Network Performance Analysis

Guido Sanguinetti (based on notes by Neil Lawrence)

Autumn Term 2007-2008

Schedule

Lectures will take place in the following locations:

Mondays 15:10 - 17:00 pm (St Georges LT1).

Tuesdays 16:10 pm - 17:00 pm (St Georges LT1).

Lab Classes In the Lewin Lab, Regent Court G12 (Monday Weeks 3 and 8).

Week 1

Monday	Lectures 1 & 2	Networks and Probability Review
Tuesday	Lecture 3	Poisson Processes

Week 2

Monday	Lecture 4 & 5	Birth Death Processes, $M/M/1$ queues
Tuesday	Lecture 6	Little's Formula

Week 3

Monday	Lab Class 1	Discrete Simulation of the $M/M/1$ Queue
Tuesday	Tutorial 1	Probability and Poisson Processes

Week 4

Monday	Lectures 7 & 8	Other Markov Queues and Erlang Delay
Tuesday	Lab Review 1	Review of Lab Class 1

Week 5

Monday	Lectures 9 & 10	Erlang Loss and Markov Queues Review
Tuesday	Tutorial 2	$M/M/1$ and Little's Formula

Week 6

Monday	Lectures 11 & 12	$M/G/1$ and $M/G/1$ with Vacations
Tuesday	Lecture 13	$M/G/1$ contd

Week 7

Monday	Lectures 14 & 15	Networks of Queues, Jackson's Theorem
Tuesday	Tutorial 3	Erlang Delay and Loss, $M/G/1$

Week 8

Monday	Lab Class 2	Simulation of Queues in NS2
Tuesday	Lecture 16	ALOHA

Week 9

Tuesday	Lab Review 2	Review of NS2 Lab Class

Week 10

Exam Practice Practice 2004-2005 Exam

Week 12

Monday	Exam Review	Review of 2004-2005 Exam

Evaluation

This course is 100% exam.

Reading List

The course is based primarily on Bertsekas and Gallager [1992]. The relevant parts of the book are as follows.

Chapter 1 For background reading and review.

Chapter 3 Section 3.1 – 3.5 For queueing models.

Chapter 4 In less detail: Section 4.1 – 4.4, Section 4.5.2, 4.5.3.

For additional background reading you may wish to consult some of the books listed below, Tanenbaum [2003] is the standard text for Introductory courses, but has little coverage on queueing theory. Chan [2000] is an alternative text for details on queues.

These Notes

These notes were written by Neil Lawrence to accompany Bertsekas and Gallager [1992], covering in extra detail areas of importance for the course and adding material where it is lacking in the course text (for example on simulation).

The first three chapters provide background material on networks and probabilities. This material is not a part of the syllabus as such but it provides a review of existing knowledge which we will build upon. The material in the course syllabus itself starts in Chapter 4 on the Poisson Process.

Please inform me of any errors in these notes using the form provided on the course website.

Chapter 1

Introduction and Background

1.1 Introduction

Welcome to Network Performance Analysis, 16 lectures and 2 lab sessions on queueing theory and network simulation.

Queueing theory dates back to the beginning of the 20th Century and the pioneering work of **Agner Krarup Erlang** (1878-1929) a Danish Telecoms Engineer who published the first paper on queueing theory Erlang [1909].

In these first two chapters we will introduce the background for the material which follows: the development of worldwide communication networks. For further reading please check `http://en.wikipedia.org` and `http://www.computerhistory.org`.

1.2 Early Milestones

Perhaps the first major development in telecommunications was *the telegraph*. **Samuel F. B. Morse** developed ideas for the electrical telegraph which exploited the relationship between magnetism and electricity. His first working system sent a short message from Baltimore to Washington D.C. (two cities that are conveniently close together!). It was sent on May 24th, 1844: 'What hath God wrought?'. Ten years later 23,000 miles of wire across the United States.

At the time, a key limitation of the telegraph was that it only provided one channel per wire for communication. Work in using a single wire for multiple channels (multiplexing) indirectly led to the invention of *the telephone* which was patented by **Alexander Graham Bell** (1847-1922) in March 7th, 1876.

The development of a *wireless telegraphy* system by the Italian born engineer **Guglielmo Marconi** (1874-1937), patented in July 1897, led to the foundation of the Wireless Telegraph & Signal Company. By December 1901 short wave transmission across the Atlantic between England and Canada (over 2,100 miles).

1.3 Automated Telephone Networks

Initially, telephone call routing was achieved manually. Operators at the exchange used patch boards to make physical connections between the callers. The invention, by **Almon Strowger** (1839-1902), of the *automatic telephone exchange* (through the Strowger switch) in 1891 brought about the need for a theoretical study of the operation of telephone exchanges.

The network of interconnected phone lines is now known as a *circuit switching network*. It is a network of lines which can be switched between two subscribers to form a connection.

Erlang's contribution was to provide theoretical studies of these networks which allowed telephone exchanges to be designed to meet demand.

1.4 Information Theory

As 'phone networks developed the need for underlying theory grew. Networks transmit 'information' but to discuss concepts such as channel capacity with out a formal definition for information is meaningless. **Claude E. Shannon** (1916-2001) wrote a seminal paper *A Mathematical Theory of Communication* Shannon [1948] which founded the field of information theory and revolutionised the way we look at telecommunication. This work enabled us to discuss concepts such as the capacity of a channel with a well founded underlying mathematical theory. We will not cover the ideas from this course here, but some students will learn more about this in the module COM6862: *Applications of Coding, Information and Learning Theories*.

1.5 Computer Networks

Circuit switched networks were designed for telephone communication between two correspondents. These were the first worldwide communication networks. However, by the early 1960s the need for a new type of communication network was arising. In circuit switching networks, a permanently open line connects the correspondents. A fixed bandwidth is available to them for communication. If they do not use this bandwidth it cannot be made available to other network users. In *packet switching networks* or store and forward switching networks the bandwidth is not reserved in this way: it is distributed across the network user base. The information a user sends (whether it be voice like in a phone conversation and VoIP[1] or a file using FTP[2]) is split into packets which are routed across the network to the desired destination. If a required communication line is in use the packet is stored in a queue until the line becomes available (thus the name store and forward switching). The ideas behind packet switching networks were developed independently by **Paul Baran** (1926-) at the RAND corporation in the U.S. and **Donald Davies** (1924-2000) of the National Physical Laboratory in the U.K.. **Leonard Kleinrock's** (1934-) 1962 thesis (later published as a book Kleinrock [1964]) provided a mathematical theory that underpinned these networks.

1.5.1 ARPANET

The Advanced Research Projects Agency (ARPA) funded a network that actually implemented these ideas. In 1969 the ARPA network connected four nodes at the University of California, Los Angeles (UCSB); the Stanford Research Institute; the University of California, Santa Barbara (UCSB) and the University of Utah. By 1975 (see Figure 1.1) the network had expanded to cover most of the major universities (61 nodes) in the U.S. as well as a node at University College London in the U.K..

ARPANET was the first network for computer communications, it was restricted to non-commercial use and funded by ARPA and NSF until 1990. With the increase in communication, protocols became far more important. Just as spellings of European languages became standardised after the introduction of the the printing press communication protocols also became more prevalent. Telnet and file transfer protocols (FTP) were developed in 1971, but perhaps the most important protocol developed at that time was the transmission control protocol (TCP) developed in the mid-seventies by **Vinton G. Cerf** (1943-) and **Robert E. Kahn** (1938-) Cerf and Kahn [1974]. These protocols allow interconnection between networks thereby enabling the formation of the internet.

1.5.2 The Internet

By 1977 Cerf and Kahn were able to send messages across networks including the 'Packet Radio net', SATNET and the ARPANET. At a similar time **Robert Metcalfe** (1946-) was developing

[1] Voice over IP, the protocol for having phone conversations over the internet.

[2] FTP is the file transfer protocol, a much older protocol which dates back to the formation of the ARPANET.

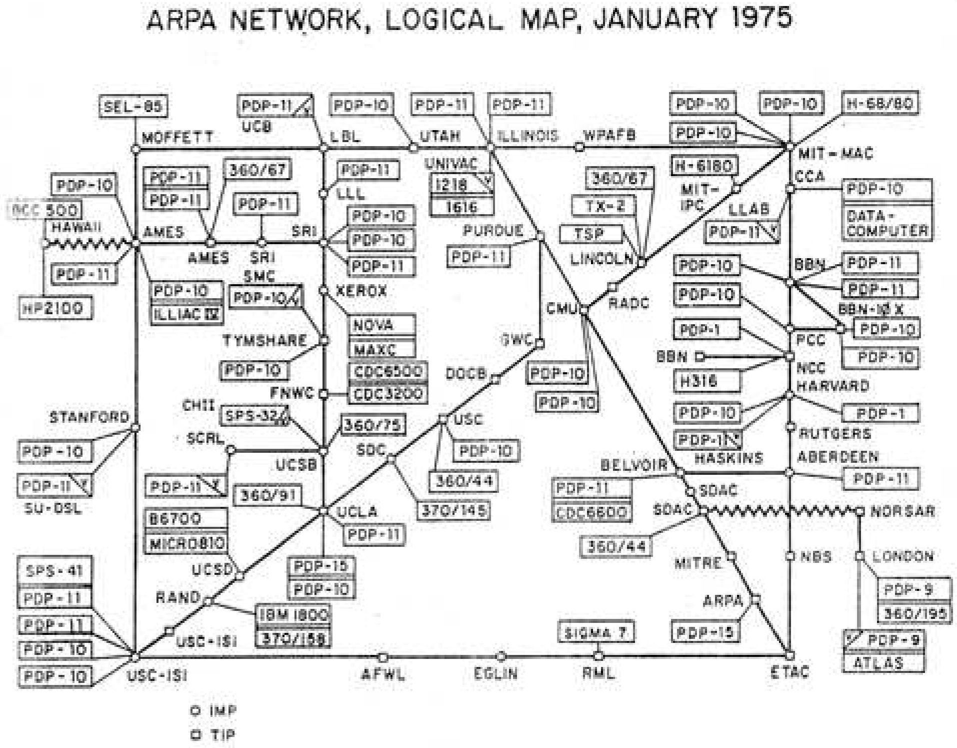


Figure 1.1: The ARPANET in 1975, image courtesy of the Computer History Museum, `http://www.computerhistory.org`. Some abbreviations: UCB - University of California at Berkeley; UCSD - University of California at San Diego; UCLA - University of California at Los Angeles; PDP-XX - the PDP machines were early mini-computers made by DEC with a price tag of around $20,000. The machines starting with 360 and 370 are IBM mainframe machines. Note the transatlantic link to London and the link over the Pacific to Hawaii. There are 61 nodes in total on this network.

Ethernet at Xerox PARC. Ethernet was inspired by the ALOHA networks at the University of Hawaii. Each machine in the ALOHA network communicated across the same frequency radio channel. This meant that if several communications where sent at the same time ‘collisions’ could occur resulting in garbled messages. A procedure for detecting and retransmitting the message was required. For Ethernet networks a cable is the medium of transmission, rather than a radio frequency. Ethernet was supported by an alliance of companies: DEC, Intel and Xerox. This led to Ethernet networks superseding other competing approaches such as token ring, FDDI and ARCNET. The cheap availability of Ethernet and computers in general contributed to an explosion in the number of networks associated with the internet.

By 1990 when ARPANET was formally shutdown it had grown to 300,000 hosts and connected countries across the globe, it evolved into the NSFNet which became the backbone of the internet. In 1991 the internet was opened to commercial traffic, the commercialisation of the internet in combination with the invention of the hyper-text markup language (HTML) and universal resource locaters (URL) by **Tim Berners-Lee** (1955-) formed the foundations of the modern internet.

Chapter 2

Network Performance Analysis

In this course we will introduce you to the tools for design and analysis of telecommunications networks. The core tools of this analysis are probability based: there is an assumption that the traffic in the system is stochastic.

The techniques we will cover are applicable beyond the design and analysis of telecommunication networks. Queues form in a variety of Information Technology scenarios: compute servers dealing with submitted jobs and databases handling queries. They also occur in our every day lives: cashpoints, supermarkets, registration, telephone call centres *etc.*. The principles we will cover apply in each of these situations.

2.1 Practical Considerations

For the moment, let's focus on, for example, network design. Broadly speaking we need to focus on two interrelated areas in our analysis:

1. **Design Issues**: when building or buying a router how much memory should we allow for? How much bandwidth do we need between routers? How much processing power should the router have so that it may exploit that bandwidth effectively?

2. **Quality Issues**: what will be the average delay for end users given our network design? What is the worst case delay? How likely is the system to drop packets?

These are important because resources are finite, there are limitations on the available bandwidth, processor speeds and memory.

2.2 Terminology

Before we cover these issues in more detail, let's introduce some standard terminology. Typically a series of communications between two users/computers is known as a *session*. The interactions themselves consist of *messages* each of which may be split into *packets*.

One example of a session would be a lecture. The session consists of a mainly one-way communication between the lecturer (the server) and the students (multiple clients). In this case most of the information transfer is one-way with the lecturer sending a series of messages, each involving a different subject: the lecturer's identity, the course's subject matter *etc.* and you acknowledging by your lively reactions. Each message consists of a series of packets, in this case perhaps the sentences that the lecturer speaks. The communication medium is the air.

2.3 Session Characteristics

Depending on the nature of the information communicated between the client and the server the characteristics of the session can vary dramatically. An FTP session may involve a few packets from the client to request a file followed by many packets from the server to send the file. A VoIP session will involve large numbers of packets between two users and no obvious client/server relationship. An hypertext transfer protocol (HTTP) session may involve long periods of inactivity as the client reads the pages they have downloaded.

Each of the following six criteria/characteristics of a session may need to be considered in the design of the protocol and the network.

1. **Message arrival rate**. We can seek to model the arrivals as: *Poisson arrivals*, *deterministic arrivals*, *uniform arrivals* or as coming from a more general class of distributions.

2. **Session holding time**: The length of a session.

3. **Expected message length** and **length distribution**, message lengths can be modelled as *exponential distributions*, *uniform length.*

4. **Allowable delay** - acceptable expected delays vary. The acceptable delay is low for VoIP and quite high for FTP.

5. **Reliability** - FTP *must* be error free but VoIP can have some bit errors or packet loss without major consequences.

6. **Message and packet ordering** - Some sessions will require that the packets arrive in the order they are sent: for example database requests.

In this course we will be particularly interested in the first five of these characteristics.

2.4 Network Types

In our review of the history of communication networks, we have already met two distinct kinds of network.

Circuit Switching: used for telephone routing. Guarantees an open ‘channel’ between two peers.

Packet Switching or Store and Forward Switching: used for data. May operate via

> **Dynamic Routing** Each packet of the message moves through the network independently.
>
> **Virtual Circuit Routing** A fixed path through the network is provided for the entire message.

The character of the two approaches are quite different.

2.4.1 Circuit Switching

Circuit switching can be wasteful in terms of bandwidth. The communicating nodes may not be using all the available bandwidth reserved for them all of the time. However, once connection is established, a guaranteed channel is provided with a known communication delay associated with it.

In a circuit switching network, the sum of rates for all sessions using a link cannot exceed the total capacity of the link. New sessions must be rejected when the link capacity is reached. The approach is used for telephone networks where the session rate, r_s, is the same for each session. Switching would be more complex for data networks and the nature of transmission would also mean that the links are inefficiently used.

2.4.2 Packet Switching

Packet switching makes efficient use of bandwidth because, whilst communication between two nodes may be idle for a period of time, packets from another communication may be sent along the same line. However because it is unknown *a priori* how much bandwidth will be available it is difficult to guarantee a maximum delay.

2.4.3 Illustrative Example

Alice and Bob both have telephone lines. Alice decides to call Bob using the phone line. Here the make use of a circuit switching network provided by their telephone supplier (Figure 2.1). The session dominates both phone lines preventing use by other 'clients'.

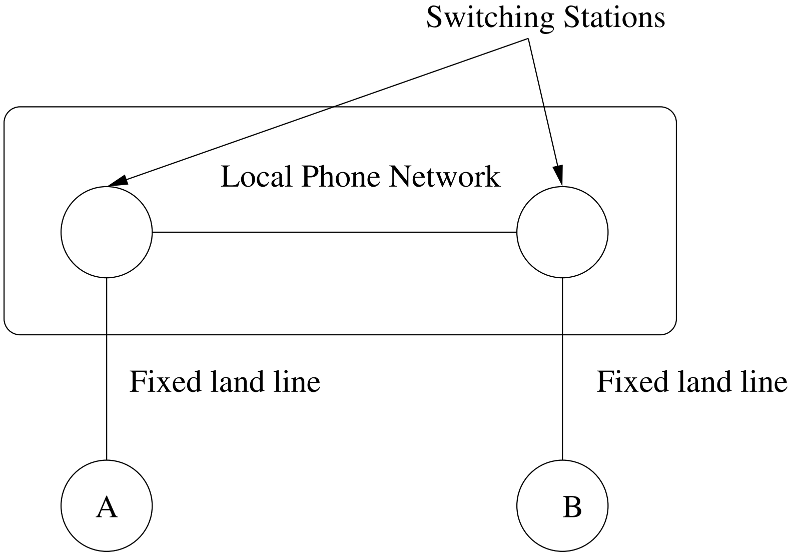


Figure 2.1: A simple circuit switching network connecting Alice and Bob.

Alternatively Alice and Bob use MSN messenger to communicate (either via voice or typing), and Bob's friend Carol shares his fixed land line connection to access David's web server (Figure 2.2).

Now the fixed land line is managed by some form of router in the local area network in Bob's house. The advantage is that Bob and Carol may both use the land line for communication the disadvantage is that if the land line becomes congested *e.g.* as a result of Carol downloading a large file from David's web server, Bob will notice a variable delay (delays also occur with circuit switching but they are known as transmission delays and should be of fixed length) in his communication with Alice due to queueing occurring at his Internet Service Provider and in his LAN's router. This can be particularly irritating for voice conversation, as anyone who has used MSN messenger for voice communication over a low bandwidth channel will attest.

2.5 The Layered Model

We now review the layered model for TCP/IP networks, you are referred to Tanenbaum (4th Edition pg. 41) for more details.

The traditional OSI Reference model contains seven layers which reflect in a network how communication occurs at different layers using different protocols. By way of reviewing the TCP model we introduce a layered representation of the TCP/IP model which underlies the internet. As we mentioned above TCP/IP was introduced in 1974 as a standard which interconnects between

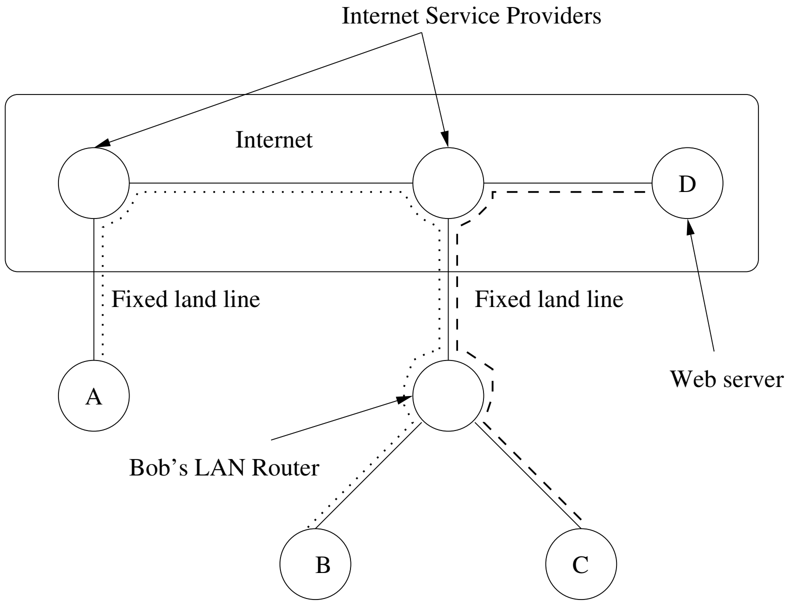


Figure 2.2: A simple packet switching network connecting Alice, Bob, Carol and David.

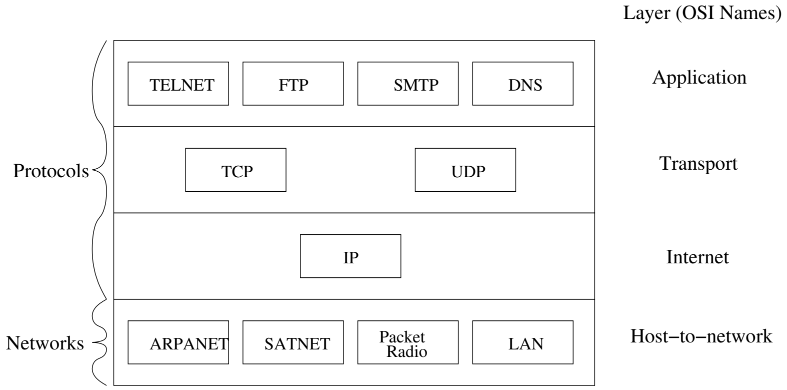


Figure 2.3: A schematic of the Layered model for TCP/IP networks.

networks. This is achieved through the Internet Protocol. The TCP/IP reference model is much more compact and will serve as a useful reminder.

