product of the average service time for a job and the effective arrival rate of jobs. As can be seen, the carried load is equal to the average number of jobs in the servers. The carried load is usually measured in *Erlangs*.

Offered load

The offered load, ρ_o , can be seen as the amount of “work” that arrives at the system. It is given by

$$\rho_o = \lambda \cdot E[X] \quad (3.14)$$

The offered load is also measured in Erlangs.

Call blocking probability

The call blocking probability, or the fraction of calls (jobs) that are blocked, is determined as

$$P_b = \frac{\lambda - \lambda_{eff}}{\lambda} \quad (3.15)$$

3.3 Stability conditions

It is important to discuss the stability conditions for a queuing system. Consider the system shown in Figure 3.3. The system is said to be stable if each job in average spends a limited amount of time in the system, that is if T is finite.

If the queue is limited, the system is always stable. Each job that enters the system spends a limited time there, since there can at maximum be $L-1$ jobs in the system when the job arrives. If there are L jobs in the system when a job arrives, the job is blocked.

If the queue is unlimited, the system is stable if the average arrival rate is less than the maximum processing capacity of the system, that is if $\lambda < c/E[X]$. In a system with unlimited queue there is no blocking, which means that all jobs enters the system. If the arrival rate of new jobs is larger than the maximum processing capacity, the queue will grow. This means that after a long time ($t \rightarrow \infty$) the queue length goes to infinity, which implies that the average time a job must spend in the system also goes to infinity.

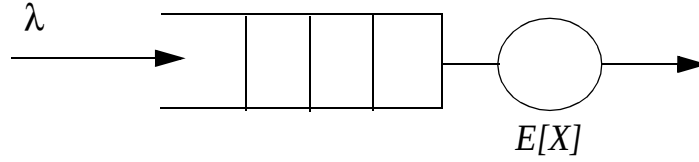


Figure 3.4. A G/G/1-system.

3.4 A general result for single server queues

Consider a single server queue, that is a G/G/1-system, with unlimited queue as shown in Figure 3.4. New jobs arrive with mean rate λ . The average service time is $E[X]$. As defined in the previous section, the carried load for this system is given by

$$\rho_c = \lambda_{eff} \cdot E[X] = \lambda \cdot E[X] \quad (3.16)$$

since no jobs are blocked. Also, the carried load is equal to the average number of jobs in the servers, that is $\bar{N}_s = \lambda E[X]$.

Further, it can be proved that, for a single server queue, **the probability that the server is idle**, denoted p_0 , is given by

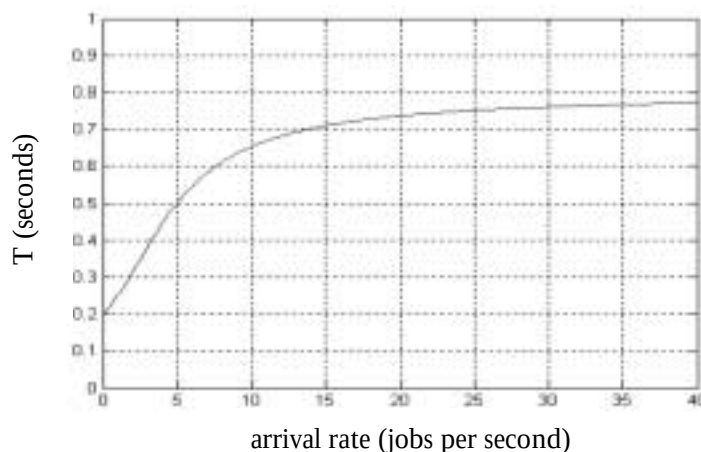
$$p_0 = 1 - \rho_c \quad (3.17)$$

Since the server is either idle or busy, we can also conclude that the average number of jobs in the server is equal to the probability that the server is busy. Note that this result is independent of the service discipline. This means, for example, that it does not matter if the server is using a First-Come-First-Served or a Last-Come-First-Served discipline.

3.5 Exercises

1. Assume that there are four employees at the telephone support in a software company. They work between 8am and 5pm, answering questions from the clients. If all employees are busy when a client calls, the client can wait in an unlimited queue. The company has 1000 clients that each calls in average once every 5th day. In average, how long time can an employee spend on a client's call for the system to be stable?
2. Consider a town with 50.000 inhabitants. Each person in the town makes in average 3 phone calls each day (24 hours). Each call lasts in average 6 minutes. We can assume that no calls are blocked. How many persons are in average talking on the phone at the same time?

-
3. We would like to model a computer system with a queuing model. We measure the traffic to the system and find that the average arrival rate is A jobs per second. Also, we measure how long time it takes for the system to process each each job, and find that the average service time for a job is B seconds.
- Assume that there is one CPU in the computer that can process jobs, and that this CPU can be modelled as a server in the queuing model. For what values of A is the system stable?
 - Instead assume that the computer is a multi-processor system, with m CPU:s that each can process the jobs (the CPU:s are modelled as m servers). For what values of A is the system now stable?
 - Assume that the system can be modelled with an infinite queue, and that the stability condition is fulfilled. What is the throughput of the system?
 - What requirements must the arrival process and the service times fulfill if we want to use an M/M/c-model during the analysis?
 - What requirements must the arrival process and the service times fulfill if we want to use an M/G/c-model?
 - We measure the number of jobs in the system, and find that, in average, there are N jobs at the same time, either beeing served or waiting in queue. Determine the average response time for a job, that is the average time it takes from a job arrives at the system until it leaves the system (assume that the queue is infinite).
4. In this exercise, we will analyse a web site. We assume that new jobs, that is HTTP requests, arrive with average λ requests per second. Further, we assume that the web site may be modelled as a single server with a finite queue with K places. The jobs are served First-Come-First-Served (FCFS). The average service time is $\bar{x}$ seconds. We have measured the average time in the system, T , for varying arrival rates. The figure below shows the average time in the system for a job that is served. Determine, from the figure, the two following metrics:
- The average service time for a job, that is $\bar{x}$.
 - The number of places in the queue, that is K .



-
5. In this exercise we investigate a computer system with one CPU and a job queue. We perform measurements on the system and find that, in average, 10 jobs per second arrive at the system, and that the average service time for a job is 0.09 seconds. The computer processes one job at a time, and the queuing discipline is First-Come-First-Served.

First, assume that the job queue is so long that the blocking probability can be assumed to be zero.

(a) Determine the carried load.

(b) Determine the throughput.

We notice that the average time in the system for a job is too long, and that there might be a problem with overload. Therefore, we decide to make the job queue smaller. Afterwards, we perform new measurements of the system and find that the throughput now is 8 jobs per second. We can assume that the total arrival rate has not changed since the last measurements.

(c) Determine the probability that a job is blocked.

(d) Determine the carried load.

(e) Say something about the user-perceived QoS in the two systems above.

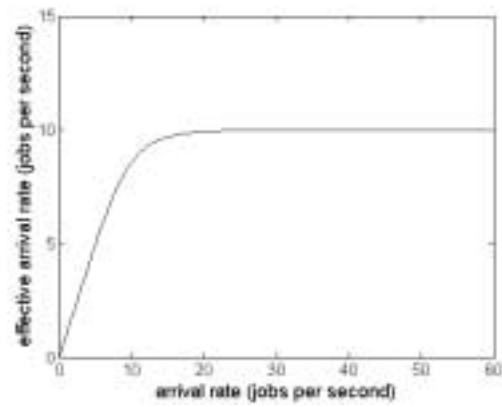
6. Measurements have been performed on a node in a telecommunication network. The results are shown in the two diagrams below. Diagram (a) shows the effective arrival rate, λ_{eff} , when the offered arrival rate, λ , is varied between 0 and 60 jobs per second. Diagram (b) shows the average number of jobs in the system, $\bar{N}$, for the same offered arrival rates. We can assume that the node can be modelled as a single server with a finite queue. Use the diagrams to determine the following performance metrics:

(a) The average service time for a job.

(b) The size of the queue.

(c) The average response time, T , for a job when $\lambda = 40$.

(a)



(b)

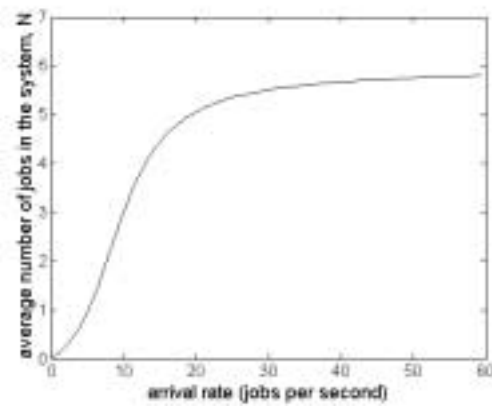


Figure 3.5. *Diagrams for Exercise 6.*

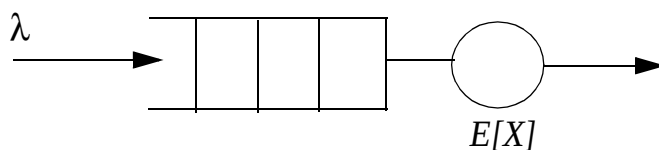
Chapter 4

M/M/c-systems

In this chapter, M/M/c-systems are analysed. In these systems both the interarrival times and the service times are exponentially distributed. All M/M/c-systems can be described with a Markov chain in which the states represent the number of jobs in the system. The number of customers present in the system, both waiting and being served, gives all the information needed to calculate the probabilities of the number of customers at any time in the future.

4.1 The M/M/1-system with unlimited queue

The M/M/1-system with unlimited queue, shown in Figure 4.1 is one of the easiest systems to analyse. New jobs arrive according to a Poisson process with mean rate λ . There is one server that process the jobs. The service times are exponential distributed with mean $E[X]$. The queue is unlimited, which means that the blocking probability for a job is zero.



Figur 4.1. An M/M/1-system

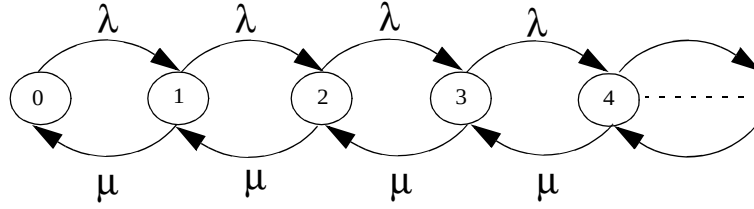


Figure 4.2. Steady state diagram for the M/M/1-system.

Markov chain

An M/M/1-system is Markovian, which means that all events in the system happen with exponentially distributed time intervals. The interarrival times for new jobs are exponentially distributed with mean $1/\lambda$. The service times are exponentially distributed with mean $E[X]$, usually denoted $1/\mu$.

This means that the stochastic behavior of the M/M/1-system can be described as a continuous-time Markov chain in which the states represent the number of jobs in the system. Since the queue is infinite, the Markov chain is also infinite. Whatever the state of the system is, a new job will arrive in the next short interval Δt with the probability $\lambda \Delta t$. If the system is not empty, the job in the queue will finish service in the next interval of length Δt with probability $\mu \Delta t$. Knowledge of the time since the last arrival or the time since the service began does not change these probabilities. An arrival means that the state of the system is increased with one. A departure means that the system state is decreased with one. The probability of having both an arrival and a departure in the same short interval Δt is assumed to be zero (or at least $o(\Delta t)$). Therefore, the system can be described by the Markov chain in Figure 4.2.

Time-dependent state probabilities

The state of the system at any time is given by the number of customers present. Denote by $p_k(t)$, the probability that there are k customers present at time t . Using knowledge of the properties of the Poisson process, the evolution of these probabilities at time $t + \Delta t$ can be studied.

Considering states with a job currently being served ($n > 0$), there are four possible occurrences in the next short interval, Δt :

- (d) There are no arrivals, and the customer being served continues service.
- (e) There is an arrival, and the customer being served continues service.
- (f) There are no arrivals, and the customer being served finishes its service.
- (g) There is an arrival, and the customer being served finishes its service.

The probability of there being more than one arrival in t is $o(\Delta t)$. Similarly, the probability that there is an arrival and the customer currently being served finishes its service is $\lambda \Delta t \cdot \mu \Delta t$, which is $o(\Delta t)$. Thus if the system is in state n at time $t + \Delta t$, then the system must have been in state $n-1$, n or $n+1$, at time t .

This means that $p_k(t+\Delta t)$ is given by:

$$\begin{aligned} p_k(t+\Delta t) = & p_k(t)[(1-\lambda\Delta t)(1-\mu\Delta t)] + \\ & p_{k+1}(t)(1-\lambda\Delta t)\mu\Delta t + p_{k-1}(t)(1-\mu\Delta t)\lambda\Delta t + o(\Delta t) \end{aligned} \quad (4.1)$$

Hence:

$$p_k(t+\Delta t) - p_k(t) = (-\lambda - \mu)\Delta t p_k(t) + \mu\Delta t p_{k+1}(t) + \lambda\Delta t p_{k-1}(t) + o(\Delta t) \quad (4.2)$$

and dividing by Δt and taking the limit as $\Delta t \rightarrow 0$, the derivative of $p_k(t)$, $p_k'(t)$, is given by:

$$p_k'(t) = (-\lambda - \mu)p_k(t) + \mu p_{k+1}(t) + \lambda p_{k-1}(t) \quad (4.3)$$

Similarly,

$$p_0'(t) = -\lambda p_0(t) + \mu p_1(t) \quad (4.4)$$

This system of differential equations can be solved, and the evolution of the system through time can be studied from any particular starting state. However this approach will not be followed because it is known from Markov process theory that in the long run the initial state does not affect the steady state probabilities. These *steady state* or *equilibrium* probabilities give the long term behaviour of the system, and this is the object of most analyses.

Steady-state probability distribution

Since by definition of equilibrium, the probabilities are not changing with respect to time, $p_k'(t)$ can be equated to 0. Let

$$p_k = \lim_{t \rightarrow \infty} p_k(t)$$

which means that p_k is the long term probability for k jobs in the system. This means that equations 4.3 and 4.4 can be written as

$$\begin{aligned} \lambda p_0 &= \mu p_1 \\ (\lambda + \mu)p_1 &= \lambda p_0 + \mu p_2 \\ (\lambda + \mu)p_2 &= \lambda p_1 + \mu p_3 \\ &\vdots \end{aligned} \quad (4.5)$$

These are the global balance equations for this system. By successive elimination, they can be rewritten as:

$$\begin{aligned}
\lambda p_0 &= \mu p_1 \\
\lambda p_1 &= \mu p_2 \\
\lambda p_2 &= \mu p_3 \\
&\vdots
\end{aligned} \tag{4.6}$$

Every probability can then be expressed in terms of p_0 :

$$\begin{aligned}
p_1 &= \frac{\lambda}{\mu} \cdot p_0 \\
p_2 &= \frac{\lambda}{\mu} \cdot p_1 = \left(\frac{\lambda}{\mu}\right)^2 \cdot p_0 \\
&\vdots
\end{aligned} \tag{4.7}$$

which means that the steady state probability for k jobs in the system is given by

$$p_k = \frac{\lambda}{\mu} \cdot p_{k-1} = \dots = \left(\frac{\lambda}{\mu}\right)^k \cdot p_0 \quad k \geq 0 \tag{4.8}$$

Since the system must be in one of the states, the axiom of probability gives that

$$\sum_{k=0}^{\infty} p_k = 1$$

This is usually referred to as the normalisation condition. Hence, using equation 4.8

$$\sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k \cdot p_0 = 1$$

Taking out the common factor p_0 on the left hand side, we get

$$p_0 \sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k = 1$$

which can be summed as a geometric series, provided that $|\lambda/\mu| < 1$. This gives

$$p_0 \cdot \frac{1}{1 - \lambda/\mu} = 1 \Rightarrow p_0 = 1 - \lambda/\mu$$

This means that the probability of the server being empty (the idle state) is $1 - \lambda/\mu$. Notice that the condition for the stability of the system, $\lambda/\mu < 1$, (and the existence of a steady state distribution) is used when solving the sum above.

Steady state diagram

In Markovian systems the global balance equations can easily be derived from the steady state diagram, shown in Figure 4.2.

When the system reaches equilibrium the net “flow” across any closed curve, or cut, must total zero. This “flow” of probability can be measured as the product of the label on the arcs and the probabilities of their starting states. Using cuts on the balance diagram is a powerful graphical technique for deriving the balance equations. In the M/M/1-system, drawing a cut surrounding each state individually leads to the first set of balance equations (equation 4.5). However if we use a cut which surrounds all the states, $0, 1, 2, \dots, k$, we derive the second form of the balance equations (equation 4.6). Examination of the steady state diagram often leads to equivalent formulations of the balance equations which are simpler to solve.

Some results

By using the probability distribution for the number of jobs in the system, the performance metrics described in Chapter 3 can be calculated. The results below are only valid for stable systems, that is, systems for which $\lambda/\mu < 1$.

The **utilisation** is the fraction of time that the server is busy, which is the same as the probability that the server is busy. This means that the utilisation can be calculated as

$$\rho = \sum_{k=1}^{\infty} p_k = 1 - p_0 = \frac{\lambda}{\mu} \quad (4.9)$$

Therefore, we will in the following equations use ρ instead of λ/μ .

The **throughput** is directly found as the arrival rate, λ , since there is no blocking in the system. Therefore, all jobs that arrives at the system are served and thereby departs sometime later.

The **average number of jobs in the system** can be calculated as follows:

$$\begin{aligned} \bar{N} &= \sum_{k=0}^{\infty} k \cdot p_k = \sum_{k=1}^{\infty} k \cdot \rho^k (1 - \rho) = \rho(1 - \rho) \sum_{k=1}^{\infty} k \cdot \rho^{k-1} \\ &= \rho(1 - \rho) \cdot \frac{d}{d\rho} \left(\sum_{k=0}^{\infty} \rho^k \right) = \rho(1 - \rho) \cdot \frac{1}{(1 - \rho)^2} = \frac{\rho}{1 - \rho} \end{aligned} \quad (4.10)$$

The **average number of jobs in the server** can be calculated in at least two ways. It can be calculated as $\bar{N}$ above:

$$\bar{N}_s = 0 \cdot p_0 + 1 \cdot \sum_{k=1}^{\infty} p_k = 1 - p_0 = \rho \quad (4.11)$$

However, it may also be calculated by using the Little's theorem on the server:

$$\bar{N}_s = \lambda_{eff} \cdot E[X] = \lambda \cdot \frac{1}{\mu} = \rho \quad (4.12)$$

Since the queue is unlimited, the **offered load** is equal to the **carried load**, which is the same as the average number of jobs in the server. Therefore, we get

$$\rho_o = \rho_c = \rho = \lambda/\mu \quad (4.13)$$

The **average number of jobs in the queue** is easily calculated as

$$\bar{N}_q = \bar{N} - \bar{N}_s = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho} \quad (4.14)$$

or in the same way as $\bar{N}$:

$$\begin{aligned} \bar{N}_q &= \sum_{k=1}^{\infty} k \cdot p_{k+1} = \sum_{k=1}^{\infty} k \cdot \rho^{k+1}(1-\rho) \\ &= \rho^2(1-\rho) \sum_{k=1}^{\infty} k \cdot \rho^{k-1} = \dots = \frac{\rho^2}{1-\rho} \end{aligned} \quad (4.15)$$

By using Little's theorem, we immediately find the **mean time a job spends in the system**

$$T = \frac{\bar{N}}{\lambda_{eff}} = \frac{\bar{N}}{\lambda} = \frac{E[X]}{1-\rho} \quad (4.16)$$

and the **mean time in the queue**

$$W = \frac{\bar{N}_q}{\lambda_{eff}} = \frac{\bar{N}_q}{\lambda} = \frac{\rho E[X]}{1-\rho} \quad (4.17)$$

It is also worth pointing out that there is a conflict of interest. Channel or server idle times is often seen by managers as an available resource, and policies are modified to reduce the idle time. Users of a service, on the other hand, see delay as the major measure of effectiveness. The utilisation of the server is given by ρ . As this utilisation increases, and approaches one, giving less and less idle time, the mean queue length increases with correspondingly longer mean waiting times. In the limit, maximum throughput can be achieved by allowing the rate of arrivals to approach the service rate, but only at the expense of allowing the mean waiting time to become unbounded.