Application Layer This layer represents the interface to the user and consists of applications which are commonly used by users. As well as the examples in the diagram packetised voice communications, electronic news, HTTP and its secure variants etc.

The Transport Layer The layer above the internet layer is known as the transport layer and contains two protocols. The Transmission Control Protocol (TCP) allows applications to communicate in a lossless manner, *i.e.* any packets which are dropped are retransmitted. It also implements flow control in an effort to prevent network congestion and reorders arrivals at the destination. The User Datagram Protocol (UDP) is an unreliable protocol for applications which wish to provide their own sequencing and flow control (e.g. voice transmission or video).

The Internet Layer The key to the success of TCP/IP is the Internet Protocol. Its objective is to allow the interfacing of different networks by allowing hosts to place packets on any network and enabling them to travel to any other network.

2.6 Communication Media

Circuit switching networks may share a communication line but this is done through multiplexing where a fixed proportion of the communication channel is dedicated to the session. The multiplexing may be time division multiplexing or frequency division multiplexing. The former splits the communication channel into time slots and allocates each to a different session, the latter uses different portions of the frequency bandwidth for each channel. Both are not 100% efficient (time multiplexing requires a small amount of time for switching to occur, and frequency division multiplexing requires a small 'buffer' band of frequencies between each session. Packet switching networks dedicate communication bandwidth to each channel as required. In a simple 'first in, first out'[1] (FIFO) system, packets arrive and, if the queue is empty, they are transmitted. Otherwise they are stored and await their turn for transmission. Thus there are two delays: a queueing delay and a transmission delay. The queueing delay is variable, whilst the transmission delay will be fixed and given by the packet length multiplied by the transmission rate of the channel.

Radio transmission is a communication medium which dates back to Marconi. Transmission may occur across several bandwidths. Traditionally radio broadcast occurs through frequency division multiplexing. Radio communication involves taking turns to communicate over the channel (half-duplex). The most basic wireless packet networks (*e.g.* ALOHA) used one channel to broadcast their packets over. Delays may occur due to collisions. A collision is when two systems both attempt to communicate on the same channel at the same time. Both packets are then lost and such losses need to be detected.

Similar principles occur for *e.g.* thinwire Ethernet, in this case the communication medium is a coaxial cable.

2.7 The Road Ahead

Probability Review Review of probability theory.

Simple Queueing models Simple models of queues based on probability theory. Little's formula, $M/M/1$ queue, $M/M/\infty$, $M/M/m$, $M/M/m/m$ and $M/G/1$.

Simulation Simulation of queueing systems.

Examples

1. As a simple example let's consider a database (perhaps MySQL). We are constructing it from existing parts. We have two CPU/motherboard systems and two separate disk systems, their expected characteristics are shown in Table 2.1. Which should we match with which? Are we better off building one faster system and one slower system? Or should we build two systems that have a similar performance?

CPU System	Expected Service Time/ms	Disk System	Expected Service Time/ms
A	423	1	215
B	210	2	390

Table 2.1: Average times for a fictional database server for three different cases.

[1] Also known as 'first come, first served' (FCFS).

Chapter 3

Probability Review

We will now review the basic concepts of probability theory, probability theory will provide the underlying structure upon which we will base our models. We will make use of a simple pictorial definition to describe what we mean by probability. For the purposes of this discussion, we are interested in trials which result in two random variables, X and Y, each of which has an 'outcome' denoted by x or y. We summarise the notation and terminology for these distributions in Table 3.1.

Terminology	Notation	Description
Joint Probability	$P(X = x, Y = y)$	'The probability that $X = x$ and $Y = y$'
Marginal Probability	$P(X = x)$	'The probability that $X = x$ regardless of Y'
Conditional Probability	$P(X = x \vert Y = y)$	'The probability that $X = x$ given that $Y = y$'

Table 3.1: The different basic probability distributions.

3.1 A Pictorial Definition of Probability

A solid understanding of the fundamentals of probability will be one of the keys to understanding the material in course. We will therefore quickly review what each of these distributions means with a simple pictorial demonstration. In Figure 3.1 we summarise the outcomes of a series of N trials in a two dimensional graph.

For Figure 3.1 the outcome of a trial is plotted on a two dimensional graph. We have drawn three further boxes on the graph (solid, dotted and dashed lines), each box is defined by the outcomes it covers. We have denoted the number of events in each box with a small n. Probabilities are defined in the limit as we take the number of trials N to infinity.

Terminology	Definition
Joint Probability	$\lim_{N\rightarrow\infty} \frac{n_{X=3,Y=4}}{N} = P(X = 3, Y = 4)$
Marginal Probability	$\lim_{N\rightarrow\infty} \frac{n_{X=5}}{N} = P(X = 5)$
Conditional Probability	$\lim_{N\rightarrow\infty} \frac{n_{X=3,Y=4}}{n_{Y=4}} = P(X = 3 \vert Y = 4)$

Table 3.2: Definition of probability distributions from Table 3.1 in terms of the system depicted in Figure 3.1.

3.1.1 Notational Details

Strictly speaking, when we write probability distributions we should write them out in full (*e.g.* for the joint distribution, $P(X = x, Y = y)$). However, in practice, we often use a shorthand

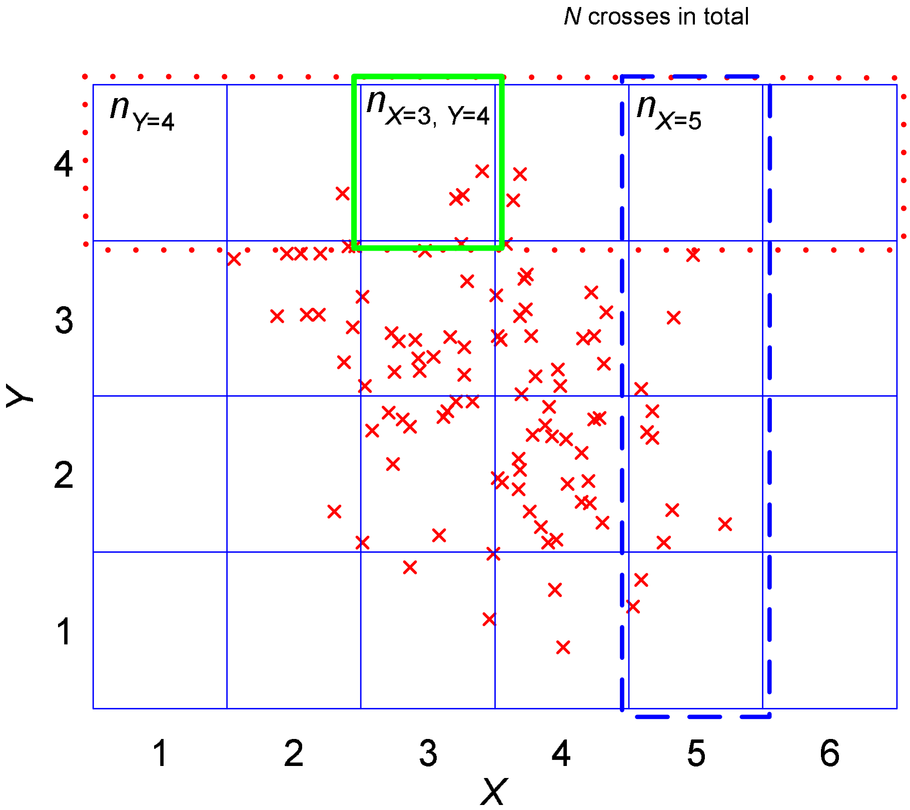


Figure 3.1: Representation of joint and conditional probabilities. In this figure the crosses are the outcomes of the trials. They live in a discrete space of boxes given by the grid. The two events, X and Y are shown across the two axes of the plot.There six possible outcomes for event X, $X = 1 \ldots 6$, and five possible outcomes for event Y, $Y = 1 \ldots 5$. The small ns in the top left hand corner of each thick lined box denote how many crosses fall inside that box.

notation: $P(x, y)$. This is convenient, but it can lead to confusion. It looks very much like we might write a multivariate function, *e.g.* $f(x, y) = \frac{x}{y}$. For a multivariate function though, $f(x, y) \neq f(y, x)$ in general. However, for our probability shorthand $P(x, y) = P(y, x)$ because $P(X = x, Y = y) = P(Y = y, X = x)$.

We have denoted the probability distribution with a capital P. In *these notes* we will use capital Ps consistently to denote discrete distributions. Later, when we come on to continuous probability density functions, we will turn to small letters.

3.2 Probability Rules

From the simple definitions we've given above we can derive some straightforward rules associated with manipulating probability distributions.

3.2.1 Normalisation

The first point to note is that probability distributions are normalised, the sum over the distribution across each of its possible value is 1. This is clear from the fact that $\sum_x n_{X=x} = N$, which gives

$$\sum_x P(x) = \lim_{N\to\infty} \frac{\sum_x n_{X=x}}{N} = \lim_{N\to\infty} \frac{N}{N} = 1.$$

A similar result can be derived for the marginal and conditional distributions.

3.2.2 The Sum Rule

The sum rule describes how we can obtain a marginal distribution from the joint distribution. The distribution is called the marginal distribution because one of the variables in the joint distribution is *marginalised*, *i.e.* it is removed from the distribution (you can think of it being written in the margin). The marginal probability $P(y)$ is given by $\lim_{N\to\infty}\frac{n_{Y=y}}{N}$ and the joint distribution $P(x,y)$ is given by $\lim_{N\to\infty}\frac{n_{X=x,Y=y}}{N}$. Now we note (using Figure 3.1) that $n_{Y=y}=\sum_x n_{X=x,Y=y}$, where the sum is over all possible values of X. We therefore find that

$$\lim_{N\to\infty}\frac{n_{Y=y}}{N}=\lim_{N\to\infty}\sum_x\frac{n_{X=x,Y=y}}{N},$$

or in other words

$$P(y)=\sum_x P(x,y).$$

This is known as the sum rule of probability.

3.2.3 The Product Rule

The product rule describes how we obtain a joint distribution from a conditional distribution and a marginal. Recall that the conditional distribution $P(x|y)$ is given by $\lim_{N\to\infty}\frac{n_{X=x,Y=y}}{n_{Y=y}}$. This can be related to the joint distribution over x and y and the marginal distribution over x using the fact that $P(x,y)$ is given by

$$\lim_{N\to\infty}\frac{n_{X=x,Y=y}}{N}=\lim_{N\to\infty}\frac{n_{X=x,Y=y}}{n_{Y=y}}\frac{n_{Y=y}}{N}$$

or in other words

$$P(x,y)=P(x|y)P(y).$$

This is known as the product rule of probability.

3.2.4 Bayes' Rule

Less relevant to our work on queueing theory, but worth pointing out since we've come so far is Bayes' Rule or Bayes' Theorem. Since the joint distribution is commutative in its arguments we have, from the product rule,

$$P(x,y)=P(y,x)=P(y|x)P(x),$$

it naturally follows that we can write

$$P(x|y)P(y)=P(y|x)P(x)$$

which leads to Bayes' Rule, which relates the two conditional probabilities associated with the variables:

$$P(x|y)=\frac{P(y|x)P(x)}{P(y)}.$$

3.2.5 Independence

In the course we will often state that two or more variables are assumed to be independent. Probabilistically independence refers to the affect of the knowledge of one variable on the distribution of another. Mathematically this means that our probability distribution over x is not affected by the value of y, or in other words

$$P\left(x\right) = P\left(x|y\right),$$

the conditional distribution of x given y is equal to the marginal distribution of x. This in turn implies that the joint distribution factorises,

$$P\left(x, y\right) = P\left(x|y\right) P\left(y\right) = P\left(x\right) P\left(y\right),$$

and furthermore that

$$P\left(y|x\right) = P\left(y\right).$$

The factorisation of the joint distribution when variables are independent turns out to be a very important property which makes a lot of analysis tractable.

3.3 Expectations

A probabilistic model of a system may summarise our beliefs about the system, but to develop further understandings of the model often want to summarise the model by its expectations.

The expected value of a function, $f\left(x\right)$, under a probability distribution $P\left(x\right)$ is

$$\left\langle f\left(x\right)\right\rangle_{P(x)} = \sum_{x} P\left(x\right) f\left(x\right).$$

Here we have introduced a particular notation for the expectation, you will also see expectations written in the form $E\left\{f\left(x\right)\right\}$.

3.3.1 Expectations under a Simple Distribution

Suppose that X can be 0, 1, or 2 and $f\left(x\right) = x^2$. We define a distribution over x in Table 3.3.

x	0	1	2
$P\left(x\right)$	0.2	0.5	0.3
$f\left(x\right) = x^2$	0	1	4

Table 3.3: Distribution over x and associated values of $f\left(x\right)$.

From the definition of an expectation we gave we can compute

$$\left\langle f\left(x\right)\right\rangle = \left\langle x^2\right\rangle = 0 \times 0.2 + 1 \times 0.5 + 4 \times 0.3 = 1.7,$$

where we have dropped the subscript $P\left(x\right)$, as we consider it clear from the context what the expectation is considered to be over. We can also see that the expectation of x under this distribution is

$$\left\langle x\right\rangle = 0 \times 0.2 + 1 \times 0.5 + 2 \times 0.3 = 1.1.$$

The expectation of x under a distribution is known as the first moment of the distribution or its *mean*. The mean can be thought of as the 'centre of probability mass' of the distribution. The expectation of x^2 is known as the second moment, it is normally used in combination with the mean to obtain the variance.

3.3.2 Variance

The variance of the distribution is given by the difference between the second moment and the square of the mean. It is often denoted by σ^2.

$$\sigma^2 = \left\langle x^2 \right\rangle - \left\langle x \right\rangle^2 .$$

It is always positive and it expresses the degree of variation around the mean value. Its square root is known as the standard deviation

$$\sigma = \sqrt{\langle x^2 \rangle - \langle x \rangle^2},$$

for our example above the variance is given by

$$1.7 - (1.1)^2 = 0.49,$$

the standard deviation is therefore

$$\sqrt{1.7 - (1.1)^2} = 0.7.$$

3.4 Distributions as Functions

In the above example we defined a probability distribution with a simple table, making sure that the values summed to 1. A more compact way to represent a distribution is in terms of a function. As an example of this we will consider the Poisson distribution.

3.4.1 The Poisson Distribution

The Poisson distribution was first published by **Siméon Denis Poisson** (1781-1840) in 1837. It is defined over the space of all non-negative integers. Since this set is countably infinite in size it is impossible to write all the numbers down that would define the distribution. The Poisson distribution is therefore defined as

$$P\left(k|\mu\right) = \frac{\mu^k}{k!} \exp\left(-\mu\right). \tag{3.1}$$

where k is any integer from 0 to ∞, and μ is a parameter of the distribution. In these notes we place μ on the right of the conditioning bar to indicate that it is a parameter. Note that in Bertsekas and Gallager [1992] a slightly different notation is used. To work out the probability of a particular value, k, in a Poisson distribution with a particular parameter, μ, we simply substitute the parameter and the value n into (3.1). For example, with $\mu = 2$ we can start filling a table (Table 3.4).

k	0	1	2	...
$P(k)$	0.135	0.271	0.271	...

Table 3.4: Some values for the Poisson distribution with $\mu = 2$.

As we mentioned in Section 3.3, the expectations under a distribution are important for summarising the distributions characteristics. Expectations of distributions that take the form of a function can also be computed. Typically we expect the resulting expectation to be a function of the distributions parameters. In the Examples section at the end of this chapter we compute the mean of a Poisson distribution.

3.5 Continuous Variables

So far we have discussed a value for discrete values of x or y. We now turn to a continuous models. First though we introduce the probability distribution's sister, the *probability density function* (PDF). In the discrete case, we defined probability distributions over a discrete number of states. However, if we wish to model a continuous value, such as the height of a computer science student, how do we represent it as a probability distribution? One way would be to 'discretise' the continuous value by splitting it into discrete regions. We effectively did this in the example in Figure 3.1. So, for our height example, we could model the probability that your height was between 1.6m and 1.7m, or alternatively we could ask what is the probability that you are over 1.8m given that you are a computer scientist? However, remodelling for each of these questions would be rather tedious. It would be more convenient if we could develop a representation which could answer *any* question we chose to ask about a computer scientist's height. This is what a probability density function represents. It does not represent probabilities directly, but from it they may be derived.

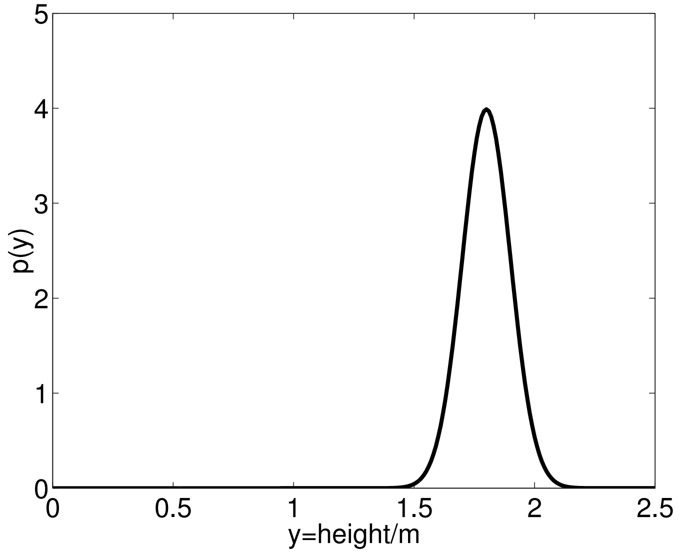


Figure 3.2: A Gaussian PDF representing heights of computer scientists.

We represent a probability density function with a small $p(\cdot)$ to differentiate it from a probability distribution which we have denoted with a large $P(\cdot)$. We define it to be a positive function whose integral over the region of interest is one[1]. One example is the Gaussian, shown above to represent a model of a computer science student's height. We can query the distribution by integrating over regions of it. If we wish to ask the question 'What is the probability that a computer science student is over 2m tall?' The answer, which we denote $P(y > 2)$ is given by integrating from 2 to ∞. Since the total area under the curve is always one, and the function is always positive, the answer will always be less than 1 and greater than 0. This fulfils our definition of probabilities. The answer to the inverse question will be $1 - P(y > 2)$. Note that this implies that the probability of a student being exactly 2m tall is zero. This may seem odd, but is a consequence of querying the system in the limit as a region tends to zero.

[1] In what follows we shall use the word distribution to refer to both discrete probabilities and continuous probability density functions.

3.5.1 Manipulating PDFs

The same rules that apply to probability distributions may also be applied to PDFs. We won't derive this here, but take it as given that

$$p(y|x) = \frac{p(x|y)\,p(y)}{p(x)}$$

where $p(x, y) = p(x|y)\,p(y)$ and for continuous variables $p(x) = \int p(x, y)\,dy$. Here the integral is over the region for which our PDF for y is defined (*e.g.* all real numbers or all positive real numbers). Additionally we may mix discrete and continuous variables. If, for example y is discrete and x is continuous, then we may write

$$P(y|x) = \frac{p(x|y)\,P(y)}{p(x)}.$$

Finally expectations under a PDF are also computed using the integral instead of the sum so

$$\langle f(x)\rangle_{p(x)} = \int f(x)\,p(x)\,dx$$

where the integral is over the region for which our PDF for x is defined.

3.5.2 Cumulative Distribution Functions

We've discussed how a PDF doesn't represent probabilities directly, but it is a mechanism by which questions about the random variable can be answered. One very common question is: what is the probability that $x < y$? It can be useful to have a function which stores the answer to this question for all y. This function is known as the cumulative distribution function (CDF). The CDF is given by the integral of the PDF across all values of x less than y. So if the probability distribution is valid in the range $-\infty < x < \infty$ the CDF is given by

$$P(x < y) = \int_{-\infty}^{y} p(x)\,dx,$$

alternatively, if the distribution is defined in the range $0 \leq x < \infty$ then the CDF is given by

$$P(x < y) = \int_{0}^{y} p(x)\,dx.$$

To complete the distribution we need the probability that $x > y$ (which is the other possibility). This is easy to compute because we know the distribution must be normalised, $P(x > y) = 1 - P(x < y)$.

In Figure 3.2 we give an example of a cumulative distribution function. We show the CDF associated with the PDF of the computer science students' height.

Finally, we note that since the cumulative density function is the integral of the probability density function the PDF can be recovered from the CDF through differentiation.

3.6 Sample Based Approximations

Recall from Section 3.3 that expectation of any given function, $f(k)$, under a distribution, $P(k)$, can be written as

$$\langle f(k)\rangle_{P(k)} = \sum_{k} f(k)\,P(k)$$

which is also written as $E\{f(k)\}$. Recall our definition of probability from Table 3.1.

$$P(k) = \lim_{S\to\infty} \frac{s_k}{S}$$

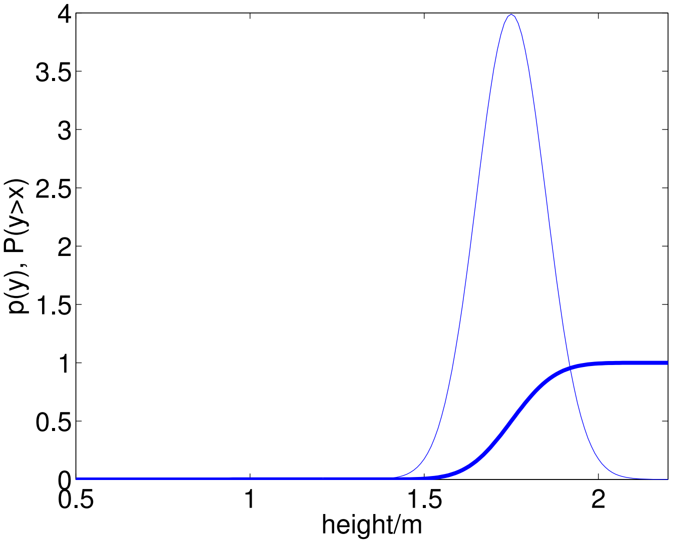


Figure 3.3: The cumulative distribution function (CDF) for the heights of computer science students. The thick curve gives the CDF and the thinner curve the associated PDF.

where we have now used S (instead of N) as the total number of samples or trials from a system and s_k is the number that have state k. When we make *sample based approximations* we hope that S is 'large enough[2]' such that it is a good approximation to say

$$\langle f(k)\rangle_{P(k)} \approx \sum_k f(k)\frac{s_k}{S}.$$

This formula may be rewritten as:

$$\sum_k f(k)\frac{s_k}{S} = \frac{1}{S}\sum_k f(k)\, s_k.$$

Now if $\mathbf{k}$ is a vector of samples from $P(k)$with elements k_i and length S then we can rewrite the above

$$\frac{1}{S}\sum_k f(k)\, s_k = \frac{1}{S}\sum_{i=1}^{S} f(k_i)$$

which provides us with a sample based approximation to the true expectation:

$$\langle f(k)\rangle_{P(k)} \approx \frac{1}{S}\sum_{i=1}^{S} f(k_i).$$

Some well know special cases of this include the *sample mean,* often denoted by $\bar{k}$, and computed as

$$\bar{k} = \frac{1}{S}\sum_{i=1}^{S} k_i,$$

this sample based mean is an approximation to the true distribution mean

$$\langle k\rangle \approx \bar{k}.$$

[2] How large S need be is dependent on both the function whose expectation we are estimating and the distribution under which we are taking the expectation.

The same approximations can be used in the context of continuous PDFs, so we have

$$\begin{aligned}\langle f(x)\rangle_{p(x)} &= \int f(x)\,p(x)\,dx \\ &\approx \frac{1}{S}\sum_{i=1}^{S} f(x_i)\,,\end{aligned}$$

where x_i are independently obtained samples from the distribution $p(x)$.

The approximation gets better for increasing S and worse if the samples from $P(k)$ are *not* independent.

Sample Means for Student Heights Returning to our example concerning the height of computer science students. If we don't know the true underlying distribution of their heights, $p(h)$, we cannot compute the mean of the distribution, but we can approximate it with the sample mean: so if we want the mean, $f(h) = h$, we use

$$\int_0^\infty hp(h)\,dh \approx \frac{1}{S}\sum_{i=1}^{S} f(k_i)\,.$$

Sample and Distribution Means In most texts the word mean is used interchangeably to describe sample based means and the true mean of a distribution. However, you should always bear in mind the important difference between the two. Later in the course we will also come across time based averages and assumptions will be made that the time based average is equal to the average under a 'steady state distribution'.

3.6.1 Mean and Variance

We compute the mean of a Poisson in the following example. The variance is an exercise in the first tutorial sheet.

Examples

1. Compute the mean of the Poisson distribution, *i.e.* the expectation of k under the distribution $P(k|\mu) = \frac{\mu^k}{k!}e^{-\mu}$.

   The variance of a Poisson is an exercise in Tutorial Sheet 1.

Chapter 4

Poisson Processes

In Section 3.4 we introduced probability distributions whose values were given by functions. In this chapter we will introduce probabilistic processes.

4.1 Stochastic Processes

A probabilistic process, also known as a random process or a stochastic process, can be viewed as a 'random function'. A normal function maps a set of inputs to a point value. A probabilistic process maps a set of inputs to a probability distribution over values or a probability density function.

Various systems are modelled with probabilistic processes. For example, the *Wiener process* is the process behind Brownian motion. Stock market data is often modelled as a stochastic process. For example the Black-Scholes formula assumes that the logarithm of the stock market data can be modelled as a Wiener Process.

We will base our definition of a random process on that given by Chan [2000].

A random process May be defined as a *random function* which, for a particular value, is a random quantity.

4.2 Poisson Process

We've already met the Poisson distribution, there is a simple process which is related to the Poisson distribution: the Poisson process.

The Poisson process gives probabilities of a number of arrivals, k, as a function of time, t, given a 'rate', λ.

$$P\left(k|t,\lambda\right)=\frac{\left(\lambda t\right)^{k}}{k!}e^{-\lambda t}. \tag{4.1}$$

Note the similarity between (3.1) and (4.1). The process is simply the distribution with μ substituted with λt.

Using this substitution the mean of the Poisson process (the first moment of k) is given by

$$\langle k\rangle=\lambda t.$$

Which is the expected number of arrivals at a given time t. The parameter λ is therefore known as the *rate* of the process. This implies that the Poisson process is a form of counting process where the counts are *expected* to increase linearly with time at a rate given by λ.

4.3 Properties

In the Chapter 5 we will derive the Poisson process from some simple assumptions about how arrivals occur, for the moment though we will highlight some important properties of the Poisson process. We will describe each in more detail below, to summarise though the fiver properties are:

1. Superposition of Poisson processes.
2. Decomposition of Poisson processes.
3. Independence of arrivals.
4. Similarity of arrival distribution.
5. Interarrival time distributions.

4.3.1 Superposition of Poissons

The first property we will consider is superposition of Poisson flows. It is a property of a Poisson process that if we combine N independent[1] processes with rates $\lambda_1, \lambda_2, \ldots, \lambda_N$ together the resulting process is also a Poisson process with a combined rate, λ, equal to the sum of each of the input rates: $\lambda = \sum_{n=1}^{N} \lambda_n$. This is shown pictorially in Figure 4.1.

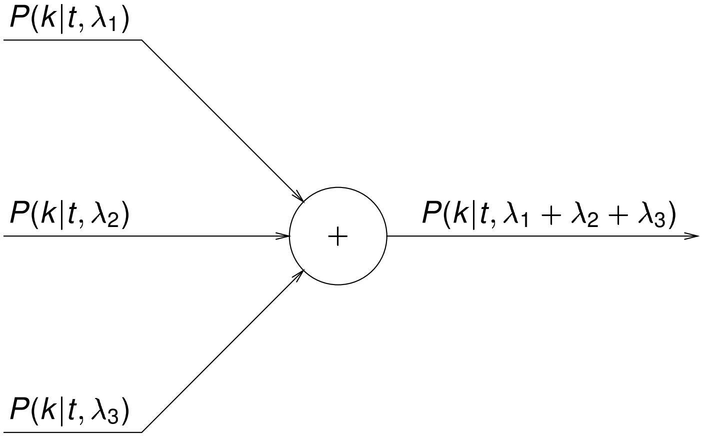


Figure 4.1: If independent Poisson processes form a single traffic flow overall process is a Poisson with a rate that s the sum of the rates of the separate processes.

4.3.2 Decomposition of Poisson Flow

Just as Poisson processes can be combined to obtain a higher rate Poisson process, they can also be decomposed to obtain lower rate Poisson processes. Assume we have one input channel where arrivals are occurring with a Poisson process with rate λ. This channel then splits into N separate output channels. Each input arrival is then *independently* assigned to output channel n with probability $P(n)$. In this case the output channels then behave like Poisson processes with rates given by $\lambda_n = P(n)\,\lambda$. This is shown in Figure 4.2.

[1] Two variables x and y are independent if their joint distributions are equal to the product of their marginal distributions $P(x,y) = P(x)\,P(y)$.

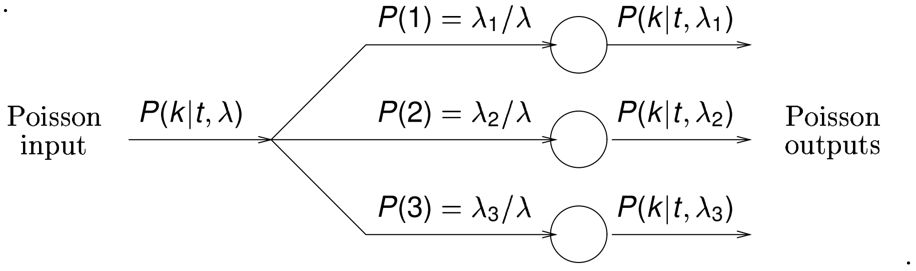


Figure 4.2: If a single Poisson process with rate λ is switched to different output channels, with each arrival being switched to channel n with probability $P(n) = \lambda_n/\lambda$, each output channel will be a Poisson process with rate λ_n.

4.3.3 Independence of Arrivals

Consider a single Poisson process, if we look within this process at two *disjoint* time intervals (*i.e.* periods of time that don't overlap). Assume we have k_1 arrivals in the first interval and k_2 arrivals in the second interval. The distribution of arrivals in the two intervals are independent of each other, *i.e.* $P(k_1, k_2) = P(k_1)P(k_2)$.

4.3.4 Similarity of Arrival Distributions for Identical Intervals

Consider a single Poisson process with rate λ, let the number of arrivals between time s and $s + t$ be given by k_1. Now consider a second independent Poisson process with the same rate λ, let the number of arrivals between time r and $r + t$ be given by k_2. The distribution of k_1 and k_2 are then equal,

$$P(k_1) = P(k_2).$$

4.3.5 Interarrival Times and Poisson Processes

The Poisson process is a simple model of arrivals that is widely used due to its analytical properties. We have already seen how a superpositions and decompositions of Poisson processes also lead to Poisson processes. Another attractive analytical property is the ease with which we can compute the *interarrival time distribution* for the Poisson process.

The interarrival time distribution is the distribution of times between arrivals. Consider waiting for the next bus at a bus stop. If you *just missed a bus* how long will you have to wait for the next bus to arrive? If the buses arrive as a Poisson process, then it is straightforward to work out the distribution of your waiting time.

The interarrival time distribution for the Poisson process turns out to be the *exponential distribution*,

$$p(X) = \lambda e^{-\lambda X}.$$

This is derived in the examples at the end of this chapter.

Stating that interarrival times are distributed exponentially is equivalent to stating that arrivals occur as a Poisson process. We will therefore use these formulations interchangeably, *i.e.* when we state that the interarrival times for students registering at the Goodwin Sports centre is given by an exponential distribution with parameter $\lambda = 2$ we are equivalently stating that these students are arriving according to a Poisson process with rate parameter $\lambda = 2$.

4.4 The Exponential Distribution

In the last section we saw that the exponential distribution gives can be derived by considering the interarrival times of the Poisson distribution. Here we give a few properties of the exponential that will be relevant in this course.

4.4.1 The Memoryless property

We have already seen in our derivation of the exponential as interarrival time distribution that it is *memoryless*. Formally speaking this means that for exponentially distributed variable X

$$P\left(X > s + t | X > t\right) = P\left(X > s\right)$$

or in English, the probability of the waiting time being greater than $s + t$ given that the waiting time is already greater than t is the same as the probability of the waiting time being greater than s.

4.4.2 Mean and Variance

Recall the exponential distribution

$$p(X) = \lambda e^{-\lambda X}$$

The mean of an exponential can be found to be,

$$\langle X \rangle_{p(X)} = \frac{1}{\lambda}$$

and the variance is

$$\left\langle X^2 \right\rangle_{p(X)} - \langle X \rangle_{p(X)}^2 = \frac{1}{\lambda^2}.$$

These are easily confirmed — they are part of Tutorial 1.

Examples

1. **Superposition of Poisson processes.** There are three queues for registration of MSc students from different courses at the University of Sheffield. Students for the first course, Physics and Fine Art, arrive with a rate of 4 students per minute. Students for the second course, The Psychology of Sociology, arrive with a rate of 2 students per minute and students for the third course, Computer Science and Aromatherapy arrive with a rate of 5 students a minute.

   Assuming the student arrivals are independent and occur as a Poisson process what is the probability that after 15 minutes there have been **a)** 90 student arrivals? **b)** 180 arrivals?

2. **Example of Poisson Decomposition.** There is one registration for all MSc courses. Students arrive in the hall and are assigned to registration desks according to the first letter of their surname. Desk 1 ($d = 1$)takes students with names from A-G, desk 2 ($d = 2$) from H-M, desk 3 ($d = 3$) from N to S and desk 4 ($d = 4$) from R to Z.

   Surnames are distributed amongst these four categories with the following probability distribution:

d	1	2	3	4
$P(d)$	0.20	0.27	0.25	0.28

   Assuming the students arrive *independently* of their surname as a Poisson process at a rate of 10 students per minute what is the probability that **a)** after 5 minutes 5 students have arrived at desk 1? **b)** after 10 minutes 20 students have arrived at desk 4?

3. **Interarrival times for the Poisson process.** In Figure 4.3 we show different arrival times, $T_1, \ldots, T_n$ for the process. You can think of these as the arrival times of the buses.

   We are interested in computing the distribution of the variables $X_i = T_i - T_{i-1}$: the times between bus arrivals. In this example you will derive this PDF for the interarrival times[2].

[2] The interarrival times, X_i, are continuous values, so they will be governed by a probability density function rather than a discrete probability distribution.

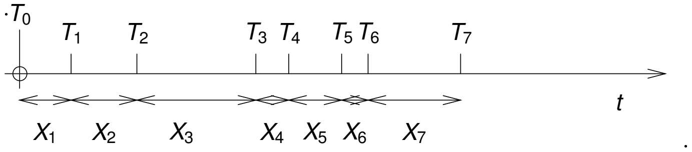


Figure 4.3: For a Poisson process, this figure shows the times of the arrivals, T_n, and the interarrival times, X_n.

Chapter 5

Markov Chains and Markov Processes

In the last chapter we introduced the Poisson process. In this chapter we will show how the Poisson process in a special case of the more general class of processes known as Markov processes. We will start by considering Markov processes as they operate in *discrete time*, otherwise known as *Markov chains.* Both Markov processes and Markov chains exhibit the *Markov property* which is an assumption about how future values drawn from the process depend on past values. We will first discuss the Markov property in the context of Markov chains, then we will show how a discrete system can be converted to a continuous system.

5.1 Markov Chains

Time is a continuous variable, but sometimes it is convenient, or natural, to discretise time. One example of a form of discretised time would be turns in a board game. As the turns progress this discretised form of time progresses.

Consider a discrete representation of time $t = 0, 1, 2....$ We associate a state variable, N, with this time such that we have values for N at $N(0)$, $N(1)$, $N(2)$, ...

5.1.1 Markov Property

If this state variable exhibits the Markov property its value at time t is dependent only on its value at time $t - 1$. The common way of putting this property is that the future is conditional independent of the past. The conditioning being on the present. In other words, the state of the system in the future is *not* dependent on what happened in the past if we know the state at the present time.

Let's consider a real-life example of a Markov chain, that of the board game of Monopoly.

5.1.2 Monopoly Board

There are forty squares on a Monopoly board. We assume that there is only one player. That player's state, $N(t)$, will be the number of the square his/her counter is on +40 for every time he/she has reached GO. For example if you have circled the board twice and are currently on GO then $N(t) = 80$. Or if you have circled the board once and are on King's Cross then $N(t) = 45$. The time t is now clearly discrete and associated with the number of turns the player has had.

In Monopoly you move your counter through rolling two six sided dice and adding them together. We will denote the result of this die roll at time t with $d(t)$. Notice that your state after turn t, $N(t)$ depends only on your die role, $d(t)$, and your state before turn t, $N(t-1)$. We write this as:

$$N(t) = N(t-1) + d(t).$$

This means that Monopoly exhibits the Markov property, as your future states are independent of your previous states given your current state.

5.1.3 State Diagrams

A Markov chain can be represented by a state diagram just as finite state machines are often represented. One important difference is that Markov chains are not necessarily finite, they can have infinite states! A further difference is that the transitions for Markov chains are associated with probabilities (or in the case of Markov processes rates).

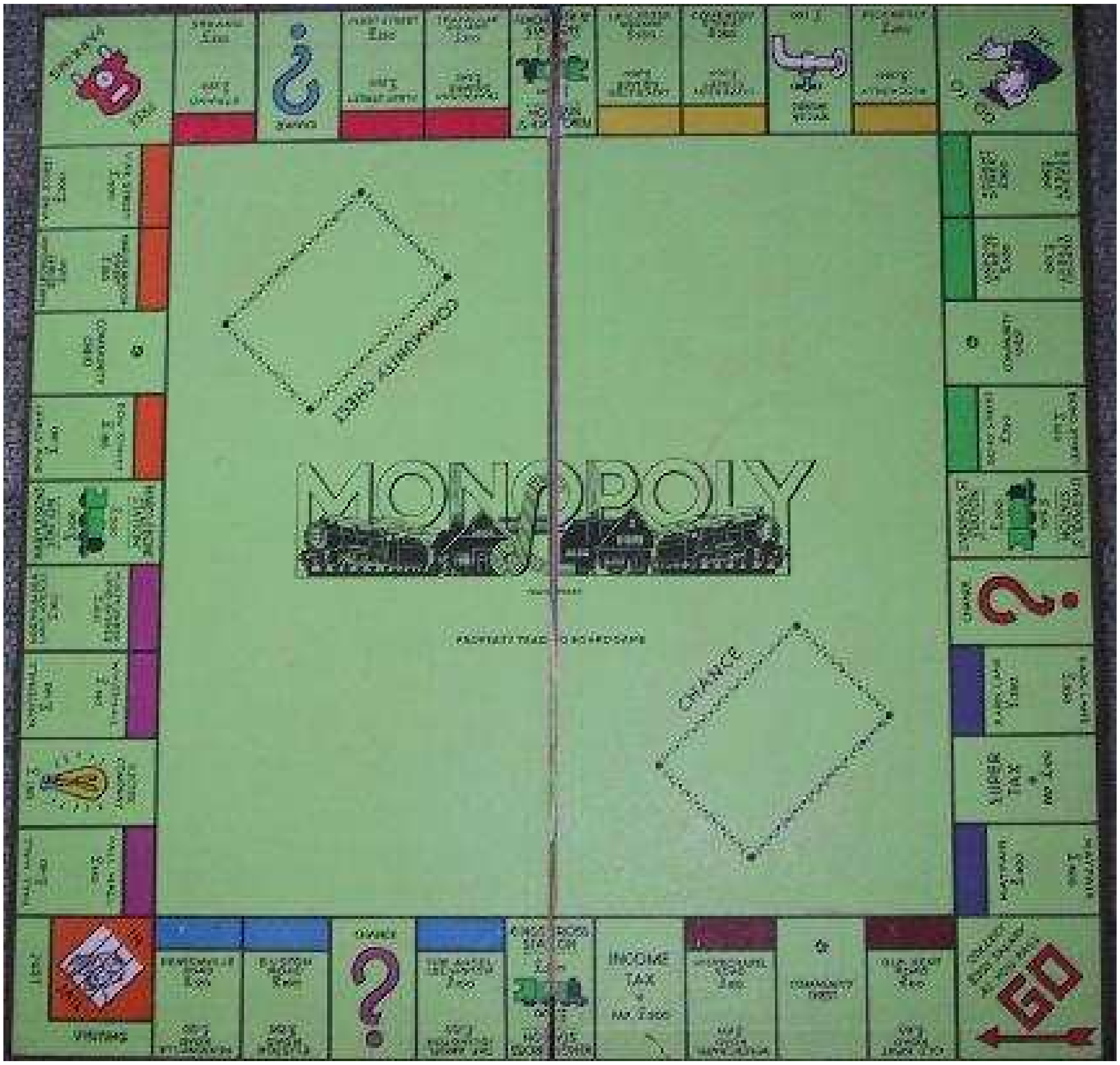


Figure 5.1: Image of a monopoly board.

5.2 Poisson Process as a Markov Process

We've introduced the game of Monopoly as a Markov chain. In this next session we shall relate Markov chains, where time is treated as a discrete variable, to Markov processes, where time is a continuous variable. We will show how the Poisson process is a special case of a Markov process where a constrained set of transitions are allowed. To do this we will first introduce a new version of Monopoly that we call 'Coin Monopoly'.

5.2.1 Coin Monopoly

In regular Monopoly each jump, $d(t)$, is given by the sum of two die rolls. Let's assume that we a Monopoly addict Anne who has lost the dice for her set of Monopoly. In danger of going into withdrawal and, finding a coin in her pocket, she comes up with the following way of playing the game.

In the coin version of Monopoly each player tosses the coin when it is their turn. Then, instead of $d(t)$ being the sum of two die roles, we take $d(t) = 0$ if a coin is tails and $d(t) = 1$ if it is heads. This is depicted as a state diagram in Figure 5.2.

There is a drawback to playing 'Coin Monopoly': it takes rather a long time. However, Anne is a keen mathematician and tries to find a way of speeding it up.