Example 4.4: Measurements on a web site has shown that the average arrival rate is 10 requests per second, and that the average service time for a request is 0.05 seconds. We decide to model the site as an M/M/1-system with infinite queue. This means that the state diagram for the system is shown in Figure 4.3.

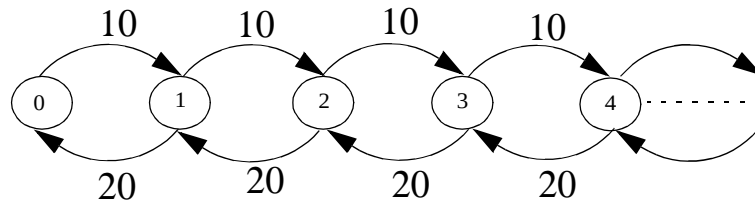


Figure 4.3. State diagram for the system in example 4.4.

An average service time of 0.05 seconds gives a average service rate of 20 requests per second. From the steady-state diagram we can determine the *global balance equations*:

$$10p_{k-1} = 20p_k \quad k \geq 1$$

The state probabilities can then be expressed in p_0 :

$$p_k = \frac{1}{2}p_{k-1} = \frac{1}{4}p_{k-2} = \dots = \left(\frac{1}{2}\right)^k p_0 \quad k \geq 0$$

p_0 is found by using the normalisation condition:

$$\sum_{k=0}^{\infty} p_k = 1 \Rightarrow \sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^k p_0 = 1 \Rightarrow p_0 \sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^k = 1 \Rightarrow p_0 = \frac{1}{1 - 1/2} = \frac{1}{2}$$

This means that the *stationary probability distribution* is given by

$$p_k = \left(\frac{1}{2}\right)^k p_0 = \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right) = \left(\frac{1}{2}\right)^{k+1} \quad k \geq 0$$

The *average number of jobs in the system* is given by

$$\bar{N} = \sum_{k=0}^{\infty} k p_k = \sum_{k=0}^{\infty} k \left(\frac{1}{2}\right)^{k+1} = \left(\frac{1}{2}\right)^2 \sum_{k=1}^{\infty} k \left(\frac{1}{2}\right)^{k-1} = \left(\frac{1}{2}\right)^2 \cdot \frac{1}{(1 - 1/2)^2} = 1$$

and the average response time for a request is found by using Little's theorem

$$T = \frac{\bar{N}}{\lambda_{eff}} = \frac{\bar{N}}{\lambda} = \frac{1}{10}$$

since the queue is infinite (no blocking).

4.2 Limited waiting room

The next example is the $M/M/1^*L$ or the $M/M/1/L$ -system, that is, the system with Poisson arrivals, exponential service times, a single server, and only enough room for L jobs to be in the system at the same time. This limited waiting room might represent a set of buffers allocated to one output channel in a packet switching computer providing a datagram service; an arriving packet which finds all buffers in use is simply discarded. The designer wishes to choose the number of buffers that are available to that circuit to provide an acceptable probability that packets will be discarded. The throughput of the system, that is the rate at which messages can be transmitted by the system, is a measure of effectiveness of the buffers. More buffers will mean fewer jobs being rejected, and should increase the throughput. In contrast, long buffers means long waiting times for those customers that are admitted so there is an argument to reducing the value of L . The cost of too few buffers can be measured by calculating the rate at which messages are refused entry to the system.

Steady state probabilities

In order to evaluate the factors discussed above, the steady state distribution of the number of jobs in the system is needed. Since processes involved are memoryless, the

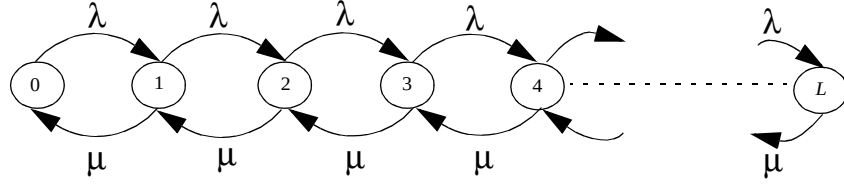


Figure 4.4. Steady state diagram for the M/M/1/L-system.

state of the system is given by the number of customers. Figure 4.4 shows the steady state diagram for the M/M/1/L -system. The state, k , can take any value between 0 and L . Transitions between states are the same as in the M/M/1-system with unlimited queue except that when the system is in state L , there are no arrivals that affect the state of the system.

The balance equations are:

$$\begin{aligned}
 \mu p_1 &= \lambda p_0 \\
 \mu p_2 &= \lambda p_1 \\
 &\vdots \\
 \mu p_L &= \lambda p_{L-1}
 \end{aligned} \tag{4.18}$$

which are easily solved to give

$$p_k = \left(\frac{\lambda}{\mu}\right)^k \cdot p_0 \quad k = 0, \dots, L \tag{4.19}$$

But now p_0 cannot be equated to $1 - \lambda/\mu$, because there is a limited number of states. From the normalisation condition

$$\sum_{k=0}^L p_k = 1$$

we can establish that

$$p_0 = \frac{1 - \lambda/\mu}{1 - (\lambda/\mu)^{L+1}} \tag{4.20}$$

Notice that because this is a finite system, no restrictions on the values λ and μ are necessary to ensure that equilibrium is reached. If λ is greater than μ , that is, jobs arrive faster than the system can serve them, a steady state is still reached, but state L has a higher probability than state 0, i.e., the system is more likely to be full than empty.

Finally, the stationary probability distribution for the M/M/1/L-system can be derived as

$$p_k = \left(\frac{\lambda}{\mu}\right)^k \cdot p_0 = \left(\frac{\lambda}{\mu}\right)^k \cdot \frac{1 - \lambda/\mu}{1 - (\lambda/\mu)^{L+1}} \tag{4.21}$$

Some results

The **effective arrival rate** to the system, λ_{eff} is calculated as the average effective arrival rate of all states:

$$\lambda_{eff} = \sum_{k=0}^L \lambda_{eff,k} \cdot p_k = \sum_{k=0}^{L-1} \lambda p_k + 0 \cdot p_L = \lambda \cdot (1 - p_L) \quad (4.22)$$

However, the effective arrival rate can also be seen as the rate at which jobs *leave* the system, that is the **throughput**, which means that

$$\lambda_{eff} = \sum_{k=0}^L \mu_k p_k = \left[\begin{array}{l} \mu_0 = 0 \\ \mu_{k>0} = \mu \end{array} \right] = 0 \cdot p_0 + \sum_{k=1}^L \mu p_k = \mu \cdot (1 - p_0) \quad (4.23)$$

In a stable system the rate of accepted arrivals, λ_{eff} will equal the rate of jobs leaving the system. Here it is easy to calculate both these quantities, and confirm the equality, but often one or other will be more difficult to find.

The **average rate of blocked jobs**, λ_b , is in the same way given by

$$\lambda_b = \sum_{k=0}^{L-1} 0 \cdot p_k + \lambda p_L = \lambda p_L \quad (4.24)$$

The **fraction of jobs that are blocked**, P_b , is calculated as the division between the average rate of blocked calls and the average arrival rate:

$$P_b = \frac{\lambda_b}{\lambda} = \frac{\lambda p_L}{\lambda} = p_L \quad (4.25)$$

As can be seen, the fraction of blocked jobs is in this case equal to the probability that the system is in state L . This will be the case for systems with an Poisson arrival process. However, for *all* other systems $P_b \neq p_L$!

Example 4.5: Assume that a computer system can be modelled as a M/M/1-system with a limited queue with 4 places. Measurements show that the average arrival rate to the system is 2 jobs per second, and that the average service time for a job is 1 second. The state diagram for this system is shown in Figure 4.5.

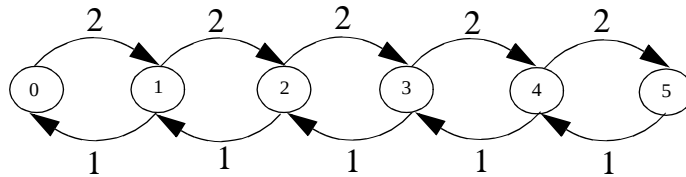


Figure 4.5. State diagram for the system in example 4.5.

The *global balance equations* are given by:

$$2p_{k-1} = p_k \quad 1 \leq k \leq 5$$

The state probabilities can be expressed in p_0 :

$$p_1 = 2p_0 \quad p_2 = 4p_0 \quad p_3 = 8p_0 \quad p_4 = 16p_0 \quad p_5 = 32p_0$$

p_0 is found by using the normalisation condition:

$$\sum_{k=0}^5 p_k = 1 \Rightarrow p_0(1 + 2 + 4 + 8 + 16 + 32) = 1 \Rightarrow p_0 = \frac{1}{63}$$

This means that the *stationary probability distribution* is given by:

$$p_k = \left[\frac{1}{63}, \frac{2}{63}, \frac{4}{63}, \frac{8}{63}, \frac{16}{63}, \frac{32}{63} \right]$$

The average number of jobs in the system can be calculated as

$$\bar{N} = \sum_{k=1}^5 k p_k = p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 = \frac{258}{63} \approx 4.1$$

The *fraction of blocked jobs* is given by

$$P_b = \frac{\lambda_b}{\lambda} = \frac{\lambda p_5}{\lambda} = p_5 = \frac{32}{63} \approx 0.51$$

which means that the effective arrival rate is

$$\lambda_{eff} = \lambda(1 - P_b) = 0.98$$

The *carried load* can be calculated as the average number of jobs in the server:

$$\rho_c = \bar{N} = 0 \cdot p_0 + 1 \cdot (p_1 + p_2 + p_3 + p_4 + p_5) = 1 - p_0 = \frac{62}{63} \approx 0.98$$

or by using the definition:

$$\rho_c = \lambda_{eff} \cdot E[X] = 0.98 \cdot 1 = 0.98$$

4.3 Finite client population

The systems that we analysed in Sections 4.1 and 4.2, both have a Poisson arrival process. In this arrival process it is assumed that the probability of an arrival in the next time interval Δt is independent of the state of the system. This means that the average arrival rate always is the same irrespective of the number of jobs in the system. However, in many systems the Poisson arrival process is not valid. Instead we need a model in which there is a finite number of clients that can send jobs to the queuing system.

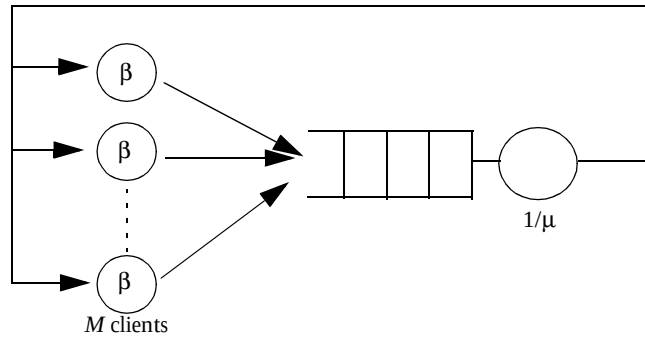
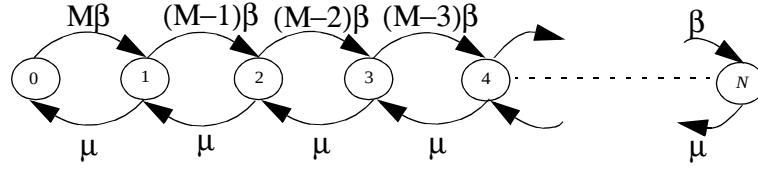


Figure 4.6. An $M/M/1$ -system with a finite number of clients.



Figur 4.7. Steady state diagram for the M/M/1-system with finite client population.

Figure 4.6 shows a queuing system that can be used to model, for example, a web-server system. In this model there are M clients sending jobs to a central computer (the web server). The client thinks for a period which is exponentially distributed with mean $1/\beta$, and then submits a job to the central computer which has an exponential distributed service requirement, with mean $1/\mu$. Jobs at the central computer are served in FCFS (First-come-first-served) order. When the service requirement has been fulfilled a reply is sent back to the client, who enters the think state again.

Steady state probabilities

Since all the processes involved in the system are memoryless (exponential), the system can be described with a Markov chain, see Figure 4.7. The state of the system is given by the number of jobs in the central computer. If there are M clients, then the states can be $0, 1, 2, \dots, M$. When there are k jobs in the central computer, there are $M - k$ clients in think state, so that new jobs join the queue at rate $(M - k)\beta$. Jobs leave the central computer at rate μ .

The balance equations are easily written down:

$$\begin{aligned}
 M\beta p_0 &= \mu p_1 \\
 (M-1)\beta p_1 &= \mu p_2 \\
 (M-2)\beta p_2 &= \mu p_3 \\
 &\vdots \\
 \beta p_{M-1} &= \mu p_M
 \end{aligned} \tag{4.26}$$

It is straight forward to solve these equations in terms of β , μ , and p_0 , to give:

$$p_k = \frac{M!}{(M-k)!} \cdot \left(\frac{\beta}{\mu}\right)^k p_0 \quad k = 0, \dots, M \tag{4.27}$$

p_0 can now be found using the normalisation condition:

$$p_0 = \left[\sum_{k=0}^M \frac{M!}{(M-k)!} \cdot \left(\frac{\beta}{\mu}\right)^k \right]^{-1}$$

Some results

The **average arrival rate** to the system is calculated as an average of all states:

$$\lambda = \sum_{k=0}^M \lambda_k p_k = [\lambda_k = (M-k)\beta] = \sum_{k=0}^{M-1} (M-k)\beta p_k \quad (4.28)$$

Since the average arrival rate is equal to the **average departure rate** (no blocked jobs), λ can also be calculated as

$$\lambda = \sum_{k=0}^M \mu_k p_k = \left[\begin{array}{l} \mu_0 = 0 \\ \mu_{k>0} = \mu \end{array} \right] = \sum_{k=1}^M \mu p_k = \mu(1-p_0) \quad (4.29)$$

The **average number of jobs in the central computer** can be calculated as

$$\bar{N} = \sum_{k=0}^M k p_k = \sum_{k=1}^M k \cdot \frac{M!}{(M-k)!} \cdot \left(\frac{\beta}{\mu}\right)^k p_0$$

After some calculations we get:

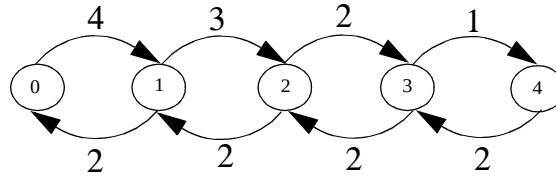
$$\bar{N} = M - \frac{\mu}{\beta} \cdot (1-p_0) \quad (4.30)$$

The **average response time** for a job can now be calculated with Little's theorem:

$$T = \frac{\bar{N}}{\lambda_{eff}} = \frac{\bar{N}}{\lambda} = \frac{M}{\mu(1-p_0)} - \frac{1}{\beta} \quad (4.31)$$

This model was also studied very early in the history of queuing theory. It is usually given the name Machine repairman problem. The clients represent machines which are operational for exponential distributed intervals, independently of one another, and then fail, and have to be repaired by a single repairman. The repair process is also exponential distributed.

Example 4.6: A central server in a company can be modelled as an M/M/1-system with 4 clients that submits jobs to the server. The think time for an idle client is 1 second, and the average service time for a job is 0.5 seconds. The state diagram for this system is shown in Figure 4.8.



Figur 4.8. State diagram for the system in Example 4.6.

The global balance equations are given by

$$\begin{cases} 4p_0 = 2p_1 \\ 3p_1 = 2p_2 \\ 2p_2 = 2p_3 \\ p_3 = 2p_4 \end{cases}$$

The state probabilities can be expressed in p_0 , and then p_0 is found by using the normalisation equation:

$$\sum_{k=0}^4 p_k = 1 \Rightarrow p_0 + 2p_0 + 3p_0 + 3p_0 + \frac{3}{2}p_0 = 1 \Rightarrow p_0 = \frac{2}{21}$$

This means that the stationary probability distribution is given by

$$p_k = \left[\frac{2}{21}, \frac{4}{21}, \frac{6}{21}, \frac{6}{21}, \frac{3}{21} \right]$$

The *average arrival rate* of jobs to the server can be determined as

$$\lambda = \sum_{k=0}^4 \lambda_k p_k = 4p_0 + 3p_1 + 2p_2 + 1p_3 + 0p_4 = \frac{38}{21} \approx 1.8$$

or by calculating the *average departure rate*

$$\lambda = \sum_{k=0}^4 \mu_k p_k = 0p_0 + 2p_1 + 2p_2 + 2p_3 + 2p_4 = \frac{38}{21} \approx 1.8$$

The *carried load* can be found by using the definition

$$\rho_c = \lambda_{eff} \cdot E[X] = \lambda \cdot E[X] = 0.9$$

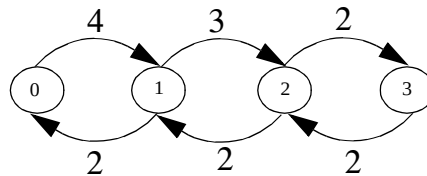
or by using the fact that the carried load for a single server system is the same as the *utilisation*

$$\rho_c = U = 1 - p_0 = \frac{19}{21} \approx 0.9$$

or by calculating the *average number of jobs in the server*

$$\rho_c = \bar{N}_s = 0p_0 + 1 \cdot (p_1 + p_2 + p_3) = 1 - p_0 \approx 0.9$$

Example 4.7: If we for the system in Example 4.6 assume that the central computer has a limited queue with 2 places, the state diagram can be drawn as in Figure 4.9.



Figur 4.9. State diagram for the system in Example 4.7.

In this case, the *stationary probability distribution* is given by

$$p_k = \left[\frac{1}{9}, \frac{2}{9}, \frac{3}{9}, \frac{3}{9} \right]$$

In this system, clients can be blocked. The *fraction of blocked jobs* is given by

$$P_b = \frac{\lambda_b}{\lambda} = \frac{1p_3}{4p_0 + 3p_1 + 2p_2 + 1p_3} = \frac{3}{19} \approx 0.16$$

As can be seen, $P_b \neq p_3 = 0,33$, which means that the probability that the system is full is larger than the probability that a job is blocked. The carried load is given by

$$\rho_c = \lambda_{eff} \cdot E[X] = (4p_0 + 3p_1 + 2p_2) \cdot E[X] = \frac{8}{9} \approx 0.89$$

As can be seen, the carried load for the limited system is almost as high as the carried load for the unlimited system.

4.4 Multiple servers

Figure 4.10 shows a system where new jobs arrive according to a Poisson process with mean rate λ , the service times are exponential distributed with mean $E[X]$, and there are c servers ($c \geq 1$) in parallel. The servers work independent of each other. If more than one server is available when a job arrives, the job randomly chooses one of the servers. The jobs join a common unlimited queue. The jobs in the queue are served FCFS.

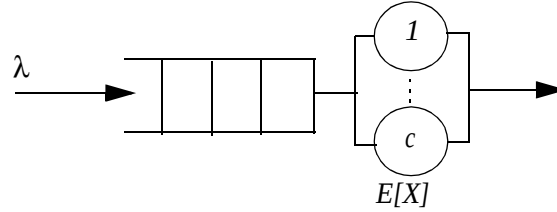


Figure 4.10. An M/M/c-system with unlimited queue.

State diagram

The M/M/c-system can be described with a Markov chain, where state k corresponds to k jobs in the system. The steady state diagram for the system is shown in Figure 4.11. As before, a new job will arrive in the next short interval Δt with the probability $\lambda \Delta t$. Also, if only one server is busy the probability that the job in this server will depart in an interval Δt is $\mu \Delta t$. However, it is necessary to calculate the probability that a job departs in the next interval Δt when *more* than one server is busy. Therefore, assume that there are k jobs, in the servers. The probability that one of k independent services will end in the next interval Δt is:

$$\sum_{i=1}^k (\mu \Delta t + o(\Delta t)) \cdot (1 - \mu \Delta t + o(\Delta t))^{k-1} = [\Delta t \cdot \Delta t = o(\Delta t)] = k\mu \Delta t + o(\Delta t)$$

This means that the time to the next departure is exponential distributed with mean $1/k\mu$ if there are k busy servers.