Currently, with each turn, we consider time, t, to change by $\Delta t = 1$. Let's pretend we have a special coin where we can change the probability that it will be heads. We denote this probability, $P(h) = \lambda \Delta t$ (which must be less than 1), giving $P(t) = 1 - \lambda \Delta t$.

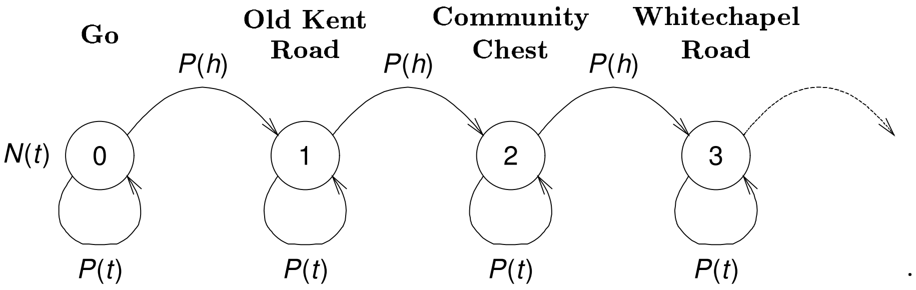


Figure 5.2: State diagram for coin monopoly. Each node (circle) represents a different possible state value. The state changes (or doesn't change) with each time step according the probabilities associated with the arrows.

Anne can compute the probability that, after n turns, the state of the system will be k using the *binomial distribution*.

5.2.2 Binomial Distribution for n Trials

Consider a system with two states (such as a coin). Let's assume that one of those states is associated with 'success' and one is associated with 'failure'. In our Coin Monopoly example heads might be considered success (because it is associated with a move forward) and tails might be associated with failure (because it is associated with no move). This distribution is known as the binomial. For n trials (equivalent to n coin tosses or n turns for Coin Monopoly) the probability of k successes (equivalent to k moves forward) is given by

$$P(k) = \left(\begin{array}{c} n \\ k \end{array} \right) P(h)^k (1 - P(h))^{n-k} \tag{5.1}$$

where $\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k!(n-k)!}$ is known as the binomial coefficient. In the examples at the end of this chapter we show that as the number of trials, $n \to \infty$, is taken to ∞ and the probability of success is driven to zero, $P(h) = \frac{\lambda}{n}$, this system becomes a Poisson process. Inserting these two limits into the equation gives:

$$P(k|\lambda, t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}. \tag{5.2}$$

In other words, we have proved that a Poisson process is a particular type of Markov chain, in fact, taking the number of trials to ∞ is equivalent to taking the limit as $\Delta t \to 0$, this system is now continuous time, it is therefore known as a Markov process rather than a Markov chain. A Poisson process is therefore also a particular instance of a Markov process.

5.2.3 Markov Process State Diagrams

When drawing state diagrams for continuous systems there a couple of small differences to the Markov chain state diagram given in Figure 5.2. We no longer include the arrows that point to the same state and we replace the probability of state transfer with a rate of transfer between states (see Figure 5.3).

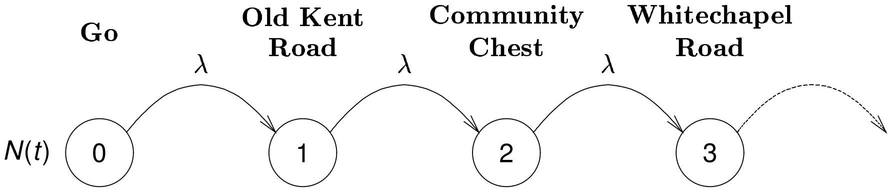


Figure 5.3: State diagram for the Markov process given by Coin Monopoly in the limit as $\Delta t \to 0$. This is the state diagram for a Poisson process. Notice in comparison to Figure 5.2, the probabilities have been replaced by rates and there are no self-transition arrows.

You've now seen that a Poisson process can be derived starting from a simple Markov chain. This gives some deeper insight to a Poisson process. In the coin monopoly model there is a fixed (constant) probability of moving forward in every turn and whether a move forward occurs in each turn is completely independent of what happened in previous turns. The Poisson process is the continuous analogue of this stem. There is a constant probability of an arrival occurring at any time which is independent of what has gone before. This gives you an intuitive explanation of the two properties of the Poisson process given in Sections 4.3.3 and 4.3.4.

Examples

1. It is possible to compute a probability distribution for a number of successes given a number of trials. This distribution is known as the binomial. For n trials (equivalent to n coin tosses or n turns for Coin Monopoly) the probability of k successes (equivalent to k moves forward) is given by

$$P(k) = \left(\begin{array}{c} n \\ k \end{array}\right) P(h)^k (1 - P(h))^{n-k} \tag{5.3}$$

where $\left(\begin{array}{c} n \\ k \end{array}\right) = \frac{n!}{k!(n-k)!}$ is known as the binomial coefficient.

By considering the limiting case as the probability of a success (or heads) goes to zero but the number of trials goes to ∞ show that a Poisson process can be recovered.

Chapter 6

Birth Death Processes and $M/M/1$ Queues

In the last chapter we introduced Markov chains. We reviewed a simple Markov chain in the context of 'Coin Monopoly'. We saw how in the limit as the number of turns goes to infinity and the probability of a 'success' goes to zero the Markov chain became a Markov process. In this chapter we will introduce the concept of a steady state distribution. First we will define the term steady state.

6.1 Steady State

Often we are interested in the steady state behaviour of a system, rather than the transient dynamics. We might ask the question: what is the average network delay when the system has reached steady state? What do we mean by steady state?

1. A stable condition that does not change over time or in which change in one direction is continually balanced by change in another.[1]
2. A steady state is a dynamic equilibrium.

A Markov process, under certain conditions, will reach a steady state distribution over k. Recall that a probabilistic process $p(k|t)$ is a 'noisy function' of time. Some probabilistic processes will converge to a probability distribution when we look at them in the limit as time $\rightarrow \infty$. Not all will though, for example a Poisson *won't*.

The steady state distribution is taken to be

$$\lim_{t \rightarrow \infty} p(k|t) = p(k)$$

the average value of k at steady state is then given by

$$\langle k \rangle_{p(k)} = \sum_{k=0}^{\infty} p(k)\, k.$$

6.1.1 Coin Monopoly Revisited

The Markov chain underlying 'Coin Monopoly' (as presented in the previous chapter) has no associated *steady state* distribution, over time the value of k will always increase. In this chapter we will introduce a slightly more complex Markov process called a birth-death processes. A birth-death process does have a steady state distribution. We will use a condition known as detailed balance to discover this distribution.

[1] `http://www.dictionary.com`.

6.2 Birth Death Process

Let's consider a model for births and deaths in a population. Recall the Markov chain underlying 'Coin Monopoly' (Figure 5.2). Consider a modification to this diagram whereby as well as moving forward in state space (a transition from state equalling k to state equalling $k+1$) you can also move backwards (a transition from state equalling k and state equalling $k-1$). The only restriction we impose is that the state cannot go negative. This system is known as a birth-death process. A birth is an increase in the state k and a death is a decrease.

We represent the discrete version of this system with a state diagrams in Figure 6.1 is with a state diagram as follows. and the continuous version with that in Figure 6.2.

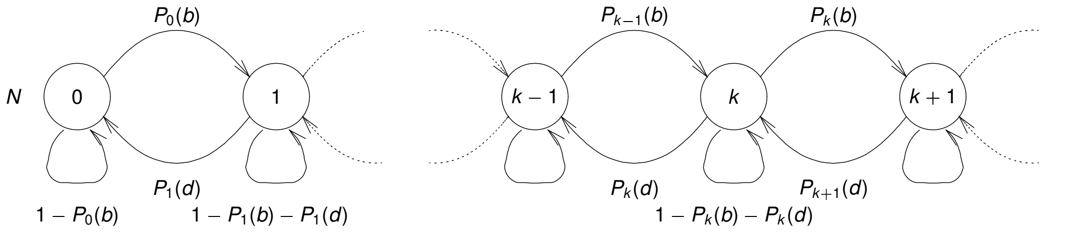


Figure 6.1: State diagram for a birth-death chain.

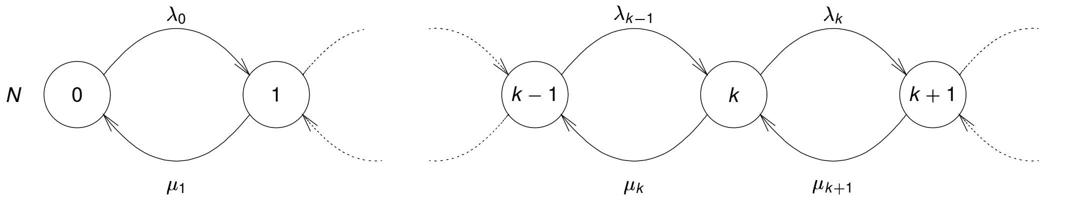


Figure 6.2: State diagram for a birth-death process.

Such processes form the basic model for many different queues. The various different queues are often summarised by a simple notation known as Kendall's notation.

6.3 Kendall's Notation

Kendall's notation provides a simple way of summarising the characteristics of the arrivals and service times of a queue, as well as the number of 'servers' and maximum number of customers. In its simplest form it represents

Arrival characteristics / Departure characteristics / number of servers / max number of customers

The arrival and departure characteristics are summarised by

- M — Markov — Poisson distributed arrivals, exponentially distributed service times.
- D — Deterministically distributed times.
- G — General distribution for inter-arrival/service times.

6.4 Global Balance

We seek a stationary distribution by looking for global balance equations. Global balance equations state that at equilibrium the frequency of transitions out of state k equals the frequency of transitions into k. (pg 260-262 of Bertsekas and Gallager [1992]).

6.4.1 Discrete Time Markov Chain

In its most general form a Markov chain can be defined by a transition probability matrix, this gives the probability of transition from any state to any other. If the probability of transition from state i to state j is given by P_{ij} then the transition probability matrix is given by

$$\mathbf{P} = \begin{bmatrix} P_{00} & \cdots & P_{i0} & \cdots & P_{N0} \\ \vdots & \ddots & & & \vdots \\ P_{0j} & & P_{ij} & & P_{Nj} \\ \vdots & & & \ddots & \vdots \\ P_{0N} & \cdots & P_{iN} & \cdots & P_{NN} \end{bmatrix}.$$

Under global balance, if the steady state probability of being in state j is $P(j)$, then we must have

$$P(j)\sum_{i=0}^{\infty} P_{ji} = \sum_{i=0}^{\infty} P(i)\,P_{ij}$$

which in English says the probability of leaving state j should be equal to the probability of arriving in state j. If this is true then the system is in global balance and there is an associated steady state distribution.

6.4.2 Continuous Time Markov Process

In Markov processes we are interested in rates of transfer rather than probabilities if the rates of transfer between states are given by a matrix whose elements q_{ij} give the rate of transfer between state i and state j, then the global balance equations give

$$P(j)\sum_{i=0}^{\infty} q_{ji} = \sum_{i=0}^{\infty} P(i)\,q_{ij}.$$

6.5 Detailed Balance

For the birth death processes we are considering transitions only occur between state k and $k+1$ or $k-1$. In other words $P_{ij} = 0$ if $i > j+1$ and $i < j-1$. This leads us to a special case of the global balance equations known as the detailed balance equations which look at the flow between two states. For the discrete case

$$P(k-1)\,P_{k-1,k} = P(k)\,P_{k,k-1}$$

and for the continuous case

$$P(k-1)\,q_{k-1,k} = P(k)\,q_{k,k-1}.$$

In our notation rate from k to state $k-1$is given by μ_k and the rate from $k-1$ to k is given by λ_{k-1}so we have

$$P(k-1)\,\lambda_{k-1} = P(k)\,\mu_k.$$

To understand what this represents it is easier to consider the flow rates at a *boundary* between two states, $k-1$ and k. This is shown in Figure 6.5.

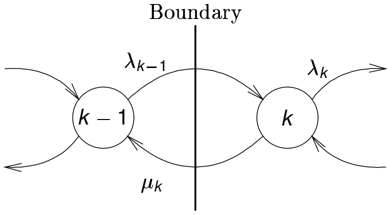


Figure 6.3: Flow rates at a Boundary for a birth-death process.

At steady state, flow from $k-1$ to k is given by $\lambda_{k-1}P(k-1)$. Flow from k to $k-1$ is given $\mu_k P(k)$. For flow to be balanced we have

$$\lambda_{k-1}P(k-1) = \mu_k P(k)$$

leading to

$$P(k) = \frac{\lambda_{k-1}}{\mu_k}P(k-1).$$

For different systems, the birth rate and death rate will differ. We will first study the $M/M/1$ queue which is a birth-death process where the birth rate and death rate are constant regardless of the state of the queue.

6.6 $M/M/1$ Queue

From Kendall's notation (Section 6.3) we know that in an $M/M/1$ that the arrivals occur as a Poisson process, and service times are exponentially distributed. This implies a birth death process with constant birth rate $\lambda_k = \lambda$ (from the Poisson process) and constant death rate (from the exponentially distributed service times) $\mu_k = \mu$. We now give a worked example showing how the stationary distribution is derived using detailed balance. The first step is to draw the state diagram.

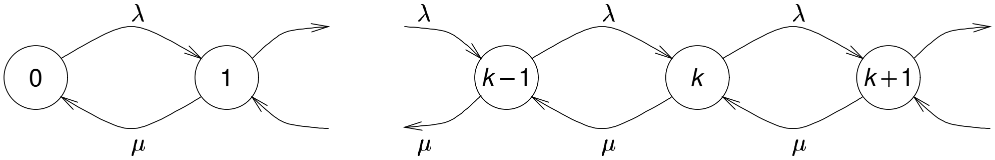


Figure 6.4: State diagram for $M/M/1$ queue.

The state diagram highlights several points.

1. There are no transitions to a state below 0.
2. There is no maximum value for the state shown, this implies that k is unbounded.
3. The birth rates and death rates for each state are indicated.
4. Transitions are only possible between neighbouring states.

Having drawn the state diagram the next step is to analyse the flow rates at a boundary. We analyse between k and $k-1$.

At steady state, the expected flow from $k-1$ to k is given by $\lambda P(k-1)$. Flow from k to $k-1$ is given $\mu P(k-1)$. For flow to be balanced we have

$$\lambda P(k-1) = \mu P(k)$$

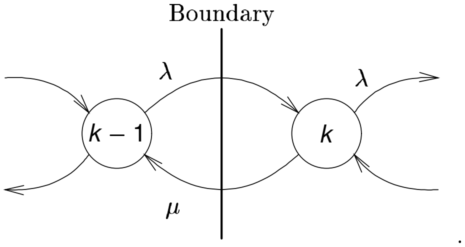


Figure 6.5: Flow rates at a boundary.

leading to

$$P(k) = \frac{\lambda}{\mu} P(k-1).$$

Thus

$$\begin{aligned} P(k) &= \left(\frac{\lambda}{\mu}\right)^k P(0) \\ &= \rho^k P(0), \end{aligned}$$

where we have defined $\rho = \frac{\lambda}{\mu}$, which for the $M/M/1$ queue will turn out to be a useful ratio called the *utilisation factor*.

6.6.1 Normalisation

To find $P(0)$ we exploit our knowledge that, for the $M/M/1$ queue $0 \leq k < \infty$. Given that the steady state distribution *must* be normalised, we can write

$$\sum_{k=0}^{\infty} P(k) = 1$$

and therefore

$$1 = P(0) \sum_{k=0}^{\infty} \rho^k$$

giving

$$P(0) = \frac{1}{\sum_{k=0}^{\infty} \rho^k}. \tag{6.1}$$

We are once again faced with an infinite series in the denominator of (6.1). This series converges if $\rho < 1$ (when the birth rate is less than the death rate),

$$\sum_{k=0}^{\infty} \rho^k = \frac{1}{1-\rho} \text{ for } \rho < 1,$$

meaning that

$$P(0) = (1-\rho). \tag{6.2}$$

We can now write down the full distribution for the *steady state* probabilities

$$P(k) = \rho^k (1-\rho).$$

6.6.2 Faster Arrivals than Service

Notice that if the arrival rate (birth rate) is greater than the service rate (death rate) then $\rho > 1$ and the series in (6.1) doesn't converge. This is indicative of the fact that if the birth rate is greater than the death rate there is no steady state distribution: the queue simply continues to grow in size.

6.6.3 Average Number in System

For network design we might be interested in the average queue length. We can use the steady state probabilities to compute this quantity.

The average number of customers in the system *is not* the same as the average queue length because a customer being served (*e.g.* a packet under transmission) is not in the queue.

The average number of customers in the system is given by the expected state of the system,

$$\langle k \rangle_{P(k)} = \sum_{k=0}^{\infty} kP(k)$$

in the examples section at the end of this chapter we show how this average can be computed as

$$\begin{aligned} \langle k \rangle_{P(k)} &= \frac{\rho}{1-\rho} \\ &= \frac{\lambda}{\mu - \lambda}. \end{aligned}$$

The variance can also be computed, it is given by

$$\mathrm{var}\,[k] = \left\langle k^2 \right\rangle - \langle k \rangle^2 = \frac{\rho}{(1-\rho)^2}.$$

Computation of the variance is an exercise in Tutorial Sheet 2.

As an exercise on the tutorial sheet, you will show that the variance of the number in the system is given by

$$\mathrm{var}\,[k] = \left\langle k^2 \right\rangle - \langle k \rangle^2 = \frac{\rho}{(1-\rho)^2}.$$

6.7 Discrete Simulation of $M/M/1$

In the first lab you will run a discrete simulation of an $M/M/1$ queue. A discrete simulation for these queues is easy to run because it takes the form of a Markov chain. The state diagram for this chain is shown in Figure 6.6. The simulation consists of a loop over discrete time steps. For each time step there are three possible events: the state of the system increases by one; the state of the system decreases by one and the state of the system stays the same. Modelling the queue is therefore a simple matter of testing which of these outcomes is true.

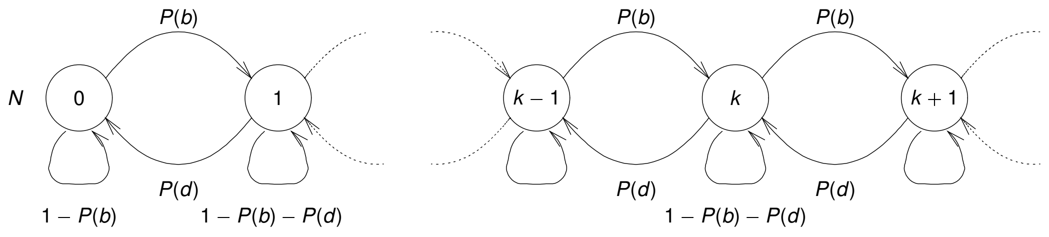


Figure 6.6: State diagram for the discrete version of the $M/M/1$ queue.

6.7.1 Approximating a Continuous $M/M/1$ Queue

We know from our derivation in Section 5.2 that a Poisson process can be derived by considering the limit, as the discrete time intervals go to zero, of a simple Markov chain. A continuous $M/M/1$ queue is also the limit of the discrete model we are using for simulation. In your lab class you can make this approximation by taking Δt to be much smaller than λ or μ. Then you may set $P(b) = \lambda \Delta t$ and $P(d) = \mu \Delta t$.

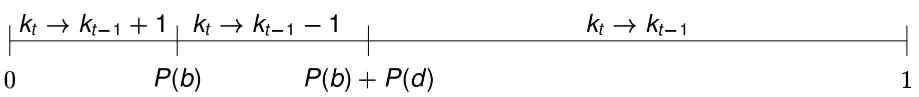


Figure 6.7: Testing for one of three outcomes. Generate a random number between 0 and 1 (in MATLAB use `rand`), see where the result lies on the line above. If it is less than $P(b)$ then state of queue increases by 1, if it is greater than $P(b)$ but less than $P(b) + P(d)$ *and the queue is not empty* then state of queue decreases by one, otherwise queue state stays the same.

6.7.2 Demonstration

In your lab class you will run a simulation of the discrete queue and answer questions on the analysis of that queue. To help you understand the simulation process we will demonstrate a simulation in the lecture. We also give you (below) histograms that are similar to those you should be producing in the lab class.

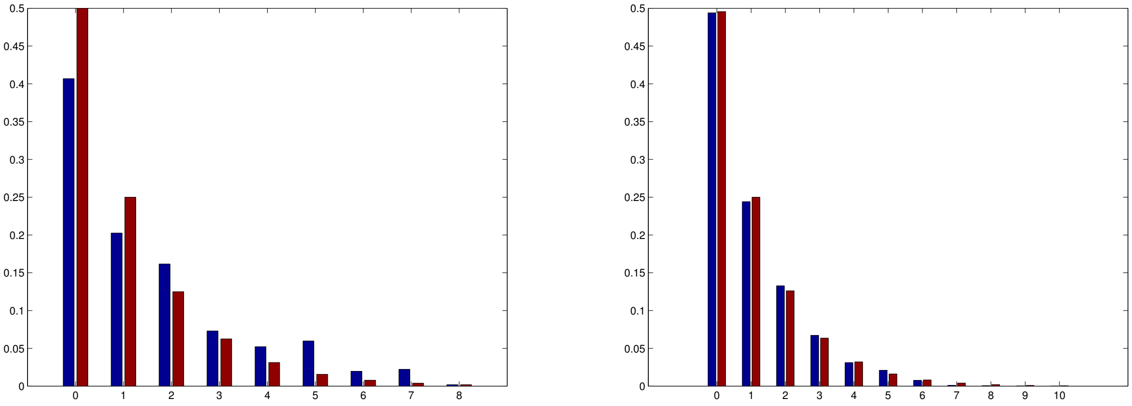


Figure 6.8: Left figure: steady state probabilities after the first 100 seconds for $\rho = 0.5$. Right figure: steady state probabilities after the first 1000 seconds for $\rho = 0.5$.

Examples

1. The mean number in the system for the $M/M/1$ queue is given by considering the expectation under the steady state distribution. Compute this value.

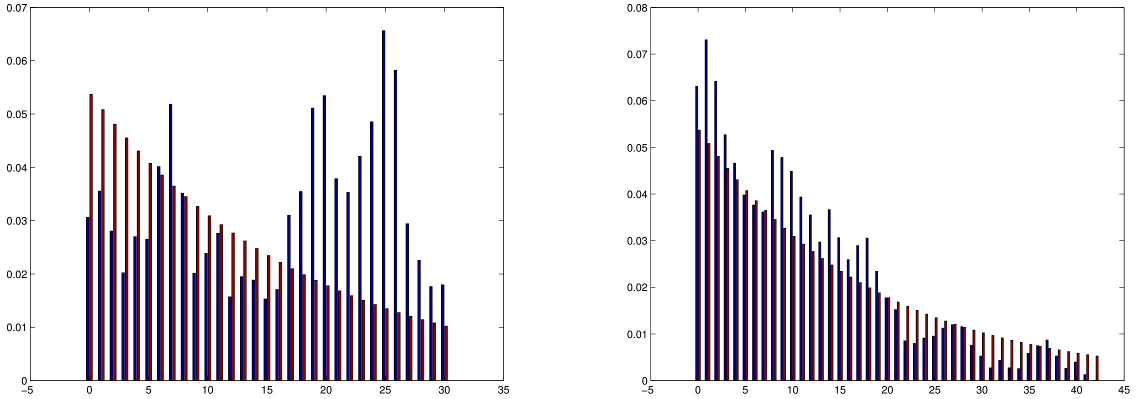


Figure 6.9: Left figure: steady state probabilities after the first 100 seconds for $\rho = 0.95$. Right figure: steady state probabilities after the first 1000 seconds for $\rho = 0.95$.

Chapter 7

Little's Formula

In the last chapter we introduced birth-death processes and demonstrated how we could compute the average number in the system for the stationary distribution. An important quality of service issue is the average delay associated with the system. We would also like to be able to compute the average number in the queue and the average time spent queueing (as opposed to queueing and being served).

In this chapter we introduce Little's formula, a very general formula which can be applied to $G/G/1$ queues. It gives the relationship between the steady-state average number of customers in the system, $\bar{N}$, the steady-state arrival rate, λ, and the steady-state customer delay, $\bar{T}$. First of all we will introduce a new type of average, the time based average, then we introduce Little's formula and review a simple derivation of the formula which highlights the conditions under which it applies.

7.1 Time and Steady State Averages

We have already introduced the concept of distribution means and sample based means. In this section we introduce a further type of average, a time based average. The time average is the average value of a particular function over time. If that function is continuous it is given by

$$\bar{f}_t = \frac{1}{t}\int_0^t f\left(\tau\right) d\tau.$$

The derivation of Little's formula that follows will rely on an assumption. We will assume that averages under the steady state distribution are equivalent to the time average in the limit as $t \rightarrow \infty$, where if the state of the system is given as a function of time as $N\left(t\right)$ the average number in the system as a function of time is given by

$$\bar{N}_t = \frac{1}{t}\int_0^t N\left(\tau\right) d\tau$$

and the time average after in the limit as $t \rightarrow \infty$ is given by

$$\bar{N} = \lim_{t\rightarrow\infty} \bar{N}\left(t\right).$$

which we assume is equivalent to $\left\langle k\right\rangle_{p(k)} \equiv \bar{N}$. See pg 156-157 in Bertsekas and Gallager [1992] for more details.

7.2 The Formula

Little's formula applies under steady state conditions (see Section 6.1). Under the steady state assumptions it states,

$$\bar{N} = \lambda\bar{T}.$$

In practice we can see what this means by considering a simple example of a nightclub queue.

Question

You are driving past a nightclub, wondering how long you will have to wait to go in. You notice on Thursday nights that the queue typically contains 20 people and that people arrive at a rate of 40 per hour.

According to Little's formula, how long would you expect to have to wait to get in?

Answer

The waiting time is given by Little's formula. $\bar{N} = 20$, $\lambda = 40$ and $\bar{T} = \frac{\bar{N}}{\lambda} = \frac{20}{40} = \frac{1}{2}$hour.

7.2.1 Intuition

When phrased in these terms the formula seems fairly obvious. If we see there are twenty people in a queue and they enter a building at a rate of 40 per hour, we might expect to have to wait half-an-hour in the queue. In the examples section at the end of this chapter we build on this intuition with a simple graphical 'proof' of the formula based upon that given in Bertsekas and Gallager [1992].

The formula applies regardless of the arrival and service time distributions, it only relies on the system being in steady state. Little's formula can be used at the level of the entire system (queueing time and service time), as we have seen, or it may be applied at a lower level.

7.3 Average Delay in $M/M/1$

In the last chapter we computed that the average state of an $M/M/1$ queue was given by

$$\bar{N} = \frac{\lambda}{\mu - \lambda} = \frac{\rho}{1 - \rho}.$$

We can use Little's formula to tell us the average delay associated with the system. From $\bar{N} = \lambda \bar{T}$ we rearrange to recover the average delay as

$$\bar{T} = \frac{1}{\mu - \lambda}$$

This is not the time spent in the queue however, this is the time spent in the system which includes *propagation delay*, the delay in transmitting the packet.

7.3.1 Little's Formula for Queueing Time and Transmission Time

The total delay for packet switching systems can be divided into queueing time, W, and service time or transmission time, X. $T = X + W$. Similarly the average delay may be decomposed as

$$\bar{T} = \bar{X} + \bar{W}.$$

Furthermore, both the queue itself and the transmission may the be analysed independently using Little's formula.

7.3.2 The Transmission

If we assume that customers/packets do not leave the queue then the rate of arrivals at the queue must equal the rate of arrivals at transmission. If we define the the average number under transmission as ρ then we can write

$$\rho = \lambda \bar{X},$$

where $\bar{X}$ is the average time spend under transmission for each packet. For the $M/M/1$ queue the average transmission time is given by the mean of an exponential distribution, $\bar{X} = \frac{1}{\mu}$, so for the $M/M/1$ queue we have

$$\rho = \frac{\lambda}{\mu},$$

a formula which we have already used in the last chapter.

The variable ρ is known as the utilisation factor. It represents the fraction of time that the transmission line (or server) is in use and is therefore a measure of efficiency. Note that it is also the steady state probability that the transmission line is in use. For a lossless system (one that doesn't drop packets) this probability is given by $1 - P(0)$. You can check that this is consistent for the $M/M/1$ by looking at equation (6.2).

Note: time under transmission is often considered proportional to packet length. For packet length of L bits and bandwidth of C bits per second transmission time is $X = \frac{L}{C}$.

7.3.3 The Queue

Given λ, an arrival rate, $\bar{N}_Q$, the average number of packets waiting in the queue, and $\bar{W}$, the average packet waiting time, Little's formula gives

$$\bar{N}_Q = \lambda \bar{W}$$

for the number in the queue. For the $M/M/1$ queue $\bar{T} = \bar{X} + \bar{W} = \frac{1}{\mu} + \bar{W}$ so the average time spent waiting in the queue is given by

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

We can therefore recover the average number in the queue

$$\bar{N}_Q = \frac{\rho^2}{1 - \rho}.$$

Examples

1. **Graphical Proof of Little's Formula.** In this example we provide a graphical proof of Little's formula.

2. **Simple example of Little's formula**. You are at a queue in a bar and notice that the bar tender spends 10% of her time cleaning between serving customers. If the typical number of people waiting to be served is 3, what is the ratio of time spent waiting to being served?

Chapter 8

Other Markov Queues

We have already met the $M/M/1$ queue, the simplest Markov queue. In this chapter we will introduce related Markov queues and compute their stationary distributions. The simplest modifications to the $M/M/1$ queue are given by a *finite length* modification which we write in Kendall's notation as $M/M/1/m$ and the *infinite server* queue, $M/M/\infty$.

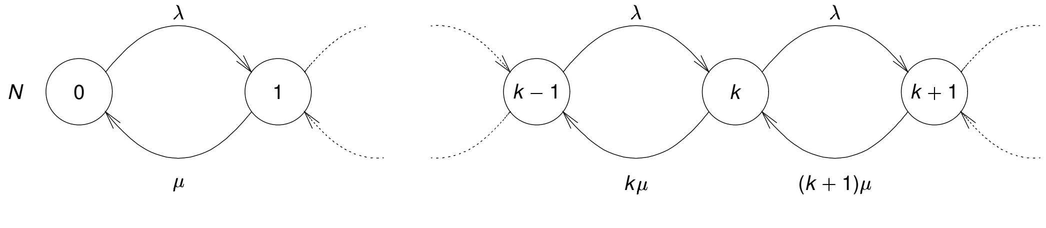
8.1 Finite Length $M/M/1/m$

In our analysis of the $M/M/1$ queue so far we have assumed that the queue can have an infinite length, in practice a restricted buffer size may mean that the queue length is also restricted. Let's assume the queue can contain at maximum $m-1$ customers (as shown in Figure 8.3). This means (because one customer can be under service) there is a maximum of m customers in the system. In the Example 1 at the end of this chapter we derive the stationary distribution for this queue. Then, in Example 2, we introduce the concept of blocking probability. The $M/M/1/m$ queue has limited capacity, that means it is a queue with 'loss'. The carried load is *not the same* as the offered load because customers are rejected when the queue is full. These customers are assumed not to return.

8.2 Infinite Serving Capacity — $M/M/\infty$

In the $M/M/1$ queue we assume that there is only one server. In some cases it can be useful to consider an unlimited number of servers so that there is one server for every customer in the system. The state diagram for this system can be drawn as in Figure 8.1

.



. .

Figure 8.1: State diagram for the $M/M/\infty$ queue.

8.2.1 Detailed Balance

We again make use of the standard detailed balance equations for this queue. From the flow rates

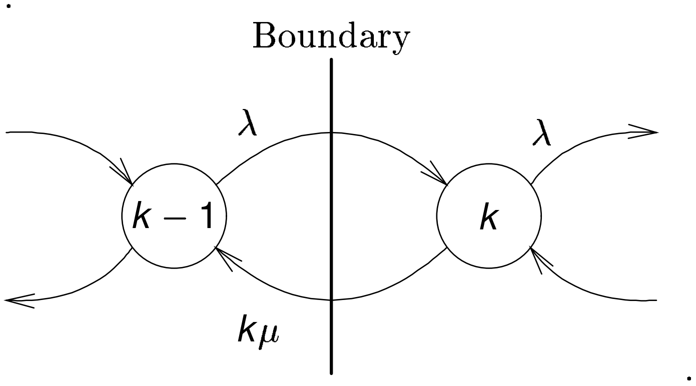


Figure 8.2: Flow rates at a boundary for $M/M/\infty$ queue.

at the boundary we have, in general,

$$\lambda_{k-1}P\left(k-1\right)=\mu_k P\left(k\right).$$

Since there is infinite service capacity in the queue the service rate at state k is simply $\mu_k = k\mu$. The rate of arrivals stays constant, $\lambda_k = \lambda$, so for this queue detailed balance gives.

$$\lambda P\left(k-1\right)=k\mu P\left(k\right)$$

Reorganising in terms of $P\left(k\right)$ then gives

$$P\left(k\right)=\frac{\lambda}{k\mu}P\left(k-1\right)$$

and therefore

$$P\left(k\right)=P\left(0\right)\left(\frac{\lambda}{\mu}\right)^k\frac{1}{k!}.$$

8.2.2 Normalisation Condition

Using the normalisation condition for this queue,

$$\sum_{k=0}^{\infty}P\left(k\right)=1,$$

we obtain $P\left(0\right)$ as

$$P\left(0\right)=\left[\sum_{k=0}^{\infty}\left(\frac{\lambda}{\mu}\right)^k\frac{1}{k!}\right]^{-1}. \qquad (8.1)$$

8.2.2.1 The Offered Load

It is convenient in this (and later queues) to define the offered load, $a=\frac{\lambda}{\mu}$, as arrival rate divided by service rate. This is a dimensionless constant, but has international standard units of Erlangs (after A. K. Erlang). Substituting this definition back into (8.1) we have,

$$P\left(0\right)=\left[\sum_{k=0}^{\infty}\frac{a^k}{k!}\right]^{-1}.$$

Recalling from Section 3.6.1 that

$$\sum_{n=0}^{\infty}\frac{x^n}{n!}=\exp\left(x\right)$$

we have

$$P(0) = \exp(-a)$$

and the steady state distribution is then

$$P(k) = \frac{a^k}{k!}\exp(-a)$$

which is recognised as a Poisson distribution with parameter a. Notice that the utilization factor for this queue, ρ, is *not* equal to $\frac{\lambda}{\mu}$ (which is referred to as the offered load). In fact for the $M/M/\infty$ queue the offered load can be much larger than 1 but we know that the utilization factor is never greater than 1. The utilization factor doesn't make much sense for this queue because there are infinite servers: but we can study the probabililty that the system is in use, which is one minus the probability that the queue is empty,

$$1 - P(0),$$

The proportion of time that any portion of the system is in use is therefore

$$1 - P(0) = 1 - \exp(-a).$$

Examples

1. Derive the steady state distribution for the $M/M/1/m$ queue.

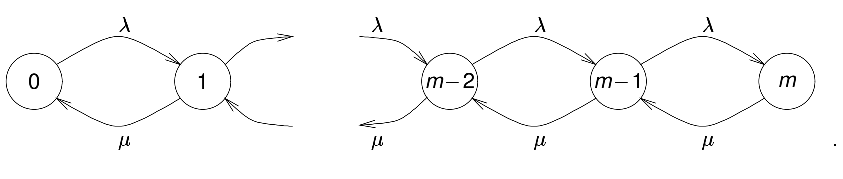


Figure 8.3: State diagram for the $M/M/1/m$ queue. Note that there are no transitions to states above the mth state.

2. **Blocking Probabilities.** Given a finite length $M/M/1/m$ queue, what is the probability that an arrival is blocked from joining the queue (and therefore lost).

Chapter 9

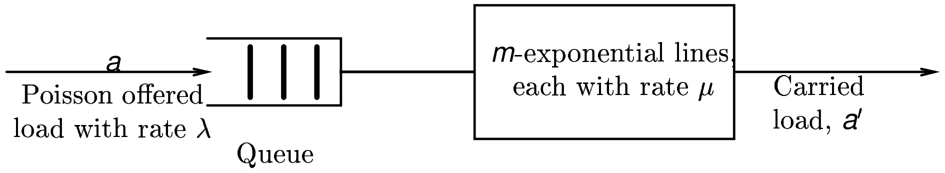
Erlang Delay $M/M/m$

In the last chapter we considered the situation were there are as many servers as there are customers. More realistic real world situations involve limited numbers of servers. This commonly arises in, for example, call centres. This system is known as the *Erlang delay* system. In Kendall's notation it is written $M/M/m$.

9.1 Stationary Distribution

In the context of the Erlang delay system it is usual to refer to the 'average holding time', $\tau = \frac{1}{\mu} \equiv \bar{X}$, which, for a network involving telephones, would be the time a call is open.

We give a schematic overview of an Erlang delay system in Figure 9.1. It shows the offered load, $a = \frac{\lambda}{\mu}$, arriving in the system through a queue. The system is *lossless*, so the carried load (sometimes denoted a') is also equal to a.



a
Poisson offered load with rate λ
Queue
m-exponential lines, each with rate μ
Carried load, a'

Figure 9.1: Queueing model of the Erlang delay system.

9.1.1 State Diagram

As with all the Markov based systems we have met so far the first step in their analysis is to write down their state diagram.

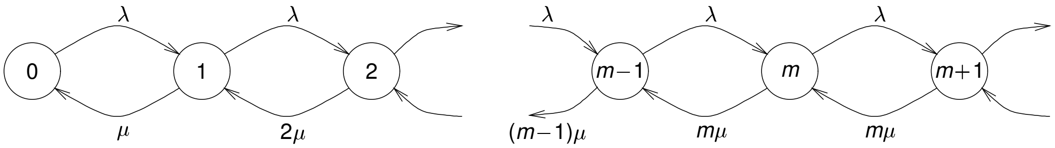


Figure 9.2: State diagram of an Erlang delay system.

The Erlang loss system may be analysed as a birth-death process with birth rate

$$\lambda_k = \lambda$$

and death rate

$$\mu_k = \begin{cases} k\mu & k = 0, 1, \ldots, m-1 \\ m\mu & k > m \end{cases}$$

we derive the detailed balance equations in Example 1 at the end of the chapter, here we give the result. The overall stationary distribution is

$$P(k) = \begin{cases} P(0)\frac{(m\rho)^k}{k!}, & k \leq m \\ P(0)\frac{m^m \rho^k}{m!} & k > m \end{cases}$$

where the normalisation constant is given by

$$P(0) = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!(1-\rho)}\right]^{-1}.$$

9.2 Probability of Waiting

For the $M/M/m$ queue a key quantity of interest is the probability of of a customer queueing, P_Q. What is the probability, when the system is at steady state, that a customer will arrive and find all the m servers busy?

$$\begin{aligned} P(k \geq m) &= \sum_{k=m}^{\infty} P(k) \\ &= P(0)\sum_{k=m}^{\infty} \frac{m^m \rho^k}{m!} \\ &= P(0)\frac{(\rho m)^m}{m!}\sum_{k=m}^{\infty} \rho^{k-m} \doteq C(m,a), \end{aligned}$$

with a little manipulation (check Example 1) this expression can be re-written

$$C(m,a) = P(0)\frac{(\rho m)^m}{m!(1-\rho)}. \tag{9.1}$$

This is known as the *Erlang C formula*, or Erlang's second formula after A.K. Erlang a pioneer of queueing theory. By making use of our definition of a (from Section 8.2.2.1), we can rewrite the offered load as

$$a = \frac{\lambda}{\mu} = m\rho$$

and substituting into (9.1) we have

$$C(m,a) = \frac{P(0)a^m}{m!\left(1-\frac{a}{m}\right)}.$$

The result of this formula can be computed on a computer: but some care must be taken when computing it. In particular when m is large overflow problems will occur for a^m and $m!$. If we naively compute the ration $\frac{a^m}{m!}$ we will obtain an overflow error. Instead we can compute the formula in log space,

$$\ln C(m,a) = \ln P(0) + m\ln a - \ln m! - \ln\left(1 - \frac{a}{m}\right).$$

Some care still needs to be taken with the $\ln P(0)$ term, but each of the other terms can be computed with appropriate functions in MATLAB. To compute a log factorial the log of the *gamma function* is used.

When we need to compute the Erlang C curve, for the purposes of exams and exercises, we will use the old fashioned method of curves.

9.3 Erlang C Curves

We have provided you with two sets of curves in Figure 9.3 and 9.4. In an exam, if they are needed, curves will be provided with the exam paper. Make sure you mark the curves and hand them in with your paper. To illustrate the use of the curves we will consider a simple telephone trunking example and a customer service centre example, see Examples 2 and 3 at the end of this chapter.