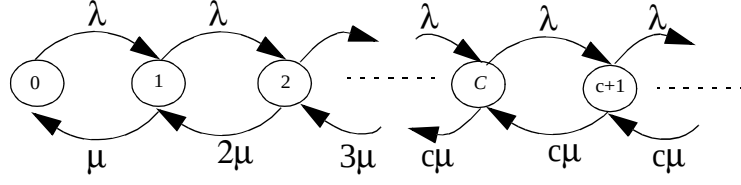


Figure 4.11. The steady state diagram for the M/M/c-system with unlimited queue.

Steady state probabilities

From the state diagram it can be seen that the balance equations are

$$\begin{aligned} \lambda p_{k-1} &= k\mu p_k & 1 \leq k \leq c \\ \lambda p_{k-1} &= c\mu p_k & k > c \end{aligned} \quad (4.32)$$

The solution of the balance equations then leads to

$$p_k = \begin{cases} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} p_0 & k \leq c \\ \left(\frac{\lambda}{\mu}\right)^k \left(\frac{1}{c}\right)^{k-c} \frac{1}{c!} p_0 & k > c \end{cases} \quad (4.33)$$

From the normalisation condition, p_0 can be calculated:

$$p_0 = \frac{1}{\sum_{k=0}^{c-1} \frac{(\lambda/\mu)^k}{k!} + \frac{(\lambda/\mu)^c}{c!} \cdot \frac{c}{c - \lambda/\mu}} \quad (4.34)$$

Some results

The **average number of jobs in the system** is given by

$$\bar{N} = \sum_{k=1}^{\infty} k p_k = \sum_{k=1}^c k p_k + \sum_{k=c+1}^{\infty} k p_k = p_0 \sum_{k=1}^c k \cdot \frac{(\lambda/\mu)^k}{k!} + p_c \sum_{k=c+1}^{\infty} k \cdot \left(\frac{\lambda}{c\mu}\right)^{k-c}$$

and after some calculation

$$\bar{N} = \frac{\lambda}{\mu} + p_c \cdot \frac{(\lambda/c\mu)}{(1 - \lambda/c\mu)^2} \quad (4.35)$$

Since there is no blocking the **effective arrival rate** is equal to λ . By using Little's theorem, the **average number of jobs in the servers** can be calculated as

$$\bar{N}_s = \lambda E[X] = \frac{\lambda}{\mu} \quad (4.36)$$

Since $\bar{N} = \bar{N}_s + \bar{N}_q$, the **average number of jobs in the queue** can be identified as

$$N_q = p_c \cdot \frac{(\lambda/c\mu)}{(1-\lambda/c\mu)^2} \quad (4.37)$$

Since the queue is unlimited, the **offered load** is equal to the carried load. Further, the **carried load** is always equal to the average number of jobs in the servers, which means that:

$$\rho_o = \rho_c = \frac{\lambda}{\mu} \quad (4.38)$$

The **utilisation** of a server is given by

$$U = \frac{\bar{N}_s}{c} = \frac{\lambda}{c\mu} \quad (4.39)$$

Since the utilisation can never be above 100%, the **stability condition** becomes $\lambda < c\mu$.

Example 4.8: We model a computer with two CPU:s as an M/M/2-system with unlimited queue. Measurements show that the average arrival rate is 10 jobs per second, and that the average service time for a job is 0.125 seconds. The state diagram for this system is shown in Figure 4.12.

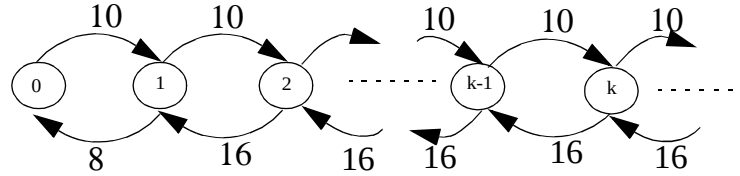


Figure 4.12. State diagram for the system in Example 4.8.

The *global balance equations* are given by

$$\begin{cases} 10p_0 = 8p_1 \\ 10p_{k-1} = 16p_k \end{cases} \quad k \geq 2$$

The state probabilities can be expressed in p_0 :

$$\begin{cases} p_1 = \frac{5}{4}p_0 \\ p_k = \left(\frac{5}{8}\right)^{k-1} \cdot \frac{5}{4}p_0 \end{cases} \quad k \geq 2$$

p_0 is found by using the normalisation condition

$$\sum_{k=0}^{\infty} p_k = 1 \Rightarrow p_0 + \frac{5}{4}p_0 + \sum_{k=2}^{\infty} \left(\frac{5}{8}\right)^{k-1} \cdot \frac{5}{4}p_0 = 1 \Rightarrow p_0 \left(\frac{9}{4} + \frac{25}{32} \cdot \sum_{k=0}^{\infty} \left(\frac{5}{8}\right)^k \right) = 1$$

which means that

$$p_0 = \frac{1}{\frac{9}{4} + \frac{25}{32} \cdot \frac{1}{1-5/8}} = \frac{3}{13}$$

Therefore, the *stationary probability distribution* is given by

$$\begin{cases} p_0 = \frac{3}{13} \\ p_k = \left(\frac{5}{8}\right)^{k-1} \cdot \frac{15}{52} & k \geq 1 \end{cases}$$

The *average number of jobs in the system* can be calculated as

$$\bar{N} = \sum_{k=1}^{\infty} k p_k = \frac{15}{52} \sum_{k=1}^{\infty} k \left(\frac{5}{8}\right)^{k-1} = \frac{15}{52} \cdot \frac{1}{(1-5/8)^2} = \frac{80}{39} \approx 2.1$$

or by using the formula above

$$\bar{N} = \frac{\lambda}{\mu} + p_c \cdot \frac{(\lambda/c\mu)}{(1-\lambda/c\mu)^2} = \frac{10}{8} + \frac{5}{8} \cdot \frac{15}{52} \cdot \frac{10/16}{(1-10/16)^2} = \frac{80}{39} \approx 2.1$$

The *carried load* is given by the definition

$$\rho_c = \lambda_{eff} \cdot E[X] = 10 \cdot 0.125 = 1.25$$

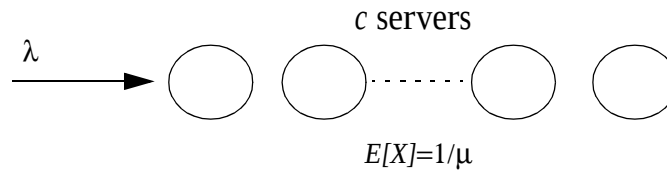
since no jobs are blocked. It can, of course, also be calculated as

$$\rho_c = \bar{N}_s = 1 \cdot p_1 + 2 \cdot \sum_{k=2}^{\infty} p_k = p_1 + 2 \cdot (1 - p_0 - p_1) = \frac{65}{52} = 1.25$$

4.5 The M/M/c-system without queue (Erlang's model)

In this system there is no facility in the system to wait and form a queue. If an arriving job finds all servers busy, it is blocked. New jobs arrive according to a Poisson process with mean rate λ . The service times are exponential distributed with mean $1/\mu$.

The M/M/c-system without queue is a so-called loss system. Erlang had originally used this loss model to investigate the distribution of busy channels in telephone systems. Such a system can handle up to c incoming calls at once. An incoming call enters the system if at least one server is free, and it is blocked if all servers are busy. The system is shown in figure 4.13.



Figur 4.13. An M/M/c-system without queue.

Steady state probabilities

The state diagram for this system is shown in figure 4.14. The system has c states, and as before the states represent the number of jobs in the system.

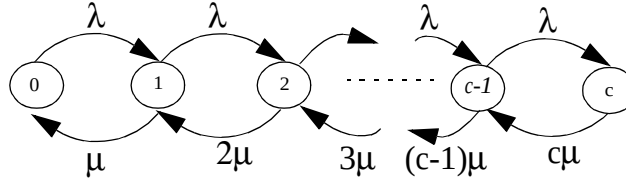


Figure 4.14. State diagram for the M/M/c-system without queue.

By using that the flow across any closed cut must total zero we find the balance equations

$$\lambda p_{k-1} = k\mu p_k \quad 1 \leq k \leq c$$

which leads to the solution

$$p_k = \frac{(\lambda/\mu)^k}{k!} p_0 \quad 0 \leq k \leq c \quad (4.40)$$

The normalisation condition gives

$$p_0 = \frac{1}{\sum_{i=0}^c \frac{(\lambda/\mu)^i}{i!}} \quad (4.41)$$

This probability distribution is defined for all values of λ and μ .

Erlang's loss formula, $E_m(\rho)$

One of the main performance metrics for a loss system is the **probability that a job is blocked**, which is given by

$$P_b = \frac{\lambda_b}{\lambda} = \frac{\lambda p_c}{\lambda} = p_c = \frac{(\lambda/\mu)^c \cdot (1/c!)}{\sum_{i=0}^c \frac{(\lambda/\mu)^i}{i!}} \quad (4.42)$$

As can be seen, the probability that a job is blocked is equal to the probability that all servers are busy. The equation above is called **Erlang's loss formula** or blocking formula. It has been shown that Erlang's loss formula holds for any distribution of service time, provided the input is Poisson. Usually, the formula is expressed as

$$E_m(\rho) = \frac{\rho^m / m!}{\sum_{i=0}^m \frac{\rho^i}{i!}} \quad (4.43)$$

where m is the number of servers and ρ is the offered load, which equals λ/μ .

It can be easily shown that

$$E_m(\rho) = \frac{\rho E_{m-1}(\rho)}{m + \rho E_{m-1}(\rho)} \quad (4.44)$$

which makes the calculations very simple, since $E_0(\rho) = 1$.

Some results

The **effective arrival rate** is given by

$$\lambda_{eff} = \lambda \sum_{k=0}^{c-1} p_k + 0 \cdot p_c = \lambda(1 - p_c) = \lambda(1 - E_c(\rho_o)) \quad (4.45)$$

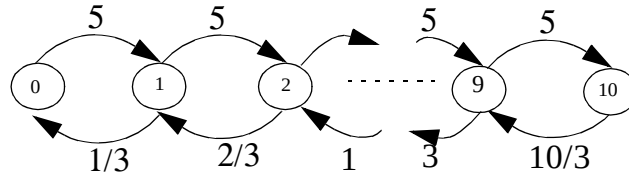
The **average number of busy servers**, which is the same as the **average number of jobs in the system**, is given by

$$\bar{N} = \bar{N}_s + \bar{N}_q = [N_q = 0] = \bar{N}_s = \lambda_{eff} \cdot E[X] = \frac{\lambda}{\mu} \cdot (1 - E_c(\rho_o)) = \rho_o(1 - E_c(\rho_o)) \quad (4.46)$$

The **carried load** is as always equal to the average number of jobs in the system, that is

$$\rho_c = \rho_o(1 - E_c(\rho_o)) \quad (4.47)$$

Example 4.9: Assume that we model the radio channels in a GSM-cell as an $M/M/c$ -system without queue. There are 10 channels in a cell, and in average 5 new calls arrive each minute. The average call holding time is 3 minutes. The state diagram for this system is shown in Figure 4.15.



Figur 4.15. State diagram for the system in Example 4.9.

The *stationary probability distribution* is given by

$$p_k = \frac{\frac{15^k}{k!}}{\sum_{i=0}^{10} \frac{15^i}{i!}}$$

The probability that a call is blocked can be found by using the special *Erlang's tables*. In those tables, the blocking probabilities can be found for varying loads and number of servers. In this case, the *blocking probability* is given by:

$$P_b = p_{10} = \frac{\frac{15^{10}}{10!}}{\sum_{i=0}^{10} \frac{15^i}{i!}} = E_{10}(15) \approx 0.410341$$

This means that the *carried load* will be

$$\rho_c = \rho_0(1 - E_{10}(\rho_0)) = 15(1 - 0.410341) \approx 8.8$$

4.6 Exercises

1. Consider an $M/M/1$ -system with parameters λ and μ . Let $\rho = \lambda/\mu$.
 - (a) Let $\rho = 0,9$. What is the average number of jobs in the system?
 - (b) Let $\rho = 0,9$. What is the average number of jobs in the queue?
 - (c) Let $\rho = 0,9$. What is the most probable state at any time?
 - (d) Assume that the queue is finite with L places. Determine the average waiting time in the queue when $\rho \rightarrow \infty$.
2. Assume that we have an $M/M/1$ -system with parameters λ and μ . Let N be the number of jobs in the system at some time. Determine $E[N]$, $E[N^2]$, and $Var[N]$.
3. During the Carneval, one of the tents could be modelled as an $M/M/1$ -system with infinite queue, since there was only room for one person at a time inside the tent. Measurements showed that $\lambda=1$ customers per minut and $\mu=1.1$ customers per minut were accurate values of the arrival rate and the service rate.
 - (a) In average, how many customers per day visited the tent if the tent was open for 12 hours a day?
 - (b) Determine the steady-state probability distribution. Note, that all steps in the calculation must be described.
 - (c) Calculate the average number of customers in the queue.
 - (d) Determine the probability that a customer entered the tent directly without queueing first.
4. A computer system is modelled as an $M/M/1$ -system with a finite queue with 3 places. The average arrival rate is 5 jobs per second, and the average service time is 0.2 seconds.
 - (a) Draw a state diagram and determine the stationary probability distribution.
 - (b) Determine the average number of jobs in the system.
 - (c) Calculate the average time a job spends in the system.
 - (d) Determine the carried load.

-
5. Assume that we model a web site with two identical servers as an M/M/2-system with one queueing place. HTTP requests arrive according to a Poisson process with mean 1 request per second. The average service time for a request is 1 second.
 - (a) Draw the state diagram.
 - (b) Determine the blocking probability for a job, and the probability that the system is full.
 - (c) Determine offered and carried load.
 - (d) Determine the utilisation of the servers.
 - (e) What is the average waiting time in the queue for a request that is not blocked?
 6. Consider an M/M/1-system, in which there is only space for one job in the queue. The average arrival rate is 2 s^{-1} , and the average service time is 1 second.
 - (a) Determine the stationary probability distribution.
 - (b) Calculate mean and variance for the number of jobs in the system.
 - (c) Let T be a random variable that correspond to the time in the system for a job that is served. Determine $P(T \leq t)$.
 7. Assume that we have an M/M/c-system with infinite queue, and parameters λ and μ . There is *one* server ($c=1$) if there are less than four jobs in the system, and *two* servers ($c=2$) if there are four or more jobs in the system. Let $\lambda=5$ and $\mu=10$.
 - (a) Determine the stationary probability distribution.
 - (b) Determine the average number of jobs in the servers.
 - (c) Determine the average time in the system for a job.
 8. Consider an M/M/1-system with arrival rate λ . Customers arrive in groups with 2 customers. The server treats one customer at a time, with service rate μ .
 - (a) Draw a state diagram, and determine the global balance equations.
 - (b) Determine $P(z) = \sum_{k=0}^{\infty} p_k z^k$ by using the balance equations.
 - (c) Calculate the average number of customers in the system.
 9. Assume that we have a system with one server, a queue with 2 places, and 4 clients that send jobs to the server. Each client thinks for an exponential distributed time with mean 2 seconds before sending a job. A client will only send one job at a time to the system. The service time for a job is exponential distributed with mean 1second.
 - (a) Draw the state diagram.
 - (b) Determine the offered and carried load.
 - (c) Calculate the blocking probability for a job, and the probability that the system is full.
-

-
10. Assume that we can model a computer system as an M/M/2-system with one queueing place and 6 clients. A client that is idle sends a new job to the system with the constant intensity $1/2 \text{ s}^{-1}$. The average service time for a job is 1 second.
- (a) Draw the state diagram for this system.
 - (b) Determine the stationary probability distribution.
 - (c) Determine the average number of blocked jobs per second.
 - (d) Calculate the blocking probability for a job, and the probability that the system is full.
 - (e) Determine the carried load.
11. A queueing system serves 4 customers. There are 2 servers and one queueing place. A customer that is not served or waits in queue, makes a call to the system with the constant intensity 1 s^{-1} . The service times are exponential distributed with mean 1 second.
- (a) Determine the stationary probability distribution.
 - (b) Determine the average number of customers in the queue.
 - (c) Determine the average blocking probability for a customers, and the probability that the system is full.
 - (d) How many customers are in average served each second?
12. In this exercise, we will analyse an M/M/ ∞ -system, that is a system with an infinite number of servers. Assume that new jobs arrive according to a Poisson process with mean λ , and that the service times are exponential distributed with mean $1/\mu$. Determine the following:
- (a) Carried load.
 - (b) Average waiting time.
 - (c) The stationary probability distribution.
 - (d) The average number of jobs in the system.
13. Assume that we can model a communication system as an M/M/ ∞ -system, where new jobs arrive in groups of two. The average arrival rate is λ groups per second. The average service time for a job is μ seconds.
- (a) Draw the state diagram, and determine the global balance equations.
 - (b) Determine the differential equation that has to be solved in order to obtain $P(z)$.
 - (c) Show that $P(z) = Ce^{\lambda(z + z^2/2)/\mu}$ is a solution to the differential equation.
 - (d) Determine C .
 - (e) Determine the average number of jobs in the system.
-

-
14. Solve the following exercises for an M/M/1-system with $\lambda=1 \text{ s}^{-1}$ and $\mu=2 \text{ s}^{-1}$.
- (a) Assume that the system is empty at time $t=20$ seconds. Calculate the probability that at least one job has arrived at time $t=22$ seconds.
 - (b) Assume that a served customer leaves the system empty. For how long, in average, will the system stay empty?
 - (c) Assume that there are two customers in the system at time $t=10$ seconds. Calculate the probability that these two customers have left the system at time $t=12$ seconds.
15. Determine the number of needed servers in an M/M/c-system without queue, so that the blocking probability is less than 1% when the offered load is 5, 10, and 20 Erlangs. Further, determine the utilisation of each server for these systems, and for the same systems when the offered load is 20% higher.
16. An M/M/10-system without queue is offered a load of 5 Erlangs. Let p_k be the probability that there are k jobs in the system. Determine p_8 , p_9 , and p_{10} .
17. Consider two M/M/c-systems without queue that are offered the loads ρ_1 and ρ_2 , respectively. The first system has c_1 servers, and the second system has c_2 servers. We would like to compare this configuration with one in which we have a loss system with c_3 servers, and an offered load of $\rho_1+\rho_2$ Erlangs. Let $\rho_1=3$ and $\rho_2=6$.
- (a) Determine c_1 , c_2 , and c_3 so that the average blocking probabilities become less than 0.1%.
 - (b) Determine the average utilisation of the servers for each of three systems.
 - (c) Determine, for each of the three systems, the probability that no more than 2 servers are idle.
18. A server system with m servers is offered a load from two sources (Poisson processes). Jobs arriving from the first source can wait in an infinite queue if all servers are busy. Jobs arriving from the second source cannot wait in a queue, which means that for them the system is a loss system. The arrival rates are λ_1 and λ_2 , respectively for each source. The service times are exponential distributed with mean $1/\mu$, for both sources.
- (a) Draw a state diagram and determine $p_k=P(k \text{ jobs in the system})$ expressed in p_0 .
 - (b) Determine an expression for the blocked load.

19. Consider a loss system with two servers, A and B. Jobs arrive from two sources that generate Poisson processes with means λ_1 and λ_2 , respectively. Jobs generated by the first source can only be processed in server A. Jobs generated by the second source can be processed by both servers, but choose server B if this server is idle. The service times are exponential distributed with mean $1/\mu$ for all jobs.

- (a) Define suitable states and draw a state diagram.
- (b) Let $\lambda_1=1$, $\lambda_2=2$, and $\mu=1$. Determine the utilisation of server A.
- (c) Determine the utilisation of server B.
- (d) Determine the carried load for the system.

20. In this exercise we will analyse a multiprocessor system with 3 processors and a central memory. We model the system as a loss system with 3 servers, to which jobs arrive according to a Poisson process with mean λ . Each job requires an exponential service time with mean $1/\mu$ seconds. However, due to the central memory the processors cannot work at maximum speed when more than one processor is busy. Therefore, we assume that the total service rate when k processors are busy is $k\mu C^{k-1}$, where C is a constant $0 < C < 1$.