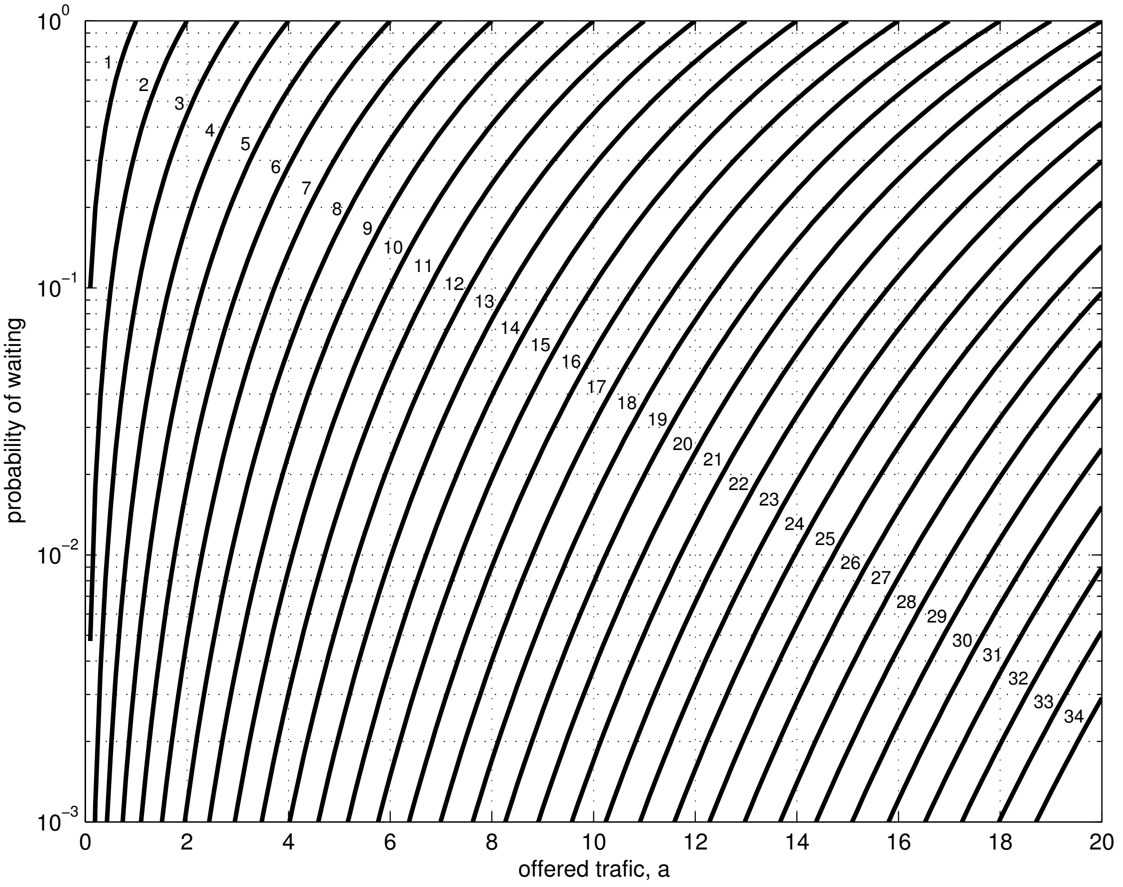


Figure 9.3: Erlang C curves for offered traffic between 0 and 20.

9.3.1 Average number in System.

By computing the expectation of k under the stationary distribution, we can also compute the average number of calls in the Erlang Delay System. The result of this computation is:

$$\bar{N} = \frac{\rho}{1-\rho} C(m,a) + m\rho.$$

These terms represent the average queue length and the average number of busy lines. The average queue length is given by

$$\bar{N}_Q = \frac{\rho}{1-\rho} C(m,a)$$

so using Little's formula the average queueing time is simply

$$\bar{W} = \frac{\bar{N}_Q}{\lambda} = \frac{C(m,a)}{m\mu(1-\rho)}.$$

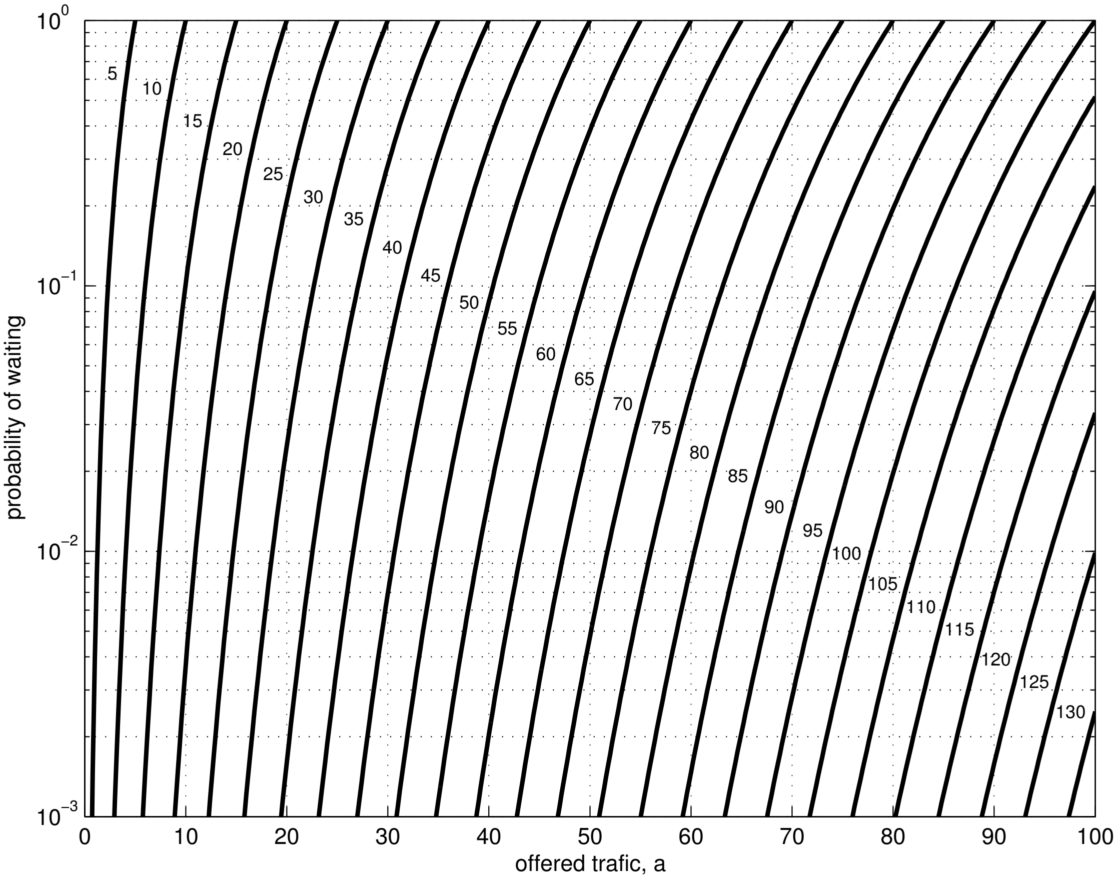


Figure 9.4: Erlang C curves for offered traffic between 0 and 100.

(see Example 4) . Note that this is the average queueing time for all customers (whether they queued or not). It might be more informative to compute the average queueing time for those who actually queue. This can be easily done using the fact that the average queueing time for those who don't queue is zero. The average queueing time can be decomposed as

$$\bar{W} = C(m,a)\,\bar{W}_Q + (1 - C(m,a))\,\bar{W}_{\sim Q},$$

where $\bar{W}_Q$ is the average queueing time for those who do queue and $\bar{W}_{\sim Q}$ is the average queueing time for those who don't queue. Since $\bar{W}_{\sim Q} = 0$ the second term in this sum is zero, so we can rewrite in terms of $\bar{W}_Q$ as

$$\bar{W}_Q = \frac{\bar{W}}{C(m,a)} = \frac{1}{m\mu(1-\rho)} = \frac{1}{\mu(m-a)}.$$

Note that the result shows that the average queueing time for those who do queue is *independent* of the probability of queueing.

Examples

1. **Erlang Delay Stationary Distribution.** Derive the stationary distribution for the Erlang delay system.

2. Consider the telephone trunking problem. A telephone exchange A is to server 10,000 subscribers in a nearby exchange as shown in Figure 9.5. Suppose that during the busy hour,

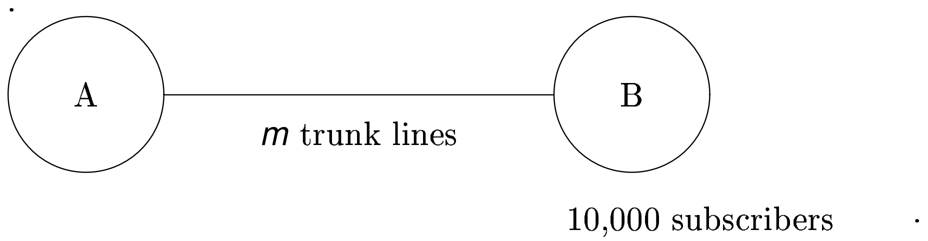


Figure 9.5: The telephone exchange.

the subscribers generate a Poisson traffic with a rate of 10 calls per minute, which requires trunk lines to exchange B for an average holding time of 3 minutes per call. Determine the number of trunk lines needed for a grade of service (probability of waiting) of 0.01.

3. A customer service centre has an average holding time, τ, for calls of 8 minutes. In the busy hour, calls arrive at the centre at a rate of one very two minutes. Inter-arrival times and service times are exponentially distributed, how many personnel are required to keep the probability of waiting at 0.1.

4. What is the average queueing time for Example 3 above? What is the average queueing time for those customers who do queue?

Chapter 10

Erlang Loss $M/M/m/m$

The Erlang delay system contained m servers, and it was assumed that, if all servers were occupied, the customer would be placed on hold until a server became available. In the $M/M/m/m$ system there is a finite number of customers.

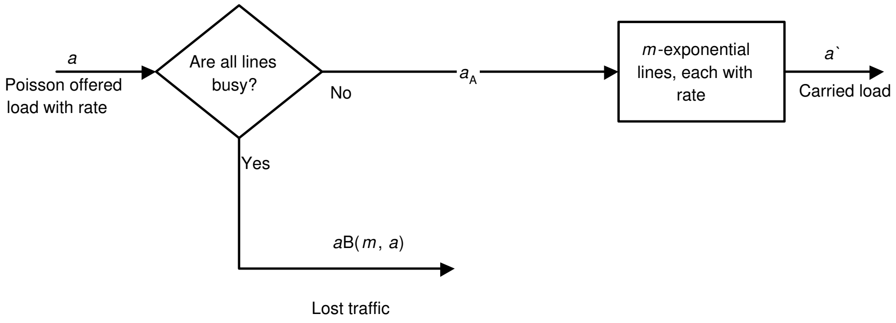

a
Poisson offered load with rate
Are all lines busy?
No
aA
m-exponential lines, each with rate
a`
Carried load
Yes
aB(m, a)
Lost traffic

Figure 10.1: Flow chart of the Erlang Loss System.

The system is a limited capacity system. The maximum number in the system is m which means the maximum queue size is zero because there are m servers. In the diagram it shows the offered load is a, while the carried load, a', is less than a because some packets are rejected, just as they were for the $M/M/1/m$ queue.

10.1 Stationary Distribution

The system is shown as a flow chart in Figure 10.1. The Erlang loss system may be analysed as a birth-death process with birth rate

$$\lambda_k = \begin{cases} \lambda & k = 0, 1, \ldots, m-1 \\ 0 & k \geq m \end{cases}$$

and death rate

$$\mu_k = \begin{cases} k\mu & k = 0, 1, \ldots, m \\ 0 & k > m \end{cases}$$

Note: when $k > m$, it doesn't matter what μ_k is as we never reach this state.

10.1.1 State Diagram

The offered load, $a = \frac{\lambda}{\mu}$, is again dimensionless with units of Erlangs, however to compute the carried load we will first have to determine what the probability of a customer being blocked is.

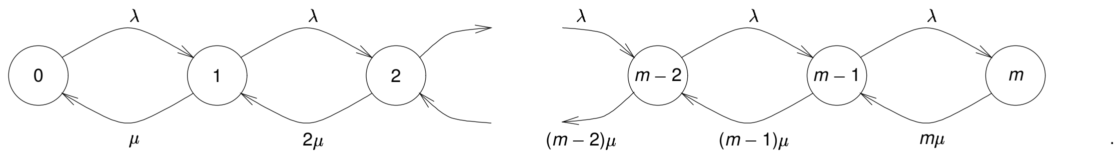


Figure 10.2: Markov chain for the Erlang loss system, $M/M/m/m$

10.2 Steady State Probabilities

When the system's state, k, is less than or equal to the number of servers we have boundary flow as in Figure 10.3, which leads to the following balance equation

$$\lambda P(k-1) = k\mu P(k).$$

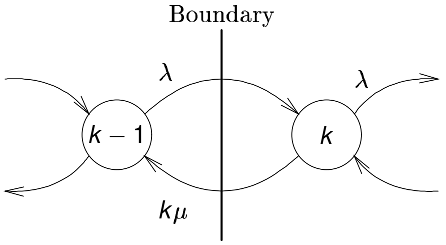


Figure 10.3: Flow rates at a boundary.

As a result, for $k = 0 \ldots m$ we may write

$$P(k) = \frac{\lambda}{k\mu} P(k-1). \tag{10.1}$$

In contrast to the $M/M/m$ queue, the system state cannot go above m so we have no need to consider the region where $k > m$. Substituting in the offered load for $\frac{\lambda}{\mu}$ we have,

$$P(k) = \frac{a}{k} P(k-1),$$

which means that

$$\begin{aligned} P(k) &= \prod_{n=1}^{k} \frac{a}{n} P(0) \\ &= \frac{a^k}{k!} P(0). \end{aligned}$$

10.2.1 Normalisation Condition

The normalisation condition gives us

$$P(0) \sum_{k=0}^{m} \frac{a^k}{k!} = 1$$

$$P(0) = \frac{1}{\sum_{k=0}^{m} \frac{a^k}{k!}},$$

which leads to the *truncated Poisson distribution*:

$$P(k) = \frac{\frac{a^k}{k!}}{\sum_{n=0}^{m} \frac{a^n}{n!}}.$$

Note if $m \to \infty$ we recover the $M/M/\infty$ queue as we would expect.

10.3 Probability of Blocking

For the $M/M/m$ queue we were interested in the probability that a customer would have to wait in the queue for service. In the $M/M/m/m$ system there is no queue because the maximum state of the system is the same as the number of servers. The probability of waiting is therefore zero. Of more interest is the probability of blocking: the probability that when we arrive the system is full and we are denied service. As we discussed for the $M/M/1/m$ queue the blocking probability is given by $P(m)$, the steady state probability that the system is full. This is given as

$$B(m, a) = P(m) = \frac{\frac{a^m}{m!}}{\sum_{n=0}^{m} \frac{a^n}{n!}}.$$

We have already discussed for the Erlang C system the problems of computing this quantity on a computer. They may be resolved in much the same way through computation in log space. Again we will turn to curves for use in exam questions and exercises. Erlang B curves plot the blocking probability for different numbers of servers as a function of the offered load.

10.4 Carried Load and Utilization Factors

Since traffic is lost in $M/M/m/m$ the carried load, a', is smaller than the offered load. The carried load is given by the offered load multiplied by the probability that the packet is not blocked.

$$a' = a(1 - P(m)) = a(1 - B(m, a)).$$

The carried load per line is the trunk occupancy,

$$\rho = \frac{a'}{m},$$

also known as the utilization factor. Note the contrast with the lossless systems where the trunk occupancy (or utilization factor) was given by

$$\rho = \frac{a}{m}$$

because the carried load is equal to the offered load $a' = a$. When computing utilization factors for lossy systems we have to take this loss into account (for example in the $M/M/1/m$ queue). Also note that the rate of service in the system is no longer given by the arrival rate, λ, we need to consider those packets that are lost. This means the rate of service is given by

$$\dot{\lambda} = \lambda(1 - P(m))$$

where $P(m)$ is the blocking probability. This has effects when studying the queue using Little's formula. When using Little's formula with a lossy system, the rate we use should be the service rate, *not* the arrival rate. Packets which are blocked by the system do not enter it and therefore have no effect on the system's state.

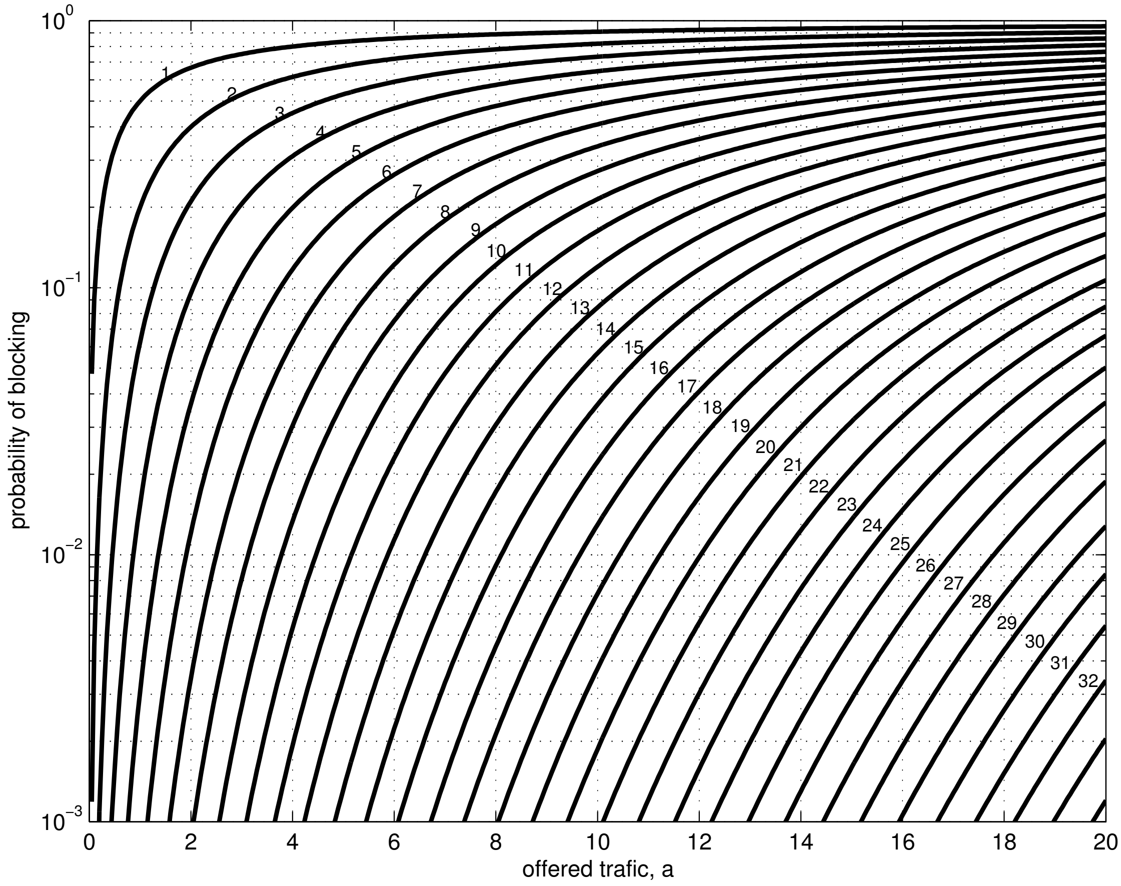


Figure 10.4: Erlang B curves.

Examples

1. **Load Tripling**. In this example we will briefly consider the effect of increasing load in an Erlang Loss system.

   (a) For a blocking probability of 0.01, what is the number of trunks needed for 6 Erlangs of offered load?

   (b) For a blocking probability of 0.01, what is the number of trunks needed for 18 Erlangs?

   (c) Comment on the increase in load compared to the increase in the number of trunks.

2. **Example** $M/M/m/n$. This example is from the Exam paper in 2002-3.

   A call centre employs thirty operators selling tickets for a major sporting event. The call centre has sixty lines in total. When all operators are active new calls are placed in a queue for the first available operator. The call centre is to be modelled as an $M/M/m/n$ queue, *i.e.* a queue in which there are m servers and a maximum state of n.

   (a) Draw a state diagram representing the queueing system.

   (b) Write down the detailed balance equations for the two cases where the queue state $k < m$ and when $m \leq k < n$.

   (c) For each of the two cases outlined in the previous part write down the steady state distribution as a function of $P(0)$, the probability of the system being empty.

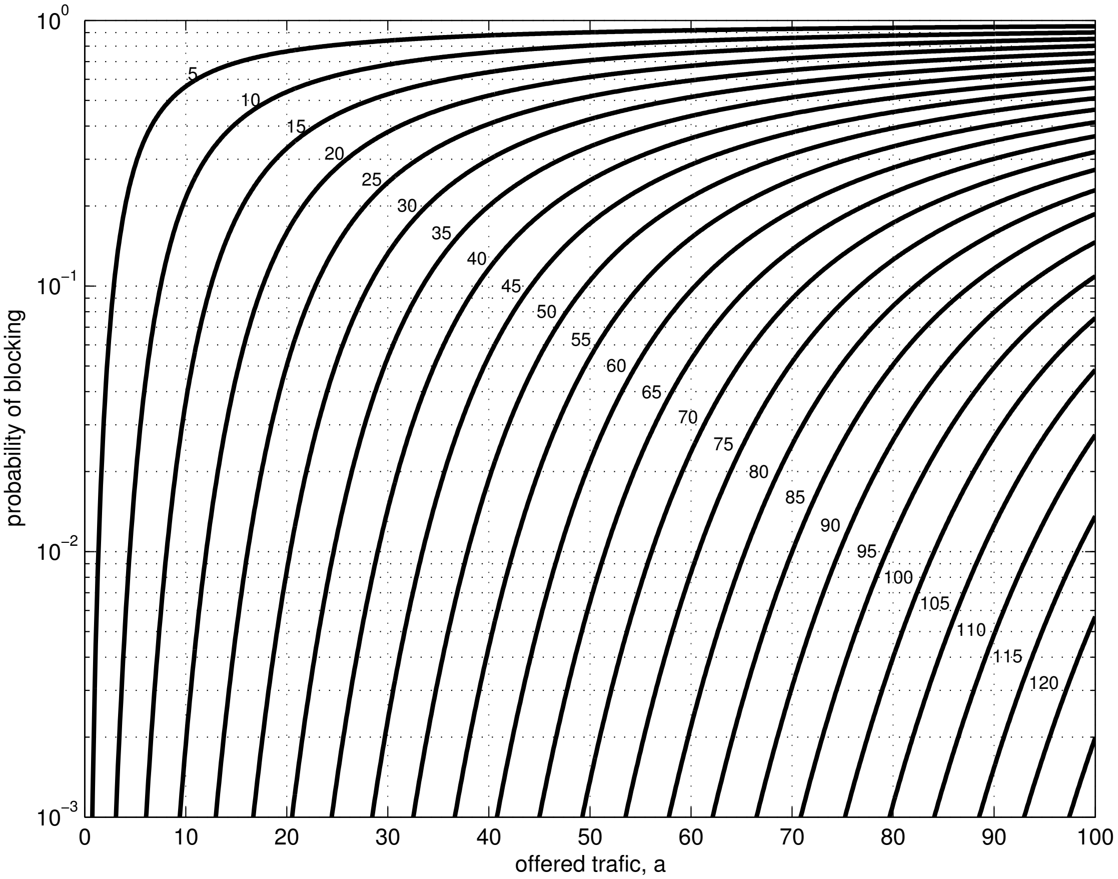


Figure 10.5: Erlang B curves.

(d) Use the fact that the steady state distribution must be normalised to show that

$$P(0) = \left[\sum_{k=0}^{m} \frac{a^k}{k!} + \sum_{k=m+1}^{n} \frac{a^k}{m! m^{k-m}} \right]^{-1}$$

where $a = \frac{\lambda}{\mu}$, λ is the arrival rate and μ is the rate of the service time distribution.

Chapter 11

$M/G/1$ Queue

If service times are *generally* distributed we cannot use the birth death process to analyse the system. In this chapter we introduce a different type of analysis based on the *mean residual service* time. This will allow us to derive the *Pollaczek-Khinchin* formula. This formula gives the expected queueing time for any distribution of service times with *finite* variance.

11.1 Pollaczek-Khinchin formula

The Pollaczek-Khinchin formula is given as

$$\bar{W} = \frac{\lambda \left\langle X^2 \right\rangle}{2(1-\rho)}$$

where $\bar{W}$ is expected waiting time,$\left\langle X^2 \right\rangle = \left\langle X^2 \right\rangle_{p(X)}$ is the second moment under the service time distribution, the utilisation factor is given as $\rho = \lambda \bar{X}$ and $\bar{X} = \left\langle X \right\rangle_{p(X)}$ is the mean under the service time distribution.

11.2 Proof of Pollaczek-Khinchin Formula

In the examples, we review the first part of a simple proof of the Pollaczek-Khinchin formula. In our proof we will make three assumptions:

1. The queue is FIFO queue, *i.e.* customers are served in the order they arrive. This assumption may be relaxed but it will simplify our proof.

2. The arrival times are statistically independent of the service times.

3. The arrival time distribution has a variance which is finite.

To formulate the proof we introduce a concept known as the mean residual service time, $\bar{R}$. We show that the average waiting time is related to the mean residual service time as follows:

$$\bar{W} = \frac{\bar{R}}{1-\rho} \tag{11.1}$$

We now have the average queueing time in terms of the mean residual service time. To complete the Pollaczek-Khinchin formula we simply need to calculate $\bar{R}$.

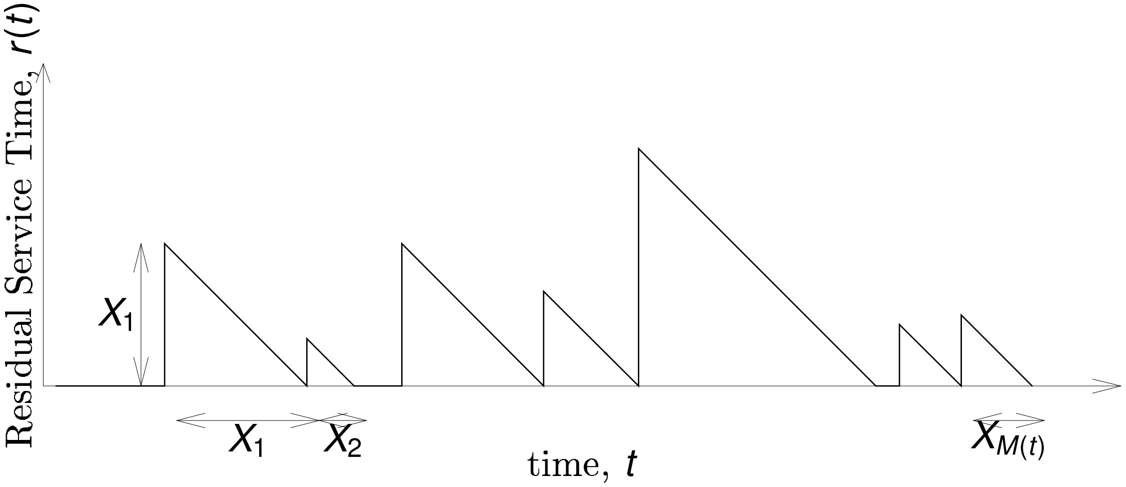


Figure 11.1: Graphical computation of mean residual time. $M(t)$ is the number of service completions in the interval $[0,\ t]$.

11.3 Residual Service Time from Departures Perspective

We again turn to a graphical proof to obtain the value of the mean residual service time. In Figure 11.1 we present a plot of the residual service time over time.

The time average of the mean residual service time can then be calculated in terms of the area under the curve, $A(t)$, which is given by the area under each of the triangles,

$$A(t) = \frac{1}{2}\sum_{i=1}^{M(t)} X_i^2.$$

The average service time is the area divided by time, t,

$$\begin{aligned}\frac{A(t)}{t} &= \frac{M(t)}{t}\frac{A(t)}{M(t)} \\ &= \frac{1}{2}\frac{M(t)}{t}\frac{\sum_{i=1}^{M(t)} X_i^2}{M(t)}.\end{aligned}$$

where we have introduced the number of arrivals after time t, as $M(t)$. As $t \to \infty$, if the variance of the service time distribution is finite, we find that

$$\bar{R} = \frac{1}{2}\lambda\left\langle X^2\right\rangle.$$

Again, assuming that time averages and ensemble averages are equal, we substitute back into (11.1) to obtain the Pollaczek-Khinchin formula

$$\bar{W} = \frac{\lambda\left\langle X^2\right\rangle}{2(1-\rho)}.$$

This implies that the total waiting time in the system is

$$\bar{T} = \bar{X} + \frac{\lambda \left\langle X^2 \right\rangle}{2(1-\rho)}$$

and through Little's formula we can obtain the expected queue length, $\bar{N}_Q$, and the expected number in the system, $\bar{N}$. **Note that derivation of this formula was Question 1 in the 2002-2003 paper**. Now consider Examples 2, 3 and 4 at the end of this chapter for the use of the Pollaczek-Khinchin formula.

11.3.1 Table of Delays and Average Numbers in System

In the table below we summarise average waiting times and various other characteristics of the systems for the Pollaczek-Khinchin formula.

Expected value	General	Exponential	Deterministic
$\bar{W}$	$\frac{\lambda}{2(1-\rho)} \left\langle X^2 \right\rangle$	$\frac{\rho}{\mu(1-\rho)}$	$\frac{\rho}{2\mu(1-\rho)}$
$\bar{T} = \bar{W} + \frac{1}{\mu}$	$\frac{\lambda}{2(1-\rho)} \left\langle X^2 \right\rangle + \frac{1}{\mu}$	$\frac{1}{\mu(1-\rho)}$	$\frac{2-\rho}{2\mu(1-\rho)}$
$\bar{N_Q} = \lambda\bar{W}$	$\frac{\lambda^2}{2(1-\rho)} \left\langle X^2 \right\rangle$	$\frac{\rho^2}{(1-\rho)}$	$\frac{\rho^2}{2(1-\rho)}$
$\bar{N} = \lambda\bar{W} + \rho =$	$\frac{\lambda^2}{2(1-\rho)} \left\langle X^2 \right\rangle + \rho$	$\frac{\rho}{1-\rho}$	$\frac{\rho(2-\rho)}{2(1-\rho)}$

11.4 Automatic Repeat Request (ARQ) System.

The $M/G/1$ queue system can be used for modelling various systems of interest. In an ARQ system packets are transmitted and the system waits for an acknowledgement. If no acknowledgement is received the system retransmits. In this section we review Example 3.15 from Bertsekas and Gallager. A schematic of the system is given in Figure 11.2.

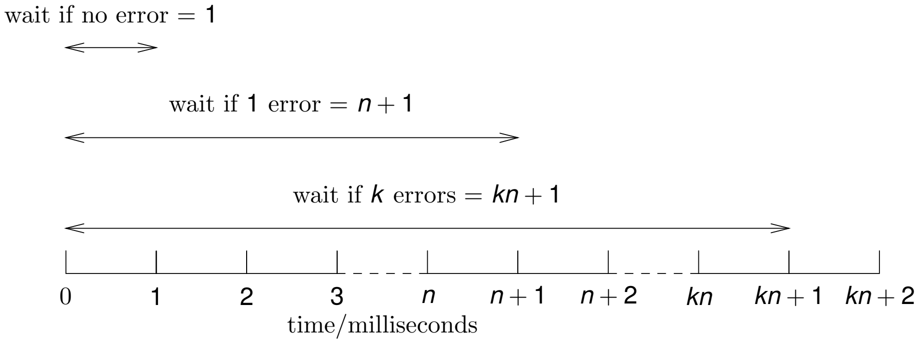


Figure 11.2: The ARQ System.

For the purposes of this analysis, we will assume

1. That each packet takes 1 millisecond to be transmitted.
2. The system waits a maximum of $n-1$ milliseconds for an acknowledgement. If there is no acknowledgement the system retransmits.
3. The acknowledgement occurs if the packet is received without errors. Otherwise acknowledgement doesn't occur.
4. The probability of an error in a packet is taken to be p and we take errors to occur independently.
5. That packets arrive as a Poisson process with rate λ.

In practice, the fourth assumption (independence of errors) is probably unrealistic . If an error occurs due to problems in the transmission channel, it is likely that several packets in a row will not be received, so the errors do not occur independently. The time interval between transmission and re-transmission will be $1 + kn$ milliseconds with probability $(1-p)\,p^k$. This time interval is associated with k retransmissions and one successful transmission. The probability of this being the service time is given by

$$P\,(X = 1 + kn) = (1-p)\,p^k$$

which is recognised as a *geometric distribution.*

11.4.1 Average Service Time

Using this distribution we can compute the expected service time which is given by

$$\begin{aligned}\langle X\rangle &= \sum_{k=0}^{\infty}(1+kn)\,(1-p)\,p^k\\ &= (1-p)\left(\sum_{k=0}^{\infty}p^k + n\sum_{k=0}^{\infty}kp^k\right)\\ &= 1+n\frac{p}{1-p}\end{aligned}$$

since $\sum_{k=0}^{\infty}p^k = \frac{1}{1-p}$ and $\sum_{k=0}^{\infty}kp^k = \frac{p}{(1-p)^2}$.

11.4.2 ARQ Utilization Factor

Strictly speaking, it probably doesn't make sense to use the term 'utilization factor' for the ARQ system. This is because during a large portion of the wait it is not actually 'utilizing' the channel, it is merely waiting for an acknowledgement. However we can still consider, $\rho = \lambda\,\langle X\rangle$, to be the proportion of time that the system will be 'busy'. This 'busy factor',

$$\rho = \lambda + \lambda n\frac{p}{1-p},$$

obviously still has to be less than 1, otherwise the steady state would not exist. This busy time is affected by three parameters: the rate of arrivals, λ, the amount of time that the system waits for an acknowledgement, n, and the probability of an error, p. Note that as $p \to 1$ the second term $\to \infty$.

11.4.3 ARQ Second Moment

The second moment of the service time is

$$\begin{aligned}\left\langle X^2\right\rangle &= \sum_{k=0}^{\infty}(1+kn)^2\,(1-p)\,p^k\\ &= (1-p)\left(\sum_{k=0}^{\infty}p^k + 2n\sum_{k=0}^{\infty}kp^k + n^2\sum_{k=0}^{\infty}k^2p^k\right)\\ &= 1+\frac{2np}{1-p}+\frac{n^2\left(p+p^2\right)}{(1-p)^2}\end{aligned}$$

since

$$\sum_{k=0}^{\infty}k^2p^k = \frac{p+p^2}{(1-p)^3}$$

These moments can be used with the Pollaczek-Khinchin formula to obtain statistics for the queue.

Examples

1. The Residual Service Time

   The residual service time is the amount of service time that remains for the customer under service from the perspective of a customer who has just arrived in the queue. The first part of the proof occurs from the perspective of the arriving customer. The arriving customer is considered to be the ith customer. In the table below we introduce some simple definitions of terms.

   W_i = Waiting time in queue of ith customer.
   R_i = 'Residual service time' seen by the ith customer.
   X_i = Service time of the ith customer.
   k_i = Number of customers in the system when the ith customer arrives.

2. **Markov Service times**. Compute the average waiting time associated with exponentially distributed service times using the Pollaczek-Khinchin formula.

3. **Deterministic Service times ($M/D/1$)**. Some systems have deterministic service times. For example, transmission of fixed length packets over a channel with a fixed bandwidth.

4. **Uniform Service Times** ($M/U/1$). Consider a uniform probability distribution is defined between 0 and b for $X > 0$ as

   $$U(X|b) = \begin{cases} \frac{1}{b} & 0 < X \leq b \\ 0 & X > b \end{cases}$$

   show that the mean of this uniform distribution is given by $\frac{b}{2}$ and its variance by $\frac{b^2}{12}$ and find the expected waiting time for an $M/U/1$ queue.

5. $M/G/1$ Queues with Vacations

   Consider a server which, when it is free, attends to some other business. For example it could transmit control and record keeping traffic. These periods are sometimes known as vacations (which is an American word that means holidays). The vacation periods are characterised by

   (a) If a new arrival arrives during a vacation it must wait until the vacation is over before being dealt with.

   (b) If a vacation ends and the server is idle it immediately goes on a new vacation.

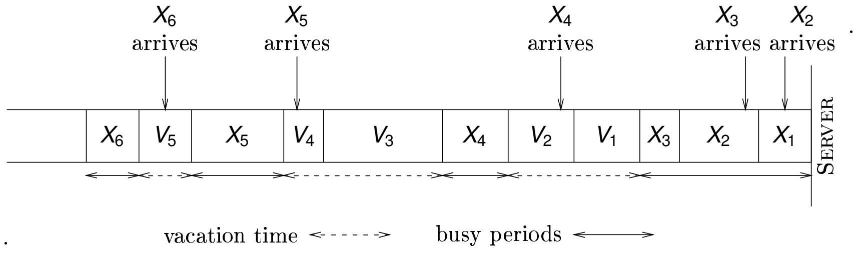


Figure 11.3: Schematic of $M/G/1$ queue

We denote the length of these successive vacations to be V_1, V_2, ... We assume that V_1,V_2, ... are independent and identically distributed (i.i.d.) random variables which are also independent of the customer times and the service times. We can then attempt to compute the mean residual service time for this queue for a *Poisson (Markov) arrival* system with *general service times* and *generally distributed vacation times*.

6. $M/G/1$ **with Vacations**. A bar has Poisson arrivals and one member of staff serving. When there are no customers at the bar, the staff member leaves the bar to check on the kitchen. Assuming these checks are exponentially distributed in length, with mean $\bar{V}$ and that customer service times are exponentially distributed with mean $\bar{X}$ what is the average waiting time in the queue?

Chapter 12

Networks of Queues and Simulation

So far in the course we have looked at queueing models in isolation. We will now briefly turn to some of the issues that arise when they are combined. We will start by considering a simple combination of two $M/M/1$ queues.

12.1 $M/M/1$ Queues in Tandem

Consider a simple system involving two transmission lines in tandem. The packet length doesn't change during transmission and transmission times are proportional to packet length. The packet lengths are exponentially distributed and arrivals occur at the first queue as a Poisson process.

The first queue may be modelled as $M/M/1$, but the second queue cannot be. A packet which is large will take a long time in the first queue. During this time the second queue will have a chance to clear. The result will be that for the second queue larger packets are likely to arrive when the queue is smaller. This means that a large packet will typically experience less delay at the second queue than a small one. One analogy for this situation is a slow moving vehicle on a road. From this vehicle's driver's perspective there is very little traffic on the road because the slow moving vehicle never catches up with any. A similar thing occurs with $M/M/1$ queues in tandem. A large packet arrives at the second queue to find it has less packets because the smaller packets have had time to be processed. The effect of the first queue is to destroy the independence of the packet length and the state of the second queue. This problem is difficult to deal with analytically. To proceed with our analysis we will need to break this dependency. This can be achieved by considering service times for each packet to be statistically independent at each queue. We will now look at the consequences of this assumption for our simple pairing of two queues, but first we must introduce the concept of time reversibility.

12.2 Time Reversibility — Burke's Theorem

In the analysis of the Markov chain systems we have seen so far ($M/M/1$, $M/M/1/m$, $M/M/\infty$, $M/M/m$ and $M/M/m/m$) we relied on detailed balance to find the steady state distribution. Detailed balance equations are a special case of global balance equations which arise when transitions only occur between neighbouring states in the Markov chain. Detailed balance can be applied to find the stationary distribution in any birth-death processes.

In this section we will briefly mention an additional special property of birth-death processes (and other Markov systems for which detailed balance applies) known as *time reversibility.* The principle with time reversibility is as follows: for a birth-death process at steady state the arrival process is Poisson with rate λ. Time reversibility states that the departure process will also be Poisson with rate λ.

We will now give some motivation for this statement, a proof is Given in Section 3.7 of Bertsekas and Gallager. Consider the following 'intuition' for the $M/M/1$ queue. The service rate for the system is given by μ. However this is not the departure rate because a departure can only occur when the system has a state $k > 0$. This condition is satisfied for a fraction of the time which is given by the utilization factor ρ. the departure rate is therefore given by $\mu\rho = \mu\frac{\lambda}{\mu} = \lambda$.

12.3 Burke's Theorem

Burke's theorem follows on from time reversibility. It states that if an $M/M/1$, an $M/M/m$ or an $M/M/\infty$ queue[1] at its stationary distribution has a Poisson arrival process with a rate λ then the departure process is also Poisson with rate λ. Furthermore the number of the customers in the system at time t is *independent* of the sequence of the departure times prior to t.

The second property is worthy of comment as it sounds counterintuitive. You might expect that if there were a rapid series of departures from the queue then the queue would be shorter than normal, but Burke's theorem tells us that this is not the case. From this we can infer that the state of a queue *before* a rapid series of departures is about to occur must be abnormally high. Otherwise it *would* be abnormally low after the departures, but Burke's theorem tells us that this is not the case.

12.4 Kleinrock's Independence Approximation

Kleinrock's independence approximation suggests assuming independence for every node in the network. The approximation is good if large numbers of packets from outside the network are introduced at every queue.

Consider a general network with labelled nodes and links between these nodes. There are several packet streams in the network, each stream has a rate of x_s. We define $f_{ij}(s)$ as the fraction of packets from stream s that cross the link between i and j. The rate associated with a link between node $i$and j is then

$$\lambda_{ij} = \sum_s f_{ij}(s)\, x_s.$$

Kleinrock suggested that merging traffic streams on a transmission line has the effect of restoring independence of arrival and service times. The approach is to assume that each link can be treated as an $M/M/1$ queue. This is a good approximation for:

1. Arrivals which occur as a Poisson process.

2. Exponentially distributed packets.

3. Densely connected networks.

4. Moderate to heavy traffic.

Given this assumption, from the $M/M/1$ queue, we know that the average number at each link is

$$\bar{N}_{ij} = \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}}$$

[1] Queues with loss are slightly more complex because the arrival process does not have rate λ due to blocked packets.

thus the average number in the entire system is

$$\begin{aligned}\bar{N} &= \sum_{ij} \bar{N}_{ij} \\ &= \sum_{ij} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}},\end{aligned}$$

average delay at each link is given by Little's theorem

$$\bar{T}_{ij} = \frac{1}{\mu_{ij} - \lambda_{ij}}.$$

In our studies we will assume that there is no further delay associated with each link (propagation, processing). In practice such delays may arise however we shall not consider them. See Bertsekas and Gallager Section 3.6.1 for more details.

12.5 Jackson's Theorem

While Kleinrock's independence assumption approximates the activities at each queue as $M/M/1$ queues, Jackson's theorem provides a result for the number in the queueing system as a whole. Some analysis of the system is given in Section 3.8 of Bertsekas and Gallager, but for our purposes it is sufficient to give the assumptions and the result. The first assumption is the premise of Kleinrock's independence assumption: there is no correlation between service times and queue lengths. The second assumption is that randomisation is used to divide the traffic along different routes through the network. Given these assumptions the average number of packets in the network is given by

$$\bar{N} = \sum_{i,j} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}}.$$

12.6 Network Simulation

The approximations we have outlined so far may be useful for 'back of the envelope' style calculations. In practice though it is highly likely that you will wish to simulate the system. There are a range of simulation packages available, we will introduce a simple one based on the Tcl scripting language called 'The Network Simulator' or ns-2 (`http://www.isi.edu/nsnam/ns/`). It is a discrete event simulator aimed at networking research. It supports a range of protocols including TCP and multicast protocols and is straightforward to extend using C++. It is a powerful research tool that has the added advantage of being freely available. It is installed in the Lewin Laboratory on the Linux systems and, if you require a personal installation, it can be installed through the Cygwin API (`http://www.cygwin.com`).