- (a) Draw a state diagram and determine the stationary probability distribution.
- (b) Calculate the blocking probability. Assume that $\lambda=1500 \text{ s}^{-1}$, $1/\mu=1 \text{ ms}$, and $C=0.8$.
- (c) Determine the carried load.
- (d) We would like to decrease the blocking probability, and therefore choose between 1) add another processor or 2) change the memory, which means that $C=0.95$. Determine which alternative that gives the lowest blocking probability.

21. Solve the following exercises:

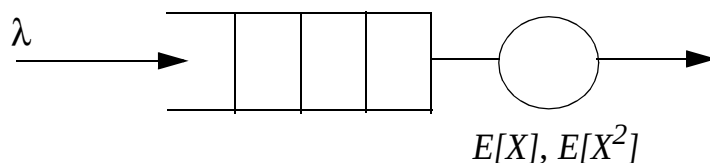
- (a) We conduct measurements on a loss system with 100 servers, in which we assume that the Erlang's model is valid. The measurements show that all 100 servers are busy for a total time of 72.3 seconds during an hour. During the same hour, exactly 99 servers were busy for a total time of 82.2 seconds. Estimate the offered load during this hour.
- (b) Calls arrive at a communication system with the mean rate 400 jobs per hour (exponential distributed and independent interarrival times). If the time interval between two successive calls is less than 0.02 seconds, the last call cannot be detected. In average, how many calls are lost per hour due to this phenomenon?

-
22. In a system without queue there are 10 servers. The service time for a job is exponential distributed with mean 1 second. There are N sources connected to the system. Each source sends job to the system according to a Poisson process with mean $1/2$ jobs per second. A source will continue to send jobs to the system with the same average rate even if there are jobs from that source already being processed in the system.
- (a) How many sources can be connected to the system if we want the job blocking probability to be less than 0.01? Motivate your answer, and then use this number of sources in the following exercises.
 - (b) Determine the carried load per server.
 - (c) Determine the probability that at least 8 servers are busy.

Chapter 5

M/G/1-systems

The M/M/1-systems that were studied before can be analysed very easily, but many systems have more complex behaviour. For example in a packet switching system the packets are not exponential distributed in length, but are either of constant length, or of a random length with some minimum and maximum values. Since transmissions occur at a fixed rate, packet length and transmission times are directly related.



Figur 5.1. An M/G/1-system

Therefore, we are now going to study a so-called M/G/1-system, shown in Figure 5.1. In this system, there is one server and the queue is unlimited. Further, the interarrival times are exponential distributed whereas the service time is generally distributed with density function $b(t)$. This means that the probability that a service in progress will end within a particular time is no longer independent of time, but depends on how long the service has lasted already. No assumptions are made about the distribution of the service time except that the first and second moments should be finite ($E[X] < \infty$, $E[X^2] < \infty$). The service time for

each job must be independent of both the arrival times and of the service requirements of other jobs.

When we discuss the stationary probability distribution, p_k , we usually mean the distribution at a random chosen time instant. However, sometimes we are instead interested in the probability distribution at an arrival to the system or a departure from the system. Therefore, we can define the following probability distributions:

1. $p_k = P(k \text{ jobs in the system at a random time instant})$
2. $r_k = P(k \text{ jobs in the system at an arrival})$
3. $d_k = P(k \text{ jobs in the system at a departure})$

It can be shown that for queueing systems with Poisson arrivals, $p_k = d_k = r_k$.

During the analysis of the M/G/1-system, we will consider the system at *departure instants*. However, due to the result above, all formulas that we find valid at departure instants are also valid at any randomly chosen time instant.

5.1 Pollaczek-Khintchin's formula

The average number of jobs in an M/G/1-system is given by the so-called Pollaczek-Khintchin's formula

$$\bar{N} = \rho + \frac{\lambda^2 \cdot E[X^2]}{2(1-\rho)} \quad (5.1)$$

where $\rho = \lambda E[X]$. The formula is sometimes expressed in terms of the squared coefficient of variation of the service time, C^2 , giving

$$\bar{N} = \rho + \frac{\rho^2(1+C^2)}{2(1-\rho)} \quad (5.2)$$

Since $\bar{N} = \bar{N}_s + \bar{N}_q$ and $\bar{N}_s = \rho$ (no blocking), the average queue length is given by

$$\bar{N}_q = \frac{\lambda^2 \cdot E[X^2]}{2(1-\rho)} \quad (5.3)$$

Note that the variance of the service time is given by

$$V[X] = E[X^2] - E[X]^2$$

This means that the average queue length in the system depends not only on the mean service time, but also on the variance of the service time. Notice that if the variance of the service time is infinite, then so is the average queue length, even though all the distributions may have finite means and the system is not saturated.

By using the Little's theorem, the average response time can also be found

$$T = \frac{\bar{N}}{\lambda_{eff}} = \frac{\bar{N}}{\lambda} = \frac{1}{\mu} + \frac{\lambda E[X^2]}{2(1-\rho)} \quad (5.4)$$

Proof

We consider the length of the queue at the instants just after a job departs, but before the next job in the queue has started service. These instants are not evenly spaced in time, since jobs have different service requirements. If the departure left the system empty, the interval until the next departure will include an idle period, until a new job arrives, as well as the service time of a job. Let us denote the number of jobs left behind by the i :th job on departure by n_i . Similarly denote the number of arrivals during the service of the i :th job as a_i . How does n_{i+1} depend on these quantities? The new queue length is the old queue length, less the job which has just finished service, plus any arrivals that occurred during the service of the job which has just departed. If the queue was empty after the last departure, the new queue length will be just those jobs that arrived during the service of the job which has just departed. This means that

$$\begin{aligned} n_{i+1} &= n_i - 1 + a_{i+1} & n_i > 0 \\ n_{i+1} &= a_{i+1} & n_i = 0 \end{aligned} \tag{5.5}$$

Defining the function $\delta(v)$ by

$$\delta(v) = \begin{cases} 1 & v > 0 \\ 0 & v = 0 \end{cases}$$

and we can rewrite the formula as

$$n_{i+1} = n_i - \delta(n_i) + a_{i+1} \tag{5.6}$$

We take the expectations of both sides, and get

$$E[n_{i+1}] = E[n_i] - E[\delta(n_i)] + E[a_{i+1}] \tag{5.7}$$

So far, none of our operations has depended on the distribution of the quantities involved, although we have implicitly assumed that the expectations exist and are finite. Since, as always, we are seeking an equilibrium solution, the expectations must be stationary, and independent of the subscript. Therefore

$$E[n_{i+1}] = E[n_i] = E[n]$$

The assumption of Poisson arrivals means that $E[a_{i+1}]$ will be independent of i , say $E[a]$. If arrivals were not Poisson, then the number of arrivals during the service of the i :th customer would depend on when the i -1st customer finished service with respect to the arrival process.

We now rearrange equation (5.7) and find

$$E[\delta(n)] = E[a]$$

Since arrivals are Poisson with rate λ the mean number of arrivals in an interval of length t is λt . Hence, the expected number of arrivals during a service time is

$$E[a] = \int_0^{\infty} \lambda t b(t) dt = \lambda \cdot E[X] = \rho$$

That is, the probability that the queue left by a customer departing after service is non-zero is equal to the expected number of arrivals during a customer's service.

To make any further progress, we examine equation (5.6) again. If we square both sides it will still be equal. We first observe that for non-negative v , $\delta(v)^2 = \delta(v)$ and $v \cdot \delta(v) = v$. After squaring both sides we arrive at

$$n_{i+1}^2 = n_i^2 + \delta(n_i) + a_{i+1}^2 - 2n_i - 2\delta(n_i)a_{i+1} + 2n_ia_{i+1}$$

and taking the expectations of both sides,

$$E[n^2] = E[n^2] + E[\delta(n)] + E[a^2] - 2E[n] - 2E[\delta(n)a] + 2E[na] \quad (5.8)$$

Since arrivals are independent of queue length,

$$E[\delta(n)a] = E[\delta(n)] \cdot E[a]$$

and

$$E[na] = E[n]E[a]$$

Rearranging equation (5.8) we find that

$$E[n] = E[a] + \frac{E[a^2] - E[a]}{2(1 - E[a])} \quad (5.9)$$

We have already calculated $E[a]$; it equals ρ . $E[a^2]$ is the only remaining unknown. If we assume that the service time of the job just departed was t , then the probability of i arrivals during the last service time is $(\lambda t)^i / i! \cdot e^{-\lambda t}$, since arrivals are Poisson. This means that

$$E[a^2|t] = \sum_{k=0}^{\infty} k^2 \cdot P(k \text{ arrivals during } t) = \dots = (\lambda t)^2 + \lambda t$$

To remove the conditioning on the value of t , we use the continuous version of the theorem of total probability, integrate with respect to $b(t)$, the density function of a service time, and find

$$E[a^2] = \int_0^{\infty} E[a^2|t]b(t)dt = \int_0^{\infty} (\lambda^2 t^2 + \lambda t)b(t)dt = \lambda^2 E[X^2] + \lambda E[X]$$

Substituting in equation (5.9), we arrive at the Pollaczek-Khintchins formula:

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

The formula just derived gives the average number of jobs in the system at departure instants. It is possible to verify that this is identical to the average number of jobs in the system at any instant.

Example 5.1: Compare two M/G/1-systems. In the first system all jobs require the deterministic service time b . In the second system, the jobs require one of two deterministic service times $b/2$ and $3b/2$, each service time occurring with probability 0.5. Then although the two systems have the same mean service time, the mean queue length will be greater in the second system with two different service times. The Pollaczek-Khintchine formula shows that the minimum queue length will be obtained when the variance of the service times is reduced to zero. This is a general phenomenon; **the queue lengths in a system will always be reduced if the variance of the random processes can be reduced.**

5.2 Stationary probability distribution

It is usually difficult to find an explicit expression for the stationary probability distribution, that is p_k . Instead, the generating function, $P(z)$, can be found. For an M/G/1-system

$$P(z) = \frac{B^*(\lambda(1-z))(1-z)(1-\rho)}{B^*(\lambda(1-z)) - z} \quad (5.10)$$

where $B^*(\lambda(1-z))$ is the Laplace transform of the service time distribution, $B^*(s)$, evaluated in $s=\lambda(1-z)$.

Proof

In the previous section, we observed, without commenting it, that the number of jobs left by a departing job depended on the number left by the previous departure and on the number of arrivals during the service. Since arrivals are Poisson, and we know the distribution of service times, we can calculate the distribution of the number that will arrive during a service. Thus we can calculate the probability of the different state changes. Further, the probability of the number of jobs left by a departing job having a particular value depends only on the number of jobs left by the previous departure, and not on its history. This means that the queue lengths at departure instants form a discrete-time Markov chain.

As before we consider the system at departure instants only. The state of the system is given by the number of jobs in the queue plus server, and we denote the steady state probability of there being j jobs with p_j . We need to find the transition probabilities, p_{ij} , from state i to state j . When these transition probabilities are known, p_i are the solutions to the balance equations:

$$p_j = \sum_{i=0}^{\infty} p_{ij} p_i \quad (5.11)$$

We have already observed that the transition probabilities depend only on the number of arrivals during a service. This is reflected in equation (5.5). Let α_i be the probability of i arrivals during a service. The discussion used above when we evaluated the mean queue length applies here, and shows that $p_{ij} = \alpha_{j-i+1}$ for $i > 0$, and $p_{0j} = \alpha_j$. Transitions from i to j with $j < i - 1$ are impossible, since they imply more than one departure from the queue. Hence equation (5.11) becomes

$$p_j = \alpha_j p_0 + \sum_{i=1}^{\infty} \alpha_{j-i+1} p_i \quad (5.12)$$

A common tool for dealing with infinite sets of equations, such as the ones above, is the generating function. Let $A(z) = \sum_{i=0}^{\infty} \alpha_i z^i$ be the generating function of the number of arrivals during a job's service time, and $P(z)$ be the generating function of the number of jobs in the system left by a departing job. We multiply each of (5.12) by z^j , and add them, which gives

$$\sum_{j=0}^{\infty} p_j z^j = \sum_{j=0}^{\infty} \alpha_j p_0 z^j + \sum_{j=0}^{\infty} \sum_{i=1}^{\infty} \alpha_{j-i+1} p_i z^j$$

Rearranging the sums, we find

$$P(z) = \sum_{j=0}^{\infty} p_j z^j = A(z) p_0 + \sum_{i=1}^{\infty} p_i z^{i-1} \cdot \sum_{j=i-1}^{\infty} \alpha_{j-i+1} z^{j-i+1}$$

which simplifies to

$$P(z) = A(z) p_0 + A(z) \cdot \frac{P(z) - p_0}{z}$$

Hence

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

Now we can evaluate $A(z)$ simply, since the arrival process is Poissonian. Given that a service lasts t time units, the probability of i arrivals, α_i , is given by $\frac{(\lambda t)^i}{i!} e^{-\lambda t}$. Thus the generating function of the number of arrivals during a service, conditional on the service having duration t is given by

$$A_t(z) = \sum_{i=0}^{\infty} \frac{(\lambda t)^i}{i!} e^{-\lambda t} \cdot z^i = e^{-\lambda t(1-z)}$$

We now have to remove the conditioning on t . The density function for the service time is given by $b(t)$. This means that

$$A(z) = \int_0^{\infty} A_t(z) b(t) dt = \int_0^{\infty} e^{-\lambda t(1-z)} b(t) dt$$

which is the same as the Laplace transform of the service time, evaluated at $s = \lambda(1 - z)$, that is

$$A(z) = B^*(\lambda(1 - z)) \quad (5.13)$$

This means that

$$P(z) = \frac{B^*(\lambda(1 - z))(1 - z)}{B^*(\lambda(1 - z)) - z} p_0 \quad (5.14)$$

Since we have a single server system, we know that $p_0 = 1 - \rho$. However, this can also be verified by using the normalisation condition $P(1) = 1$. Direct substitution of $z = 1$ leads to an indeterminate form 0/0. Using l'Hospital's rule, we find:

$$-\lambda B^{*'}(0) - 1 = p_0[(1 - z)(-\lambda B^{*'}(0)) - B^*(0)]$$

Now $-B^{*'}(0) = E[X]$, and $B^*(0) = 1$, so $p_0 = 1 - \rho$ as expected.

This means that

$$P(z) = \frac{B^*(\lambda(1 - z))(1 - z)(1 - \rho)}{B^*(\lambda(1 - z)) - z} \quad (5.15)$$

Using this formula, we can confirm the Pollaczek-Khintchin's formula arrived at in the previous section, by differentiating $P(z)$ and evaluating it at $z = 1$.

5.3 Waiting time distributions

We have seen that the mean residence time for a job can be found by evaluating the mean number in the system and using Little's theorem. However, this approach tells us nothing about the distribution of the residence time, only its mean.

The jobs left in the system by a departing job are just those jobs which have arrived during the residence time of the job which has just departed. Let us assume that we know the density function of the time in the system, $s(t)$, and its Laplace transform, $S^*(s)$. Consider a departing job, which has been in the system for a time t . It leaves behind i jobs, all of which arrived during its residence in the system. Since arrivals are Poisson, the distribution of i satisfies $P(\text{No of jobs left} = i) = e^{-\lambda t} (\lambda t)^i / i!$. Hence, the generating function of the number of jobs left by a departing job whose system residence time was t is given by

$$P(z, t) = e^{-\lambda t(1 - z)}$$

In order to find the generating function of the number left in the system by a random departing job, independent of its residence time, we must integrate with respect to the system residence time density function, $s(t)$:

$$P(z) = \int_0^\infty P(z, t) s(t) dt = \int_0^\infty e^{-\lambda t(1 - z)} s(t) dt$$

which is the definition of the Laplace transform of $s(t)$ evaluated at $s = \lambda(1 - z)$. Hence $P(z)$, the generating function of the number of jobs left in the system, is given by

$S^*(\lambda(1-z)) \cdot P(z)$ is already known from equation (5.15). Hence, we can find $S^*(s)$ by making the change of variable $s=\lambda(1-z)$

$$S^*(s) = \frac{(1-\rho)sB^*(s)}{s-\lambda+\lambda B^*(s)} \quad (5.16)$$

Using the formula above, we can also find the Laplace transform for the waiting time in the queue, $W^*(s)$. We know that the residence time for a job is the sum of the waiting time and the service time. Since the waiting time in the queue and the service time for a job are independent, the Laplace transform for the residence time is given by

$$S^*(s) = W^*(s) \cdot B^*(s)$$

Therefore,

$$W^*(s) = \frac{(1-\rho)s}{s-\lambda+\lambda B^*(s)} \quad (5.17)$$

5.4 Exercises

1. Use Pollaczek-Khintcin's formula to show the well-known formula for the number of jobs in an M/M/1- system.
2. In Figure 5.2, the mean number of jobs in the system, $\bar{N}$, has been plotted as a function of the offered load, ρ . The results from four different systems are shown in the diagram. Determine which of the systems that belong to which curve. The four systems are:
 - M/M/1 with $E[X]=1$.
 - M/D/1 with $E[X]=1$.
 - M/ E_2 /1 with $E[X]=1$.
 - M/ H_2 /1 with $\mu_1=0.5$, $\mu_2=2$, and $p_1=0.34$.

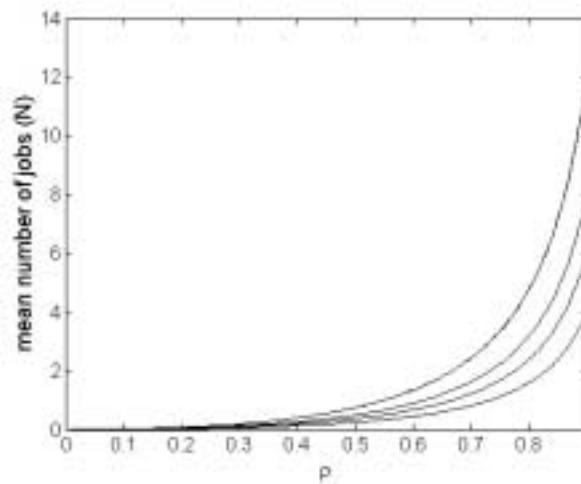


Figure 5.2. Diagram for Exercise 2.

-
3. Use Pollaczek-Khintchin's formula and find the average time a job spends in the system, T , if the arrival rate $\lambda=0.1s^{-1}$ and the service time is
 - (a) exponential distributed with $E[X] = 3$ seconds.
 - (b) deterministic. All services take exactly 3 seconds.
 - (c) 1 second with the probability 0.25, 3 seconds with the probability 0.5, and 5 seconds with the probability 0.25.
 4. Use Pollaczek-Khintchin's formula to find the average time a job spends waiting in the queue, W , if the arrival rate is λ , the mean service time is $E[X]$ and the service time is
 - (a) exponentially distributed.
 - (b) deterministic.
 - (c) compare the results from (a) and (b).
 5. A communication line capable of transmitting at a rate of 50 kbit/s will be used to accommodate a session that generates 1500 packets/min. Find the average number of packets in queue, the average number of packets in the system and the average time in the system for a packet if
 - (a) the packet lengths are exponentially distributed with mean 1000 bits.
 - (b) 10% of the packets are 100 bits long and the rest are 1500 bits long.
 6. Assume that we can model a customer support desk as an M/G/1-system with $\lambda=10$ calls per hour. The service time for each call is, initially, exponential distributed with mean 5 minutes. However, the customer server is sent on a course in order to improve the service. After the course, his average service time has unfortunately *increased* with 10%, however, the standard deviation of the service time has *decreased* with 20%. Determine, whether the general performance of the system has improved or not.
 7. Use equation (5.10) to find the generating function for an
 - (a) M/M/1-system.
 - (b) M/D/1-system.
 8. Use equation (5.17) to find the Laplace transform for the waiting time in the queue, for an
 - (a) M/M/1-system.
 - (b) M/D/1-system.
-

-
9. Consider an $M/E_r/1$ -system, with mean arrival rate λ and mean service time $1/\mu$.
- Determine the Laplace transform for the waiting time in queue, $W^*(s)$, expressed in s , λ , μ , and r .
- Let $r=5$, $\lambda=0.8s^{-1}$, and $\mu=1s^{-1}$. Determine the following metrics:
- The average waiting time in queue (W).
 - The average waiting time in queue for those jobs that have to wait.
 - The average number of jobs in the queue ($\bar{N}_q$).
 - The average time in the system for a job (T).
10. Consider an $M/G/1$ -system *without* queue. Determine the stationary probability distribution, p_k .
11. We have a computer that processes two types of jobs, A and B. New jobs arrive according to two Poisson processes, with mean 1 job per second for type A jobs and mean 0.007 jobs per second for type B jobs. The service times are exponential distributed with mean 0.7 seconds for type A jobs and 30 seconds for type B jobs. The queueing discipline is FCFS.
- Determine the Laplace transforms for the waiting time in queue for type A and type B jobs.
 - Determine the Laplace transform for the time in the system for type B jobs.
 - Find the average time in the system for type B jobs, using the transform in (b).
12. Consider an computer system, which receives interrupt signals from an external process. We assume that the system can be modelled as an $M/G/1$ -system. The interrupt signals are of two types and they use the same waiting queue. Type 1 signals arrives according to a Poisson process with mean 20 signals per second, and each type 1 signal starts the execution of a sub routine with 27 instructions. Type 2 signals arrives according to a Poisson process with mean 2 signals per second, and each type 2 signal starts the execution of a sub routine with 150 instructions. The execution time for one instruction is 1 ms.
- Determine the density function for the interarrival times of signals, irrespective of type.
 - Determine the density function for the service time of a random signal, irrespective of signal type.
 - Calculate the utilisation of the server.
 - Calculate the average waiting time in the queue for a signal.
 - Calculate the average number of signals in the queue.
 - Determine the probability that a signal must wait in the queue.
 - Determine the average waiting time in the queue for those signals that must wait.
-

13. Consider an M/G/1-system, where a served customer immediately returns to the queue with probability α , and leaves the system with probability $1-\alpha$. The service time for a returning customer is independent of the previous service times for this customer. Let $B^*(s)$ be the Laplace transform for one service time, and $B_{sys}^*(s)$ be the Laplace transform for a customer's *total* service time.

- (a) Determine $B_{sys}^*(s)$ expressed in $B^*(s)$ and α .
- (b) Calculate the mean and variance of a customer's total service time. Let the mean of one service time be $\bar{x}$.

14. Assume that we have a queueing system with a single server and an infinite queue. Further, we assume that jobs arrive in groups. New groups arrive with independent, and exponentially distributed interarrival times with mean λ groups per second. Let $g_i = P(\text{number of jobs in a group} = i)$, where $g_0 = 0$, and $G(z) = \sum_{i=1}^{\infty} g_i z^i$. The density function for the service time of a job is $b(t)$, with Laplace transform $B^*(s)$.

- (a) Show that the z-transform for the number of jobs that arrive during a time interval t is $e^{-\lambda t(1-G(z))}$.
- (b) Show that the z-transform for the random variable v = the number of jobs that arrive during a job's service time is $B^*(\lambda - \lambda G(z))$.

Chapter 6

Queueing networks

All the queues that we have studied in the previous chapters were studied in isolation. Jobs arrive at the queue by some 'magic' Poisson process, are served, and then disappear for ever from the system. In real life this is rarely the case. The input process to one queue is usually formed by the output of a number of other queues, possibly combined with entirely new jobs. After service at a queue most jobs join some other queue to await further service. A small proportion may disappear for ever.

For example, in a packet switching network, most packets are in transit from one packet switch to another; at each switch, a few new packets may be added representing new messages entering the network, and a few packets may leave the system representing the messages that have reached their destination. This is an appropriate model in many cases, particularly if the population of potential customers is large, and they act independently of one another.

6.1 Feed-forward networks

A feed-forward network is a queueing network in which each queue can have several input streams and in which the output of any queue can be split. The only restriction is that feedback must be impossible; that is, it must not be possible for jobs which have left a queue to ever visit the same queue later in their passage through the network.

Two tandem M/M/1-systems

The simplest possible network, two M/M/1 queues in tandem, is shown in Figure 6.1.

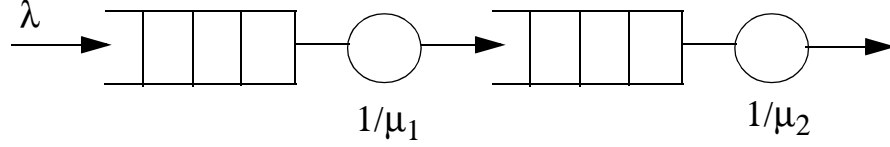


Figure 6.1. Two M/M/1-systems in tandem.

All jobs arrive at the first server, are served by it, and then go to the second, queuing before either server if necessary. We will assume that there is no limit to the number of jobs that can be waiting for either server. The arrivals at the first queue form a Poisson stream with parameter λ . Service times are exponential distributed with mean $1/\mu_1$ at server 1 and $1/\mu_2$ at server 2. A job's service time at server 2 is independent of its service time at server 1. Intuitively, the condition for the system to reach an equilibrium is that the service rate of the slower server must be greater than the arrival rate.

Since all the processes are Markovian, the system state can be represented by a pair of non-negative integers representing the number of customers at each server. The balance equations are easy to derive, given that departures from server 1 go directly to queue 2. Writing $p_{i,j} = P(i \text{ jobs at server 1 and } j \text{ jobs at server 2})$, the balance equations are

$$\begin{aligned} \lambda p_{0,0} &= \mu_2 p_{0,1} \\ (\lambda + \mu_1) p_{1,0} &= \lambda p_{0,0} + \mu_2 p_{1,1} \\ (\lambda + \mu_2) p_{0,1} &= \mu_1 p_{1,0} \\ (\lambda + \mu_1 + \mu_2) p_{1,1} &= \lambda p_{0,1} + \mu_1 p_{2,0} + \mu_2 p_{1,2} \\ &\vdots \\ (\lambda + \delta(i)\mu_1 + \delta(j)\mu_2) p_{i,j} &= \lambda p_{i-1,j} + \delta(j)\mu_1 p_{i+1,j-1} + \delta(j+1)\mu_2 p_{i,j+1} \end{aligned}$$

where $\delta(k)$ is the Kronecker delta function, equal to 1 if $k > 1$, and 0 otherwise. We can try to guess a solution. Since queue 1 is unaffected by what happens in the second queue, the marginal probability of i customers at the first system, $P_1(i)$, is well known:

$$P_1(i) = \rho_1^i (1 - \rho_1) \quad \rho_1 = \lambda / \mu_1$$

The second queue is more problematic. The arrival process is certainly a renewal process, and assuming that the whole system is stable, then it must have mean rate λ . Its distribution is not known though. If the distribution was Poisson, then the second system would be independent of the first, and

$$P_2(j) = \rho_2^j (1 - \rho_2) \quad \rho_2 = \lambda / \mu_2$$

and the joint probability of i customers in system 1 and j customers in system 2 would be

$$p_{i,j} = \rho_1^i (1 - \rho_1) \rho_2^j (1 - \rho_2)$$

However, the joint probabilities above satisfies the balance equations so the queues act as if they are independent. The second queue appears to be independent of the first, i.e. the state of the second is identical to that of an isolated M/M/1-system with input parameter λ . It appears as if the output from an M/M/1-system forms a Poisson stream with the same rate as the input rate, λ . Although the queue length probabilities are the same in the second queue as they would be if the output from the first queue was Poisson, that equality does not prove that the output stream is Poisson. We shall now prove this is the case.

Departure process from an M/M/1-system

What is the interval between successive departures from an M/M/1 system? If the previous departure left jobs in the system, then the next departure will occur when the job now starting service finishes its service. In this case, the interdeparture interval is distributed identically to the service time. If the previous departure left the system empty, the interval will be an interarrival interval plus a service interval. This interval will be distributed as the sum of the two random variables. The Laplace transform of a sum of two random variable is the product of the Laplace transforms of the individual variables.

Assuming that the arrival rate is λ , the service rate μ , and $\rho = \lambda/\mu$, we find that the Laplace transform of the interdeparture distribution, $D^*(s)$, is given by

$$D^*(s) = (1 - \rho)A^*(s)B^*(s) + \rho B^*(s)$$

where $A^*(s)$ is the Laplace transform for the interarrival time distribution and $B^*(s)$ is the Laplace transform for the service time distribution. Since the interarrival and service times are exponential distributed, their Laplace transforms are given by

$$A^*(s) = \frac{\lambda}{\lambda + s} \quad B^*(s) = \frac{\mu}{\mu + s}$$

and hence

$$D^*(s) = (1 - \rho) \frac{\lambda}{\lambda + s} \frac{\mu}{\mu + s} + \rho \frac{\mu}{\mu + s} = \dots = \frac{\lambda}{\lambda + s} = A^*(s)$$

which means that the distribution of interdeparture times is identical to the distribution of interarrival times. Also, the departure process is a Poisson process, since the interdeparture times are independent, which is not shown here.

Therefore, the second server can be analysed independently of the first. This property of M/M/1 systems was first proved by P.J. Burke, and is often referred to as Burke's theorem. Other systems which have the property of preserving the Poisson nature of arrival stream in the departure stream include the M/M/m-system and the M/G/ ∞ -system. This property is also called the “Markov-implicate-Markov” property.

Merging and splitting of Poisson processes

A network of queues that have the “Markov-implicate-Markov” property, which are connected in tandem can be analysed as if each queue was independent. More complex con-

nections between the queues can be allowed, so long as we maintain the Poisson nature of the flows.

Merging of Poisson processes. The merged stream resulting when several independent Poisson streams are joined is itself a Poisson stream. This can be proved by considering the definition of a Poisson stream of rate λ in terms of an arrival during a short interval Δt .

$$P(1 \text{ arrival in } \Delta t) = \lambda \Delta t + o(\Delta t)$$

$$P(0 \text{ arrivals in } \Delta t) = 1 - \lambda \Delta t + o(\Delta t)$$

$$P(\text{multiple arrivals in } \Delta t) = o(\Delta t)$$

Consider the output process formed by the merging of two independent Poisson streams with parameter λ_1 and λ_2 :

$$P(1 \text{ output in } \Delta t) = P(1 \text{ arrival in stream 1 and 0 arrivals in stream 2}) \\ + P(1 \text{ arrival in stream 2 and 0 arrivals in stream 1})$$

$$= (\lambda_1 \Delta t + o(\Delta t))(1 - \lambda_2 \Delta t + o(\Delta t)) + (\lambda_2 \Delta t + o(\Delta t))(1 - \lambda_1 \Delta t + o(\Delta t))$$

Performing the multiplication, we see that no terms larger than $o(\Delta t)$ remain, except $(\lambda_1 + \lambda_2)\Delta t$. Similarly, the probability of more than one arrival is $o(\Delta t)$, and of no arrivals is $1 - (\lambda_1 + \lambda_2)\Delta t + o(\Delta t)$. These are the axioms for a Poisson process with rate $\lambda_1 + \lambda_2$.

Splitting of Poisson processes. A similar analysis can be used to show that a Poisson stream can be split into several separate streams, and so long as the choice of which output stream to add a job is made at random, the output streams are also Poisson. The restriction to random splitting means that if a Poisson stream is split by sending alternate customers to alternate streams, the output stream will not be Poisson.

Consider a Poisson stream with rate λ , in which successive arrivals are independently assigned to output stream 1 with probability p , and to output stream 2 with probability $q=1-p$. Let $N_i(t)$ be the probability of i arrivals being assigned to stream 1 in an interval of length t .

$$N_1(\Delta t) = P(1 \text{ arrival in } \Delta t \text{ and that arrival chooses stream 1}) + \\ + P(\text{multiple arrivals in } \Delta t \text{ and only one arrival chooses stream 1}) = \\ = P(1 \text{ arrival in } \Delta t) \cdot P(\text{arrival chooses stream 1}) + o(\Delta t) = \\ = (\lambda \Delta t + o(\Delta t)) \cdot p + o(\Delta t) \\ = p\lambda \Delta t + o(\Delta t)$$

Similarly the probability of no arrival being allocated to stream 1 is given by

$$N_0(\Delta t) = P(\text{no arrival in } \Delta t) + P(1 \text{ arrival in } \Delta t \text{ and that arrival chooses stream 2}) + \\ + P(\text{multiple arrivals in } \Delta t \text{ and all the arrivals choose stream 2}) = \\ = (1 - \lambda \Delta t) + \lambda \Delta t \cdot (1 - p) + o(\Delta t) \\ = (1 - p\lambda \Delta t) + o(\Delta t)$$

Clearly, $N_i(\Delta t) = o(\Delta t)$ for $i \geq 2$, and these probabilities are just the axiomatic definition of a Poisson stream with rate $p\lambda$.

Thus a network of queues in which each queue can have several input streams and in which the output of any queue can be split, can still be analysed as if each queue were independent.

6.2 Networks with feedback

Feed-forward networks are applicable to a fairly restricted set of circumstances. When we introduce feedback into the network, so that a job may visit the same queue more than once, the flows in the network need not to be Poisson any more.

An M/M/1-system with feedback

Consider the simple network shown in the Figure 6.2. Jobs arrive in a Poisson stream, and receive an exponential distributed service. After service they leave the system with probability p , or return to the queue with the probability $1 - p$.

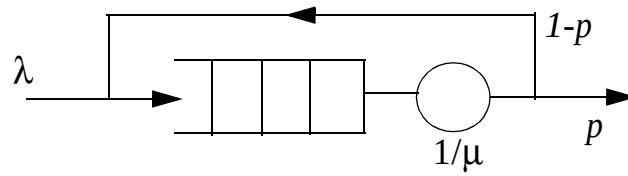


Figure 6.2. An M/M/1-system with feedback.

Although external arrivals form a Poisson stream, the arrivals at the queue do not. They include not only the external arrivals, which are Poisson and independent of the service process, but also the jobs which have been fed back after service. These fed back arrivals only occur at service completion instants and cannot be independent of service times. The actual distribution of interarrival times at the queue is non-exponential. Similarly the intervals between jobs leaving the server, and between jobs being fed back, are not exponentially distributed.

However, since both the interarrival times of new jobs and the service times are exponential distributed, the system can be analysed with a Markov chain. The steady-state diagram for this system is shown in Figure 6.3. As can be seen from the steady state diagram, as far as queue length is concerned, the system behaves like an M/M/1 queue with arrival rate λ and service rate $p\mu$. This means that the steady state probabilities for the number of jobs in the system, p_k , are given by

$$p_k = \left(\frac{\lambda}{p\mu}\right)^k \left(1 - \frac{\lambda}{p\mu}\right) \quad k \geq 0 \quad (6.1)$$

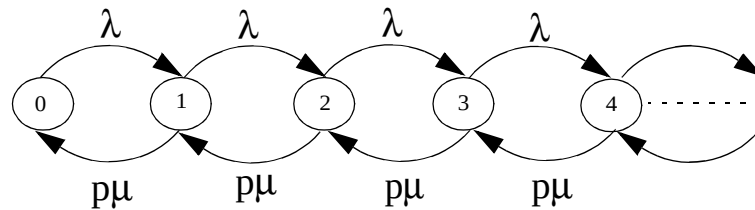


Figure 6.3. Steady-state diagram for an M/M/1-system with feedback

The system is stable when $\lambda < p\mu$.

Jackson networks

Consider a network consisting of M $M/M/1$ -systems. Jobs arrive from outside the system joining queue i at rate λ_i . The service times in system i are exponential distributed with parameter μ_i . After service in system i , a job either leaves the system with probability p_{i0} , or goes to system j , with probability p_{ij} . Clearly

$$\sum_{j=0}^M p_{ij} = 1$$

since each job must go somewhere. This type of queuing network is often known as “Jackson networks” after J.R Jackson, who first published mathematical results about this type of systems. A Jackson network is a so-called open network, since the population of jobs is not fixed, but can vary as new jobs arrive at the network and as jobs leave the network. An example of a Jackson network is shown in Figure 6.4.

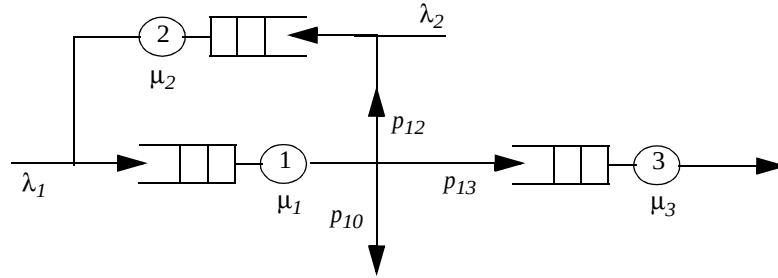


Figure 6.4. A Jackson network.

The system as a whole is clearly Markovian, with the state being given by the vector of non-negative integers representing the number of jobs in each $M/M/1$ -system. The vector $\mathbf{n}=(n_1, n_2, \dots, n_M)$ represents the state of having n_1 jobs in system 1, n_2 jobs in system 2, etc. We assume that a steady state is reached and write $P(\mathbf{n})$ for the steady state probability of state $\mathbf{n}$. The balance equations are given by

$$\begin{aligned} P(\mathbf{n}) \left(\sum_{i=0}^M \lambda_i + \sum_{i=1}^M \delta(n_i) \mu_i \right) &= \sum_{i=1}^M \delta(n_i) \lambda_i P(\mathbf{n} - \mathbf{e}_i) \\ &+ \sum_{i=1}^M p_{i0} \mu_i P(\mathbf{n} - \mathbf{e}_i) + \sum_{i=1}^M \sum_{j=1}^M \delta(n_j) p_{ij} \mu_i P(\mathbf{n} + \mathbf{e}_i - \mathbf{e}_j) \end{aligned} \quad (6.2)$$

The vector $\mathbf{e}_j$ has all components zero, except the j th, which is 1. Notice that the term $\delta(n_i) p_{ii} \mu_i P(\mathbf{n})$ appears on both sides of the equation, representing jobs that leave system i and immediately rejoin system i .

First we find the total throughput of system i . It consists of those jobs that arrive at system i from outside the network, at rate λ_i , plus all those jobs that are transferred to system i after completing service at system j for all systems j in the network. If γ_j is the throughput for system j , then the rate at which jobs arrive at system i from system j is $p_{ji} \gamma_j$. Hence the throughput of system i , γ_i , must satisfy the traffic equations:

$$\gamma_i = \lambda_i + \sum_{j=1}^M p_{ji} \gamma_j \quad (6.3)$$

These equations have a solution provided that p_{i0} is non-zero for some i .

The steady state probabilities for the network are given by

$$P(\mathbf{n}) = \prod_{i=1}^M \rho_i^{n_i} (1 - \rho_i) \quad (6.4)$$

where $\rho_i = \gamma_i / \mu_i$. This can be easily verified by substitution in the balance equations (6.2). This means that the systems can be analysed as if they are independent M/M/1-systems. It is important to note that the steady state probabilities behave *as if* the flows were Poisson, but the flows are *not*.

Example 6.1: An outgoing link in a router is modelled as the queueing network shown in Figure 6.5. The network consists of three single server systems with infinite queues (1, 2, and 3). New packets arrive at system 1 and 2 according to two Poisson processes with means λ_1 and λ_2 . When a packet has been served in system 1 or 2, it proceeds to system 3. All service times are exponentially distributed with mean $1/\mu$.

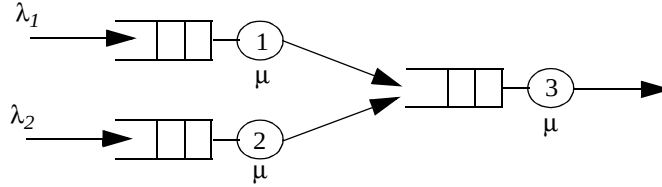


Figure 6.5. Queueing network for Example 6.1.

Since the arrival processes are Poissonian, and all three systems have infinite queues and exponential service times, we can analyse each system independently of the others.