12.7 Network Simulation with NS-2

At the second lab class you will simulate some simple queues in NS2 and compare the simulations with approximate results obtained using Kleinrock's independence assumption. The lab class will be completed in Linux so please make sure you know your Unix password for the DCS system before the lab starts. The lab sheet has some notes on simple commands you will need. If you are unfamiliar with Linux it is suggested that you try them out before starting the lab.

Examples

1. Figure 12.1 shows a small network with five nodes. In the network flow only occurs from lower numbered nodes to higher numbered nodes. The connections in the network have average service times in milliseconds as follows:

$\bar{X}$	1	2	3	4	5
1	-	-	-	-	-
2	0.05	-	-	-	-
3	0.05	-	-	-	-
4	0.0333	-	-	-	-
5	-	0.05	0.05	0.033	-

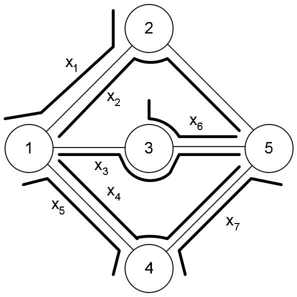


Figure 12.1: A small network with five nodes.

Different routes through the network are indicated with thick lines. Each route has a rate associated with it as follows

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
λ	5	10	7	12	4	8	4

According to the Kleinrock independence assumption, what is the most congested link? What is the quickest route between node 1 & 5? In your answers you should consider propagation or processing delays at each node to be negligible.

2. $M/M/1$ Queues in Tandem Revisited

   Let's return to the $M/M/1$ queues in tandem. Burke's theorem tells us that the departure process from the first queue will be a Poisson with rate λ. Now if we allow service times at the second queue to be independent of service times at the first queue we can break the dependence of the queue length at the second queue on the service time at the second queue. In this case the system can be modelled as two $M/M/1$ queues. What are the distributions for the states of the two systems?

Chapter 13

Multiaccess Systems

The final topic we will cover is a brief review of multiaccess systems. These are systems where there is one communication channel and multiple stations (or nodes) making use of it. Examples of multiaccess systems include Satellite systems, where one Satellite serves many ground stations; packet radio networks, where many stations communicate across the same radio bandwidths; a multitapped bus, where several computers are attached to the same transmission line.

A very common example of multi-access channels is the local area network. In a local area network transmission rates are around 10 Mbps to 1Gbps. Transmission distances range from a few meters to several kilometres. The required error rates are low: typically from 10^{-8} to 10^{-11}, that's about one error in every Gigabyte.

Typically we are interested in the throughput of the network — the amount of data or messages passing through the network each second. If the number of overhead bits (associated with network administration *etc.*) is low then this is approximately equal to the utilization factor ρ.

Until the introduction of the ALOHA network the standard approach to splitting a channel was through multiplexing. We briefly review multiplexing in the examples section at the end of this chapter, recreating Example 3.9 from Bertsekas and Gallager.

13.1 Channel Multiplexing

Let's assume we have m statistically independent Poisson streams. Each has arrival rate $\frac{\lambda}{m}$. The packets to be transmitted are assumed to have an exponentially distributed length which, for a given bandwidth leads to a service time distribution for the whole transmission line which is an exponential with a rate parameter μ.

13.1.1 Statistical Multiplexing

If we merge the Poisson streams together, through statistical multiplexing the superposition property of the Poisson distribution (from Section 4.3.1) tells us that they become a Poisson stream with rate λ. If the transmission times are exponentially distributed the average delay, from the $M/M/1$ queue, will be

$$\bar{T} = \frac{1}{\mu - \lambda}.$$

13.1.2 Time or Frequency Division Multiplexing

With the alternative approach of providing each communication channel with a separate circuit the transmission line needs to be split into m separate channels. Each of these separate channels will now have service time distributions with a reduced rate $\frac{\mu}{m}$.

Delay in this system will now be m times larger than that for statistical multiplexing

$$\bar{T} = \frac{1}{\frac{\mu}{m} - \frac{\lambda}{m}} = \frac{m}{\mu - \lambda}.$$

13.1.3 Why does this happen?

The reduction in delay arises from the fact that when only one of the m channels is transmitting with statistical multiplexing it can use the entire bandwidth of the channel. With time or frequency multiplexing $\frac{m-1}{m}$ of the bandwidth is wasted.

13.2 Multiaccess Systems

In multiaccess systems splitting the channel using time division multiplexing is known as time division multiple access (TDMA). Similarly splitting the channel using frequency division multiplexing is known as frequency division multiple access (FDMA). A further technique is known as code division multiple access (CDMA) but we will not discuss that here.

We have already shown that statistical multiplexing can make more efficient use of bandwidth than time or frequency based multiplexing. However in a multiaccess system it can be difficult to implement in the manner discussed above. We assumed that all the transmissions could be queued in a single $M/M/1$ queue. In a local area network the transmitting stations (or nodes) are distributed across different portions of the network as are the receiving nodes. Forming a single queue is therefore not possible without some form of centralised control. TDMA and FDMA can be implanted by dedicating one channel for each station to transmit on, but allowing each station to receive on any channel. This then allows for an analysis of the type we described above. However, it is not clear how to perform statistical multiplexing in these circumstances.

13.3 Random Access

One way of performing statistical multiplexing when we have stations distributed across a single communication channel would be to have a centralised computer attempting to control which computer is able to access our transmission lines at what times. However, we may wish communication to occur without centralised control. In this case we may choose a random access solution.

We will consider random access networks where there is one communication channel on which all the nodes communicate[1]. Each station sends a packet immediately whenever it receives a packet to send. this means that packets can *collide* and may have to be resent.

13.3.1 Pure ALOHA Networks

The ALOHA network was developed at the University of Hawaii, it was a packet radio network which communicated between different stations on the Hawaiian islands. For the ALOHA network (and Ethernet, the modern LAN protocol that was inspired by ALOHA) all stations attempt to communicate on the same channel. When a station attempts to transmit, it detects if its transmission has interfered with an existing transmission: this is known as a collision. If a collision occurs the station becomes *backlogged* and attempts to transmit at a later time. If both stations attempted to retransmit after the *same* period of time they would simply collide again after retransmission. To avoid this problem the stations normally wait for a random period of time before retransmitting.

For this network we assume that each station, after a given propagation delay, can tell whether the transmitted packets were correctly received. Furthermore, we assume that each packet requires

[1] It could be the case that only a subset of the nodes have access to the channel. For example in a packet radio network the range between stations may mean that only a subset of nodes are able to receive from any particular node. However this complicates analysis so we will ignore this possibility.

1 'time unit' for transmission. This means that if a packet of interest is sent at time t, then any other packet sent between time $t-1$ and time $t+1$ will collide with our packet.

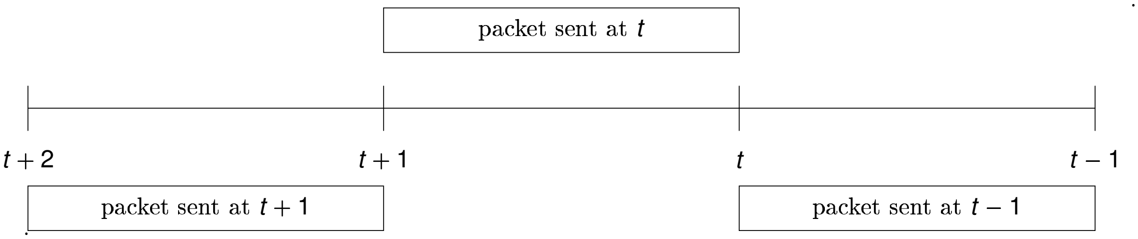


Figure 13.1: Pure ALOHA, packets collide with any other packet transmitted within one time unit of their transmission. The diagram shows a packet sent at t and packets sent at $t+1$ and $t-1$. Any packet sent after $t-1$ and before $t+1$ will collide with the packet sent at t.

We will now look at a simple analysis of the ALOHA system. Let's assume that the total arrival rate for packets at all stations is a Poisson process with rate λ. We assume that the number of stations is very large (infinite) so that an individual station's packet rate is an insignificant proportion of λ, *i.e.*

$$\frac{\lambda_n}{\lambda} = \lim_{\delta \to 0} = 0.$$

A station is considered backlogged between the time that it became aware that its previous transmission failed and the time of a successful transmission. We denote the number of backlogged stations at any particular time as n. Once a station is aware that a transmission was unsuccessful it waits for T_i time units to retransmit. The waiting time is sampled from a probability density function given by

$$p\left(T_i\right) = \mu \exp\left(-\mu T_i\right).$$

So if n nodes are backlogged then (from superposition of Poissons) the arrivals from the backlogged nodes are a Poisson process with rate[2] $n\mu$. So the overall arrival process, from the perspective of the channel, is Poisson with a rate that is a function of n, $G\left(n\right) = \lambda + n\mu$.

The probability that there are no arrivals in an interval of 2 time units for a Poisson process with rate $G\left(n\right)$ is given by

$$p\left(k = 0 | G\left(n\right), 2\right) = \exp\left(-2G\left(n\right)\right)$$

as we derived in Section 4.2. This is the probability that there is no collision between the packet of interest and any other packets in the network, thus it is the probability of a successful transmission. The rate of attempted transmissions is $G\left(n\right)$ so the rate of successful transmissions is given by

$$G'\left(n\right) = G\left(n\right) \exp\left(-2G\left(n\right)\right),$$

this is known as the throughput. It is the number of packets that successfully reach their destination. The throughput is plotted as a function of G in Figure 13.2.

Note that the curve goes up to a maximum point and drops down thereafter. We can find the maximum of the throughput by looking for points where the gradient of the throughput is zero. In the examples below we find that the maximum throughput for the system occurs when $G = \frac{1}{2}$. The throughput at this point is $\frac{1}{2e} \approx 0.184$. This maximum throughput shows that at best the channel is in use only about 18% of the time[3]. The system will be in equilibrium if the throughput equals λ. In other words if the throughput is equal to the arrival rate of the packets.

[2] In Bertsekas and Gallager they use x as the parameter instead of μ.

[3] Note that since the service time is fixed at $X_n = 1$ for all n then the utilization for this system is given by $\rho = \lambda \bar{X} = \lambda$.

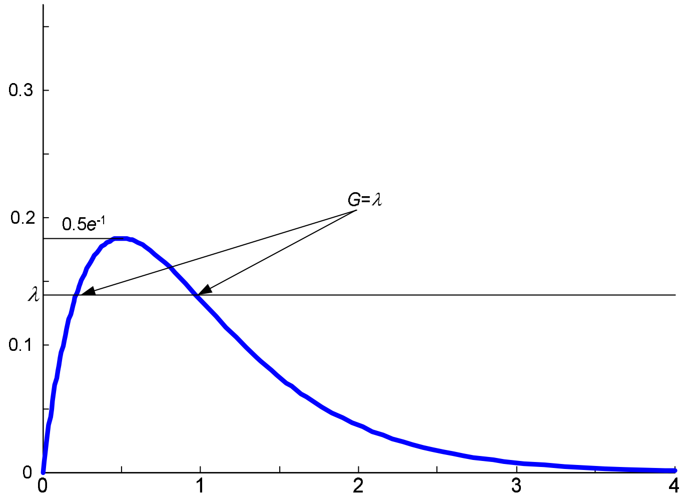


Figure 13.2: Throughput for pure ALOHA as a function of G.

13.3.2 Slotted ALOHA

A simple improvement to the ALOHA system comes from constraining when packets can be sent. Instead of allowing stations to transmit immediately they receive a packet each station waits for the beginning of a 'slot' to start transmitting. These slots are synchronised across the entire network. In this system a collision occurs if another station attempts to transmit in the same slot. This happens if another station received a packet for transmission in the preceding slot. If each slot is one time unit in length the probability of this event is given by

$$p\left(k=0|G\left(n\right),2\right)=\exp\left(-G\left(n\right)\right).$$

The probability is smaller than for pure ALOHA because the 'vulnerable period' for a collision is smaller (half the length of before). A collision now only occurs if another station attempts to transmit in the same slot. As a result the rate of successful transmissions is given by

$$G'\left(n\right)=G\left(n\right)\exp\left(-G\left(n\right)\right),$$

this is known as the throughput. It is the number of packets that successfully reach their destination. The throughput is plotted as a function of G in Figure 13.3.

The smaller vulnerable period leads to a greater maximum throughput which is now found at $G=1$. The maximum throughput is $\frac{1}{e}$. Slotted ALOHA therefore can be twice as efficient as pure ALOHA.

13.3.3 Analysis of Backlog Delay in Pure ALOHA

Recall our assumption that a backlogged node waits for a time, T_i, before attempting to retransmit,

$$p\left(T_i\right)=\mu\exp\left(-\mu T_i\right).$$

Given this assumption, can we compute the delay associated with this backlogged node? Since the node may experience several collisions, the wait associated with the backlog will involve a sum over the random variable T_i. It turns out to be an *Erlang distribution*.

13.3.4 Sums of Exponential Variables and the Erlang Distribution

If a random variable, x, is exponentially distributed with rate parameter b. The sum of a random variables, each of which is exponentially distributed with rate parameter b is distributed as an

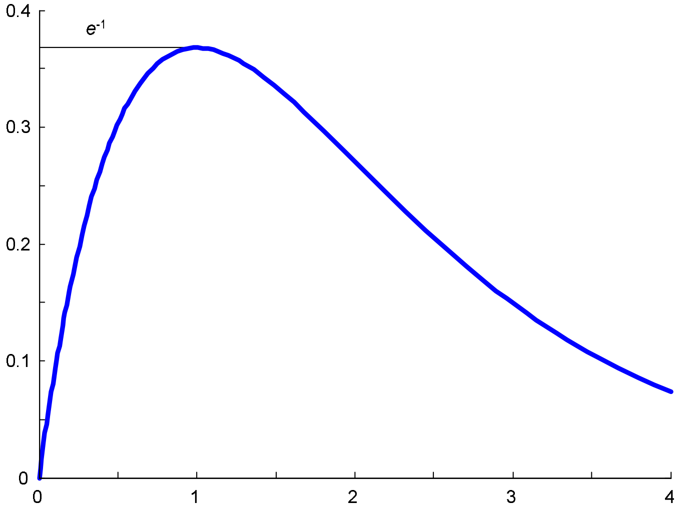


Figure 13.3: Throughput for slotted ALOHA as a function of G.

Erlang distribution. Consider variables a set of a variables$(x_1 \ldots x_a)$ which are each governed by the same exponential distribution,

$$p\left(x_i|b\right) = b\exp\left(-bx_i\right).$$

We define the sum of the values to be, $z = \sum_{i=1}^{a} x_i$. The distribution governing z is then an *Erlang distribution*, given by

$$p\left(z|a,b\right) = \frac{b^a}{(a-1)!}z^{a-1}\exp\left(-bz\right).$$

Note that this distribution is a generalisation of the exponential distribution because the exponential is the special case where $a = 1$. This clearly should be the case because if $a = 1$ then the 'sum' of that variable will simply be the variable itself.

13.3.5 Erlang Distribution Mean and Variance

The mean of the Erlang distribution is given by

$$\langle z\rangle_{p(z|a,b)} = \frac{a}{b}$$

and the variance is

$$\mathrm{var}\left(z\right) = \frac{a}{b^2}.$$

13.3.6 Delay Associated with Backlogs

If the node retransmits k times then the distribution over the total time required, $W = \sum_{i=1}^{k} T_i$, comes from the sum of k independent exponentially distributed variables. From the property of the exponential distribution given in Section 13.3.4 we know that the resulting variable X is Erlang distributed with parameters μ and k,

$$p\left(W|k,\mu\right) = \frac{\mu^k}{(a-1)!}W^{k-1}\exp\left(-\mu W\right).$$

The probability of transmitting after k attempts is given by

$$p\left(k\right) = q\left(1-q\right)^{k-1}.$$

where q is the probability of a successful transmission[4]. For the pure ALOHA system we have, from above,

$$q = \exp\left(-2G\right).$$

.

We are now interested in the distribution of the waiting time, using the product rule and the sum rule of probability we know that the probability distribution for W is given by

$$p\left(W|\mu\right) \quad = \quad \sum_{k=1}^{\infty} p\left(W|k,\mu\right) p\left(k\right),$$

substituting in our distributions gives,

$$p\left(W|\mu\right) = \sum_{k=1}^{\infty} q\left(1-q\right)^{k-1} \frac{\mu^k}{(k-1)!} W^{k-1} \exp\left(-\mu W\right),$$

now substituting $\dot{k} = k - 1$

$$p\left(W|\mu\right) = \sum_{k=0}^{\infty} q\left(1-q\right)^{\dot{k}} \frac{\mu^{\dot{k}+1}}{\dot{k}!} X^{\dot{k}} \exp\left(-\mu W\right),$$

extracting the q, the exponential term and one of the μs from the product we rewrite as,

$$p\left(X|\mu\right) = q\mu \exp\left(-\mu W\right) \sum_{\dot{k}=0}^{\infty} \frac{\left(\mu\left(1-q\right)W\right)^{\dot{k}}}{\dot{k}!}, \tag{13.1}$$

Recalling the following equality,

$$\exp\left(x\right) = \sum_{k=0}^{\infty} \frac{x^{\dot{k}}}{\dot{k}!},$$

we know we can write

$$\sum_{k=0}^{\infty} \frac{\left(\mu\left(1-q\right)W\right)^{\dot{k}}}{\dot{k}!} = \exp\left(\mu\left(1-q\right)W\right),$$

which can be substituted back into (13.1) to obtain

$$\begin{aligned} p\left(W|\mu\right) &= q\mu \exp\left(-\mu W\right) \exp\left(\mu\left(1-q\right)W\right), \\ &= q\mu \exp\left(-q\mu W\right) \end{aligned}$$

which is recognised as an exponential distribution with rate parameter $\mu' = q\mu = \mu \exp\left(-2G\right)$.

Using the fact that G is a function of μ we can write

$$\mu' = \mu \exp\left(-2\left(\lambda + n\mu\right)\right). \tag{13.2}$$

This means that for a given value of λ and n there is an optimal value for μ. This value of μ can be found by seeking the maximum rate, this is done by differentiating with respect to μ,

$$\frac{d\mu'}{d\mu} = \left(1 - 2n\mu\right) \exp\left(-2\left(\lambda + n\mu\right)\right)$$

and setting the result to zero to find the stationary points we have

$$\left(1 - 2n\mu\right) \exp\left(-2\left(\lambda + n\mu\right)\right) = 0.$$

[4] Compare this with the ARQ system we analysed in the chapter on $M/G/1$ queues. There we were considering errors (unsuccessful transmissions) so $p = (1 - q)$.

There are two solutions to this equation. The first is $\exp\left(-2\left(\lambda + n\mu\right)\right) = 0$, meaning that $\lambda + n\mu \to \infty$. This solution is a minimiser . The other solution is given when

$$0 = 1 - 2n\mu$$

or

$$\mu = \frac{1}{2n},$$

this turns out to be a maximum of (13.2). A maximum is what we want because by maximising the 'rate' parameter μ' we minimise the average delay associated with the backlog, which is given by $\bar{W} = \frac{1}{\mu}$, so by setting $\mu = \frac{1}{2n}$ we find a minimum for the average delay which is given by

$$\bar{W}_{\min} = 2n \exp\left(2\lambda + 1\right).$$

In practice it may be difficult to set the rate μ to something that is dependent on the number of backlogged nodes. this is because the number of backlogged nodes can only be known by nodes communicating their status over the network and we have not allowed for such a mechanism in our analysis. Note that the minimum average delay goes up linearly with the number of backlogged nodes and *exponentially* with the rate of arrivals for the system, λ.

Examples

1. Throughput for pure ALOHA system. Find the maximum throughput for the pure ALOHA system by seeking the stationary point of the throughput curve.

Bibliography

D. P. Bertsekas and R. G. Gallager. *Data Networks.* Prentice-Hall, Upper Saddle River, NJ, second edition, 1992. ISBN 0-13-200-916-1.

V. G. Cerf and R. E. Kahn. A protocol for packet network intercommunication. *IEEE Transactions on Communications*, COM-22(5):637–648, 1974.

W.-C. Chan. *Performance Analysis of Telecommunications and Local Area Networks.* Kluwer, Dordrecht, The Netherlands, 2000. ISBN 0-7923-7701-X.

A. K. Erlang. The theory of probabilities and telephone conversations. *Nyt Tidsskrift for Matematik B*, 20, 1909.

L. Kleinrock. *Communication Nets: Stochastic Message Flow and Delay.* McGraw-Hill, New York, NY, 1964.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423 and 623–656, 1948.

A. S. Tanenbaum. *Computer Networks.* Prentice-Hall, Upper Saddle River, NJ, fourth edition, 2003. ISBN 0-13-038488-7.