The arrival rate for system 3, $\lambda_3 = \lambda_1 + \lambda_2$, since all jobs that have been served in systems 1 and 2 proceed to system 3. The average number of jobs in system i , $\bar{N}_i$, is given by the formula for M/M/1-systems:

$$\bar{N}_i = \frac{\rho_i}{1 - \rho_i}$$

where $\rho_i = \lambda_i / \mu_i$. The average total number of jobs in the system, $\bar{N}_{tot} = \bar{N}_1 + \bar{N}_2 + \bar{N}_3$. The average time a job spends in the network, T_{tot} , can be calculated with Little's formula:

$$T_{tot} = \frac{\bar{N}_{tot}}{\lambda_1 + \lambda_2}$$

Finally, the throughput of the network is equal to $\lambda_3 = \lambda_1 + \lambda_2$, since no jobs are blocked in the network.

Example 6.2: A web site consists of a web server and a database server. We decide to model the site as the queueing network in Figure 6.6. The network consists of two single server systems with infinite queues, representing the two servers. The service times are exponentially distributed with mean $\bar{x}_w$ for the web server and $\bar{x}_d$ for the database server. New requests arrive at the web server according to a Poisson process with mean λ jobs per second. When a request has been processed in the web server, it proceeds to the database server with probability α , or leaves the system with probability $1-\alpha$. A job that has been processed in the database server must always be processed once more in the web server.

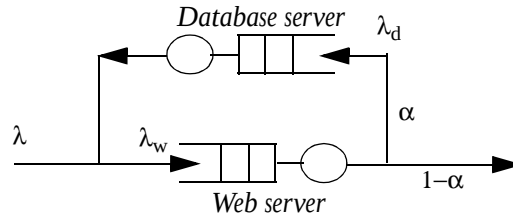


Figure 6.6. A queueing model of a web site.

This is a Jackson network, which means that the two systems behave as two independent M/M/1-systems. The arrival rate to the web server, λ_w , and the database server, λ_d , are given by

$$\begin{cases} \lambda_w = \lambda + \lambda_d \\ \lambda_d = \alpha \lambda_w \end{cases} \Rightarrow \begin{cases} \lambda_w = \frac{\lambda}{1-\alpha} \\ \lambda_d = \frac{\alpha \lambda}{1-\alpha} \end{cases}$$

The utilisations of the two servers are of course equal to the carried load

$$\begin{cases} U_w = \rho_{cw} = \lambda_w \cdot \bar{x}_w \\ U_d = \rho_{cd} = \lambda_d \cdot \bar{x}_d \end{cases}$$

The average number of requests in the site is given by

$$\bar{N}_{tot} = \bar{N}_w + \bar{N}_d = \frac{\rho_{cw}}{1-\rho_{cw}} + \frac{\rho_{cd}}{1-\rho_{cd}}$$

The throughput of the site is always equal to the effective arrival rate, which in this case is the same as the “ordinary” arrival rate since no requests are blocked:

$$Throughput = \lambda_{eff} = \lambda$$

The average time a request spends in the site before completed can be determined with Little's formula:

$$T_{tot} = \frac{\bar{N}_{tot}}{\lambda_{eff}} = \frac{\bar{N}_{tot}}{\lambda}$$

6.3 Exercises

1. A simple telephone switch could be modelled as the queueing network in figure 6.7. The network consists of three M/M/1-systems with infinite queues. The mean service time for system i is $1/\mu_i$. New jobs arrive at system 1 according to a Poisson process with mean λ_1 . When a job has been served in system 1 it proceeds to system 2 with probability α , and to system 3 with probability $1-\alpha$.

Let $\lambda_1=10s^{-1}$, $\mu_1=15s^{-1}$, $\mu_2=10s^{-1}$, $\mu_3=3s^{-1}$, and $\alpha=0.7$.

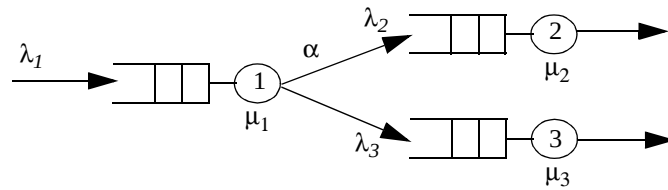


Figure 6.7. Queueing network in Exercise 1.

- (a) Which of the servers has the highest utilisation? The answer must be well motivated.
 - (b) Determine the average total number of jobs in the switch.
 - (c) Determine the average total waiting time in the queues for a job.
2. Assume that we have modelled a computer system as a queueing network, shown in figure 6.8. Systems 1 and 2 are single server systems with infinite queues, whereas system 3 is a loss system with 2 servers. The service times in all systems are exponentially distributed with means $\bar{x}_1$, $\bar{x}_2$, and $\bar{x}_3$, respectively. All jobs arrive at system 1 according to a Poisson process with mean λ . When a job has been served in system 1, it continues to system 2 with probability β and to system 3 with probability $1-\beta$.

Let $\lambda=1/15$, $\bar{x}_1 = 10$, $\bar{x}_2 = 30$, $\bar{x}_3 = 20$, and $\beta=0.2$.

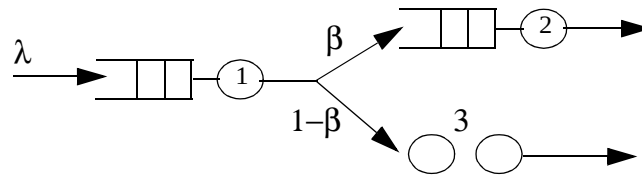


Figure 6.8. Queueing network in Exercise 2..

Determine the following metrics:

- (a) Arrival rates to systems 2 and 3.
- (b) Average number of jobs in systems 1, 2, and 3.
- (c) Average number of blocked jobs per time unit at system 3.
- (d) Average total time in the system for a job that is not blocked.
- (e) Average total waiting time in the queues for a job that is not blocked.

3. Consider the queueing network shown in Figure 6.9. The network consists of two systems, 1 and 2. System 1 is an M/M/1-system with infinite queue, and system 2 is an M/M/3-system without queue. New jobs arrive according to a Poisson process with mean λ jobs per second. The average service time is in system 1 $\bar{x}_1$ seconds, and in system 2 $\bar{x}_2$ seconds. When a job has been served in system 1, it leaves the network with probability α , and proceeds to system 2 with probability $1-\alpha$.

Let $\lambda=1/20$, $\bar{x}_1 = 10$, $\bar{x}_2 = 60$, and $\alpha=0.2$.

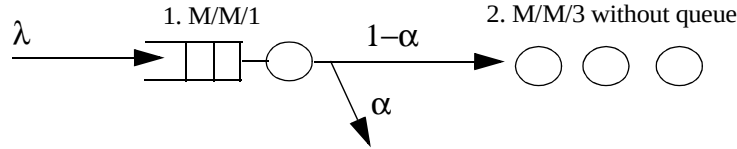


Figure 6.9. Queueing network in Exercise 3.

Determine the following metrics:

- The average arrival rate and offered load for system 2.
 - The probability that a job is blocked from system 2.
 - Average number of jobs in systems 1 and 2.
 - Carried and blocked load in system 2.
 - Average number of served jobs per second that leave system 2.
4. Assume that we have modelled a computer system as the queueing network in figure 6.10. The network consists of five systems, of which four have finite queues (1, 2, 3, and 5). System 4 is a loss system with three servers. All service times are exponentially distributed with mean $1/\mu_i$ seconds for system i . New jobs arrive according to two Poisson processes, with means λ_1 and λ_2 jobs per second. Jobs that have been served in system 3 proceed to system 4 with probability α , and to system 5 with probability $1-\alpha$.

Let $\lambda_1=4$, $\lambda_2=2$, $\mu_1=\mu_2=5$, $\mu_3=8$, $\mu_4=2$, $\mu_5=6$, and $\alpha=2/3$.

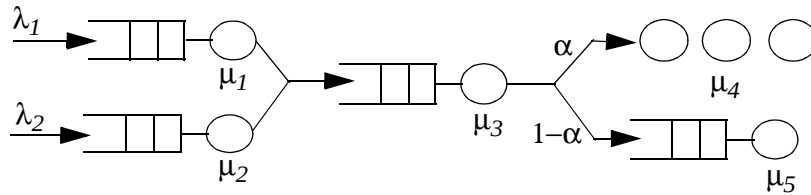


Figure 6.10. Queueing network in exercise 4.

Determine:

- The average number of jobs in each of the systems.
- The average number of blocked jobs per second.
- The average time in the network for a job that is not blocked.
- The average total waiting time in the queues for a random job.

5. In this exercise we will analyse a so-called “overflow system” that consists of two loss systems, see figure 6.11. System 1 has m_1 servers, and system 2 has m_2 servers. New calls arrive at system 1 according to a Poisson process with mean λ . Those calls that are blocked continue to system 2. If a call is blocked also in system 2, it leaves the system. The average service time in both systems is $\bar{x}$. Determine the carried load for system 2.

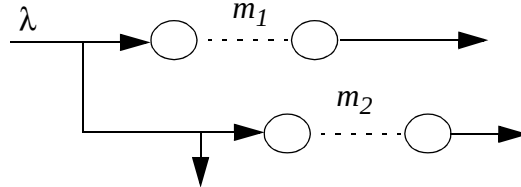


Figure 6.11. The overflow system in exercise 5.

6. Jobs arrive according to a Poisson process with mean 1 to a loss system with one server. The jobs that are served continue to a queueing system with one server and a finite queue with one place. The service times in both systems are exponentially distributed, independent of each other, and they have the mean 1.
- Determine the average number of jobs in system 1 and 2.
 - Determine the average time in system 2 for a job that is served in this system.
 - Is the number of jobs in system 1 independent of the number of jobs in system 2?
The answer must be motivated mathematically.
7. We consider a closed network with two single server systems with infinite queues, see figure 6.12. When a job has been served in the second system, it immediately returns to the first system for further processing. Assume that there are M jobs that circulate in the network. The service times are exponentially distributed with mean $1/\mu_1$ in system 1 and mean $1/\mu_2$ in system 2.

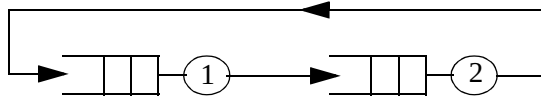


Figure 6.12. The network in exercise 7.

- Determine the joint stationary probability distribution, $p(k_1, k_2) = P(k_1 \text{ jobs in system 1 and } k_2 \text{ jobs in system 2})$.
- Determine the utilisation of server 1.
- Determine the “throughput” in the system, that is the number of jobs per second that leaves system 2.
- Let $\mu_1=1$, and $\mu_2=1/10$. Determine the metrics in (b) and (c) when $M=1, 2, 3, 4$, and 5. What happens with the utilisation of the two servers?

8. The queueing network in figure 6.13 represents a distributed web site. The site consists of three systems (1, 2, and 3) with single servers and infinite queues. All service times are exponentially distributed with mean $1/\mu_k$ for system k . New HTTP requests arrive at the site according to a Poisson process with mean λ . A request may need to be processed several times before completed. All requests are first processed in system 1. After processing, a request is completed with probability α , and therefore leave the site. With probability $1-\alpha$, a request remains in the site for further processing. Those requests proceed to system 2 with probability β , and to system 3 with probability $1-\beta$.

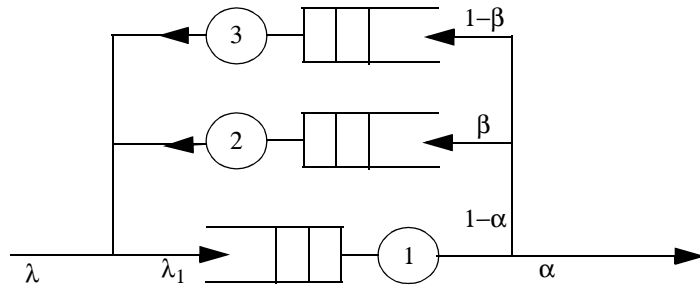


Figure 6.13. A distributed web site

- (a) Denote the average arrival rate for system k with λ_k . Determine λ_k , for $k=1,2$, and 3.
- (b) Determine the average number of requests in the site.
- (c) Determine the average time a job spends in the site before completed.
- (d) Determine the average total waiting time in queues for a request.
- (e) Determine the average number of jobs that leave the site per time unit.
9. Figure 6.14 shows a queueing network with 4 systems. System 1 and 2 have single servers with infinite queues. System 3 is a loss system with 10 servers, and system 4 is also a loss system with 5 servers. All service times are exponentially distributed with mean $1/\mu_i$ for system i . New jobs arrive at system 1 according to a Poisson process with mean λ . A job that has been served in system 2 immediately returns to this system with probability α . A job that has left system 2, proceeds to system 3 with probability β and to system 4 with probability $1-\beta$. Let $\lambda=10$, $\mu_1=20$, $\mu_2=40$, $\mu_3=3$, $\mu_4=1$, $\alpha=0.6$, and $\beta=0.75$.

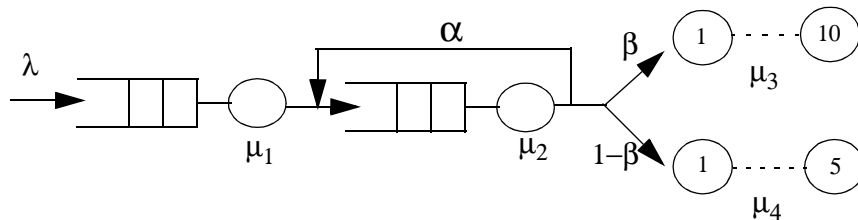


Figure 6.14. Queueing network in exercise 9.

Determine:

- The average number of jobs in each system.
- The average number of blocked jobs per time unit.
- The average total time a random job spends in the system.
- The average total time a job that is not blocked spends in the system.

10. The queueing network in figure 6.15 represents a computer system. The network consists of three systems (1, 2, and 3). System 1 and 2 have single servers with infinite queues, whereas system 3 is a loss system with two servers. New jobs arrive according to a Poisson process with mean λ_{in} . A job that has been served in system 1 proceeds to system 2 with probability α , and to system 3 with probability $1-\alpha$. All service times are exponentially distributed with mean $1/\mu_i$ for system i .

Let $\alpha=0.4$, $\lambda_{in}=0.75$, $\mu_1=1.5$, $\mu_2=2$, and $\mu_3=3$.

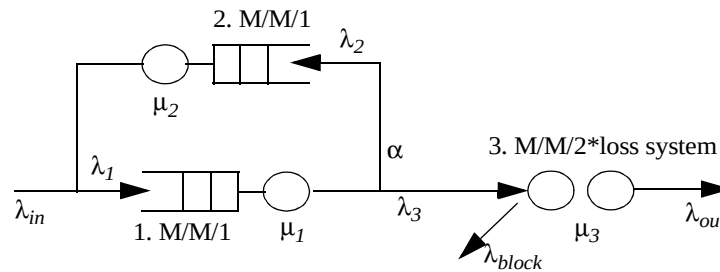


Figure 6.15. A computer system

Determine:

- λ_1 , λ_2 , λ_3 , λ_{out} , and λ_{block} .
- The average number of jobs in the network.
- The average total time a job that is not blocked spends in the network.

11. Assume that we will start a web store that sells computer items. In order to dimension the site accurately, we first make a mathematical model. We model the site as the queueing network in Figure 6.16. The site consists of three queueing systems, A-C. System A represents the web server that processes the HTTP requests that the customers send to the site. System B represents the authentication server that is needed when a customer wants to buy something. Systems A and B are both M/M/1-systems with infinite queues. System C represents a support service that is implemented in the site. System C is an M/M/1-system with a finite queue with 2 places..

New customers arrive according to a Poisson process with average λ customers per second. A customer that visits the store will send one or more HTTP requests in order to receive information about the items sold by the store. When a customer has received a reply, that is when the request has been served in system A, the customer will send a new HTTP request with probability α . Before the request is sent, the customer has a think time. This think time is modelled as a M/M/ ∞ -system (system D), that is a queueing system with infinite number of servers.

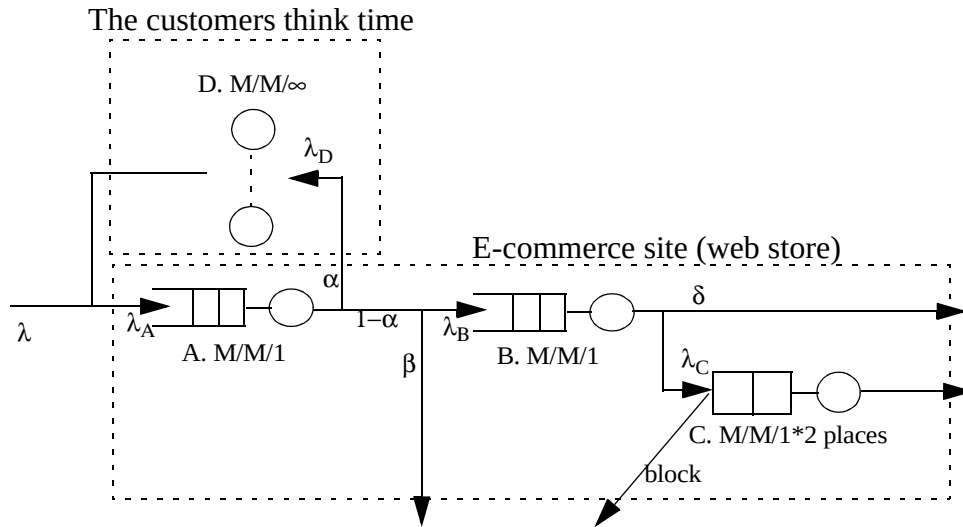


Figure 6.16. A model of an E-commerce site

A customer that has been served in system A, will not send any more requests with probability $1-\alpha$. These customers will leave the site without buying anything with probability β . With probability $1-\beta$, they will buy something and must thereby be served in system B.

A customer that has bought something (has been served in system B), will leave the site with probability δ . With probability $1-\delta$, the customer wants to use the support service (system C). If system C is full when a customer arrives, this customer leaves the system.

The average service time in system k is $1/\mu_k$ seconds.

Use the following values: $\lambda = 10 \text{ s}^{-1}$, $\mu_A = 60 \text{ s}^{-1}$, $\mu_B = 5 \text{ s}^{-1}$, $\mu_C = 0,1 \text{ s}^{-1}$, $\mu_D = 0,1 \text{ s}^{-1}$, $\alpha=0.8$, $\beta=0.9$ och $\delta=0.7$.

Note that the departure process from an M/M/∞-system is Poissonian.

Determine the following:

- λ_k for $k=A,B,C$ och D, that is the arrival rate for each system.
- The average number of customers that have their think time at a random time instant, that is the average number of customers in system D.
- The total average service time in the site for a customer. Note that system D does not belong to the site.

12. A multiprocessor system with M processors could be modelled as the queueing network in figure 6.17. The network consists of M single server systems with infinite queues. All service times are exponentially distributed with mean $1/\mu$. New jobs arrive according to a Poisson process with mean λ . A job is served by system i with probability p_i . A job that has been served leaves the network with probability $1-\alpha$ ($0 < \alpha < 1$). With probability α the job must be served again.

Let $p_i = 1/M$, $i = 1, \dots, M$.

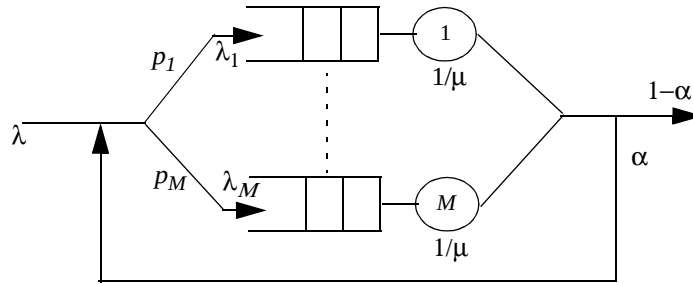


Figure 6.17. A multiprocessor system.

Determine:

- λ_i for $i = 1, \dots, M$
- The average total time a job spends in the network.
- The probability that a job *never* is served by system 1.

Chapter 7

Probability theory

This chapter contains a short review of some important issues in probability theory.

7.1 Basic principles

The mathematical basis for probability has three important components:

1. A *sample space*, Ω , which represents all possible outcomes of an experiment.
2. A collection of *events*, E , which are sets of points in the sample space.
3. A mapping, P , from events to real numbers. P is a probability mapping if it satisfies the following three conditions:

- For all events A , $A \in E$, the mapping P is defined, and satisfies
$$0 \leq P(A) \leq 1$$
- $P(\Omega)=1$.
- If A and B are mutually exclusive, that is they contain no sample points in common, then

$$P(A \cup B) = P(A) + P(B)$$

The probability of the union of two events, A and B , which are *not* mutually exclusive is given by

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \quad (7.5)$$

The probability of the complement of event A , denoted by $\neg A$, is

$$P(\neg A) = 1 - P(A) \quad (7.6)$$

An event that cannot occur has probability 0.

If we know that some event has occurred, then this affects the probability that other events have occurred. The *conditional probability* of A occurring given that B has occurred is

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (7.7)$$

Obviously if A and B are mutually exclusive $P(A|B) = 0$. If B must have occurred in order for A to have occurred, then $P(A \cap B) = P(A)$.

Two events are *independent* if knowledge of the occurrence of one of them tells us nothing about the probability of the other, that is:

$$P(A|B) = P(A)$$

An equivalent way of expressing the independence of two events is:

$$P(A \cap B) = P(A) \cdot P(B)$$

It is often the case that one can calculate the probability of an event of interest, for example A , conditioned on one of the events B_i having occurred. If the complete set of B_i exhausts the sample space, or at least exhausts that subset of it in which A can occur, then

$$P(A) = \sum_i P(A|B_i)P(B_i) \quad (7.8)$$

This is known as the *theorem of total probability*.

7.2 Random variables

A *random variable* is a mapping from the sample space, Ω , to the real numbers. Usually we associate probabilities with random variables in the natural manner. That is, the probability that a random variable, X , is equal to m is just the probability of the union of all events that lead to $X=m$.

Discrete random variables

Discrete random variables take on values which are usually from the natural numbers, although any finite or countable set of values will do. The probability that a discrete random variable, X , has value x_i is given by the *probability distribution* p_i , that is

$$p_i = P(X = x_i) \quad \forall i$$

The *distribution function* for X is given by

$$F_X(x_i) = P(X \leq x_i) = \sum_{k=0}^i p_k \quad \forall i$$

The *expectation* of X is given by

$$E[X] = \sum_{\forall i} x_i \cdot p_i \quad (7.9)$$

Loosely, it might be termed the average value that the random variable will have.

Further information can be given in terms of the *moments* of the distribution. The *2nd moment* is given by

$$E[X^2] = \sum_{\forall i} x_i^2 \cdot p_i \quad (7.10)$$

The *variance* of X is given by

$$V[X] = E[X^2] - E[X]^2 \quad (7.11)$$

The variance measures how far from the expected value, the values of the random variable are spread. The actual value of the variance depends on the magnitude of the values of the random variables. In order to make the measure independent of scale, the *squared coefficient of variance* is often used. It is defined as follows:

$$C^2 = \frac{V[X]}{E[X]^2} \quad (7.12)$$

Two discrete random variables, X and Y , are independent if

$$P(X = m, Y = n) = P(X = m) \cdot P(Y = n)$$

that is if the joint probability distribution is equal to the product of the individual probability distributions.

Example 7.1: A Bernoulli trial is an experiment which has two possible outcomes. Arbitrarily we can call them success, S , and failure, F . Conventionally, success occurs with probability p and failure with probability $q=1-p$, on any trial independently of any other trial.

Consider sequences of n Bernoulli trials. Define the random variable X to be the number of successes after n trials. Clearly there are 2^n different sequences. $\binom{n}{k}$ of them have k successes. Each of the sequences with k successes occurs with the probability $p^k q^{n-k}$. Therefore, the probability that $X=k$, p_k , is given by

$$p_k = P(X = k) = \binom{n}{k} p^k q^{n-k}$$

We can see that the set $\{p_k\}$ is a probability distribution since

$$\sum_{k=0}^n p_k = \sum_{k=0}^n \binom{n}{k} p^k q^{n-k} = (p+q)^n = 1$$

This distribution is called the *binomial distribution*.

Example 7.2: The *Poisson distribution* is given by

$$p_k = \frac{m^k}{k!} \cdot e^{-m} \quad k \geq 0$$

where m is a positive constant. As we will see in later chapters, the Poisson distribution is one of the classical distributions when modelling arrivals to telecommunication systems.

Example 7.3: When sending a packet over a data link, the packet either arrives correct at the receiver or incorrect if there is one or more bit errors. The packet must be retransmitted until it is received correctly. Assume that the probability that probability of a packet error is p .

Define the random variable X to be the number of *retransmissions* needed before the packet arrives without bit errors at the receiver. The probability distribution for X , p_k , is given by

$$p_k = P(X = k) = p^k (1-p) \quad k \geq 0$$

This distribution is known as the *geometrical distribution*.

If we instead define X to be the total number of transmission needed, the probability distribution is given by

$$p_k = P(X = k) = p^{k-1} (1-p) \quad k \geq 1$$

which is known as the *modified geometrical distribution*.

Continuous random variables

For some sample spaces, discrete values of random variables are not appropriate. For example, if we are interested in the number of requests that arrives at the web site each hour then a discrete random variable is ideal. Alternatively, the time it takes to process a request can take almost any non-negative value, so a continuous random variable is indicated.

Since a continuous random variable have an infinite number of sample points, we cannot assign a probability to each sample point in a meaningful way. Therefore, we define the probabilities in terms of the *distribution function*, which is a function giving the probability that the random variable is less or equal than its argument. If the distribution function of a random variable, X , is $F(x)$, then

$$P(X \leq x) = F(x)$$

In order to be a probability distribution function, a function must satisfy four conditions:

1. $F(x_1) \leq F(x_2)$ if $x_1 \leq x_2$.
2. $F(x) \rightarrow 0$ as $x \rightarrow -\infty$.
3. $F(x) \rightarrow 1$ as $x \rightarrow \infty$.
4. $F(x)$ is right continuous.

The derivative of the distribution function is known as the *density function*, where it exists

$$f(x) = \frac{dF(x)}{dx}$$

It can be convenient to regard the infinitesimal quantity $f(x)dx$ as the probability that the value of X lies in the interval $(x, x + dx)$, that is

$$P(x < X \leq x + dx) = F(x + dx) - F(x) = \frac{F(x + dx) - F(x)}{dx} \cdot dx = \frac{d}{dx}F(x) \cdot dx = f(x)dx$$

This simplifies many arguments, without compromising their validity.

If $F(x)$ is continuous with derivative $f(x)$ then the expectation of X is given by

$$E[X] = \int_{-\infty}^{\infty} xf(x)dx$$

The second moment of X is given by

$$E[X^2] = \int_{-\infty}^{\infty} x^2 f(x)dx$$

The definitions of the variance and squared coefficient of variance of X are identical to those for a discrete random variable.

Two continuous random variables, X and Y , with density functions, $f_X(x)$ and $f_Y(y)$, are independent if

$$f_{XY}(x, y) = f_X(x) \cdot f_Y(y)$$

that is if their joint density function is equal to the product of the individual density functions.

Example 7.4: The most important continuous distribution for performance modelling of telecommunication systems is the *exponential distribution*, which has density function

$$f(x) = ae^{-ax} \quad x \geq 0 \quad (0 \text{ otherwise})$$

and distribution function $F(x) = 1 - e^{-ax}$. It is also known as the *negative exponential distribution*. The mean can easily be calculated:

$$E[X] = \int_0^{\infty} xf(x)dx = \int_0^{\infty} xae^{-ax} = \dots = \frac{1}{a}$$

Example 7.5: The time required to process an HTTP request in a web server can be modelled with a *uniform distribution*. A variable is uniformly distributed between a and b if its distribution function is

$$F(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x > b \end{cases}$$

The density function, $f(x)$, is identically 0, except in the range $a \leq x \leq b$, where it is the constant $1/(b-a)$

Example 7.6: The value of an *hyperexponential- r* random variable, X , is with probability p_i exponentially distributed with mean $1/a_i$ ($i=1,..,r$), where $a_i \neq a_j$ for $\forall i \neq j$. Usually $r=2$, which means that the density function for X is given by

$$f_X(x) = p_1 a_1 e^{-a_1 x} + (1-p_1) a_2 e^{-a_2 x}$$

The mean, variance, and squared coefficient of variance for X are given by

$$\begin{aligned} E[X] &= p_1/a_1 + (1-p_1)/a_2 \\ V[X] &= \frac{p_1(2-p_1)}{a_1^2} + \frac{(1-p_1)^2}{a_2^2} - \frac{2p_1(1-p_1)}{a_1 a_2} \\ C^2 &= \frac{2(p_1 a_2^2 + (1-p_1) a_1^2)}{(p_1 a_2 + (1-p_1) a_1)^2} - 1 \end{aligned}$$

One can rather easily show that $1 \leq C^2 < \infty$.

7.3 Z-transforms of probability distributions

Although we can often derive the full set of probability values for a discrete distribution, if the sample space is infinite, we have a rather unwieldy set of formulas to manipulate. A convenient tool for manipulating infinite series is the generating function.

If the values $p_0, p_1, \dots$ form a probability distribution, then the distribution's *probability generating function* is given by

$$P(z) = E[z^i] = \sum_{i=0}^{\infty} z^i p_i \quad (7.13)$$

The convergence of series is guaranteed for all z less than or equal to 1, since the probabilities must sum to 1. That property also ensures that $P(1)=1$. Unless otherwise stated, all generating functions used in this book are probability generating functions. The probability generating function is also called the *z-transform*.

The z-transform contains all the information that was in the individual point probabilities. In principle, we can extract the value of any probability using repeated differentiation, since

$$p_j = \frac{1}{j!} \cdot \frac{d^j P(z)}{dz^j}$$

More importantly, the moments of the distribution can also be calculated from the z-transform. For example, the expectation (or first moment) is given by

$$E[X] = \sum_{i=0}^{\infty} i p_i = \sum_{i=0}^{\infty} i p_i z^{i-1} \Big|_{z=1} = \frac{d}{dz} \sum_{i=0}^{\infty} p_i z^i \Big|_{z=1} = P'(1)$$

Similarly, the second moment is given by

$$\begin{aligned} E[X^2] &= \sum_{i=0}^{\infty} i^2 p_i = \sum_{i=0}^{\infty} i(i-1) p_i + \sum_{i=0}^{\infty} i p_i \\ &= \sum_{i=0}^{\infty} i(i-1) p_i z^{i-2} \Big|_{z=1} + \sum_{i=0}^{\infty} i p_i z^{i-1} \Big|_{z=1} = P''(1) + P'(1) \end{aligned}$$

It is intuitively obvious that there is one to one correspondence between functions and their z-transforms. If two z-transforms are identical, then the underlying distributions must also be identical.

Example 7.7: The z-transform for the geometrical distribution is given by

$$P(z) = \sum_{k=0}^{\infty} p^k (1-p) z^k = (1-p) \sum_{k=0}^{\infty} (pz)^k = (1-p) \cdot \frac{1}{1-pz} = \frac{1-p}{1-pz}$$

7.4 Laplace transforms of distribution functions

There are a number of transforms which can be used to manipulate the distribution function of continuous random variables. We shall only consider the *Laplace transform*, and restrict consideration to non-negative random variables. A non-negative random variable, X , with distribution function $F(x)$, has a Laplace transform denoted by $F^*(s)$,

$$F^*(s) = E[e^{-sX}] = \int_0^{\infty} e^{-sx} dF(x) = \int_0^{\infty} e^{-sx} f(x) dx \quad (7.14)$$

The moments of the distribution can be retrieved from the Laplace transform. The expectation of X is given by

$$E[X] = \int_0^{\infty} x f(x) dx = \int_0^{\infty} x e^{-sx} f(x) dx \bigg|_{s=0} = -\frac{d}{ds} \int_0^{\infty} e^{-sx} f(x) dx \bigg|_{s=0} = -F^{*'}(0)$$

Similarly, one can derive the second moment $E[X^2] = F^{*''}(0)$ and so on. The Laplace transform has a similar property to z-transforms. There is a one to one correspondence between distributions and their Laplace transforms.

Extraction of the distribution function, rather than its moments, from the transform is more difficult than extracting individual state probabilities from a z-transform. Inversion of arbitrary Laplace transforms is well defined and involves integrating round a contour in the complex plane. The simplest technique, however, is to try to identify similar Laplace transforms of a similar form in a table of transforms, and hence deduce the form of the distribution function.

Example 7.8: The Laplace transform for the exponential distribution is given by

$$F^*(s) = \int_0^{\infty} a e^{-at} e^{-st} dt = a \int_0^{\infty} e^{-(a+s)t} dt = a \cdot \frac{1}{a+s} = \frac{a}{a+s}$$

7.5 Sum of two random variables

Consider two random variables, X and Y . Also, consider the random variable Z , which is the sum of X and Y , that is $Z=X+Y$.

First, the expectation of a number of random variables is the sum of their individual expectations. This means that the expectation of Z is given by

$$E[Z] = E[X + Y] = E[X] + E[Y]$$

This is true even if X and Y are dependent in some way.

If X and Y are discrete *independent* random variables with $P(X=k)=p_k$ and $P(Y=k)=g_k$, then the probability distribution for Z can be calculated as

$$\begin{aligned} P(Z=i) &= P(X+Y=i) = \sum_{n=0}^{\infty} P(X+Y=i|Y=n) \cdot P(Y=n) \\ &= \sum_{n=0}^{\infty} P(X=i-n) \cdot P(Y=n) = \sum_{n=0}^{\infty} p_{i-n} g_n \end{aligned}$$

that is the convolution of the probability distributions for X and Y . However, instead of calculating the convolution of X and Y , one can use the z-transforms, $P_X(z)$ and $P_Y(z)$. The z-transform of Z , $P_Z(z)$, is given by

$$P_Z(z) = E[z^{X+Y}] = E[z^X \cdot z^Y] = E[z^X] \cdot E[z^Y] = P_X(z) \cdot P_Y(z)$$

since X and Y are independent.

If X and Y instead are two continuous independent random variables, with density functions, $f_X(t)$ and $f_Y(t)$, the density function for Z , $f_Z(t)$, is similarly given by

$$f_Z(t) = \int_{-\infty}^{\infty} f_X(t-s) \cdot f_Y(s) ds$$

that is the convolution of the density functions for X and Y . Also here the Laplace transform can be used. The Laplace transform of Z , $F_Z^*(s)$, is given by

$$F_Z^*(s) = E[e^{-sZ}] = E[e^{-s(X+Y)}] = E[e^{-sX} \cdot e^{-sY}] = E[e^{-sX}] \cdot E[e^{-sY}] = F_X^*(s) \cdot F_Y^*(s)$$

where $F_X^*(s)$ and $F_Y^*(s)$ are the Laplace transforms of X and Y respectively

Example 7.9: Assume that X and Y are two independent exponentially distributed random variables with the same parameter a . The Laplace transform for $Z=X+Y$ is given by

$$F_Z^*(s) = F_X^*(s) \cdot F_Y^*(s) = \frac{a}{a+s} \cdot \frac{a}{a+s} = \frac{a^2}{(a+s)^2}$$

Example 7.10: One distribution sometimes used in queueing theory is the *Erlangian- r* distribution. An Erlangian- r distributed random variable, X , with mean $1/a$ can be seen as the sum of r independent exponential distributed random variables, X_i ($i=1, \dots, r$), each with mean $1/ra$. This means that the Laplace transform for X is given by

$$F_X^*(s) = \prod_{i=1}^r F_{X_i}^*(s) = \left(\frac{ra}{ra+s} \right)^r$$

The density function is found by inversion of the Laplace transform:

$$f_X(t) = ra \cdot \frac{(rat)^{r-1}}{(r-1)!} e^{-rat}$$

7.6 Exercises

1. Let X and Y be two random variables, that can have the values 0 and 1. We know that:

$$\begin{aligned}P(X = 0, Y = 0) &= 1/2 & P(X = 1, Y = 0) &= 1/8 \\P(X = 0, Y = 1) &= 1/4 & P(X = 1, Y = 1) &= 1/8\end{aligned}$$

Determine:

- (a) $P(X = 0)$
 - (b) $P(X = 1|Y = 0)$
 - (c) $P(Y = 0|X = 1)$
 - (d) $E[Y]$
 - (e) $E[XY]$
 - (f) $E[X|Y = 0]$
 - (g) $E[Y|X = 1]$
2. Calculate the mean and variance of the geometrical distribution.
3. Calculate the squared coefficient of variance, C^2 , for the
- (a) Exponential distribution with mean $1/\lambda$.
 - (b) Erlangian-r distribution with mean $1/\lambda$.
 - (c) What are the lower and upper limits of C^2 in (b)?
4. Assume that we have coin for which the probability is p that “head” comes up when the coin is thrown. Let us throw the coin an infinite number of times. Let x_n be a random variable for the number of throws necessary until “head” comes up for the n :th time. Determine $E[x_n]$.
5. Determine the probability distribution for the positive discrete random variable V when

$$\begin{aligned}P(V = k|X = x) &= \frac{(\lambda x)^k}{k!} \cdot e^{-\lambda x} \\P(X \leq x) &= 1 - e^{-\mu x}\end{aligned}$$

where X is a positive continuous random variable.

6. For a random variable N , that is Poissonian distributed with parameter m , determine the z-transform, the mean, and the variance.
7. Assume that X and Y are independent Poissonian distributed random variables, with parameters λ_1 and λ_2 respectively. Determine the probability distribution for $Z=X+Y$.

Chapter 8

Stochastic processes

This chapter contains an introduction to stochastic processes, and in particular Markov processes. Informally, a stochastic process is a random process which is changing in time. For example, the number of passengers waiting in a bus queue is a stochastic process. It increases when new passengers join the queue, and may decrease when a bus arrives.

Formally, a stochastic process is defined as a set of random variables $\{X(t) | t \in T\}$, for some index set T . The set T is usually interpreted as a set of time instants. When T is countable, we have a discrete-time stochastic process. When the elements of T are a continuous set of real numbers, the process is a continuous-time stochastic process. $X(t)$ is called the state of the process at time t . A stochastic process is said to be stationary if the distribution of $X(t + s) - X(t)$ is independent of t . We will only analyse stationary stochastic processes.

8.1 The Poisson process

A Poisson process is one of the simplest interesting stochastic processes. It can be defined in a number of ways, which are equivalent. The following axiomatic definition seems most appropriate to our needs. A Poisson process is a so-called counting process that records the number of events which occur in an interval. A Poisson process with parameter λ satisfies the following properties:

-
1. $P(\text{a single event occurs in an interval of duration } \Delta t) = \lambda \Delta t + o(\Delta t)$, independently of when the last event occurred, and of all other intervals.
 2. $P(\text{more than one event will occur in an interval of duration } \Delta t) = o(\Delta t)$.

From the axioms of probability, the probability that no events occur in the interval of length Δt is $1 - \lambda \Delta t + o(\Delta t)$.

From this definition, we can derive the probability of n events in an interval of length t , denoted $P_n(t)$. Consider the interval divided into m equal subintervals ($m > n$). Either the events occur so that n of the intervals have a single event, and $m-n$ have no events, or there are some intervals with more than one event. Denoting the first possibility with $P_n^{(a)}(t)$ and the second by $P_n^{(b)}(t)$, since these two possibilities are mutually exclusive, we have:

$$P_n(t) = P_n^{(a)}(t) + P_n^{(b)}(t)$$

Consider $P_n^{(a)}(t)$, and choose the n subintervals in advance. Then the probability of this occurrence is:

$$\begin{aligned} & P(\text{given } n \text{ intervals have 1 event each and remaining } m-n \text{ have none}) = \\ & = \left(\lambda \frac{t}{m}\right)^n \cdot \left(1 - \lambda \frac{t}{m} + o\left(\frac{t}{m}\right)\right)^{m-n} \end{aligned}$$

The choice of the n intervals can be made in $\binom{m}{n}$ distinct ways. Hence

$$\begin{aligned} P_n^{(a)}(t) &= \frac{m!}{(m-n)!n!} \left(\lambda \frac{t}{m}\right)^n \left(1 - \lambda \frac{t}{m} + o\left(\frac{t}{m}\right)\right)^{m-n} \\ &= \frac{(\lambda t)^n}{n!} \cdot \frac{m!}{(m-n)!m^n} \cdot \left(1 - \lambda \frac{t}{m} + o\left(\frac{t}{m}\right)\right)^{-n} \cdot \left(1 - \lambda \frac{t}{m}\right)^m \cdot \left(1 + o\left(\frac{t}{m}\right)\right)^m \end{aligned}$$

Now we take the limit as $m \rightarrow \infty$ and find

$$P_n^{(a)}(t) = \frac{(\lambda t)^n}{n!} \cdot 1 \cdot 1 \cdot e^{-\lambda t} \cdot 1 = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

Recalling that $P_n(t) = P_n^{(a)}(t) + P_n^{(b)}(t)$ and that since $P_n(t)$ is a probability distribution,

$$\sum_{n=0}^{\infty} P_n(t) = \sum_{n=0}^{\infty} P_n^{(a)}(t) + \sum_{n=0}^{\infty} P_n^{(b)}(t) = e^{-\lambda t} \sum_{n=0}^{\infty} \frac{(\lambda t)^n}{n!} + \sum_{n=0}^{\infty} P_n^{(b)}(t) = 1 + \sum_{n=0}^{\infty} P_n^{(b)}(t)$$

Hence $P_n^{(b)}(t) = 0$, since no probability can be negative, and

$$P_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t} \tag{8.15}$$

which is the Poisson distribution with parameter λt .

If the number of events in an interval t is denoted $N(t)$, the average number of events in this interval is given by

$$E[N(t)] = \sum_{k=1}^{\infty} k P_k(t) = \sum_{k=1}^{\infty} k \frac{(\lambda t)^k}{k!} e^{-\lambda t} = \lambda t \cdot \sum_{k=1}^{\infty} \frac{(\lambda t)^{k-1}}{(k-1)!} e^{-\lambda t} = \lambda t$$

The distribution of the time interval between successive events can also be calculated. Let $A(t)$ be the distribution function of τ , a random variable representing the time between events. This means that $A(t) = P(\tau \leq t)$. Then $P(\tau > t)$ is the probability that no events occur in an interval of length t . This means that

$$P(\tau > t) = 1 - A(t) = P_0(t) = e^{-\lambda t}$$

Hence

$$A(t) = 1 - e^{-\lambda t} \quad (8.16)$$

This means that the interevent time distribution is an exponential distribution.

The Poisson and exponential distributions have a special place in queuing theory and performance modelling because of their so-called *memoryless property*. It is clear from the definition of the Poisson process that the number of events in distinct intervals is dependent only of the interval lengths, and not of the behaviour of the event process in preceding or subsequent intervals. The memoryless property of the exponential distribution is so called because the time to the next event is independent of when the last event occurred. Suppose that the last event was at time 0, what is the probability that the next event will be after $t+s$, given that time t has elapsed since the last event, and no events have occurred?

$$P(\tau > t + s | \tau > t) = \frac{P(\tau > t + s \cap \tau > t)}{P(\tau > t)} = \frac{e^{-\lambda(s+t)}}{e^{-\lambda t}} = e^{-\lambda s}$$

Thus the knowledge that there have been no events from a Poisson stream for a period of time does not alter the distribution of time until the next event. Conversely, the knowledge that there has been a large number of events in the recent past does not change the distribution of time until the next event.

8.2 Markov chains

A *Markov process* is a stochastic process in which the distribution at any time in the future depends only on the current state of the process, and *not* on how that state was reached. Formally, a stochastic process, $X(t)$, is a Markov process if

$$P(X(t) = x | X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \dots, X(t_0) = x_0) = P(X(t) = x | X(t_n) = x_n) \\ \text{for } t_0 < t_1 < \dots < t_n < t \quad (8.17)$$

The set of values assumed by $X(t)$, the state space, can be discrete or continuous. The elements of the index set, t , can also be discrete or continuous. It is possible for these condi-

tional probabilities to depend on time, but *time-homogeneous* processes, in which the conditional probabilities are independent of time, are much more common in practical modelling situations.

When the state space, $X(t)$, is discrete, the Markov process is called a *Markov chain*. In a Markov chain, the system can occupy a finite or countable infinite number of states $e_1, e_2, \dots, e_j, \dots$ such that the future evolution of the process, once it is in a given state, depends only on the present state and not on how it arrived at that state. This is the *Markov, or memoryless, property*.

8.3 Discrete-time Markov chains

In a discrete-time Markov chain time is discretised as $t = 0, 1, \dots, n, n+1, \dots$. The necessary condition that the process at time n , X_n , must satisfy can be written as:

$$P(X_{n+1} = m | X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_0 = x_0) = P(X_{n+1} = m | X_n = x_n) \quad (8.18)$$

Again, these probabilities can depend on the value of n .

Transition probabilities

Given that a chain is in state i at time n , then the one-step transition probabilities, $p_{ij}(n)$, are given by:

$$p_{ij}(n) = P(X_{n+1} = j | X_n = i)$$

We will only deal with chains with stationary transition probabilities. For these homogeneous Markov chains, the dependence on n will be omitted and we write the one-step transition probabilities as p_{ij} .

Notice that the one-step transition probabilities give all the information we need to predict the (probabilistic) behaviour of the process at all future times. For example, if the process is in state i at time n , what is the probability that it will be in state k at time $n+2$? Denote this two-step probability by $p_{ik}^{(2)}$, and observe that to reach state k in two steps, the process must make a transition to some state j in one step with probability p_{ij} (j may equal i or k), and then make the transition to state k , with probability p_{jk} . The two-step transition probability is given by summing over all possible intermediate states:

$$p_{ik}^{(2)} = \sum_j p_{ij} \cdot p_{jk}$$

The one-step probabilities can be written in the form of a matrix $\mathbf{P}=(p_{ij})$. This *transition matrix* is a stochastic matrix, that is, the sum of the elements in each row is 1. Writing $\mathbf{1}$ for the column vector equal to one in every component:

$$\mathbf{P} \cdot \mathbf{1} = \mathbf{1}$$

Premultiplying each side of the relationship by $\mathbf{P}$:

$$\mathbf{P} \cdot \mathbf{P} \cdot \mathbf{1} = \mathbf{P}^2 \cdot \mathbf{1} = \mathbf{P} \cdot \mathbf{1} = \mathbf{1}$$

it is seen that any power of a stochastic matrix is itself stochastic. This property holds, even if the state space is countable infinite. Writing the state probabilities at time n as a row vector, $\mathbf{p}(n) = (p_0(n), p_1(n), \dots)$. The evolution of the process can then be expressed as:

$$\mathbf{p}(n+1) = \mathbf{p}(n) \cdot \mathbf{P}$$

This relationship can be iterated, and we find that

$$\mathbf{p}(n) = \mathbf{p}(0) \cdot \mathbf{P}^n$$

so that the initial probability distribution, $\mathbf{p}(0)$, and the transition matrix, $\mathbf{P}$, are sufficient to calculate the state probabilities at all times in the future. The two-step transition probabilities can be expressed in matrix form as $\mathbf{P}^2$. In general, the n -step transition matrix is given by $\mathbf{P}^n$.

State classification

A state j is said to be *accessible* from a state i , if $p_{ij}^{(n)} > 0$, for some $n > 0$. If state i is accessible from state j , and state j is accessible from state i , then state i and j are said to *communicate*. If all states in a Markov chain communicate, the chain is said to be *irreducible*.

A state i , is said to have period, d , if $p_{ii}^{(n)} = 0$ for all n that are not divisible by d , and d is the largest integer with this property. A state with period 1 is said to be *aperiodic*. All states that communicate have the same period. Therefore, an irreducible chain can be referred to as periodic or aperiodic according to the classification of its states.

A state is said to be *recurrent*, if it will be entered again with probability 1. If the probability of return to the state is less than 1, then the state is *transient*. If the mean number of steps until the state is returned to is finite, the state is said to be *positive recurrent*.

An irreducible, aperiodic Markov chain of which all states are positive-recurrent is an *ergodic* Markov chain. If the chain is ergodic, every state in the chain can be reached from every other state.

Stationary probability distribution

It can be proved that an ergodic Markov chain reaches a unique stationary probability distribution that is independent of the starting state. In this case, the stationary distribution $\mathbf{p} = (p_1, p_2, \dots)$ satisfies

$$\mathbf{p} = \mathbf{p} \cdot \mathbf{P}$$

Example 8.1: Consider a discrete-time Markov chain with

$$\mathbf{P} = \begin{bmatrix} 2/5 & 3/5 & 0 \\ 1/5 & 2/5 & 2/5 \\ 0 & 1/5 & 4/5 \end{bmatrix}$$

The chain has 3 states. For all Markov chains, one can draw a so-called *state diagram* showing the states and the transition probabilities. The state diagram for this Markov chain is shown in Figure 8.1.

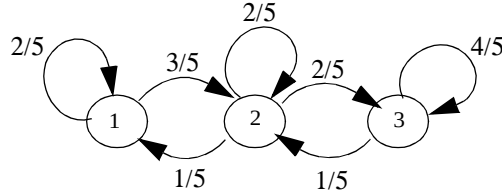


Figure 8.1. A state diagram for the Markov chain in Example 8.1.

Since the chain is ergodic (all states are accessible and positive recurrent), we can calculate the stationary distribution, $\mathbf{p}$, by setting $\mathbf{p} = \mathbf{pP}$:

$$(p_1, p_2, p_3) = (p_1, p_2, p_3) \cdot \begin{bmatrix} 2/5 & 3/5 & 0 \\ 1/5 & 2/5 & 2/5 \\ 0 & 1/5 & 4/5 \end{bmatrix}$$

which gives the following equations to solve:

$$\begin{aligned} p_1 &= \frac{2}{5} \cdot p_1 + \frac{1}{5} \cdot p_2 \\ p_2 &= \frac{3}{5} \cdot p_1 + \frac{2}{5} \cdot p_2 + \frac{1}{5} \cdot p_3 \\ p_3 &= \frac{2}{5} \cdot p_2 + \frac{4}{5} \cdot p_3 \end{aligned}$$

The solution to these equations is $\mathbf{p} = (p_1, 3p_1, 6p_1)$. In order to find the unique solution, we must use the axiom of probability

$$\sum_k p_k = 1$$

since the system have to be in one of the states at all times. This is usually referred to as the *normalisation condition*. By using the normalisation condition the stationary probability distribution is found:

$$\sum_{k=1}^3 p_k = 1 \Rightarrow p_1 + 3p_1 + 6p_1 = 1 \Rightarrow p_1 = \frac{1}{10}, p_2 = \frac{3}{10}, p_3 = \frac{6}{10}$$

8.4 Continuous-time Markov chains

In a continuous-time Markov chain, the time index t varies continuously. If the state of the system at time t , $X(t)$, is i , then

$$P(X(t + \Delta t) = j | X(t) = i) = q_{ij}\Delta t + o(\Delta t) \quad i \neq j$$

where all q_{ij} are known as the *transition densities* of the process. The Markov chain can, therefore, be characterized by its so-called *infinitesimal generator matrix* $\mathbf{Q}=(q_{ij})$. The state classification is the same as for discrete-time Markov chains. We will assume that the chain is ergodic.

If we denote the probability that the system is in state k at time t by $p_k(t)$, then the distribution a short time Δt later is given by:

$$p_k(t + \Delta t) = p_k(t) \left(1 - \sum_{j \neq k} q_{kj} \Delta t \right) + \sum_{i \neq k} q_{ik} \Delta t \cdot p_i(t) + o(\Delta t)$$

Defining

$$q_{ii} = -\sum_{j \neq i} q_{ij}$$

the expression above can be rewritten as

$$p_k(t + \Delta t) = p_k(t) + \sum_{i=1}^n q_{ik} \Delta t \cdot p_i(t) + o(\Delta t) \quad (8.19)$$

where n is the number of states. By rearranging this expression and taking the limits as $\Delta t \rightarrow 0$ gives

$$\frac{dp_k}{dt} = \sum_{i=1}^n q_{ik} p_i(t)$$

Denoting the row vector $(p_1(t), p_2(t), \dots, p_n(t))$ by $\mathbf{p}(t)$ these equations can be expressed in matrix notation as

$$\frac{d\mathbf{p}(t)}{dt} = \mathbf{p}(t) \mathbf{Q}$$

Steady-state is reached when $\frac{d\mathbf{p}(t)}{dt} = 0$, which means that

$$\mathbf{p}\mathbf{Q} = 0$$

where $\mathbf{p}$ is the steady-state distribution vector. Although this presentation has been given in terms of a finite state space, infinite state space processes can also be expressed in terms of infinite $\mathbf{Q}$ -matrices. These are well defined so long as the infinite sums that arise in calculating vector products are well defined.

Example 8.2: Assume that we have modelled a computer system as a continuous-time Markov chain, with

$$\mathbf{Q} = \begin{bmatrix} -6 & 6 & 0 \\ 1 & -5 & 4 \\ 0 & 1 & -1 \end{bmatrix}$$

The states are numbered 0, 1, and 2 where state k means that there are k jobs in the system.

As with discrete-time Markov chains, we can draw a state-diagram that shows the states and the *transition intensities*, q_{ij} , see figure 8.2. However, here we only draw the intensities for *moving* to another state.

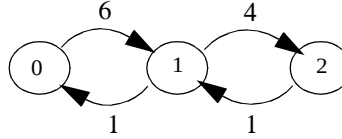


Figure 8.2. A state diagram for the Markov chain in Example 8.2.

Since the chain is ergodic, we can calculate the stationary probability distribution, $\mathbf{p}=(p_0, p_1, p_2)$, by setting $\mathbf{pQ}=0$:

$$\mathbf{pQ} = 0 \Rightarrow \begin{cases} -6p_0 + p_1 = 0 \\ 6p_0 - 5p_1 + p_2 = 0 \\ 4p_1 - p_2 = 0 \end{cases}$$

which has the solution $\mathbf{p}=(p_0, 6p_0, 24p_0)$. The stationary distribution is then found by using the normalisation condition:

$$\sum_{k=0}^2 p_k = 1 \Rightarrow p_0 + 6p_0 + 24p_0 = 1 \Rightarrow p_0 = \frac{1}{31}, p_1 = \frac{6}{31}, p_2 = \frac{24}{31}$$

Also, If we define the random variable N to be the number of jobs in the system we can calculate the expectation (or mean value) of N :

$$E[N] = \sum_{k=0}^2 k p_k = 0p_0 + 1p_1 + 2p_2 = \frac{54}{31} \approx 1,74$$

The second moment of N is:

$$E[N^2] = \sum_{k=0}^2 k^2 p_k = 0p_0 + 1p_1 + 4p_2 \approx 3,29$$

which means that the variance of N is

$$V[N] = E[N^2] - E[N]^2 \approx 0,26$$

Finding balance equations from the steady state diagram

When the Markov chain has an infinite number of states, the Q -matrix becomes impractical. Instead, the global balance equations can easily be derived from the steady state dia-

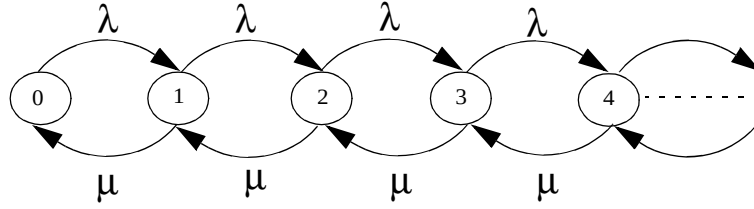


Figure 8.3. Steady state diagram.

gram. Figure 8.3 shows the steady state diagram for a system with an infinite number of states. For state k , the following is given:

$$q_{k, k+1} = \lambda \quad k \geq 0$$

$$q_{k, k-1} = \mu \quad k \geq 0$$

When the system reaches equilibrium the net “flow” across any closed curve, or cut, must total zero. This “flow” of probability can be measured as the product of the label on the arcs and the probabilities of their starting states. Using cuts on the balance diagram is a powerful graphical technique for deriving the balance equations.

Drawing a cut (or a circle) surrounding each state individually leads to the following balance equations:

$$\lambda p_0 = \mu p_1$$

$$(\lambda + \mu)p_1 = \lambda p_0 + \mu p_2$$

$$(\lambda + \mu)p_2 = \lambda p_1 + \mu p_3$$

This method is also called the “*flow-in-flow-out-method*”.

However if we instead use a cut which surrounds all the states, $0, 1, 2, \dots, k$, we derive the following balance equations:

$$\lambda p_0 = \mu p_1$$

$$\lambda p_1 = \mu p_2$$

$$\lambda p_2 = \mu p_3$$

This method is also called the “*cut-method*”.

Examination of the steady state diagram often leads to equivalent formulations of the balance equations which are simpler to solve.

8.5 Exercises

1. Consider a discrete-time Markov chain with

$$\mathbf{P} = \begin{bmatrix} 1/2 & 1/2 \\ 3/4 & 1/4 \end{bmatrix}$$

- (a) Draw the state diagram.
 - (b) Determine the stationary probability distribution $\mathbf{p}=(p_1, p_2)$.
 - (c) Assume that $\mathbf{p}(0)=(1,0)$. Determine $\mathbf{p}(1)$, $\mathbf{p}(2)$, $\mathbf{p}(3)$.
2. Assume that we have developed a model of a computer system. Using the model, the system can be described by the following $\mathbf{Q}$ -matrix:

$$\mathbf{Q} = \begin{pmatrix} -3 & 3 & 0 \\ 2 & -4 & 2 \\ 0 & 2 & -2 \end{pmatrix}$$

The states are numbered 0, 1 och 2, where state k corresponds to k jobs in the system.

- (a) Draw the state diagram for this system.
 - (b) Determine the stationary probability distribution.
 - (c) Determine the z-transform for the stationary probability distribution.
 - (d) Calculate the mean and variance for the number of jobs in the system.
3. Assume a system with four states, 1..4, and with the following $\mathbf{Q}$ -matrix:

$$\mathbf{Q} = \begin{pmatrix} -\lambda & \lambda & 0 & 0 \\ 0 & -\lambda & \lambda & 0 \\ 0 & 0 & -\lambda & \lambda \\ 0 & 0 & \mu & -\mu \end{pmatrix}$$

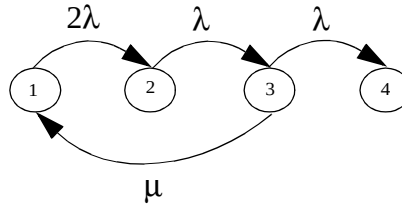
- (a) Draw the state diagram for this system.
- (b) Determine $p_1(t)$ and $p_2(t)$, that is the transient state probabilities for the states 1 and 2.
- (c) Determine the stationary probability distribution.

-
4. Assume that a computer system can be modelled as a continuous-time Markov chain in which state k corresponds to k jobs in the system. For state k , the following is given:

$$q_{k, k+1} = \frac{\alpha}{k+1} \quad k \geq 0$$

$$q_{k, k-1} = \beta \quad k \geq 1$$

- (a) Draw the state diagram.
(b) Determine the stationary probability distribution.
(c) Calculate the average number of jobs in the system.
5. Assume a system that can be modelled as the following continuous-time Markov chain:



Determine the following:

- (a) The stationary probability distribution.
(b) The total transition intensity for state 3.
(c) The distribution for the time spent in state 3 before leaving the state.
(d) The average times spent in state 1, 2, and 3 before leaving them.
(e) Assume that the system is in state 3. Determine $P(\text{transition from 3 to 4})$ and $P(\text{transition from 3 to 1})$.
(f) The average time it takes for the system to move to state 4 if it is started in state 1.

References

- [1] Jerry Banks, John S. Carson, and Barry L. Nelson, “Discrete-event Simulation”, Prentice Hall, 1996.
- [2] Peter J.B. King, “Computer and Communication Systems Performance Modelling”, Prentice Hall International Series in Computer Science, 1990.
- [3] Ulf Körner, “Köteori och tillförlitlighetsteori”, Studentlitteratur, 2002.
- [4] Averill M. Law and W. David Kelton, “Simulation Modeling and Analysis”, McGraw-Hill, 2000.
- [5] Alberto Leon-Garcia and Indra Widjaja, “Communication Networks: Fundamental Concepts and Key Architectures”, MacGraw-Hill Series on Computer Science, 2000.
- [6] Athanasios Papoulis and S. Unnikrishna Pillai, “Probability, Random Variables and Stochastic Processes”, Mc Graw Hill, 2002.
